



クイック検索

検索

Copyright © 2016 NTT DATA INTRAMART
CORPORATION

目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の内容
 - 2.2. 対象読者
 - 2.3. 表記規則
 - 2.4. その他
- 3. 基本
 - 3.1. 事前準備
 - 3.1.1. 開発環境構築
 - 3.1.2. S2JDBC
 - 3.2. 初めてのプログラミング
 - 3.2.1. プログラミング
 - 3.2.2. 動作確認
 - 3.3. 仕様解説
 - 3.3.1. Gadget
 - 3.3.2. TMTable
 - 3.3.3. Accessor
 - 3.3.4. Ajax
 - 3.3.5. Mode&Formated
 - 3.3.6. Dialog
 - 3.3.7. ListData
 - 3.3.8. FS2List
 - 3.3.9. Manager
 - 3.3.10. Conductor
 - 3.3.11. Log
 - 3.3.12. MultiLocales
 - 3.3.13. Helper&Util
 - 3.3.14. Exception
- 4. 応用
 - 4.1. マスタメンテナンス
 - 4.1.1. 機能説明
 - 4.1.2. チュートリアル
 - 4.1.3. サーバサイド
 - 4.2. マスタ検索
 - 4.2.1. 機能説明
 - 4.3. ガジェット(経費旅費)
 - 4.3.1. 機能説明
 - 4.3.2. チュートリアル
 - 4.3.3. クライアントサイド
 - 4.3.4. サーバサイド
 - 4.4. ガジェット(勤務管理)
 - 4.4.1. 機能説明
 - 4.4.2. チュートリアル
 - 4.5. その他
 - 4.5.1. 機能説明
- 5. サンプル
 - 5.1. ガジェット(経費旅費)
 - 5.1.1. 経費明細
 - 5.1.2. 経費明細(複数通貨)
 - 5.1.3. 旅程明細
 - 5.1.4. 旅程明細(複数通貨)
 - 5.1.5. 宿泊明細
 - 5.1.6. 宿泊明細(複数通貨)
 - 5.1.7. 日当明細
 - 5.1.8. 日当明細(複数通貨)
 - 5.2. マスタ検索
 - 5.2.1. プロジェクトマスタ検索
 - 5.2.2. 経費区分マスタ検索
 - 5.3. マスタメンテナンス
 - 5.3.1. プロジェクトマスタメンテナンス
 - 5.3.2. 経費区分マスタメンテナンス
 - 5.4. インポートジョブ
 - 5.4.1. プロジェクトマスタインポート
 - 5.4.2. 経費区分マスタインポート
- 6. リファレンス
 - 6.1. バリデーション
 - 6.1.1. 項目
 - 6.1.2. マスタ
 - 6.1.3. その他
 - 6.2. タグライブラリ
 - 6.2.1. エレメント

- 6.2.2. リストデータ
- 6.2.3. マスタ検索
- 6.2.4. その他

改訂情報

変更年月日	変更内容
2013-11-21	初版
2014-06-09	第2版 下記を追加しました。 intra-mart Accel Kaiden! 勤務管理 の記載を追加
2015-02-27	第3版 下記の追加・変更を行いました。 本書の構成や記載内容を整備 サンプル を追加

はじめに

本書の内容

本書では、次の様な構成でintra-mart Accel Kaiden!の基本的なプログラミング方法や注意点を記載しています。

- [基本](#)
intra-mart Accel Kaiden!を使用したプログラミングの基本的な内容を記載しています。
プログラミング前の準備や、簡単なプログラミング方法、各種仕様を記載していますので、初めてintra-mart Accel Kaiden!を使用したプログラミングを行う方を対象に記載しています。
- [応用](#)
intra-mart Accel Kaiden!を使用したプログラミングの応用的な内容を記載しています。
機能別の解説やチュートリアル、カスタマイズ方法を記載していますので、intra-mart Accel Kaiden!上で機能を開発する方や、標準機能のカスタマイズを行う方を対象に記載しています。
- [サンプル](#)
intra-mart Accel Kaiden!のサンプルソースです。
- [リファレンス](#)
本書のリファレンスを記載しています。



コラム

本書では、特に記載のない限り次の製品（バージョン）を前提に記載しています。

- intra-mart Accel Platform 2014 Winter(Iceberg) (8.0.9)
- intra-mart Accel Kaiden! Core Module 2.0.2
- intra-mart Accel Kaiden! 経費旅費 2.0.2
- intra-mart Accel Kaiden! 勤務管理 2.0.2

対象読者

本書では次の読者を対象としています。

- intra-mart Accel Kaiden! で初めてプログラミングを行う開発者
- intra-mart Accel Kaiden! の各機能を利用したい開発者



注意

本書では次の知識を有していることを前提に解説しています。

- intra-mart Accel Platform のSAstruts+S2JDBCフレームワークでの開発の知識/経験
- intra-mart Accel Platform に同梱されているjQueryおよびjQueryの各種プラグインの知識/経験



コラム

intra-mart Accel Kaiden!はSAstruts+S2JDBCの開発モデルで構築されています。
本書を読み進める前に、『[SAstruts+S2JDBC プログラミングガイド](#)』の理解を深めることを推奨します。

表記規則

本書内での表記方法は、次の凡例に従って記述しています。

- 例:『○○ リリースノート』
二重カギ括弧(『』)は、他のマニュアルなどの文献名を示しています。
- 例:「△△画面」「××ボタン」
大括弧(「」)は、画面、ダイアログボックス、ボタン、リンクなどの画面上の要素を示しています。

その他

- 本書内で記載されている外部URLは、2015年02月現在のものとなります。

基本

本項では、intra-mart Accel Kaiden!を使用したプログラミングの基本的な内容を記載しています。

事前準備

本項では、intra-mart Accel Kaiden!でプログラミングする際の準備内容を説明します。
本書では、本項の事前準備が完了していることを前提に記載しています。

- 開発環境構築
 - intra-mart e Builder for Accel Platformの準備
 - モジュール・プロジェクトの作成
 - convention.diconの設定
 - kaiden_tutorial.diconの作成
 - kaiden.diconの編集
- S2JDBC
 - S2JDBC-Genのセットアップ
 - 補足
 - GenerateEntity
 - GenerateService
 - AutoTimestamp
 - AutoUpdateUser
 - s2jdbc-gen-build.xml/s2jdbc-gen-build.properties

開発環境構築

本書のプログラミング方法は、intra-mart e Builder for Accel Platformを前提としています。
本項では、intra-mart e Builder for Accel Platformのインストール、および開発プロジェクトの準備を行います。

intra-mart e Builder for Accel Platformの準備

intra-mart e Builder for Accel Platformのインストール、およびデバッグサーバの構築を行います。
構築手順の詳細は、『intra-mart e Builder for Accel Platform セットアップガイド』を参照してください。



コラム

デバッグサーバの構築時（WARファイルの作成時）には、利用するintra-mart Accel Kaiden!のモジュールを追加してください。
また、WARファイルのデプロイ後、テナント環境の構築まで完了させてください。

モジュール・プロジェクトの作成

intra-mart e Builder for Accel Platform上にモジュール・プロジェクトを作成し、プロジェクトの設定を行います。
プロジェクトの作成・設定の方法に関しては、『intra-mart e Builder for Accel Platform ユーザ操作ガイド』のモジュール・プロジェクト作成、およびプロジェクトの設定を参照してください。



コラム

モジュール・プロジェクト作成ウィザードでは次の情報を参考に入力してください。
あくまで参考値ですので、別の値を入力していただいても問題ありません。

- プロジェクト名 :tutorial
- グループID :jp.co.slcs.kaiden2
- アーティファクトID:tutorial
- バージョン :1.0.0

convention.diconの設定

モジュール・プロジェクトの作成後、convention.dicon でルートパッケージの設定をします。
SAStruts フレームワークでは、ルートパッケージを設定することで、「ルートパッケージ.action」パッケージに配置されたJavaクラスにアクセスすることができます。



コラム

例

ルートパッケージを「jp.co.xxx.yyy」と設定した場合に、Javaクラス「jp.co.xxx.yyy.action.foo.BarAction」にアクセスするためのURLは「http://ホスト名/war名/foo/bar/」となります。

「convention.dicon」を「src/main/resource」に配置し、「addRootPackageName」を追加します。
本項以降のチュートリアルでは、ルートパッケージを「jp.co.slcs.kaiden2.tutorial.feature」と設定したものとして記載しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN" "http://www.seasar.org/dtd/components24.dtd">
<components>
  <component class="jp.co.intra_mart.framework.extension.seasar.convention.IMNamingConventionImpl">
    <initMethod name="addRootPackageName">
      <arg>"org.seasar.framework.container.warmdeploy"</arg>
    </initMethod>
    <initMethod name="addRootPackageName">
      <arg>"jp.co.slcs.kaiden2.tutorial.feature"</arg>
    </initMethod>
  </component>
  <component class="org.seasar.framework.convention.impl.PersistenceConventionImpl"/>
</components>
```



コラム

convention.diconなどの各種設定を行うdiconファイルの仕様に関しては、seasar2のホームページを参照してください。

kaiden_tutorial.diconの作成

チュートリアルで使用する「kaiden_tutorial.dicon」を作成します。

「kaiden_tutorial.dicon」を「src/main/resource」に作成し、次の様に編集します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">

<components>

  <!-- manager -->

  <!-- helper -->

  <!-- validator -->

  <!-- service -->

</components>
```

kaiden.diconの編集

「kaiden_tutorial.dicon」を作成後、「kaiden.dicon」に設定を追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN" "http://www.seasar.org/dtd/components24.dtd">
<components xmlns:xi="http://www.w3.org/2001/XInclude">
  <include path="s2jdbc.dicon"/>
  <component name="kaidenSqlLogRegistry" class="org.seasar.extension.jdbc.SqlLogRegistry">@org.seasar.extension.jdbc.SqlLogRegistryLocator@getInstance()</component>

  <xi:include href="/kaiden_base.dicon" />
  <xi:include href="/kaiden_extension_imw.dicon" />
  <xi:include href="/kaiden_product_workflow.dicon" />
  <xi:include href="/kaiden_product_labormgr.dicon" />

  <!-- ↓を追加します -->
  <xi:include href="/kaiden_tutorial.dicon" />
  <!-- ↑を追加します -->
</components>
```



コラム

環境構築時に「kaiden.dicon」以外を利用している場合は、利用したdiconファイルに設定を追加してください。

設定するdiconファイルの設定は『[intra-mart Accel Kaiden! セットアップガイド](#)』のSAStruts用設定ファイルを参照してください。

- intra-mart Accel Kaiden! Core Moduleのみを利用する場合
→ **kaiden_framework.dicon**を編集します。
- intra-mart Accel Kaiden! 経費旅費を利用する場合
→ **kaiden_workflow.dicon**を編集します。
- intra-mart Accel Kaiden! 勤務管理を利用する場合
→ **kaiden_labormgr.dicon**を編集します。
- intra-mart Accel Kaiden!の全てのモジュールを利用する場合
→ **kaiden.dicon**を編集します。

S2JDBC

intra-mart Accel Kaiden!のデータベースプログラミング方法は、S2JDBCを前提としています。

本項では、S2JDBCのエンティティやサービスクラスを自動生成するS2JDBC-Genの準備を行います。

S2JDBC-Genのセットアップ

S2JDBC-Genのダウンロード、およびセットアップを行います。

セットアップの詳細はSeasarプロジェクトのサイト (http://s2container.seasar.org/2.4/ja/s2jdbc_gen/) を参照してください。

- エンティティ

intra-mart Accel Kaiden!が使用するエンティティは、S2JDBC-Genを使ってデータベース上のテーブル定義から自動生成します。
自動生成時にエンティティのスーパークラスに「jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity」を指定してください。

! 注意

エンティティを自動生成するには、必ずスーパークラス「GenerateEntity」を指定する必要がありますが、自動生成時に「GenerateEntity」がない場合、ビルドエラーが発生する場合があります。
次項の「GenerateEntity」をS2JDBC-Genのプロジェクト内に配置し、エンティティの自動生成を行うようにしてください。

i コラム

intra-mart Accel Kaiden!のエンティティはOracle Database上のテーブルから自動生成しています。
本書でも、エンティティを自動生成する際のデータベースは、Oracle Databaseを推奨します。

自動生成されるエンティティのプロパティの型は、各データベースのデータ型により生成される内容が変わります。
詳細はSeasarプロジェクトのサイトのデータベースのデータ型とJavaの型の対応表 (http://s2container.seasar.org/2.4/ja/s2jdbc_gen/tasks/gen_entity.html) を参照して

- サービス

エンティティに対する操作を格納するクラスをサービスと呼びます。
intra-mart Accel Kaiden!が使用するサービスは、S2JDBC-Genを使ってエンティティから自動生成します。
自動生成後にサービスのスーパークラスを「jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService」に変更してください。

i コラム

サービス生成時に「GenerateService」を自動設定する場合は、S2JDBC-Genのカスタマイズが必要です。
サービス生成時に抽象サービスクラスが出力される場合がありますが、スーパークラスを「GenerateService」にする場合は生成された抽象サービスクラスは不要です。

補足

S2JDBC-Genプロジェクトへ配置するソースや設定ファイルは次の項を参照してください。

```
src
├── main
│   └── java
│       ├── jp
│       │   ├── co
│       │   │   ├── slcs
│       │   │   │   ├── kaiden2
│       │   │   │   │   ├── base
│       │   │   │   │   │   ├── foundation
│       │   │   │   │   │   │   ├── model
│       │   │   │   │   │   │   │   ├── annotation
│       │   │   │   │   │   │   │   │   ├── AutoTimestamp.java
│       │   │   │   │   │   │   │   │   ├── AutoUpdateUser.java
│       │   │   │   │   │   │   │   ├── entity
│       │   │   │   │   │   │   │   │   ├── GenerateEntity.java
│       │   │   │   │   │   │   │   └── service
│       │   │   │   │   │   │   │       GenerateService.java
```

GenerateEntity

- Path

src/main/java/jp/co/slcs/kaiden2/base/foundation/model/entity/GenerateEntity.java

```

package jp.co.slcs.kaiden2.base.foundation.model.entity;

import java.sql.Timestamp;
import javax.persistence.Column;
import javax.persistence.MappedSuperclass;
import javax.persistence.Version;
import jp.co.slcs.kaiden2.base.foundation.model.annotation.AutoTimestamp;
import jp.co.slcs.kaiden2.base.foundation.model.annotation.AutoUpdateUser;

/**
 * s2jdbc-genを利用して自動生成したEntityのスーパークラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@MappedSuperclass
public abstract class GenerateEntity {

    /** 登録日時. */
    @AutoTimestamp(insert = true, update = false)
    @Column(name = "entry_ts", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp entryTs;

    /** 登録者コード. */
    @AutoUpdateUser(insert = true, update = false)
    @Column(name = "entry_user_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String entryUserCd;

    /** 更新カウンタ. */
    @Version
    @Column(name = "renew_cnt", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long renewCnt;

    /** 更新日時. */
    @AutoTimestamp(insert = true, update = true)
    @Column(name = "renew_ts", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp renewTs;

    /** 更新者コード. */
    @AutoUpdateUser(insert = true, update = true)
    @Column(name = "renew_user_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String renewUserCd;
}

```

GenerateService

- Path
src/main/java/jp/co/slcs/kaiden2/base/foundation/model/service/GenerateService.java

```

package jp.co.slcs.kaiden2.base.foundation.model.service;

import org.seasar.extension.jdbc.AutoSelect;
import org.seasar.extension.jdbc.DbmsDialect;
import org.seasar.extension.jdbc.IterationCallback;
import org.seasar.extension.jdbc.IterationContext;
import org.seasar.extension.jdbc.JdbcManager;
import org.seasar.extension.jdbc.OrderByItem;
import org.seasar.extension.jdbc.SqlFileSelect;
import org.seasar.extension.jdbc.SqlFileUpdate;
import org.seasar.extension.jdbc.SqlLogRegistry;
import org.seasar.extension.jdbc.manager.JdbcManagerImpl;
import org.seasar.extension.jdbc.manager.JdbcManagerImplementor;
import org.seasar.extension.jdbc.operation.Operations;
import org.seasar.extension.jdbc.service.S2AbstractService;
import org.seasar.extension.jdbc.util.LikeUtil;
import org.seasar.extension.jdbc.where.SimpleWhere;
import org.seasar.extension.sql.Node;
import org.seasar.extension.sql.SqlParser;
import org.seasar.extension.sql.node.BindVariableNode;
import org.seasar.extension.sql.node.EmbeddedValueNode;
import org.seasar.extension.sql.node.ParenBindVariableNode;
import org.seasar.extension.sql.parser.SqlParserImpl;
import org.seasar.util.beans.util.BeanMap;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.lang.reflect.Field;
import java.net.URL;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;

```

```

import javax.persistence.Id;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;
import jp.co.slcs.kaiden2.base.foundation.model.util.AutoFieldSetter;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.log.KaidenLogger;

/**
 * s2jdbc-genを利用して自動生成したServiceのスーパークラス。
 * <p>
 * 当クラスで実行されたSQL文はすべてログファイルにレベルDebugで出力されます。<br>
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 * @param <ENTITY> エンティティの型
 */
public abstract class GenerateService<ENTITY> extends S2AbstractService<ENTITY> {

    /** SQLファイル読み込時に使用する文字エンコーディング。 */
    private static final String SQL_READ_CHARSET_STR = "UTF-8";

    /**
     * ログリボジトリ。
     */
    @Resource
    public SqlLogRegistry kaidenSqlLogRegistry;

    /**
     * {@inheritDoc}
     */
    @Override
    protected final void setEntityClass(Class<ENTITY> entityClass) {
        super.setEntityClass(entityClass);
        super.sqlFilePathPrefix = "META-INF/sql/" + StringUtil.replace(this.getClass().getName(), ".", "/" + "/");
    }

    /**
     * エンティティのクラスを返却。
     *
     * @return エンティティクラス
     */
    public Class<ENTITY> getEntityClass() {
        return super.entityClass;
    }

    /**
     * エンティティにフィールドが含まれるかを判定。
     *
     * @param fieldName フィールド名
     * @return 判定結果 true:含まれる、false:含まれない
     */
    public boolean hasField(String fieldName) {
        return BeanUtil.hasField(entityClass, fieldName);
    }

    /**
     * PKフィールドを返却。
     *
     * @return PKフィールド
     */
    public List<String> getPkFiled() {
        List<String> result = new ArrayList<String>();
        Field[] fields = entityClass.getFields();
        for (Field fld : fields) {
            Id id = fld.getAnnotation(Id.class);
            if (id != null) {
                result.add(fld.getName());
            }
        }
        return result;
    }

    /**
     * dtoから主キー条件のkey-value(MAP)を返却。
     * <p>
     * すべてイコール条件になります。主キー項目がDTOに存在しない、または主キー条件がnull(空文字)の場合は、RuntimeExceptionをスローします。
     * </p>
     *
     * @param dto 対象dto
     * @return 主キーのkey-value(MAP)
     */
    public Map<String, Object> getPKCriteria(Object dto) {

```

```

public Map<String, Object> getCriteria(Object dto) {
    Map<String, Object> params = new HashMap<String, Object>();
    Field[] fields = entityClass.getFields();

    for (Field fld : fields) {
        Id id = fld.getAnnotation(Id.class);
        if (id != null) {
            try {
                Field dtoField = BeanUtil.getField(dto, fld.getName());
                if (dtoField == null) {
                    throw KaidenIllegalParametersException.makeInstance();
                }

                // フィールドの値
                Object value = dtoField.get(dto);
                if (value instanceof String && StringUtil.isEmpty((String) value)) {
                    throw KaidenIllegalParametersException.makeInstance();
                } else if (value == null) {
                    throw KaidenIllegalParametersException.makeInstance();
                }
                params.put(fld.getName(), value);

            } catch (IllegalArgumentException e) {
                throw KaidenRuntimeException.makeInstance(e);
            } catch (IllegalAccessException e) {
                throw KaidenRuntimeException.makeInstance(e);
            } catch (SecurityException e) {
                throw KaidenRuntimeException.makeInstance(e);
            }
        }
    }
    return params;
}

/**
 * dtoから検索条件のkey-value(MAP)を返却.
 * <p>
 * エンティティのフィールドと名称のフィールドのみ条件が作成されます.<br />
 * すべてイコール条件になります。条件値がnull(空文字)の場合は、無視されます。
 * </p>
 *
 * @param dto 対象dto
 * @return 検索条件のMap
 */
public Map<String, Object> getCriteria(Object dto) {
    Map<String, Object> params = new HashMap<String, Object>();
    Field[] fields = entityClass.getFields();

    for (Field fld : fields) {
        Field dtoField;
        try {
            dtoField = BeanUtil.getField(dto, fld.getName());
            if (dtoField != null) {

                // フィールドの値
                Object value = dtoField.get(dto);
                if (value != null) {
                    params.put(fld.getName(), value);
                }
            }
        } catch (SecurityException e) {
            throw KaidenRuntimeException.makeInstance(e);
        } catch (IllegalArgumentException e) {
            throw KaidenRuntimeException.makeInstance(e);
        } catch (IllegalAccessException e) {
            throw KaidenRuntimeException.makeInstance(e);
        }
    }
    return params;
}

/**
 * ORDER BYをAutoSelectで利用できる形式で作成.
 *
 * @param order ORDER BY条件
 * @return ORDER BY
 */
private OrderByItem[] makeOrderBy(List<OrderDto> order) {
    List<OrderByItem> rv = new ArrayList<OrderByItem>();

    for (OrderDto that : order) {
        if (that.orderType == OrderType.ASC) {
            rv.add(Operations.asc(that.fieldName));
        } else {
            rv.add(Operations.desc(that.fieldName));
        }
    }
}

```



```

    }

    return rv.toArray(new OrderByItem[rv.size()]);
}

/**
 * ORDER BYをSqlFileSelectで利用できる形式で作成.
 *
 * @param order ORDER BY条件
 * @return ORDER BY
 */
private String makeOrderByString(List<OrderDto> order) {
    List<String> rv = new ArrayList<String>();

    for (OrderDto that : order) {
        if (that.orderType == OrderType.ASC) {
            rv.add(Operations.asc(StringUtil.camelToSnake(that.fieldName)).getCriteria());
        } else {
            rv.add(Operations.desc(StringUtil.camelToSnake(that.fieldName)).getCriteria());
        }
    }

    if (rv.size() > 0) {
        return StringUtil.join(rv.toArray(new String[rv.size()]), ",");
    }
    return null;
}

/**
 * generalSelect用のAtutoSelectを返却.
 *
 * @param criteria 検索条件
 * @param order ORDER BY
 * @return AtutoSelect
 */
public AutoSelect<ENTITY> getAutoSelect(Map<String, Object> criteria, List<OrderDto> order) {
    AutoSelect<ENTITY> autoSelect = select().where(criteria);
    OrderByItem[] orderby = makeOrderBy(order);

    if (orderby.length > 0) {
        autoSelect = autoSelect.orderBy(orderby);
    }
    return autoSelect;
}

/**
 * 汎用検索.
 *
 * @param criteria 検索条件
 * @return 検索結果
 */
public List<ENTITY> genericSelectList(Map<String, Object> criteria) {
    return genericSelectList(criteria, new ArrayList<OrderDto>());
}

/**
 * 汎用検索.
 *
 * @param criteria 検索条件
 * @param order ORDER BY
 * @return 検索結果
 */
public List<ENTITY> genericSelectList(Map<String, Object> criteria, List<OrderDto> order) {
    List<ENTITY> ret = getAutoSelect(criteria, order).getResultList();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * 汎用検索(ロック).
 *
 * @param criteria 検索条件
 * @param order ORDER BY
 * @return 検索結果
 */
public List<ENTITY> genericSelectList4Update(Map<String, Object> criteria, List<OrderDto> order) {
    List<ENTITY> ret = getAutoSelect(criteria, order).forUpdate().getResultList();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * generalSelect用のAtutoSelectを返却.
 *
 * @param where 検索条件
 * @param order ORDER BY
 * @return AtutoSelect
 */

```

```

    private AutoSelect<ENTITY> getAutoSelect(SimpleWhere where, List<OrderDto> order) {
        AutoSelect<ENTITY> autoSelect = select().where(where);
        OrderByItem[] orderby = makeOrderBy(order);

        if (orderby.length > 0) {
            autoSelect = autoSelect.orderBy(orderby);
        }
        return autoSelect;
    }

    /**
     * 汎用検索.
     *
     * @param where 検索条件
     * @return 検索結果
     */
    public List<ENTITY> genericSelectList(SimpleWhere where) {
        return genericSelectList(where, new ArrayList<OrderDto>());
    }

    /**
     * 汎用検索.
     *
     * @param where 検索条件
     * @param order ORDER BY
     * @return 検索結果
     */
    public List<ENTITY> genericSelectList(SimpleWhere where, List<OrderDto> order) {
        List<ENTITY> ret = getAutoSelect(where, order).getResultList();
        KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
        return ret;
    }

    /**
     * 汎用検索(ロック).
     *
     * @param where 検索条件
     * @param order ORDER BY
     * @return 検索結果
     */
    public List<ENTITY> genericSelectList4Update(SimpleWhere where, List<OrderDto> order) {
        List<ENTITY> ret = getAutoSelect(where, order).forUpdate().getResultList();
        KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
        return ret;
    }

    /**
     * generalSelectAll用のAtutoSelectを返却.
     *
     * @param order ORDER BY
     * @return AtutoSelect
     */
    private AutoSelect<ENTITY> getAutoSelect(List<OrderDto> order) {
        AutoSelect<ENTITY> autoSelect = select();
        OrderByItem[] orderby = makeOrderBy(order);

        if (orderby.length > 0) {
            autoSelect = autoSelect.orderBy(orderby);
        }
        return autoSelect;
    }

    /**
     * 汎用検索(全検索).
     *
     * @param order ORDER BY
     * @return 検索結果
     */
    public List<ENTITY> genericSelectAll(List<OrderDto> order) {
        List<ENTITY> ret = getAutoSelect(order).getResultList();
        KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
        return ret;
    }

    /**
     * 汎用検索(全検索)(ロック).
     *
     * @param order ORDER BY
     * @return 検索結果
     */
    public List<ENTITY> genericSelectAll4Update(List<OrderDto> order) {
        List<ENTITY> ret = getAutoSelect(order).forUpdate().getResultList();
        KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
        return ret;
    }
}

```

```

/**
 * SQL ファイルを利用して検索条件から、limitとoffsetを除外。
 * <p>
 * S2JDBCにてlimitやoffsetは予約であり、SQL検索条件にあると自動的にlimitやoffsetを利用してページングをするため、意図しないページングが行われないよう無害化します
 * </p>
 *
 * @param criteria 検索条件
 * @return 無害化した検索条件
 */
private Map<String, Object> sanitizeCriteria4SqlFile(Map<String, Object> criteria) {
    if (criteria != null) {
        if (criteria.containsKey("limit")) {
            criteria.remove("limit");
        }
        if (criteria.containsKey("offset")) {
            criteria.remove("offset");
        }
    }
    return criteria;
}

/**
 * SqlFileSelectを返却。
 *
 * @param sqlfile sqlファイル名
 * @param criteria 検索条件
 * @param resultClass 検索結果のクラス
 * @param order ORDER BY
 * @param <T> 検索結果のクラス
 * @return SqlFileSelect
 */
public <T> SqlFileSelect<T> getSqlFileSelect(Class<T> resultClass, String sqlfile, Map<String, Object> criteria,
    List<OrderDto> order) {
    String orderBy = makeOrderByString(order);
    if (StringUtil.isEmpty(orderby)) {
        criteria.put("orderBy", orderby);
    }

    Map<String, String> args = getSqlArguments("/" + super.sqlFilePathPrefix + sqlfile);
    for (Map.Entry<String, String> e : args.entrySet()) {
        if (!criteria.containsKey(e.getKey())) {
            criteria.put(e.getKey(), null);
        }
    }

    return selectBySqlFile(resultClass, sqlfile, sanitizeCriteria4SqlFile(criteria));
}

/**
 * SQL ファイルを解析し、SQLファイルで指定されている変数(検索条件)を返却。
 *
 * @param sqlPath SQLファイル
 * @return 引数リスト(valueにはnullが設定されています。キーのみ利用してください)
 */
private Map<String, String> getSqlArguments(String sqlPath) {

    StringBuffer sql = new StringBuffer();
    String sqlPathDialect = null;
    if (super.jdbcManager instanceof JdbcManagerImplementor) {
        DbmsDialect dialect = ((JdbcManagerImplementor) super.jdbcManager).getDialect();
        if ((StringUtil.isEmpty(sqlPath)) && (dialect != null) && (StringUtil.isEmpty(dialect.getName()))) {
            int posDot = sqlPath.lastIndexOf(".");
            if (posDot > 0) {
                sqlPathDialect = sqlPath.substring(0, posDot) + "_" + dialect.getName() + sqlPath.substring(posDot);
            } else {
                sqlPathDialect = sqlPath + "_" + dialect.getName();
            }
        }
    }
    URL sqlFile = null;
    if (StringUtil.isEmpty(sqlPathDialect)) {
        sqlFile = GenerateService.class.getResource(sqlPathDialect);
    }
    if (null == sqlFile) {
        sqlFile = GenerateService.class.getResource(sqlPath);
    }
    InputStream is = null;
    BufferedReader br = null;
    try {
        is = sqlFile.openStream();
        br = new BufferedReader(new InputStreamReader(is, SQL_READ_CHARSET_STR));
        while (br.ready()) {
            sql.append(br.readLine());
        }
    }

```

```

    } catch (IOException e) {
        throw KaidenRuntimeException.makeInstance(e);
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                throw KaidenRuntimeException.makeInstance(e);
            }
        }
    }
    if (is != null) {
        try {
            is.close();
        } catch (IOException e) {
            throw KaidenRuntimeException.makeInstance(e);
        }
    }
}

Map<String, String> argMap = new HashMap<String, String>();
SqlParser parser = new SqlParserImpl(sql.toString());
Node sqlNode = parser.parse();
sqlNodeParse(sqlNode, argMap);
return argMap;
}

/**
 * SQLを解析する.
 *
 * @param node SQL解析Node
 * @param argMap 変数MAP ※当メソッド内で追加します。
 */
private void sqlNodeParse(Node node, Map<String, String> argMap) {

    if (BindVariableNode.class.isInstance(node)) {
        argMap.put(((BindVariableNode) node).getExpression(), null);
    } else if (ParenBindVariableNode.class.isInstance(node)) {
        argMap.put(((ParenBindVariableNode) node).getExpression(), null);
    } else if (EmbeddedValueNode.class.isInstance(node)) {
        argMap.put(((EmbeddedValueNode) node).getExpression(), null);
    }

    for (int i = 0; i < node.getChildSize(); i++) {
        sqlNodeParse(node.getChild(i), argMap);
    }
}

/**
 * SQLファイル検索.
 * <p>
 * Like条件の%等はエスケープされます
 * </p>
 *
 * @param sqlfile sqlファイル名
 * @param criteria 検索条件
 * @param resultClass 検索結果のクラス
 * @param order ORDER BY
 * @param <T> 検索結果のクラス
 * @return 検索結果
 */
public <T> List<T> selectBySql(Class<T> resultClass, String sqlfile, Map<String, Object> criteria,
    List<OrderDto> order) {
    List<T> ret = getSqlFileSelect(resultClass, sqlfile, criteria, order).getResultList();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * SQLファイル検索.
 * <p>
 * Like条件の%等はエスケープされます
 * </p>
 *
 * @param sqlfile sqlファイル名
 * @param criteria 検索条件
 * @return 検索結果
 */
public List<BeanMap> selectBySql(String sqlfile, Map<String, Object> criteria) {
    return selectBySql(BeanMap.class, sqlfile, criteria, new ArrayList<OrderDto>());
}

/**
 * SQLファイル検索.
 * <p>
 * Like条件の%等はエスケープされます
 * </p>
 *

```

```

* @param sqlfile sqlファイル名
* @param criteria 検索条件
* @param order ORDER BY
* @return 検索結果
*/
public List<BeanMap> selectBySql(String sqlfile, Map<String, Object> criteria, List<OrderDto> order) {
    return selectBySql(BeanMap.class, sqlfile, criteria, order);
}

/**
* SQLファイル検索後、Filterクラスによって内容をフィルタリングし返却。
* <p>
* 引数「limit」が0より大きい場合は以下のように動作します。
* <p>
* <ul>
* <li>返却データは「limit」の数までです</li>
* <li>「numberOfPage」が指定されている場合にはそのページ番号のデータのみを返却します。</li>
* </ul>
*
* @param sqlfile sqlファイル名
* @param criteria SQL検索条件 ※Like条件の%等はエスケープされます
* @param resultClass 検索結果のクラス
* @param order ORDER BY
* @param filterClasses フィルタリングクラス
* @param limit フィルタリング結果の返却最大数
* @param numberOfPage フィルタリング結果の返却ページ
* @param <T> 検索結果(リスト)のクラス
* @return 検索結果
*/
@SuppressWarnings("unchecked")
public <T> Map<String, Object> selectBySqlFilter(Class<T> resultClass, String sqlfile,
    Map<String, Object> criteria, List<OrderDto> order, SqlResultFilterIF[] filterClasses, long limit,
    long numberOfPage) {

    if (criteria.containsKey("limit")) {
        criteria.remove("limit");
    }
    numberOfPage = numberOfPage <= 1 ? 1L : numberOfPage;

    FilterIterationCallBack filterIterationCallBack = new FilterIterationCallBack(filterClasses, limit,
        numberOfPage);
    Map<String, Object> ret = getSqlFileSelect(resultClass, sqlfile, criteria, order).iterate(
        (IterationCallback<T, Map<String, Object>>) filterIterationCallBack);
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力

    // 結果なし
    if (ret == null) {
        return this.selectBySqlFilterNoDataResult();
    }

    if (limit > 0) {
        Long matchCount = (Long) ret.get("matchCount");
        long pageTtl = (matchCount / limit) + ((matchCount % limit) > 0 ? 1 : 0);

        if (matchCount >= (limit * (numberOfPage - 1) + 1)) {
            // 指定ページが存在する
            ret.put("resultPage", numberOfPage);
        } else {
            // 指定ページが存在しないため、最大ページ番号で再検索
            FilterIterationCallBack reFilterIterationCallBack = new FilterIterationCallBack(filterClasses, limit,
                pageTtl);
            ret = getSqlFileSelect(resultClass, sqlfile, criteria, order).iterate(
                (IterationCallback<T, Map<String, Object>>) reFilterIterationCallBack);
            ret.put("resultPage", Long.valueOf(pageTtl));
        }
    }

    matchCount = (Long) ret.get("matchCount");
    pageTtl = (matchCount / limit) + ((matchCount % limit) > 0 ? 1 : 0);

    // 返却ページ総数の計算
    if (ret.get("matchCount") != null) {
        ret.put("resultPageTotal", Long.valueOf(pageTtl));
    }
    else {
        // limitが0以下なら全件返却のため、総ページ数、ページ番号とも1を返却
        ret.put("resultPageTotal", Long.valueOf(1L));
        ret.put("resultPage", Long.valueOf(1L));
    }

    return ret;
}

/**
* SQLファイル検索後、Filterクラスによって内容をフィルタリングし返却。
* <p>

```

```

* {@link #selectBySqlFilter(Class, String, Map, List, SqlResultFilterIF[], long, long)}
* と同様の機能で、戻り値を特定のDTOでなくBeanMapで返却します。
* </p>
*
* @param sqlfile sqlファイル名
* @param criteria SQL検索条件 ※Like条件の%等はエスケープされます
* @param order ORDER BY
* @param filterClasses フィルタリングクラス
* @param limit フィルタリング結果の返却最大数
* @param numberOfPage フィルタリング結果の返却ページ
* @param <T> 検索結果 (リスト) のクラス
* @return 検索結果
* @return
*/

public <T> Map<String, Object> selectBySqlFilter(String sqlfile, Map<String, Object> criteria,
    List<OrderDto> order, SqlResultFilterIF[] filterClasses, long limit, long numberOfPage) {
    return selectBySqlFilter(BeanMap.class, sqlfile, criteria, order, filterClasses, limit, numberOfPage);
}

/**
* 検索後、Filterクラスによって内容をフィルタリングし返却。
*
* @param resultClass 検索結果のクラス
* @param criteria SQL検索条件 ※Like条件の%等はエスケープされます
* @param order ORDER BY
* @param filterClasses フィルタリングクラス
* @param limit フィルタリング結果の返却最大数
* @param numberOfPage フィルタリング結果の返却ページ
* @param <T> 検索結果 (リスト) のクラス
* @return 検索結果
*/

@SuppressWarnings("unchecked")
public <T> Map<String, Object> selectListByFilter(Class<T> resultClass, Map<String, Object> criteria,
    List<OrderDto> order, SqlResultFilterIF[] filterClasses, long limit, long numberOfPage) {

    if (criteria.containsKey("limit")) {
        criteria.remove("limit");
    }
    numberOfPage = numberOfPage <= 1 ? 1L : numberOfPage;

    FilterIterationCallBack filterIterationCallBack = new FilterIterationCallBack(filterClasses, limit,
        numberOfPage);

    Map<String, Object> ret = getAutoSelect(criteria, order).iterate(
        (IterationCallBack<ENTITY, Map<String, Object>>) filterIterationCallBack);
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力

    // 結果なし
    if (ret == null) {
        return this.selectBySqlFilterNoDataResult();
    }

    if (limit > 0) {
        Long matchCount = (Long) ret.get("matchCount");
        long pageTtl = (matchCount / limit) + ((matchCount % limit) > 0 ? 1 : 0);

        if (matchCount >= (limit * (numberOfPage - 1) + 1)) {
            // 指定ページが存在する
            ret.put("resultPage", numberOfPage);
        } else {
            // 指定ページが存在しないため、最大ページ番号で再検索
            FilterIterationCallBack reFilterIterationCallBack = new FilterIterationCallBack(filterClasses, limit,
                pageTtl);
            ret = getAutoSelect(criteria, order).iterate(
                (IterationCallBack<ENTITY, Map<String, Object>>) reFilterIterationCallBack);
            ret.put("resultPage", Long.valueOf(pageTtl));
        }
    }

    matchCount = (Long) ret.get("matchCount");
    pageTtl = (matchCount / limit) + ((matchCount % limit) > 0 ? 1 : 0);

    // 返却ページ総数の計算
    if (ret.get("matchCount") != null) {
        ret.put("resultPageTotal", Long.valueOf(pageTtl));
    }
} else {
    // limitが0以下なら全件返却のため、総ページ数、ページ番号とも1を返却
    ret.put("resultPageTotal", Long.valueOf(1L));
    ret.put("resultPage", Long.valueOf(1L));
}

List<ENTITY> resultList = (List<ENTITY>) ret.get("resultList");
if (CollectionUtil.isEmpty(resultList)) {
    ret.put("resultList", CollectionUtil.changeDtoListClass(resultClass, resultList));
}

```

```

    return ret;
}

/**
 * 検索条件で、Like検索のワイルドカードをエスケープします。
 * <p>
 * % → $%
 * </p>
 *
 * @param criteria 検索条件
 * @param likeTarget LIKE検索対象カラム
 */
public void escapeWildcard(Map<String, Object> criteria, String[] likeTarget) {

    if (likeTarget == null || likeTarget.length == 0) {
        return;
    }

    for (String that : likeTarget) {
        if (criteria.containsKey(that)) {
            Object value = criteria.get(that);

            if (value instanceof String) {
                criteria.put(that, LikeUtil.escapeWildcard((String) value));
            }
        }
    }
}

/**
 * {@inheritDoc}
 * <p>
 * {@link AutoFieldSetter#autoTimestampByUpdate(Object)}, {@link AutoFieldSetter#autoUpdateuserByUpdate(Object)}
 * を実行後に、<code>super.update(entity)</code>を実行します
 * </p>
 */
@Override
public int update(ENTITY entity) {
    AutoFieldSetter.autoTimestampByUpdate(entity);
    AutoFieldSetter.autoUpdateuserByUpdate(entity);
    int ret = super.update(entity);
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * エンティティを更新。
 * <p>
 * 指定のプロパティを更新対象から除外します。<br>
 * {@link AutoFieldSetter#autoTimestampByUpdate(Object)}, {@link AutoFieldSetter#autoUpdateuserByUpdate(Object)}
 * を実行後に、<br>
 * <code>super.jdbcManager.update(entity).excludes(propertyNames).execute()</code>を実行します
 * </p>
 *
 * @param entity 更新対象エンティティ
 * @param propertyNames 更新対象から除外するプロパティ名の配列
 * @return 更新件数
 */
public int updateExcludes(ENTITY entity, String[] propertyNames) {
    AutoFieldSetter.autoTimestampByUpdate(entity);
    AutoFieldSetter.autoUpdateuserByUpdate(entity);
    int ret = super.jdbcManager.update(entity).excludes(this.adjustExcludesPropertyNames(propertyNames)).execute();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * エンティティを更新。
 * <p>
 * 指定のプロパティのみを更新対象とします。<br>
 * {@link AutoFieldSetter#autoTimestampByUpdate(Object)}, {@link AutoFieldSetter#autoUpdateuserByUpdate(Object)}
 * を実行後に、<br>
 * <code>super.jdbcManager.update(entity).includes(propertyNames).execute()</code>を実行します
 * </p>
 *
 * @param entity 更新対象エンティティ
 * @param propertyNames 更新対象とするプロパティ名の配列
 * @return 更新件数
 */
public int updateIncludes(ENTITY entity, String[] propertyNames) {
    AutoFieldSetter.autoTimestampByUpdate(entity);
    AutoFieldSetter.autoUpdateuserByUpdate(entity);
    int ret = super.jdbcManager.update(entity).includes(adjustIncludesPropertyNames(propertyNames)).execute();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

```

```

}

/**
 * {@inheritDoc}
 * <p>
 * {@link AutoFieldSetter#autoTimestampByInsert(Object)}, {@link AutoFieldSetter#autoUpdateuserByInsert(Object)}
 * を実行後に、<code>super.insert(entity)</code>を実行します
 * </p>
 */
@Override
public int insert(ENTITY entity) {
    AutoFieldSetter.autoTimestampByInsert(entity);
    AutoFieldSetter.autoUpdateuserByInsert(entity);
    int ret = super.insert(entity);
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
 * 一括挿入.
 *
 * @param dataList 挿入対象リスト
 * @return 挿入件数の配列 ※S2AbstractService#insert(ENTITY)の仕様準拠
 */
public int[] insertBatch(List<ENTITY> dataList) {
    List<Integer> rvList = new ArrayList<Integer>();
    for (ENTITY entity : dataList) {
        rvList.add(this.insert(entity));
    }

    return CollectionUtil.toIntArray(rvList);
}

/**
 * 一括更新.
 * <p>
 * 指定のプロパティを更新対象から除外します。<br>
 * </p>
 *
 * @param dataList 更新対象リスト
 * @return 更新件数(Ingeger)の配列
 */
public int[] updateBatch(List<ENTITY> dataList) {
    List<Integer> rvList = new ArrayList<Integer>();
    for (ENTITY entity : dataList) {
        rvList.add(this.update(entity));
    }

    return CollectionUtil.toIntArray(rvList);
}

/**
 * 一括更新.
 * <p>
 * 指定のプロパティのみを更新対象とします。<br>
 * </p>
 *
 * @param dataList 更新対象リスト
 * @param propertyNames 更新対象から除外するプロパティ名の配列
 * @return 更新件数(Ingeger)の配列
 */
public int[] updateExcludesBatch(List<ENTITY> dataList, String[] propertyNames) {
    List<Integer> rvList = new ArrayList<Integer>();
    for (ENTITY entity : dataList) {
        rvList.add(this.updateExcludes(entity, propertyNames));
    }

    return CollectionUtil.toIntArray(rvList);
}

/**
 * 一括更新.
 *
 * @param dataList 更新対象リスト
 * @param propertyNames 更新対象とするプロパティ名の配列
 * @return 更新件数(Ingeger)の配列
 */
public int[] updateIncludesBatch(List<ENTITY> dataList, String[] propertyNames) {
    List<Integer> rvList = new ArrayList<Integer>();
    for (ENTITY entity : dataList) {
        rvList.add(this.updateIncludes(entity, propertyNames));
    }

    return CollectionUtil.toIntArray(rvList);
}

```



```

/**
 * 一括削除.
 *
 * @param dataList 更新対象リスト
 * @return 削除数(Integer)の配列
 */
public int[] deleteBatch(List<ENTITY> dataList) {
    List<Integer> rvList = new ArrayList<Integer>();
    for (ENTITY entity : dataList) {
        rvList.add(delete(entity));
        KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    }
    return CollectionUtil.toIntArray(rvList);
}

/**
 * 内容に応じて挿入/更新を切り替えて実行.
 * <p>
 * エンティティのrenewCntがnullの場合はinsert、そうでない場合はupdateを行います
 * </p>
 *
 * @param data 更新対象
 * @return 挿入/更新件数.
 */
public int autoInsUpd(ENTITY data) {
    if (((GenerateEntity) data).renewCnt == null) {
        return this.insert(data);
    } else {
        return this.update(data);
    }
}

/**
 * 内容に応じて挿入/更新を一括実行.
 * <p>
 * エンティティのrenewCntがnullの場合はinsert、そうでない場合はupdateを行います
 * </p>
 * <p>
 * エラーが発生した場合はnullを返却
 * </p>
 *
 * @param dataList 更新対象リスト
 * @return 挿入/更新件数.
 */
public int[] autoInsUpdBatch(List<ENTITY> dataList) {
    List<Integer> rvList = new ArrayList<Integer>();
    genericLock(dataList);
    for (ENTITY data : dataList) {
        rvList.add(this.autoInsUpd(data));
    }
    return CollectionUtil.toIntArray(rvList);
}

/**
 * 汎用ロック.
 * <p>
 * 注意&#65281;引数のリストを主キーの昇順にソートしますので、注意してください
 * </p>
 *
 * @param dataList 対象エンティティのリスト
 */
public void genericLock(List<ENTITY> dataList) {
    // PKでソート
    List<String> pk = this.getPkFiled();
    List<OrderDto> order = new ArrayList<OrderDto>();
    for (String fld : pk) {
        OrderDto that = new OrderDto();
        that.fieldName = fld;
        that.orderType = OrderType.ASC;
        order.add(that);
    }
    CollectionUtil.dtoSort(dataList, order);

    // PKキー順にロック
    for (ENTITY that : dataList) {
        this.genericSelectList4Update(getPkCriteria(that), new ArrayList<OrderDto>());
    }
}

/**
 * 主キーの昇順でソート定義を生成.
 *
 * @return ソート定義(主キーの昇順)
 */
public List<OrderDto> makeOrderListFromPk() {
    List<String> pk = this.getPkFiled();

```

```

        List<String> pk = ums.getPKField();
        List<OrderDto> order = new ArrayList<OrderDto>();
        for (String fld : pk) {
            OrderDto that = new OrderDto();
            that.fieldName = fld;
            that.orderType = OrderType.ASC;
            order.add(that);
        }
        return order;
    }

/**
 * getSelectBySqlFilter時の返却値なしの状態のMapを返却.
 * <p>
 * 主にSQL実行前に戻り値を返却する場合に利用してください
 * </p>
 *
 * @return 返却値なしの状態のMap
 */
public Map<String, Object> getSelectBySqlFilterNoDataResult() {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put("resultList", new ArrayList<Object>()); // フィルター結果(データ) ※最大でもlimit数まで
    result.put("matchCount", Long.valueOf(0L)); // フィルターにマッチした数 ※limit以上もカウント
    result.put("sqlTotal", Long.valueOf(0L)); // フィルター前のカウント
    result.put("resultPageTotal", Long.valueOf(0L)); // 返却ページtotal
    result.put("resultPage", Long.valueOf(0L)); // 返却ページ番号
    return result;
}

/**
 * selectListFilter用コールバック.
 * <p>
 * 返却値 (MAP)には以下の情報が格納されています。
 * <table border="1">
 * <tr>
 * <th>key</th>
 * <th>value</th>
 * </tr>
 * <tr>
 * <td>resultList</td>
 * <td>List<Object>; フィルター結果(指定ページのデータのみ)</td>
 * </tr>
 * <tr>
 * <td>matchCount</td>
 * <td>Long フィルター一致数</td>
 * </tr>
 * <tr>
 * <td>sqlTotal</td>
 * <td>Long SQLヒット数</td>
 * </tr>
 * <tr>
 * <td>resultPageTotal</td>
 * <td>Long 返却ページ総数</td>
 * </tr>
 * <tr>
 * <td>resultPage</td>
 * <td>Long 返却ページ番号</td>
 * </tr>
 * </table>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private class FilterIterationCallBack implements IterationCallback<Object, Map<String, Object>> {

    /** フィルタークラス. */
    private final SqlResultFilterIF[] filters;
    /** 返却結果の最大数. */
    private final long limit;
    /** 返却ページ番号. */
    private final long numberOfPage;

    /** 戻り値. */
    private final Map<String, Object> result = new HashMap<String, Object>();

    /**
     * コンストラクタ.
     *
     * @param filters フィルタークラス配列
     * @param limit 返却最大件数
     * @param numberOfPage 返却ページ番号
     */
    public FilterIterationCallBack(SqlResultFilterIF[] filters, long limit, long numberOfPage) {
        this.filters = filters;
        this.limit = limit;
        this.numberOfPage = numberOfPage <= 1 ? Long.valueOf(1L) : numberOfPage;
        // 戻り値の準備
        result.put("resultList", new ArrayList<Object>()); // フィルター結果(データ) ※最大でもlimit数まで
    }

```

```

result.put("resultList", new ArrayList<Object>()); // フィルター 結果リストの要素をeliminate
result.put("matchCount", Long.valueOf(0L)); // フィルターにマッチした数 ※limit以上もカウント
result.put("sqlTotal", Long.valueOf(0L)); // フィルター前のカウント
result.put("resultPageTotal", Long.valueOf(0L)); // 返却ページtotal
result.put("resultPage", Long.valueOf(0L)); // 返却ページ番号
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> iterate(Object selectEntity, IterationContext ctx) {
    boolean match = true;
    for (SqlResultFilterIF sqlFilter : this.filters) {
        match = sqlFilter.filter(selectEntity) && match;
        if (!match) {
            break;
        }
    }

    // インクリメント
    this.result.put("sqlTotal", ((Long) this.result.get("sqlTotal")) + 1);
    if (match) {
        Long matchCount = ((Long) this.result.get("matchCount")) + 1;
        this.result.put("matchCount", (matchCount));

        if (limit <= 0
            || (matchCount <= (limit * numberOfPage) && matchCount >= (limit * (numberOfPage - 1) + 1))) {
            @SuppressWarnings("unchecked")
            List<Object> lst = (List<Object>) this.result.get("resultList");
            lst.add(selectEntity);
        }
    }
    return this.result;
}

/**
 * 更新除外対象プロパティ配列補正.
 * <p>
 * 更新除外対象プロパティ配列に下記プロパティを含む場合、当該要素を削除します。
 * </p>
 * <li>{@link EntityCommonFieldNames.RENEW_USER_CD 更新者コード}</li> <br>
 * <li>{@link EntityCommonFieldNames.TS 更新日時}</li>
 *
 * @param arg 更新除外対象プロパティ
 * @return 補正後プロパティ配列
 */
private String[] adjustExcludesPropertyNames(String[] arg) {
    List<String> list = new ArrayList<String>(Arrays.asList(arg));
    int targetIdx = list.lastIndexOf(EntityCommonFieldNames.RENEW_USER_CD);
    if (targetIdx != -1) {
        list.remove(targetIdx);
    }
    targetIdx = list.lastIndexOf(EntityCommonFieldNames.RENEW_TS);
    if (targetIdx != -1) {
        list.remove(targetIdx);
    }
    return list.toArray(new String[list.size()]);
}

/**
 * 更新対象プロパティ配列補正.
 * <p>
 * 更新対象プロパティ配列に下記プロパティを含まない場合、当該要素を追加します。
 * </p>
 * <li>{@link EntityCommonFieldNames.RENEW_USER_CD 更新者コード}</li> <br>
 * <li>{@link EntityCommonFieldNames.TS 更新日時}</li>
 *
 * @param arg 更新対象プロパティ
 * @return 補正後プロパティ配列
 */
private String[] adjustIncludesPropertyNames(String[] arg) {
    List<String> list = new ArrayList<String>(Arrays.asList(arg));
    int targetIdx = list.lastIndexOf(EntityCommonFieldNames.RENEW_USER_CD);
    if (targetIdx == -1) {
        list.add(EntityCommonFieldNames.RENEW_USER_CD);
    }
    targetIdx = list.lastIndexOf(EntityCommonFieldNames.RENEW_TS);
    if (targetIdx == -1) {
        list.add(EntityCommonFieldNames.RENEW_TS);
    }
    return list.toArray(new String[list.size()]);
}

/**
 * SQLファイル更新.

```

```

*
* @param sqlfile sqlファイル名
* @return 更新件数
*/
public int updateBySql(String sqlfile) {
    return updateBySql(sqlfile, null);
}

/**
* SQLファイル更新.
*
* @param sqlfile sqlファイル名
* @param criteria 更新条件
* @return 更新件数
*/
public int updateBySql(String sqlfile, Map<String, Object> criteria) {
    int ret = getSqlFileUpdate(sqlfile, criteria).execute();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
* SqlFileUpdateを返却.
*
* @param sqlfile sqlファイル名
* @param criteria 検索条件
* @return SqlFileUpdate
*/
public SqlFileUpdate getSqlFileUpdate(String sqlfile, Map<String, Object> criteria) {

    Map<String, String> args = getSqlArguments("/") + super.sqlFilePathPrefix + sqlfile);
    for (Map.Entry<String, String> e : args.entrySet()) {
        if (!criteria.containsKey(e.getKey())) {
            criteria.put(e.getKey(), null);
        }
    }

    return updateBySqlFile(sqlfile, criteria);
}

/**
* 汎用検索結果件数取得.
*
* @return 検索結果件数
*/
@Override
public long getCount() {
    return getCount(new HashMap<String, Object>());
}

/**
* 汎用検索結果件数取得.
*
* @param criteria 検索条件
* @return 検索結果件数
*/
public long getCount(Map<String, Object> criteria) {
    long ret = select().where(criteria).getCount();
    KaidenLogger.debug(kaidenSqlLogRegistry.getLast().getCompleteSql()); // SQL文をログに出力
    return ret;
}

/**
* Sqlファイルの検索結果件数を返却.
*
* @param sqlfile sqlファイル名
* @return Sqlファイルの検索結果件数
* @see #getCountBySql(String, Map)
*/
public long getCountBySql(String sqlfile) {
    return getCountBySql(sqlfile, null);
}

/**
* Sqlファイルの検索結果件数を返却.
*
* @param sqlfile sqlファイル名
* @param criteria 検索条件
* @return Sqlファイルの検索結果件数
*/
public long getCountBySql(String sqlfile, Map<String, Object> criteria) {

    if (null == criteria || criteria.isEmpty()) {
        return jdbcManager.getCountBySqlFile(sqlFilePathPrefix + sqlfile);
    }
}

```

```

Map<String, String> args = getSqlArguments("'" + super.sqlFilePathPrefix + sqlfile);

for (Map.Entry<String, String> e : args.entrySet()) {
    if (!criteria.containsKey(e.getKey())) {
        criteria.put(e.getKey(), null);
    }
}

return jdbcManager.getCountBySqlFile(sqlFilePathPrefix + sqlfile, criteria);
}

/**
 * JdbcManagerを設定.
 *
 * @param jdbcManager JdbcManager
 */
public void setJdbcManager(JdbcManager jdbcManager) {
    this.jdbcManager = jdbcManager;
    // @formatter:off
    KaidenLogger.info(getClass().getSimpleName() + "に jdbcManagerが設定されました。[" + ((JdbcManagerImpl) jdbcManager).getDialect().getName() + "]");
    // @formatter:on
}
}

```

AutoTimestamp

- Path
src/main/java/jp/co/slcs/kaiden2/base/foundation/model/annotation/AutoTimestamp.java

```

package jp.co.slcs.kaiden2.base.foundation.model.annotation;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;

/**
 * 自動現在時刻設定.
 * <p>
 * S2JDBCのエンティティにアノテーションを付与することで、insert時/update時に自動的に現在日付時刻を設定します。<br>
 * ただし自動的に設定されるのは、{@link GenerateService}の{@link GenerateService#insert(Object) insert}/
 * {@link GenerateService#update(Object) update}を利用した場合のみです。
 * <p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
public @interface AutoTimestamp {

    /** insert時に現在日付時刻を設定. */
    boolean insert();

    /** update時に現在日付時刻を設定. */
    boolean update();
}

```

AutoUpdateUser

- Path
src/main/java/jp/co/slcs/kaiden2/base/foundation/model/annotation/AutoUpdateUser.java

```

package jp.co.slcs.kaiden2.base.foundation.model.annotation;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;

/**
 * 自動ログインユーザ設定.
 * <p>
 * S2JDBCのエンティティにアノテーションを付与することで、insert時/update時に自動的にログインユーザを設定します。<br>
 * ただし自動的に設定されるのは、{@link GenerateService}の{@link GenerateService#insert(Object) insert}/
 * {@link GenerateService#update(Object) update}を利用した場合のみです。
 * <p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
public @interface AutoUpdateUser {

    /** insert時にログインユーザを設定. */
    boolean insert();

    /** update時にログインユーザを設定. */
    boolean update();
}

```

[s2jdbc-gen-build.xml/s2jdbc-gen-build.properties](#)

- 次ファイルは参考としてお使いください。
ご利用中の環境によっては、動作しない場合がありますので、ご注意ください。
- **s2jdbc-gen-build.xml**

```

<project name="example-s2jdbc-gen" default="generate-src" basedir=".">
  <property file="s2jdbc-gen-build.properties" />

  <path id="classpath">
    <pathelement location="${classpathdir}" />
    <pathelement location="${resourcesdir}" />
    <fileset dir="${librarydir}" />
  </path>
  <taskdef resource="s2jdbc-gen-task.properties" classpathref="classpath"/>
  <property name="templateFilePrimaryDir" value="resources/tempaltes"/>
  <property name="pluralFormFile" value="${templateFilePrimaryDir}/plural.properties"/>

  <target name="generate-src">
    <gen-entity
      tableNamePattern="${tableNamePattern}"
      ignoreTableNamePattern="${ignoreTableNamePattern}"
      entitySuperclassName="jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity"
      pluralFormFile="${pluralFormFile}"
      useTemporalType="false"
      useAccessor="false"
      applyDbCommentToJava="true"
      showTableName="true"
      showColumnName="true"
      showColumnDefinition="true"
      showJoinColumn="true"
      rootPackageName="${rootPackageName}"
      entityPackageName="entity"
      templateFilePrimaryDir="${templateFilePrimaryDir}"
      overwrite="true"
      classpathref="classpath"
    />
    <javac
      srcdir="${srcdir}"
      destdir="${classpathdir}"
      encoding="UTF-8"
      fork="true"
      debug="on"
      includeantruntime=""
      classpathref="classpath"
    />
    <gen-names
      classpathDir="${classpathdir}"
      rootPackageName="${rootPackageName}"
      namesPackageName="entity"
      templateFilePrimaryDir="${templateFilePrimaryDir}"
      generateNamesAggregateClass="false"
      overwrite="true"
      classpathref="classpath"
    />
    <gen-service
      classpathDir="${classpathdir}"
      rootPackageName="${rootPackageName}"
      servicePackageName="service"
      templateFilePrimaryDir="${templateFilePrimaryDir}"
      overwrite="true"
      classpathref="classpath"
    />
    <gen-condition
      classpathDir="${classpathdir}"
      rootPackageName="${rootPackageName}"
      conditionPackageName="service"
      templateFilePrimaryDir="${templateFilePrimaryDir}"
      overwrite="true"
      classpathref="classpath"
    />
    <javac
      srcdir="${srcdir}"
      destdir="${classpathdir}"
      encoding="UTF-8"
      fork="true"
      debug="on"
      includeantruntime=""
      classpathref="classpath"
    />
  </target>
</project>

```

- s2jdbc-gen-build.properties

```
# classpath
srcdir=src/main/java
classpathdir=target/classes
resourcesdir=resources
librarydir=lib

# 対象とするテーブル名の正規表現です。
# (デフォルト値:".*")
tableNamePattern=.*

# 対象としないテーブル名の正規表現です。
# (デフォルト値:"^(SCHEMA_INFO|.*$).*")
ignoreTableNamePattern=

# ルートパッケージ名です。
# エンティティクラスは、ルートパッケージ名と「entity」をピリオドで連結したパッケージに配置します。
# 名前クラスは、ルートパッケージ名と「entity」をピリオドで連結したパッケージに配置します。
# サービスクラスは、ルートパッケージ名と「service」をピリオドで連結したパッケージに配置します。
# 条件クラスは、ルートパッケージ名と「service」をピリオドで連結したパッケージに配置します。
rootPackageName=aaa.bbb.ccc.foundation.model
```


初めてのプログラミング

intra-mart Accel Kaiden!は、SAStruts+S2JDBCの開発モデルで構築しています。

本項では、intra-mart Accel Kaiden!での開発の基本として、SAStrutsフレームワークを使用したプログラミングの流れを体験します。

本項の内容は『SAStruts+S2JDBC プログラミングガイド』と同じような内容となりますが、
後続の解説で本項で作成したソースを基にintra-mart Accel Kaiden!のプログラミングを体験してください。

- [プログラミング](#)
 - [入力画面の作成](#)
 - [出力画面の作成](#)
 - [Formの作成](#)
 - [Dtoの作成](#)
 - [Actionの作成](#)
- [動作確認](#)
 - [メニュー設定](#)
 - [画面表示](#)

プログラミング

SAStrutsフレームワークを使用し、Hello Worldを表示するプログラムを作成します。



注意

本項で作成したソースはすべて文字コードを **UTF-8** にして保存してください。

入力画面の作成

入力画面 (index.jsp) を作成します。

事前準備で作成したモジュール・プロジェクトの src/main/webapp に、
/WEB-INF/view/sample フォルダを作成し、作成したフォルダに **index.jsp** という名前で次のソースを実装します。

```
<%@page pageEncoding="UTF-8"%>
<%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui"%>
<%@ taglib prefix="s" uri="http://sastruts.seasar.org" %>
<%@ taglib prefix="f" uri="http://sastruts.seasar.org/functions" %>

<!-- HEADタグ -->
<imui:head>
  <title>Hello, World</title>
  <script type="text/javascript">
    $(function() {
      $('#button').click(function() {
        $('#helloForm').submit();
      });
    });
  </script>
</imui:head>

<!-- 画面上に表示されるタイトル -->
<div class="imui-title">
  <h1>Hello, World (SAStruts)</h1>
</div>

<!-- 入力画面 -->
<div class="imui-form-container-narrow">
  <p>
    <label for="name">Please input the name. </label>
  </p>
  <!-- 入力フォームの設定
       actionに結果表示画面へのパスを入力(Helloアクションのoutputメソッドを呼び出す) -->
  <s:form action="/sample/hello/output" styleId="helloForm">
    <div>
      <!-- テキストボックス -->
      <imui:textbox type="text" value="" id="name" name="name" size="10" />
    </div>
    <!-- submitボタン -->
    <imui:button name="button" value="Hello!!" class="mt-10" id="button"></imui:button>
  </s:form>
</div>
```

出力画面の作成

出力画面 (output.jsp) を作成します。

事前準備で作成したモジュール・プロジェクトの src/main/webapp/WEB-INF/view/sample に

output.jsp という名前で次のソースを実装します。

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui"%>
<%@ taglib prefix="s" uri="http://sastruts.seasar.org" %>
<%@ taglib prefix="f" uri="http://sastruts.seasar.org/functions" %>

<!-- HEADタグ -->
<imui:head>
  <title>Hello, World </title>
  <script type="text/javascript">
    function back() {
      $('#back-form').submit();
    };
  </script>
</imui:head>

<!-- 画面上に表示されるタイトル -->
<div class="imui-title">
  <h1>Hello, World (SAStruts)</h1>
</div>

<!-- ツールバー(前の画面へと戻るボタンを配置) -->
<div class="imui-toolbar-wrap">
  <div class="imui-toolbar-inner">
    <ul class="imui-list-toolbar">
      <li>
        <!-- 「戻る」ボタンのアイコン、HelloActionのindexメソッドが呼び出される。 -->
        <a href=${f:url("/sample/hello")} class="imui-toolbar-icon" title="back">
          <span class="im-ui-icon-common-16-back"></span>
        </a>
      </li>
    </ul>
  </div>
</div>

<!-- 出力結果 -->
<div class="imui-form-container-narrow">
  <label>
    <imui:textbox type="text" value="${f:h(helloDto.outputString)}" id="name" name="name" size="10" class="imui-text-readonly" readonly />
  </label>
</div>
```

Formの作成

入力情報を保持するFormクラスを作成します。

事前準備で作成したモジュール・プロジェクトの src/main/java に
パッケージ名は jp.co.slcs.kaiden2.tutorial.feature.form.sample 、クラス名は HelloForm として次のソースを実装します。

```
package jp.co.slcs.kaiden2.tutorial.feature.form.sample;

/**
 * index.jspで入力された情報を保持するFormクラスです。
 */
public class HelloForm {

    public String name; // テキストエリアに入力された値

}
```

Dtoの作成

出力情報を保持するDtoクラスを作成します。

事前準備で作成したモジュール・プロジェクトの src/main/java に
パッケージ名は jp.co.slcs.kaiden2.tutorial.feature.dto.sample、クラス名は HelloDto として次のソースを実装します。

```
package jp.co.slcs.kaiden2.tutorial.feature.dto.sample;

/**
 * 結果表示画面に渡すDTOクラス
 */
public class HelloDto {

    public String outputString; // 結果を保持する文字列

}
```

Actionの作成

入力画面、および出力画面に対する処置を行うActionクラスを作成します。

事前準備で作成したモジュール・プロジェクトの src/main/java に

パッケージ名は jp.co.slcs.kaiden2.tutorial.feature.action.sample、クラス名は HelloAction として次のソースを実装します。

```
package jp.co.slcs.kaiden2.tutorial.feature.action.sample;

import javax.annotation.Resource;
import javax.servlet.http.HttpServletRequest;

import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;

import jp.co.slcs.kaiden2.tutorial.feature.dto.sample.HelloDto;
import jp.co.slcs.kaiden2.tutorial.feature.form.sample.HelloForm;

/**
 * Hello Worldの処理を行うActionクラス
 */
public class HelloAction {

    @Resource
    @ActionForm
    public HelloForm helloForm; // 入力情報を保持するFormクラス

    @Resource
    public HelloDto helloDto; // 出力情報を保持するDTOクラス(publicにすること)

    /**
     * 入力ページのパスを返却します。
     * @return 入力ページのパス
     */
    @Execute validator = false
    public String index() {
        return "/sample/index.jsp";
    }

    /**
     * 結果表示画面のパスを返却します。
     * @return 結果表示画面のパス
     */
    @Execute validator = false
    public String output() {
        helloDto.outputString = "Hello, " + helloForm.name; // 結果の情報を格納
        return "/sample/output.jsp";
    }
}
```

動作確認

作成したプログラムの動作確認を行います。

メニュー設定や認可設定などの詳細は『[SAStruts+S2JDBC プログラミングガイド](#)』を参照してください。

メニュー設定

作成したプログラムをサイトマップに登録します。

Resinを起動し、テナント管理者としてintra-mart Accel Platformにログインします。

ログイン後、メニュー設定画面を表示し、メニューを次の様に設定します。

メニューアイテム名(日本語)	Hello World for SA
URL	sample/hello

メニューを閲覧できるようにするため、作成したメニューに認可を設定します。

画面表示

設定したメニュー使用し、画面表示を行います。

サイトマップから登録したメニューをクリックすると入力画面が表示されます。

入力画面テキストボックスに名前を入力し「Hello!!ボタン」をクリックすると、出力画面が表示され、入力内容が表示されます。

出力画面の上部にある「戻る」アイコンを押下すると、入力画面へと戻ります。

- 入力画面

- 出力画面



仕様解説

本項では、intra-mart Accel Kaiden!の各種仕様を解説します。

Gadget

本項では、intra-mart Accel Kaiden!におけるガジェットの仕様を解説します。

- [ガジェットとは？](#)
- [ガジェットの識別子](#)
- [ガジェットの構造](#)
- [ガジェットのプログラミング](#)
- [ガジェット間通信とは？](#)
- [ガジェット間通信のイベント](#)
- [ガジェット間通信のプログラミング](#)

ガジェットとは？

intra-mart Accel Kaiden!におけるガジェットとは、特定の機能を部品化したものを指します。
intra-mart Accel Kaiden!の多くの画面では、次の様にガジェットを組み合わせで構成しています。

申請書マスタメンテナンスの様に、ユーザが独自に組合せを変更できる機能も用意しています。

- 申請画面（旅費精算申請）
申請書マスタメンテナンスの設定により、ガジェットの組合せを変更することができます。

旅費精算申請

申請

一時保存

申請書コピー

2013/08/27

申請会社

サンプル会社

申請組織

申請者(代理申請者)

青柳 辰巳

出張情報

出張期間

出張目的

出張先

出張先住所

備考

精算情報

案件番号

案件名

仮払残高

精算予定額

精算対象の事前申請書を追加

精算手段

銀行振込

精算内訳

費用合計

会社負担額合計

精算予定額合計

精算額

0

0

0

0

旅程明細

日付

利用区間

補足

経費区分

勘定科目

税区分

部門

プロジェクト

支払分類

支払先

片通金額

金額

経費検索連携

宿泊明細

宿泊期間

宿泊先

宿泊先Tel

経費区分

勘定科目

税区分

部門

プロジェクト

支払分類

支払先

金額

日当明細

日当種別

単価

日数

金額

経費区分

勘定科目

税区分

部門

プロジェクト

経費明細

発生日

経費内容

経費区分

勘定科目

税区分

部門

プロジェクト

支払分類

支払先

金額

申請

一時保存

- 登録／更新画面（マスタメンテナンス）（プロジェクトマスタ）
ガジェットの組合せはプログラムで固定となります。

プロジェクトマスタメンテナンス

期間一覧

開始日

終了日

1900/01/01

2999/12/31

会社

サンプル会社

プロジェクトコード

000001

プロジェクト名

日本語

情報システム株式会社ワークフローシステム再構築案

英語

D Company Workflow System Restructuring Project

中国語(中華人民共和国)

信息系统公司 工作流系统重新构筑项目

上位プロジェクト

更新

削除

一覧に戻る

- 勤務表画面
ガジェットの組合せはプログラムで固定となります。

34

intra-mart Top Workflow テナント管理 サンプル サイトマップ 青柳辰巳

勤務表 - 青柳辰巳 (100001)

提出 2014/03/21 - 2014/04/20

点検ステータス - 未提出

通常(一般)

日付	勤務状況	勤務時間	時間外	申請ステータス	摘要	caution
3/21 (金)						
22 (土)						
23 (日)						
24 (月)	出勤	09:00 - 18:00 (08:00)	1:00	承認済み (実績)		
25 (火)	出勤	09:00 - 17:00 (07:00)	-	差戻し (実績)		
26 (水)	出勤	09:00 - 19:00 (09:00)	2:00	申請中 (実績)		
27 (木)	出勤	09:00 - 17:00 (07:00)	-	申請中 (予定)		
28 (金)	(未入力)					
29 (土)						
30 (日)						
31 (月)	有給休暇	-		承認済み (予定)		
4/1 (火)	(未入力)					
2 (水)	(未入力)					
3 (木)	(未入力)					

- 勤務入力画面
ガジェットの色合わせはプログラムで固定となります。

2014/03/28(金) 青柳辰巳

保存 予定申請 実績申請

勤務入力一括登録

勤務状況 出勤 点検ステータス 申請ステータス caution 客観時間

IN OUT

勤務時間 09:00 - 17:00 時間外 0:00 摘要

休憩・控除・時間外休を追加

09:00

12:00 - 13:00 : 休憩

17:00

予定労働時間	労働時間(内:深夜)	休憩時間	控除	控除免除	時間外	時間外
予定労働日	法定休日	予定休日			法定内	法定外
7:00	7:00(-)	-(-)	-(-)	1:00	-	-

即時更新

- 勤務表PDF
ガジェットの色合わせはプログラムで固定となります。

勤務表 出力日 2014.05/15

青柳辰巳(100001)
2014.03.21 - 2014.04.20

サンプル会社 サンプル課 1 1

通常(一般)

日付	勤務状況	勤務時間	時間外	摘要
3/21 (金)	振替出勤(所定)	09:00 - 17:00 (07:00)		
22 (土)				
23 (日)				
24 (月)	出勤	09:00 - 18:00 (08:00)	1:00	
25 (火)	出勤	09:00 - 17:00 (08:00)	1:00	
26 (水)	出勤	09:00 - 19:00 (09:00)	2:00	
27 (木)	出勤	09:00 - 17:00 (07:00)		
28 (金)	出勤	09:00 - 17:00 (07:00)		
29 (土)				
30 (日)				
31 (月)	有給休暇			
4/1 (火)	出勤	09:00 - 17:00 (07:00)		
2 (水)	出勤	09:00 - 17:00 (07:00)		
3 (木)	出勤	09:00 - 17:00 (07:00)		
4 (金)	出勤	09:00 - 17:00 (07:00)		
5 (土)				
6 (日)				
7 (月)	出勤	09:00 - 17:00 (07:00)		
8 (火)	出勤	09:00 - 17:00 (07:00)		
9 (水)	出勤	09:00 - 17:00 (07:00)		
10 (木)	出勤	09:00 - 17:00 (07:00)		
11 (金)	出勤	09:00 - 17:00 (07:00)		
12 (土)				
13 (日)				

ガジェットの識別子

ガジェットは次の様な識別子で構成されています。

識別子	説明
ガジェットクラス	ガジェットの種別を表します。
ガジェットバリエーション	同一ガジェット種別で画面レイアウト違いを表します。
ガジェットインスタンス	特定のガジェットを同時に複数利用する場合に使用します。 ガジェットクラス、ガジェットバリエーションでガジェットを特定することが可能ですが、申請書などで、特定のガジェットを同時に複数利用する可能性があるため連番を設けています。 同時複数利用を行わない画面では、原則「0」となり、同時複数利用を行う画面では、「0」からの連番となります。
ガジェットID	ガジェットクラス、ガジェットバリエーション、ガジェットインスタンスの3つをアンダースコア(_)で繋げた文字列です。 ガジェットIDがガジェットを一意に示す値として使用します。

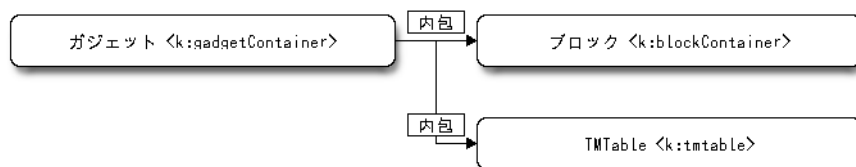
ガジェットの構造

ガジェットは次の様な構造となります。

ガジェットは1つのJSP内に複数のガジェットを定義することが可能です。

ガジェット `<k:gadgetContainer>` に内包する要素として次の2種類のブロックがあります。

- ブロック `<k:blockContainer>`
- `TMTable <k:ttable>`



i コラム

ガジェット内には一つだけ`<form>`を配置することができます。
バリデーションを行うためには、バリデーション対象の要素が`<form>`内に配置されている必要があります。

i コラム

ガジェット内には最低ひとつのブロック `<k:blockContainer>`、または `TMTable <k:ttable>` を配置してください。
テキストボックス `<k:text>` などの要素は、必ずブロック `<k:blockContainer>`、または `TMTable <k:ttable>` 内に配置してください。

! 注意

次の様な構造は実装できませんので、ご注意ください。

- ブロック `<k:blockContainer>` が `TMTable <k:ttable>` を内包
- `TMTable <k:ttable>` が ブロック `<k:blockContainer>` を内包

ガジェットのプログラミング

ガジェットのプログラミングに必要な機能はタグライブラリとして用意しております。

本項では、タグライブラリを使用して簡単なガジェットを作成する方法を解説します。

- プログラミング例
次のソースは、一つのテキストボックスを持つガジェットとなります。


```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>

<c:set var="gadgetClass">sampleGadget</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">0</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>

<k:gadgetContainer
  gadgetClass="${gadgetClass}"
  gadgetVariation="${gadgetVariation}"
  gadgetInstance="${gadgetInstance}"
  gadgetTitle="ガジェットタイトル"
  gadgetId="${gadgetId}">

  <k:blockContainer id="inputArea">
    <k:text name="textbox" mode="0" />
  </k:blockContainer>
</k:gadgetContainer>

```

■ 解説

<c:set ///>の部分は、ガジェットの識別子をそれぞれ変数に設定しています。
特にガジェットIDは、JavaScriptのコーディングでは重要な役割を担いますので、
"\${gadgetId}"のコーディングだけで済むよう変数に設定すること推奨します。

📘 コラム

ガジェットの初期化処理で、ガジェット内のname(id)属性はガジェットやブロックを考慮した値に変更します。
変更前の値は「data-kaiden-org-id」「data-kaiden-org-name」に保持していますが、
変更前の値でアクセスできるようJavaScriptライブラリ([Accessor](#))を用意していますので、
プログラミングの際に変更後の値を意識する必要はありません。

⚠️ 注意

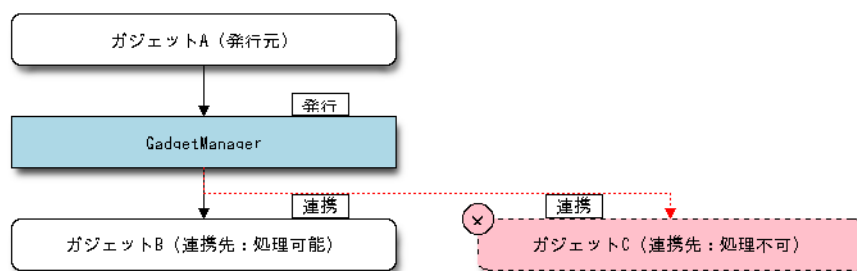
エレメントのname属性およびid属性は、[ガジェット <k:gadgetContainer>](#)内で一意な値を設定してください。

[ブロック <k:blockContainer>](#)や[TMTable <k:tmtable>](#)内で一意であれば問題はありませんが、
メンテナンス性などを考慮し、[ガジェット <k:gadgetContainer>](#)内で一意な値を設定するようにしてください。

ガジェット間通信とは？

ガジェット間通信とは、ガジェットとガジェットの間にイベントを発行し、処理やデータなどを連携する仕組みです。

ガジェット間通信はGadgetManagerを仲介させてイベントのやり取りを行います。
GadgetManagerを仲介させることで、ガジェット間の依存をなくすメリットがあります。



図の様に、ガジェットAから発行されたイベントは、GadgetManagerを通じガジェットBやガジェットCに連携します。
連携したイベントの処理は受け側のガジェットに委任されます。

たとえば、図の様にガジェットBはイベントを処理する用意があるので、そのままガジェットBでは処理を実行します。
ガジェットCはイベントを処理する用意がないため、発行されたイベントを無視します。(エラーにもなりません)

ガジェット間通信のイベント

ガジェット間通信は、次の2種類のイベントがあります。

- **PUSHイベント**
他のガジェットに何かを通知したり、処理を実行させるためのイベントです。
- **PULLイベント**
他のガジェットにデータなどを要求するイベントです。
イベントを処理する側のガジェットでは、何らかの値を返却する必要があります。

📘 コラム

ガジェット間通信のイベントは2種類ありますが、厳密な違いはありません。
 PUSHイベントで値を返却することも可能ですが、メンテナンス性向上の観点から次の様な分類を推奨します。

- PUSHイベント: 返却値なし
- PULLイベント: 返却値あり(処理結果や処理ガジェットが保有するデータを返却)

ガジェット間通信のプログラミング

ガジェット間通信のプログラミングは、次の様に行います。

イベントの発行

他のガジェットへイベントを発行する場合は、次の様にプログラミングします。

```
// PUSHイベントを発行
KAIDEN.gadgetMan.execPush("pushイベントキー", "引数", "呼び出し元ガジェットID");

// PULL イベントを発行
var ret = KAIDEN.gadgetMan.execPull("pullイベントキー", "引数", "呼び出し元ガジェットID");
```

イベントの受信

他のガジェットからのイベントを受信する場合は、次の様にプログラミングします。

“arg”にはイベント発行時の引数が渡されます。

```
// EventManagerの取得
var eventMan = KAIDEN.getGadget("ガジェットID").eventManager;

// PUSHイベントの受信
eventMan.registPushListener("pushイベントキー", function(arg) {
  // イベント処理
});

// PULL イベントの受信
eventMan.registPullListener("pullイベントキー", function(arg) {
  // イベント処理
  return "戻り値";
});
```

イベントの戻り値

他のガジェットへ発行したイベントは、戻り値を返却する場合があります。
 返却された戻り値は、複数のガジェットの処理結果を格納するため、配列として返却します。
 返却された戻り値の配列には、イベントの戻り値と処理を行ったガジェットの情報を返却します。

```
// PULL イベントを発行
KAIDEN.gadgetMan.execPush("pushイベントキー", "引数", "呼び出し元ガジェットID");

// PULL イベントを発行
var ret = KAIDEN.gadgetMan.execPull("pullイベントキー", "引数", "呼び出し元ガジェットID");
```



コラム

上記の場合、retには以下の情報が配列で格納されています。

- result ※処理自体の戻り値が格納されています。
- gadgetAttr.gadgetId
- gadgetAttr.gadgetClass
- gadgetAttr.gadgetVariation
- gadgetAttr.gadgetInstance

TMTable

前項の *Gadget* ではガジェットの仕様を解説しました。

本項では、ガジェットの内包要素であるTMTableの仕様を解説します。

- TMTableとは?
- TMTableのプログラミング
- タプルとは?
- タプル操作のプログラミング
- TMTable操作パネル
- TMTableの関連API
 - タプルへのデータ設定
 - タプルのデータ取得
 - タプルのイベント処理

TMTTableとは？

TMTableとは、いわゆる帳票形式で表現する場合に利用するブロックです。
動的に入力欄を増やしたり、一覧表示を行う際に使用します。

次の画面では、TMTTableを利用して機能を実現しています。

- #### ■ 明細ガジェット(経費明細)

経費明細				
☐	発生日 *	経費内容 *		
☐	経費区分 * 勘定科目 税区分	部門	プロジェクト	
☐	支払分類 *	支払先	金額 *	
☐ 1				
☐				
	立替			

- 一覧画面(マスタメンテナンス)(プロジェクトマスタ)

プロジェクトマネジメント

新規登録

検索

検索オプション
 検索基準日: 2015/01/01
 会社: サンプル会社
 ロケール: 日本語
 ☐ ステータスが無効のデータも検索

No	▲ プロジェクトコード	プロジェクト名	上位プロジェクト名	ステータス
1	000001	情報システム株式会社 ワークフローシステム再構築プロジェクト		●
2	000002	社内専任研修プロジェクト		●
3	000003	<<社内向才育成>>経費申請・承認ワークフローシステム(SAPフロント)導入		●
4	000004	△社 システム保守(2013年度)		●
5	000005	システム更新対応(社内)		●
6	000006	○社 ワークフローシステム保守		●
7	000007	H25年度事務運用(問合せ案)		●
8	000008	■社 システム購置様式支援		●
9	000009	新製品構築プロジェクト		●
10	000010	品質保証プロジェクト		●

[最初へ前へ](#) |
 1 |
 [次へ最後へ](#)

TMTableのプログラミング

TMTTableは、タグライブラリの *TMTTable* <code><tmtable></code>を使用します。

TMTTable <code><tmtable></code>は、<code><table></code>に変換されますので、基本的なプログラミング方法は<code><table></code>と同様ですが、<code><thead></code>と<code><tbody></code>を明確に分けて作成してください。

- プログラミング例

```
<k:tmtable id="sampleTmtable" class="imui-form">
  <thead>
    <tr>
      <th><k:surface surfaceKey="入力欄1" requiredMark="true"/></th>
    </tr>
    <tr>
      <th><k:surface surfaceKey="入力欄2" requiredMark="true"/></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><k:text name="inputArea1" mode="0"/></td>
    </tr>
    <tr>
      <td><k:text name="inputArea2" mode="0"/></td>
    </tr>
  </tbody>
</k:tmtable>
```

- ## ■ 解説

<thead>
ヘッダ行の扱いになりますので、特に何も変化しません。

<tbody>

動的に増減する対象で、複数行（<tr>が複数）あってもそれが1セットの行として増減します。

たとえば、プログラミング例の様にJSPにてあらかじめ2行（2つの<tr>）を作成しておく、この2行単位で動的に増減されます。

タプルとは？

タプル、またはタプルIDは次の様に定義しています。

- **タプル**
前項の通り、<tbody>にあらじめ<tr>2つある場合は、この2行単位で動的に増減していきます。TMTableでは、この2行の単位をタプルと定義しています。
- **タプルID**

タブルIDは、タブルに対して付与される一意な連番です。

タブル削除がされると欠番となりますし、「上に挿入」などを行った場合には画面表示順とタブルIDの順序は一致しくなくなります。
TMTTableの操作は、このタブルIDを利用して操作します。

タブル操作のプログラミング

JavaScriptにてタブル操作を行う場合のプログラミング方法を解説します。

次の様な実装で、タブルに対する各種操作が行えます。

■ 最下行に追加

```
// TMTTableの取り出し
var tmttable = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"];
// 最下行に1行(1タブル)追加
tmttable.append();
```

■ 下に挿入

```
// TMTTableの取り出し
var tmttable = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"];
// 指定したタブルIDの下にタブルを挿入
tmttable.insertUnder("タブルID");
```

■ 上に挿入

```
// TMTTableの取り出し
var tmttable = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"];
// 指定したタブルIDの上にタブルを挿入
tmttable.insertUpper("タブルID");
```

■ タブルを1つ削除

```
// TMTTableの取り出し
var tmttable = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"];
// 指定したタブルを削除
tmttable.del("タブルID");
```

■ タブルをすべて削除

```
// TMTTableの取り出し
var tmttable = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"];
// タブルをすべて削除
tmttable.deleteAll();
```



コラム

タブル操作の詳細は、『intra-mart Accel Kaiden! JsDoc』を参照してください。

TMTTable操作パネル

ユーザ操作で行追加や行削除などを行う場合は *TMTTable操作パネル* `<k:tmttableOperationPanel>` を使用してください。

次の様に *TMTTable操作パネル* `<k:tmttableOperationPanel>` を配置するだけで、
行追加・行削除などのタブルに対する基本的な操作が行えるようになります。

■ プログラミング例

```
<k:tmttable id="sampleTmttable" class="imui-form">
  <thead>
    <tr>
      <td>
        <k:tmttableOperationPanel position="header" allowAdd="true" allowDel="true" />
      </td>
      <th><k:surface surfaceKey="入力欄1" requiredMark="true"/></th>
      <th><k:surface surfaceKey="入力欄2" requiredMark="true"/></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>
        <k:tmttableOperationPanel position="detail" allowAdd="true" allowDel="true" />
      </td>
      <td><k:text name="inputArea1" mode="0"/></td>
      <td><k:text name="inputArea2" mode="0"/></td>
    </tr>
  </tbody>
</k:tmttable>
```

本項では、TMTableのタブルの操作やデータの設定・取得の方法を紹介します。
各APIの詳細は『[intra-mart Accel Kaiden! JsDoc](#)』を参照してください。

タブルへのデータ設定

タブル内のエレメント(テキストボックスなど)に値を設定する場合には、次のAPIを使用してください。

- **append(data, exclude)**
最下行にタブルを追加し、そのタブルに対しdataを設定
- **insertUpper(tupleId, data, exclude)**
指定タブルの上にタブルを挿入、そのタブルに対しdataを設定
- **insertUnder(tupleId, data, exclude)**
指定タブルの下にタブルを挿入、そのタブルに対しdataを設定
- **setTupleData(tupleId, data, exclude)**
指定タブルに対しdataを設定
- **replaceAll(datas, exclude)**
全行削除後、datas (配列) の分だけ行追加しデータを設定します ※タブルIDを復元する
- **replaceAllNewTupleId(datas, exclude)**
全行削除後、datas (配列) の分だけ行追加しデータを設定します ※タブルIDを復元せずに新たなタブルIDを付与
- **appendAllNewTupleId(datas, exclude)**
datas (配列) の分だけ行追加しデータを設定します ※タブルIDを復元せずに新たなタブルIDを付与

i コラム

次のソースは、TMTableに行を追加し、text1に"txt1"をtext2に"txt2"を設定する例です。
excludeにtext3が指定されているため、text3は値が設定されません。

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
//タブルを追加するとともにデータを設定
tmtable.append({text1:"txt1", text2:"txt2", text3:"txt3"}, {text3:true});
```

タブルのデータ取得

タブルのデータを取得する場合には、次のAPIを使用してください。

- **getTupleData(tupleId, includeLabel)**
特定のタブルのデータをjavascriptオブジェクトの形式で返却
- **getCount()**
Tmtableの行数を返却
- **getAvailabilityTupleIds()**
現在有効なタブルIDを配列で返却
- **getDataSet()**
タブルデータの配列をjavascriptオブジェクトの形式で返却
- **getColumnValues(fieldName)**
タブルのうち特定のフィールドキー (もしくはフィールドキーの配列) の入力データをすべて返却

タブルのイベント処理

タブルのイベント処理は次の様に実装してください。

- **タブル追加(挿入)直前**

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
//タブル追加(挿入)直前イベント
tmtable.addEventPreNewRow(function(tupleId, tmtableAccessor) {
  //引数:tupleId      :挿入起点となるタブルID
  //引数:tmtableAccessor:当該TMtableへのAccessor
});
```

- **タブル追加(挿入)直後(データ設定前)**

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
//タブル追加(挿入)直後(データ設定前)イベント
tmtable.addEventPostNewRowBeforeSetData(function(tupleId, tmtableAccessor, data) {
  //引数:tupleId      :新たに挿入されたタブルID
  //引数:tmtableAccessor:当該TMtableへのAccessor
  //引数:data         :挿入されたタブルに設定しようとしている値
});
```

- **タブル追加(挿入)直後(データ設定後)**

```
// TTableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTTableID"];
//タブル追加(挿入)直後(データ設定後)イベント
tmtable.addEventPostNewRow(function(tupleId, tmtableAccessor, data) {
  //引数:tupleId      :新たに挿入されたタブルID
  //引数:tmtableAccessor:当該TTableへのAccessor
  //引数:data         :挿入されたタブルに設定された値
});
```

- **タブル削除時**

```
// TTableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTTableID"];
//タブル削除時イベント
tmtable.addEventDecreaseRowFunc(function() {
});
```

Accessor

本項では、intra-mart Accel Kaiden!のJavaScriptライブラリ(Accessor)の仕様を解説します。

- [Accessorとは?](#)
- [Accessorの取得](#)
- [各要素へのアクセス](#)
- [各要素の主な操作](#)
- [イベントの追加](#)
- [隣接要素へのアクセス](#)

Accessorとは？

intra-mart Accel Kaiden!のガジェット(ブロック・TMTTable)では、ガジェットの初期化処理でエレメントのname(id)属性をガジェットやブロックを考慮した値に変更します。そのため、JSPで id="txt"と設定したテキストボックスが \$("#txt") では取得できません。

Accessorとは、変更前のname(id)属性でアクセスできるようにするJavaScriptライブラリです。上記に加え、値の取得や設定などよく使う機能を用意しています。



コラム

Accessorの詳細は、『[intra-mart Accel Kaiden! JsDoc](#)』を参照してください。

Accessorの取得

ガジェットが初期化された後に、次の様にAccessorを取得できます。

```
// ブロックに対するAccessor
var blockAccessor = KAIDEN.getGadget("ガジェットID").blockAccessors["ブロックID"];
// TMTTableに対するAccessor
var tmttableAccessor = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"].tmttableAccessor;
```



コラム

ガジェットの初期化は、</k:gadgetContainer>の直後に行われます。そのため、ガジェットに対する操作は、</k:gadgetContainer>の後続に<script>にて記載します。

```
<k:gadgetContainer
  gadgetClass="{gadgetClass}"
  gadgetVariation="{gadgetVariation}"
  gadgetInstance="{gadgetInstance}"
  gadgetTitle="ガジェットタイトル"
  gadgetId="{gadgetId}"
>
  <k:blockContainer id="inputArea">
    <k:text name="txt" mode="0" />
  </k:blockContainer>
</k:gadgetContainer>

<script type="text/javascript">
  jQuery(function($) {
    //この時点ではガジェットが初期化されてますので、各種操作が可能です。
  })
</script>
```

各要素へのアクセス

各要素がブロック内にあるのか、TMTTable内にあるのかにより引数の違いはありますが、idでアクセスする場合にはjQueryと同様に先頭に"#"を付与します。name属性の場合には"[name=]"という装飾をせずにそのまま指定できます。

各ブロックへのAccessorを利用することで、例えばガジェットが異なるがname属性やid属性が同じものが存在した場合でも、確実に該当ガジェットの要素へアクセスできます。

■ ガジェットブロックの場合

```
// ブロックに対するAccessor
var blockAccessor = KAIDEN.getGadget("ガジェットID").blockAccessors["ブロックID"];

var findName = blockAccessor("name属性"); //name属性でのアクセス
var findId = blockAccessor("#id属性"); //id属性でのアクセス
```

■ TMTTableの場合

```
// TMTTableに対するAccessor
var tmttableAccessor = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"].tmttableAccessor;

var findName = tmttableAccessor("name属性", "タブID"); //name属性でのアクセス
var findId = tmttableAccessor("#id属性", "タブID"); //id属性でのアクセス
```

各要素の主な操作

Accessorを用いてよく利用する操作を解説します。

■ 値を設定する／取得する

```
//値を設定
blockAccessor("name属性").setValue("設定する値");

//値を取得
blockAccessor("name属性").getValue();
```

- スタイルシート(class)を適用する／適用を外す

```
//classを適用する
blockAccessor("name属性").addClass("適用するclass");

//classの適用を外す
blockAccessor("name属性").removeClass("適用を外すclass");
```

- 無効にする／無効を解除する

```
//無効にする
blockAccessor("name属性").disabled();

//無効を解除する
blockAccessor("name属性").removeDisabled();
```

イベントの追加

Accessorのメソッドelem()を利用するとjQueryオブジェクトそのものが返却されますので、jQueryを通じて様々な操作が行えます。

次の例では、エレメントの"on"イベントを追加しています。

- “on(change)”イベントの追加

```
// ブロックに対するAccessor
var blockAccessor = KAIDEN.getGadget("ガジェットID").blockAccessors["ブロックID"];

blockAccessor("name属性").elem().on("change", function() {
  //changeイベントの処理
});
```

Accessorのメソッドelem()は、その時点でエレメントが存在しない、または複数存在する場合にエラーが発生しますが、elem(true)とすると、エラーの発生を抑止することができます。そのため、“live”イベントを付与する場合などに便利に使用できます。

- “live(change)”イベントの追加

```
// TMTTableに対するAccessor
var tmttableAccessor = KAIDEN.getGadget("ガジェットID").tmttables["TMTTableID"].tmttableAccessor;

tmttableAccessor("name属性").elem(true).live("change", function() {
  //changeイベント
});
```



コラム

TMTTableはタブルが動的に増減するため、タブルが存在しない(0件)場合があります。そのため、elem(true)としてjQueryオブジェクトを取り出す必要があります。



コラム

通常、TMTTableに対するAccessorでは、タブルIDを指定して対象エレメントを特定しますが、ここではあえてタブルIDを指定せずに「tmttableAccessor("name属性")」とすることで、全てのタブル内エレメントを対象とすることができます。

隣接要素へのアクセス

同一ブロック、またはTMTTableの同一タブル内の別要素を隣接要素と呼びます。

たとえば、下記のようにTMTTable内にテキストボックスが2つある場合、“text1”と“text2”が隣接要素となります。


```
<k:tmtable id="sampleTmtable" class="imui-form">
  <thead>
    <tr>
      <th><k:surface surfaceKey="入力欄1" requiredMark="true"/></th>
      <th><k:surface surfaceKey="入力欄2" requiredMark="true"/></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><k:text name="text1" mode="0"/></td>
      <td><k:text name="text2" mode="0"/></td>
    </tr>
  </tbody>
</k:tmtable>
```

"text1"の値が変更された場合に、"text2"へ値をコピーする様な実装が必要な場合は、隣接要素へのAccessorを使用してください。
隣接要素へのAccessorは、「KAIDEN.gadgetMan.getNeighboringAccessor」にて取得します。

```
// TMTTableに対するAccessor
var tmtableAccessor = KAIDEN.getGadget("ガジェットID").tmtables["sampleTmtable"].tmtableAccessor;
tmtableAccessor("text1").elem(true).live("change", function() {
  KAIDEN.gadgetMan.getNeighboringAccessor(this, "text2").setValue($(this).val());
});
```



コラム

隣接要素へのAccessorは、ブロック、TMTTableのどちらでも使用可能です。

隣接要素へのAccessor (NeighboringAccessor) の詳細は、『[intra-mart Accel Kaiden! JsDoc](#)』を参照してください。

Ajax

intra-mart Accel Kaiden!では簡単にAjax通信が行えるような機構を用意しています。
本項では、intra-mart Accel Kaiden!を使用した基本的なAjax通信の方法を解説します。

- 基本的なプログラミング方法
- Ajax通信の応用

基本的なプログラミング方法

簡単な例を用い、Ajax通信の基本的なプログラミング方法を解説します。

Actionクラス

「jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto」をJSONに変換したデータを返却してください。

```
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.base.foundation.conf.LogicResultStatus;
import net.arx.jsonic.JSON;

@Execute(validator = false)
public String index() {

    /**～略～**/
    LogicResultDto result = new LogicResultDto();
    result.data = "クライアントへの返却データ";
    result.status = LogicResultStatus.SUCCESS.getName(); //成功ステータス
    result.message = "処理が成功しました。";
    ResponseWriter.writeJson(JSON.encode(result));
    return null;
}
```

JSP (JavaScript)

```
var ajax = KAIDEN.Ajax(); //Ajaxライブラリの呼び出し
var criteria = {}; //サーバーに送信する値

//Ajax実行 (POST通信)
ajax.postAuto("/URL/")
, criteria
, function(returnData) {
    //正常時の処理
}
, function(returnData) {
    //エラー時の処理
}
, function(returnData) {
    //ワーニング時の処理
}
);
```

JSP側で、Ajax通信が終了すると次の様な処理を行います。

1. メッセージが設定されている場合は、LogicResultDto.statusの内容に応じて自動的にメッセージを表示します。

“success”の場合、imuiShowSuccessMessage()を利用して成功メッセージを表示
 “fail”の場合、imuiShowErrorMessage()を利用してエラーメッセージを表示
 “warn”の場合、imuiShowWarningMessage()を利用してワーニングメッセージを表示

2. LogicResultDto.statusの内容に応じて、それぞれメソッドを呼び出します。
 この時、引数には LogicResultDto.dataで設定した値が渡されます。



コラム

ajax.postAuto()の、正常時・エラー時・ワーニング時の処理が不要で、サーバーに値を送信する必要がなければ、
 ajax.postAuto("/URL/"); だけで実行できます。



コラム

GET通信を行う場合には、ajax.getAuto()で同様の処理が行えます。

Ajax通信の応用

同期通信やGET/POSTの切り替えなどのプログラミング方法は『intra-mart Accel Kaiden! JsDoc』のKAIDEN.base.foundation.Ajaxを参照してください。

メッセージ表示やその他の処理を細かく制御したい場合には、次のメソッドを使用してください。
 (どちらのメソッドも\$.ajax()を返却しますので、処理を細かく制御することができます。)

- KAIDEN.base.foundation.Ajax#post (String url, Object data, Function success) : XMLHttpRequest
- KAIDEN.base.foundation.Ajax#get (String url, Object data, Function success) : XMLHttpRequest

Mode&Formatted

intra-mart Accel Kaiden!ではテキストボックスなどのモードやフォーマットを簡単に切り替える機構を用意しています。
 本項では、モードやフォーマットの仕様を解説します。

- モードとは？
- フォーマットとは？

モードとは？

intra-mart Accel Kaiden! のタグライブラリでは、mode属性に設定した値によってモードを変更することができます。

モードはタグライブラリのmode属性に設定した値により自動的に設定され、mode属性に"0"を設定した場合は入力モードとなり、"1"を設定した場合は表示モードとなります。

主に申請書の入力画面（申請画面）と処理画面（承認画面）で使用している機能で、入力画面ではmode属性に"0"を、処理画面ではmode属性に"1"を設定し、一つのJSPで入力画面と処理画面を実現しています。

- モードが指定可能なタグライブラリ

タグライブラリ	入力モードの状態 mode="0"	表示モードの状態 mode="1"
テキストボックス <k:text>	<input type="text">	<label>
テキストボックス (数値) <k:numericText>	<input type="text">	<label>
テキストボックス (コード) <k:codeText>	<input type="text">	<label>
テキストボックス (Hidden) <k:hidden>	<input type="hidden">	<label>
テキストボックス (日付) <k:date>	<input type="text">	<label>
テキストボックス (時間) <k:time>	<input type="text">	<label>
テキストボックス (国際化対応) <k:text-locales>	<input type="text">	<label>
テキストボックス (パスワード) <k:password>	<input type="password">	<label>
テキストエリア <k:textarea>	<textarea>	<label>
テキストエリア (国際化対応) <k:textarea-locales>	<textarea>	<label>
ドロップダウンリスト <k:select>	<select>	<label>
チェックボックス <k:checkbox>	<input type="text">	<label>
アンカー <k:a>	<a>	非表示 (何も出力しません)
ボタン <k:button>	<input type="button">	非表示 (何も出力しません)
ラジオボタン <k:radio>	<input type="radio">	<label>



コラム

上記以外にも [マスタ検索](#) のタグライブラリにもモードの設定が可能です。
マスタ検索のタグライブラリは [テキストボックス <k:text>](#) と同様になります。
タグライブラリの詳細は [タグライブラリ](#) を参照してください。

フォーマットとは？

intra-mart Accel Kaiden! では未フォーマットの値とフォーマット済の値の両方をクライアント (厳密にはAccessor) へ渡します。
渡された値は、タグライブラリのformatted属性により、表示する値を変更することができます。

- formatted属性により使用される値

formatted属性の値	mode="0"の場合	mode="1"の場合
"input"	未フォーマットの値	未フォーマットの値
"view"	未フォーマットの値	フォーマット済の値
"both"	フォーマット済の値	フォーマット済の値

- JSPのコーディング例

```
<k:text name="txt1" mode="0" formatted="view" />
<k:text name="txt2" mode="1" formatted="view" />
```

```
// ブロックに対するAccessor
var blockAccessor = KAIDEN.getGadget("ガジェットID").blockAccessors["ブロックID"];

// 値を設定する
blockAccessor("txt1").setValue(["2000/01/01","2000年01月01日"]); //txt1には"2000/01/01"が設定される
blockAccessor("txt2").setValue(["2000/01/01","2000年01月01日"]); //txt2には"2000年01月01日"が設定される
```

- Javaのコーディング例

jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSourceにフォーマット変換を行う機能があり、このクラスをサブクラス化し以下のようにformatFormattedValueをオーバーライドすることで、特定のフィールドのフォーマット変換と配列化を行うことができます。

```

class Converter extends ConvertForJsonSource {

  @Override
  protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    // 標準フォーマット変換
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 開始日のフォーマット変換
    Object startDate = map.get("startDate");
    if (startDate != null) {
      String formatted = /*フォーマット変換後の文字列*/"";
      result.put("startDate", formatted);
    }

    // 終了日のフォーマット変換
    Object endDate = map.get("endDate");
    if (endDate != null) {
      String formatted = /*フォーマット変換後の文字列*/"";
      result.put("endDate", formatted);
    }

    // フォーマット変換した文字列のMapを返却することで、
    // 「未フォーマットの値」と「フォーマット済の値」の配列が作成されます。
    return result;
  }
}

/* ～略～ */

// フォーマット変換・コンバート実行
Converter converter = new Converter();
Map<String, Object> convertedMap = converter.mapConvert(/*フォーマット変換対象Map*/);

// String[] { /*未フォーマットの値*/, /*フォーマット済の値*/ } が返却されます。
convertedMap.get("startDate");

```

Dialog

intra-mart Accel Kaiden!ではダイアログを作成するためのタグライブラリを用意しています。
 本項では、タグライブラリ([ダイアログ <k:dialog>](#))を使用したプログラミング方法を解説します。

- [プログラミング](#)
- [基本操作](#)
- [バリデーション](#)

プログラミング

ダイアログのプログラミングはタグライブラリ([ダイアログ <k:dialog>](#))を使用します。

プログラミングしたダイアログは、[ブロック <k:blockContainer>](#)と同様に扱われるため、[ガジェット <k:gadgetContainer>](#)内に配置する必要があります。

また、[ブロック <k:blockContainer>](#)と [TMTable <k:tmtable>](#)の関係の様に、互いに内包することはできません。

■ プログラミング例

```
<k:dialog title="サンプルダイアログ" id="sampleDialog" modal="true">
  <%-- ダイアログの内容 --%>
</k:dialog>
```

基本操作

ダイアログは次の様な形式で基本操作を行います。

■ プログラミング例

```
//ダイアログオブジェクトの取得
var dialog = KAIDEN.getGadget("ガジェットID").dialogs["sampleDialog"];

//ダイアログにボタンを登録
dialog.addButton("ボタン名", function() {
  //ボタン押下時の処理
});

//ダイアログを開く
dialog.open();

//ダイアログを閉じる
dialog.close();
```



コラム

dialog.addButton()で指定する関数について詳しくは、jQuery UIのDialogのbuttonオプションを参照してください。

バリデーション

[ダイアログ <k:dialog>](#)を使用して作成したダイアログは、ダイアログを自動的に<form>で囲みます。そのため、次の様な形式でバリデーションを実行できます。

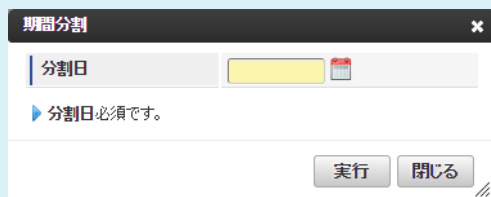
■ プログラミング例

```
if (dialog.validate("バリデーションキー")) {
  //エラーがなかった場合の処理
}
```



コラム

エラーメッセージはダイアログ内に表示されます。



ListData

intra-mart Accel Kaiden!ではドロップダウンリストを作成するためのタグライブラリを用意しています。本項では、タグライブラリ([ドロップダウンリスト <k:select>](#))を使用したプログラミング方法を解説します。

- リストデータとは？
- リストデータの作成
- リストデータの動的変更
- リストデータの動的変更 (Ajax)
- タグライブラリ
- ドロップダウンリストとの連携

リストデータとは？

リストデータとは、ドロップダウンリストで使用する選択肢(<option>)となるデータを指します。
 リストデータはJavaScriptで作成する方法や、タグライブラリ([リストデータ](#))で生成する方法などがあります。

リストデータの作成

リストデータは、KAIDEN.base.foundation.SelectListData.storeListを使用し、作成することができます。

- **プログラミング例**
 第一引数がリストデータ名、第二引数はリストデータとなり、
 第一引数のリストデータ名は、HTMLページ全体で一意な名称で扱われます。

```
// リストデータの作成
jQuery(function($) {
  KAIDEN.base.foundation.SelectListData.storeList("listName"
    , [{key:"key1", value:"value1", deleteFlag:"false"}
    , {key:"key2", value:"value2", deleteFlag:"false"}
    , {key:"key3", value:"value3", deleteFlag:"true"}]);
});
```



コラム

intra-mart Accel Kaiden!では、上記の様なプログラミング以外にもタグライブラリとしてリストデータを用意しています。
 詳細は、[タグライブラリ - リストデータ](#)を参照してください。

リストデータの動的変更

リストデータの動的な変更は、前項と同様に、KAIDEN.base.foundation.SelectListData.storeListを使用します。

- **プログラミング例**
 第一引数のリストデータ名を同じ名前で再定義することで、
 関連づいている<select>すべての<option>値が新しく書き換わります。

```
// リストデータの作成
jQuery(function($) {
  KAIDEN.base.foundation.SelectListData.storeList("listName"
    , [{key:"key1", value:"値1", deleteFlag:"false"}
    , {key:"key2", value:"値2", deleteFlag:"false"}
    , {key:"key3", value:"値3", deleteFlag:"true"}]);
});
```



コラム

リストデータが動的に変更された場合、選択済みのコードが変更後のリストデータにもあれば復元(選択状態)します。
 選択済みのコードが変更後のリストデータになければ、valueが空文字の<option>が選択されるか、存在しなければselectedIndex = -1を設定します。

リストデータの動的変更(Ajax)

KAIDEN.base.foundation.SelectListData.storeListには、Ajax通信を利用しリストデータを動的に変更する仕組みが用意されています。

- **プログラミング例**
 refreshList()を実行することで、追加条件({criteriaData:"2013/01/01"})を付与してAjax通信を行い、
 その戻り値に応じて<select>の<option>を変更します。

```
// リストデータの作成
jQuery(function($) {
  KAIDEN.base.foundation.SelectListData.storeList("listName"
    , [{key:"key1", value:"値1", deleteFlag:"false"}
    , {key:"key2", value:"値2", deleteFlag:"false"}
    , {key:"key3", value:"値3", deleteFlag:"true"}]
    , "ajaxURL"
    , {companyCd:"comp_sample_01"});
});

// リストデータのリフレッシュ
KAIDEN.base.foundation.SelectListData.refreshList("listName", {criteriaData:"2013/01/01"});
```



コラム

Ajax通信の戻り値は、JSON形式のリストデータを返却するようにしてください。
 storeList()を実行した時点でもURLを指定しますが、この時点ではAjax通信は行われません。

タグライブラリ

会社マスタや項目マスタなどのリストデータはタグライブラリを使用して作成します。

- **プログラミング例**

```
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_base" uri="http://kaiden.slcs.co.jp/taglib/listdata/base" %>

<klist_base:listDataCompany name="readParmitCompany" type="readParmitCompany" />
<k:select name="companyCd" mode="0" dataName="readParmitCompany"/>
```

i コラム

詳細は、[タグライブラリ・リストデータを参照してください](#)。

ドロップダウンリストとの連携

前項までで作成したリストデータを、[ドロップダウンリスト](#) `<k:select>`に連携させ、`<option>`を生成します。

■ プログラミング例

```
<%-- ドロップダウンリストの作成 --%>
<k:select name="list" mode="0" dataName="listName" />
```

■ HTML

前項までで作成したリストデータを連携させた場合、次の様なHTMLを生成します。

```
<select name="list">
  <option value="key1">value1</option>
  <option value="key2">value2</option>
</select>
```

i コラム

リストデータのdeleteFlagが"true"の場合は、`<option>`は生成されません。
[ドロップダウンリスト](#) `<k:select>`にincludeDisabled="true"を設定すると、
 deleteFlagが"true"のデータの`<option>`も生成します。

[ドロップダウンリスト](#) `<k:select>`にmode="1"を設定した際に、
 Accessorを使用してsetValue("コード値")すると、deleteFlagを無視してデータを表示します。

FS2List

intra-mart Accel Kaiden!では2段ヘッダリストを作成するためのタグライブラリを用意しています。
 本項では、タグライブラリ ([2段ヘッダリスト](#) `<k:fs2List>`)を使用したプログラミング方法を解説します。

- FS2Listとは？
- FS2Listのプログラミング
 - JSP
 - XML
 - Java
- FS2Listの関連API
 - FS2Listのデータ取得
 - FS2Listのイベント処理

FS2Listとは？

FS2Listは、2段ヘッダのリストを表現する場合に利用します。
ヘッダやデータに表示する項目は、xmlで設定します。

次の画面では、FS2Listを利用して機能を実現しています。

- 勤務表検索(勤務表照会)

FS2Listのプログラミング

FS2Listは、タグライブラリの **2段ヘッダリスト** `<k:fs2List>` を使用します。
2段ヘッダリスト `<k:fs2List>` は、Ajax通信でデータを取得した結果をHTMLに生成します。

JSP

- プログラミング例(HTML)

ajaxAccessUrlは一覧検索を行う際のURLを指定します。

```
<k:fs2List id="fs2ListId" ajaxAccessUrl="fs2List/search"/>
```

- プログラミング例(JavaScript)

ajaxAccessUrlは一覧検索を行う際のURLを指定します。

```
// FS2List検索を実行
jQuery(function($) {
    var fs2List = KAIDEN.Fs2List("fs2ListId");
    // 検索条件
    var criteria = {"companyCd":companyCd};
    fs2List.search(criteria);

});
```



コラム

JSP内に複数のFS2Listを生成することが可能です。



注意

複数のFS2Listを生成する場合、idは画面内で一意となるように指定してください。

XML

FS2Listは、2段ヘッダリストの設定はxmlファイルにて実装します。
各設定値については **2段ヘッダリスト** `<k:fs2List>` を参照してください。
表示項目はテナント全体の設定、および会社個別の設定を行うことができます。

- プログラミング例(XML)


```

<defaultSetting>
  <limit>20</limit>
  <rownumberSurfaceKey></rownumberSurfaceKey>
  <useSelectedMark>true</useSelectedMark>
  <useRownumber>true</useRownumber>
  <colsHeight>32px</colsHeight>
  <fs2ListScrollHeight>350px</fs2ListScrollHeight>
  <!-- 検索画面の固定列カラム設定 -->
  <fixedCols>
    <fixedCol>
      <fieldKey>fixedColumn</fieldKey>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey></headerSurfaceKey>
      <colPosition></colPosition>
      <colspan></colspan>
      <rowspan>2</rowspan>
      <sortable>false</sortable>
    </fixedCol>
  </fixedCols>
  <!-- 検索画面の可変列カラム設定 -->
  <variableCols>
    <variableCol>
      <fieldKey>variableColumn</fieldKey>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey></headerSurfaceKey>
      <colPosition></colPosition>
      <colspan></colspan>
      <rowspan>2</rowspan>
      <sortable>false</sortable>
    </variableCol>
  </variableCols>
  <orders>
    <order>
      <fieldKey>A</fieldKey>
      <type>asc</type>
    </order>
  </orders>
</defaultSetting>
<!-- 会社毎の設定 -->
<companySetting>
  <companyCd>%会社Aのコード値%</companyCd>
  <settings>
    %会社Aの設定%
  </settings>
</companySetting>

```

Java

FS2Listを生成するには、Form、Dto、Actionクラスを作成します。

FS2Listを生成するFormクラスを作成します。

FS2ListのFormクラスを実装する場合、jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListForm を継承します。

■ プログラミング例(Form)

```

package jp.co.slcs.kaiden2.base.feature.form.kaiden;

import jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListForm;

/**
 * FS2ListのFormクラスです。
 */
public class Fs2ListForm extends AbstractFs2ListForm {
    // 実装なし
}

```

FS2Listを生成するDtoクラスを作成します。

FS2ListのDtoクラスを実装する場合、jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListDto を継承します。

■ プログラミング例(Dto)

```

package jp.co.slcs.kaiden2.base.feature.dto.kaiden;

import jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListDto;

/**
 * FS2ListのDTOクラスです。
 */
public class Fs2ListDto extends AbstractFs2ListDto {
    // 実装なし
}

```

FS2Listを生成するActionクラスを作成します。

FS2ListのActionクラスを実装する場合、jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListAction を継承します。

■ プログラミング例(Action)

```
package jp.co.slcs.kaiden2.base.feature.action.kaiden;

import java.util.HashMap;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.feature.dto.kaiden.Fs2ListDto;
import jp.co.slcs.kaiden2.base.feature.form.kaiden.Fs2ListForm;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListAction;
import jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListDto;
import jp.co.slcs.kaiden2.base.foundation.fs2.AbstractFs2ListForm;
import org.seasar.struts.annotation.ActionForm;

/**
 * FS2ListのActionクラスです。
 */
public class Fs2ListAction extends AbstractFs2ListAction {

    /** Form. */
    @Resource
    @ActionForm
    public Fs2ListForm fs2ListForm;

    /** Dto. */
    @Resource
    public Fs2ListDto fs2ListDto;

    /**
     * 一覧検索結果ページの返却.
     *
     * @return 一覧検索結果ページ
     * @throws KaidenSystemException システムエラー
     */
    public String search() throws KaidenSystemException {
        return super.index();
    }

    /**
     * 検索結果をサブクラスから返却.
     *
     * @return 検索結果
     * @throws KaidenSystemException システムエラー
     */
    @Override
    protected Map<String, Object> getSearchResultData() throws KaidenSystemException {
        Map<String, Object> criteria = new HashMap<String, Object>();
        criteria.put("key01", "value01");
        criteria.put("key02", "value02");
        return criteria;
    }

    /**
     * Formのインスタンスを返却.
     *
     * @return 対象検索用Form
     */
    @Override
    protected AbstractFs2ListForm getForm() {
        return fs2ListForm;
    }

    /**
     * DTOのインスタンスを返却.
     *
     * @return 対象検索用DTO
     */
    @Override
    protected AbstractFs2ListDto getDto() {
        return fs2ListDto;
    }
}
```

FS2Listの関連API

本項では、FS2Listの操作やデータの設定・取得の方法を紹介します。

各APIの詳細は『[intra-mart Accel Kaiden! JsDoc](#)』を参照してください。

FS2Listのデータ取得

FS2Listのデータを取得する場合には、次のAPIを使用してください。

- **getRowData(rowNumber)**
特定のタブルのデータをjavascriptオブジェクトの形式で返却
- **countAll()**
検索した結果の全件数を返却
- **pageCount()**
検索した結果のページ数を返却
- **numberOfPage()**
検索した結果の表示中のページ番号を返却
- **countDisplay()**
検索した結果の表示中の件数を返却

FS2Listのイベント処理

FS2Listのイベント処理は次の様に実装してください。

- 1行追加直後

```
// FS2Listの取り出し
var fs2List = KAIDEN.Fs2List("fs2ListId");
// 1行追加直後イベント
fs2List.postRowEachEvent(function(event, rowNumber, fs2ListAccessor, rowData) {
  //引数:event      :イベントオブジェクト
  //引数:rowNumber   :挿入起点となる行番号
  //引数:fs2ListAccessor:当該FS2ListへのAccessor
  //引数:rowData     :行番号に設定された値
});
```

- 全一覧データ表示直後

```
// FS2Listの取り出し
var fs2List = KAIDEN.Fs2List("fs2ListId");
// 全一覧データ表示直後イベント
fs2List.displayFs2ListSuccessEvent(function(event, fs2ListAccessor) {
  //引数:event      :イベントオブジェクト
  //引数:fs2ListAccessor:当該FS2ListへのAccessor
});
```

Manager

本項では、intra-mart Accel Kaiden!におけるManagerについて解説します。

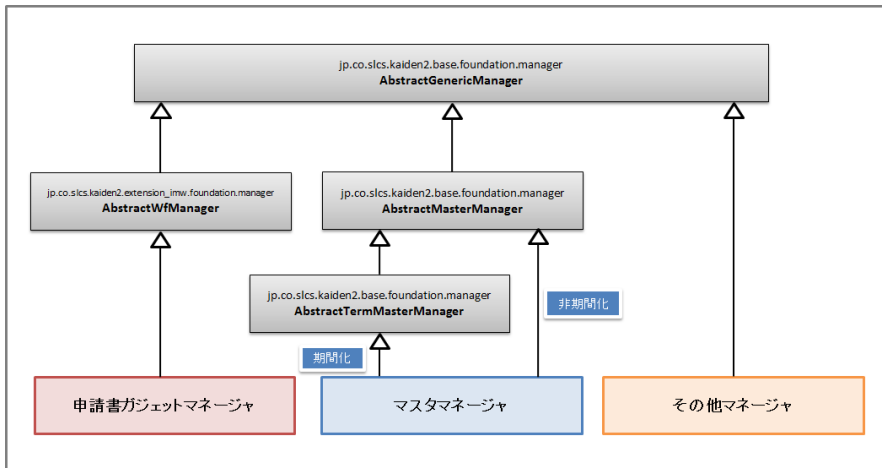
- マネージャとは？
- マネージャの構成
- マネージャの呼び出し
- マネージャの作成
 - 会社認可
 - SQLファイル検索
 - キーワード検索

マネージャとは？

intra-mart Accel Kaiden!のマネージャとは、データベース操作のAPIを提供しています。
 マスタメンテナンスや申請書ガジェット等の機能ごとに様々なマネージャーが存在します。

マネージャの構成

intra-mart Accel Kaiden!におけるマネージャは、継承するクラスの違いにより
 「マスタマネージャ」「申請書ガジェットマネージャ」「その他マネージャ」に分類できます。



マネージャの呼び出し

呼び出す場所については特に制限はなく、どこからでも使用できます。
 intra-mart Accel Kaiden!のプロジェクトマスタマネージャクラスを呼び出すサンプルを以下に掲載します。

- @Resourceアノテーションによるコンポーネント取得

```

@Resource
public ProjectManager projectManager;

public void xxxMethod() {
    projectManager.xxx();
}
  
```

- コンポーネント名を指定してコンポーネント取得

```

public ProjectManager projectManager;

public void xxxMethod() {
    projectManager = SingletonS2Container.getComponent("projectManager");
    projectManager.xxx();
}
  
```

i コラム

intra-mart Accel Kaiden!のマネージャクラスでは、@Resourceアノテーションを使用しない場合、
 コンポーネント名を指定して取得しています。

マネージャの作成

最上位クラスのjp.co.slcs.kaiden2.base.foundation.manager.AbstractGenericManagerで提供している機能について、ポイントのみ解説していきます。
 基本的な検索・登録・更新・削除のメソッドについては言及しません。

会社認可

会社認可を参照するメソッド、しないメソッドが用意されていますので、必要に応じてマネージャに実装できます。

会社認可を参照するAPIはjp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDtoを引数で受け取ります。
 AuthorityCriteriaDtoには以下の内容が設定されており、認可設定で許可されているかどうかを判定します。

- アクション
- 認可判定基準日
- 認可判定ユーザ（省略時はログインユーザを利用）
- 認可判定ロケール

SQLファイル検索

複雑なSQL文を実行するためにSQLファイルによる検索メソッドが用意されています。

サービスクラスとSQLファイル名(パス除く)を引数で受け取ります。

「"WEB-INF/classes/META-INF/sql" + "サービスクラスのパッケージ" + "サービスクラス名"」のSQLファイルを intra-mart Accel Kaiden! 基底クラスが取得しています。

また、引数のlikeTargetにカラムを指定すると、部分一致検索条件に変換されます。

キーワード検索

キーワード検索を実行するメソッドを用意しています。

intra-mart Accel Kaiden!ではマスタメンテナンス画面の汎用検索画面で使用しており、一つのテキストボックスに入力した値を複数カラムに対して検索を実施できます。

Conductor

本項では、intra-mart Accel Kaiden!におけるConductorについて解説します。

- [コンダクターとは？](#)
- [コンダクターの呼び出し](#)

コンダクターとは？

intra-mart Accel Kaiden!におけるコンダクターとは、複数の画面から呼び出されるビジネスロジック処理のAPIを提供しています。
intra-mart Accel Kaiden!のコンダクターでは、前項の *Manager* クラスを複数呼び出し、共通処理を実装しています。



コラム

コンダクタークラスの詳細は『intra-mart Accel Kaiden! JavaDoc』を参照してください。

コンダクターの呼び出し

呼び出す場所については特に制限はなく、どこからでも使用できます。
intra-mart Accel Kaiden!の勤務表コンダクタークラスを呼び出すサンプルを以下に掲載します。

- @Resourceアノテーションによるコンポーネント取得

```
@Resource
public WorkReportConductor workReportConductor;

public void xxxMethod() {
    workReportConductor.xxx();
}
```

- コンポーネント名を指定してコンポーネント取得

```
public WorkReportConductor workReportConductor;

public void xxxMethod() {
    workReportConductor = SingletonS2Container.getComponent("workReportConductor");
    workReportConductor.xxx();
}
```



コラム

intra-mart Accel Kaiden!のコンダクタークラスでは、@Resourceアノテーションを使用しない場合、
コンポーネント名を指定して取得しています。

Log

intra-mart Accel Kaiden!でのログ出力はjp.co.slcs.kaiden2.base.foundation.util.log.KaidenLoggerクラスを使用します。
以下のログレベルで出力できます。

- ログレベル
 - trace (最も軽微)
 - debug
 - info
 - warn
 - error (最も重大)
- プログラミング例

```
String logMsg = "ログメッセージ";

// traceログを出力
KaidenLogger.trace(logMsg);

// debugログを出力
KaidenLogger.debug(logMsg);

// infoログを出力
KaidenLogger.info(logMsg);

// warnログを出力
KaidenLogger.warn(logMsg);

// errorログを出力
KaidenLogger.error(logMsg);
```



コラム

KaidenLoggerクラスを使用すると、ローガー名は「KAIDEN_LOG」となり、ログ設定は「WEB-INF/conf/log/kaiden_logger.xml」で設定されます。



コラム

KaidenLoggerクラスの詳細は『intra-mart Accel Kaiden! JavaDoc』を参照してください。

intra-mart Accel Kaiden!でのメッセージ、サーフェスはjp.co.slcs.kaiden2.base.foundation.helper.PropertyHelperクラスを使用して取得できます。

```
/** PropertyHelper. */
@Resource
public PropertyHelper propertyHelper;

public void xxx() {
    // ロケールを指定してメッセージを取得
    propertyHelper.getMessage(Locale.JAPANESE, "メッセージキー", "置換文字列");

    // ロケールを指定しないでメッセージを取得
    propertyHelper.getMessage("メッセージキー", "置換文字列");

    // ロケールを指定してサーフェスを取得
    propertyHelper.getSurface(Locale.JAPANESE, "サーフェスキー", "置換文字列");

    // ロケールを指定しないでサーフェスを取得
    propertyHelper.getSurface("サーフェスキー", "置換文字列");
}
```

コラム

メッセージ設定ファイル名は「モジュールID_messages_言語ID.xml」、サーフェス設定ファイル名は「モジュールID_surfaces_言語ID.xml」となります。
ファイルの配置場所は「WEB-INF/classes/jp/co/slcs/kaiden2/」です。

注意

メッセージ設定ファイルやサーフェス設定ファイルの設定では、異なるファイルであってもキーが重複しないように注意してください。

コラム

PropertyHelperクラスの詳細は『intra-mart Accel Kaiden! JavaDoc』を参照してください。

Helper&Util

本項では、intra-mart Accel Kaiden!のHelperクラスとUtilクラスについて解説します。

intra-mart Accel Kaiden!では、様々なHelperクラスとUtilクラスを用意しています。

- ヘルパーとは？
- intra-mart Accel Kaiden!で提供しているヘルパー
 - 基盤モジュール
 - IM-Workflow連携モジュール
 - ワークフローモジュール
 - 勤務管理モジュール
- ヘルパーの呼び出し
- ヘルパーの作成
- ユーティリティとは？
- intra-mart Accel Kaiden!で提供しているユーティリティ
 - 基盤モジュール

ヘルパーとは？

intra-mart Accel Kaiden!のヘルパーとは、各種処理支援を行うAPIを提供しています。
 また、ユーティリティのうち主にintra-mart Accel PlatformのAPIを利用もしくはラッピングしたクラスです。
 ヘルパーはDIコンテナにsingletonにて登録されます。

intra-mart Accel Kaiden!で提供しているヘルパー

intra-mart Accel Kaiden!で提供しているヘルパークラスは以下の通りです。
 各APIの詳細は『[intra-mart Accel Kaiden! JavaDoc](#)』を参照してください。

基盤モジュール

クラス名	説明
CacheHelper	キャッシュ機能を扱うクラスです。
CalendarInfoManagerHelper	intra-mart Accel PlatformのCalendarInfoManagerクラスをラッピングしています。
CompanyManagerHelper	intra-mart Accel PlatformのCompanyManagerクラスをラッピングしています。
CompanySettingsHelper	会社設定に関する機能を扱うクラスです。
ContextHelper	intra-mart Accel Platformのコンテキストに関する操作を扱うクラスです。
CurrencyHelper	通貨に関する機能を扱うクラスです。
DateHelper	日付に関する機能を扱うクラスです。
ExportHelper	Export処理を扱う機能を扱うクラスです。
ExtExportHelper	拡張版Export処理を扱う機能を扱うクラスです。
ExtImportHelper	拡張版Import処理を扱う機能を扱うクラスです。
FormatHelper	書式変換に関する機能を扱うクラスです。
IdentifierHelper	intra-mart Accel PlatformのIdentifierクラスをラッピングしています。
ImportHelper	Import処理を扱う機能を扱うクラスです。
LockHelper	intra-mart Accel PlatformのNewLockクラスをラッピングしています。
ModuleHelper	Module判定用機能を扱うクラスです。
NumberHelper	数値に関する機能を扱うクラスです。
PermissionAthCompanyHelper	所属会社の権限保持を判定する機能を扱うクラスです。
PropertyHelper	XMLプロパティへのアクセス機能を扱うクラスです。
RoleInfoManagerHelper	intra-mart Accel PlatformのRoleInfoManagerクラスをラッピングしています。
StorageFileHelper	ファイル操作を扱う機能を扱うクラスです。
SystemLocaleHelper	システムロケールを扱う機能を扱うクラスです。
TransferHelper	intra-mart Accel PlatformのTransferクラスをラッピングしています。
UserManagerHelper	intra-mart Accel PlatformのUserManagerクラスをラッピングしています。
ValidateMapHelper	各ValidatorクラスのvalueMapにアクセスする機能を扱うクラスです。
XmlOutputHelper	XML出力を行います。
XMLResourceBundle	XMLを扱うリソースバンドルクラスです。
XMLResourceBundleControl	ResourceBundle.Controlクラスにて、XMLを扱う実装クラスです。
XMLResourceHelper	XMLプロパティファイルを扱うクラスです。

IM-Workflow連携モジュール

クラス名	説明
UserMatterPropertyHelper	ユーザデータ案件プロパティ情報操作を扱うクラスです。

ワークフローモジュール

クラス名	説明
FbExportHelper	FBデータファイルのExportの機能を扱うクラスです。

勤務管理モジュール

クラス名	説明
LaborDateHelper	intra-mart Accel Kaiden! 勤務管理モジュールの日付関連を機能を扱うクラスです。

ヘルパーの呼び出し

呼び出す場所については特に制限はなく、どこからでも使用できます。

intra-mart Accel Kaiden!の通貨関連ヘルパークラスを呼び出すサンプルを以下に掲載します。

- @Resourceアノテーションによるコンポーネント取得

```
@Resource
public CurrencyHelper currencyHelper;

public void xxxMethod() {
    String currencyCd = currencyHelper.getCompanyCurrencyCd("会社コード");
}
```

- クラス名を指定してコンポーネント取得

```
public CurrencyHelper currencyHelper;

public void xxxMethod() {
    currencyHelper = SingletonS2Container.getComponent(CurrencyHelper.class);
    String currencyCd = currencyHelper.getCompanyCurrencyCd("会社コード");
}
```

ヘルパーの作成

業務要件に応じて作成してください。

作成する際はintra-mart Accel Kaiden!が提供しているヘルパークラスを拡張するのではなく、新規のヘルパークラスを作成してください。



注意

intra-mart Accel Kaiden!ではクラス名を指定してヘルパークラスのコンポーネント取得をしています。
ヘルパークラスが拡張されることを想定していません。

ユーティリティとは？

intra-mart Accel Kaiden!のユーティリティとは、

主に簡易な便利機能を提供し、staticな共通の処理のメソッドを集めたクラスです。

intra-mart Accel Kaiden!で提供しているユーティリティ

intra-mart Accel Kaiden!で提供しているユーティリティクラスは以下の通りです。

各APIの詳細は『[intra-mart Accel Kaiden! JavaDoc](#)』を参照してください。

基盤モジュール

クラス名	説明
ArrayUtil	配列のユーティリティクラスです。
BeanUtil	Beanユーティリティクラスです。 Beanへの汎用アクセスを提供します。
CollectionUtil	コレクションに関するユーティリティクラスです。
CollectionUtilDtoSortComparator	DTOをソートするComparatorクラスです。
CollectionUtilDtoSortComparator4Locale	システムロケール順にソートするComparatorクラスです。 対象DTOにロケールIDが存在しない場合はソートしません。
ConvertForJsonSource	JSON出力用の形式に変換するユーティリティクラスです。
HtmlEncodeUtil	HTML形式での表示を目的とする文字列変換ユーティリティクラスです。
JsonUtil	JSON操作用のユーティリティクラスです。
KaidenPropertyReader	intra-mart Accel Kaiden!のプロパティファイルを読み込むユーティリティクラスです。
ListDivider	リスト分割ユーティリティクラスです。 DTOリストを指定の主キー毎に分割します。
ListDividerForMap	リスト分割ユーティリティクラスです。 Mapリストを指定の主キー毎に分割します。
PropertyJspUtil	クライアントからのプロパティファイル読み込みユーティリティクラスです。
ResponseWriter	HttpResponseへの出力機能を扱うユーティリティクラスです。
StringUtil	String型オブジェクトのユーティリティクラスです。

Exception

本項では、intra-mart Accel Kaiden!のExceptionクラスについて解説します。

- intra-mart Accel Kaiden!で用意しているExceptionクラス
 - [intra-mart Accel Kaiden! 基盤モジュール](#)
 - [intra-mart Accel Kaiden! IM-Workflow連携モジュール](#)
 - [intra-mart Accel Kaiden! 勤務管理モジュール](#)

intra-mart Accel Kaiden!で用意しているExceptionクラス

intra-mart Accel Kaiden!で用意しているExceptionクラスは以下の通りです。
各APIの詳細は『[intra-mart Accel Kaiden! JavaDoc](#)』を参照してください。

intra-mart Accel Kaiden! 基盤モジュール

クラス名	説明
KaidenApplicationException	アプリケーションエラー発生時にスローされます。
KaidenDuplicationEntryException	重複登録が行われた際にスローされます。
KaidenHaventAuthorityException	データに対するアクセス権を持たない際にスローされます。
KaidenHaventAuthorityToCompanyException	会社に対する認可設定による権限を持たない際にスローされます。
KaidenIllegalParametersException	引数や項目値不正でエラーが発生した場合にスローされます。
KaidenNotFoundTargetDataException	対象データが見つからなかった場合にスローされます。
KaidenOptimisticLockException	楽観的ロックで競合が発生した場合にスローされます。
KaidenRuntimeException	実装バグが原因でエラーが発生した場合にスローされます。
KaidenSystemException	システム上続行するには問題があるエラーをキャッチしたときにスローされます。
KaidenValidateException	サーバサイドバリデートエラー発生時にスローされます。

intra-mart Accel Kaiden! IM-Workflow連携モジュール

クラス名	説明
KaidenWorkflowExternalException	IM-Workflowの申請や承認などのアクション処理、到達処理などでエラーが発生した場合にスローされます。



注意

IM-Workflowのユーザプログラム以外では使用できません。

intra-mart Accel Kaiden! 勤務管理モジュール

クラス名	説明
KaidenLaborHaventAgentAuthorizationException	代理処理権限を持っていない場合にスローされます。
KaidenLaborHaventOperateAuthorityException	処理実行権限を持っていない場合にスローされます。

応用

次の項ではintra-mart Accel Kaiden!の各機能の概要や構成を解説します。

マスタメンテナンス

本項では、マスタメンテナンス機能の概要、チュートリアル、サーバサイドを解説します。

機能説明

本項では、プロジェクトマスタメンテナンスを参考に、マスタメンテナンス機能の概要や構成を解説します。

概要

各機能で使用するマスタデータの新規登録や登録済みデータの更新、削除を行う機能です。
intra-mart Accel Kaiden!のマスタメンテナンス機能をプログラミングする際には、次のモジュールが必要となります。

- intra-mart Accel Kaiden! 基盤モジュール

intra-mart Accel Kaiden!のマスタは「期間化マスタ」「非期間化マスタ」の2種類があり、
次の条件に該当するマスタを「期間化マスタ」、該当しないマスタを「非期間化マスタ」と定義しています。

- システムの開始日から終了日までの全期間のデータを重複や間断なく連続して保持していること。
- マスタデータに削除フラグを保持し、削除フラグの切り替えで、マスタデータの有効無効が切り替えられること。

構成

intra-mart Accel Kaiden!のマスタメンテナンス機能は「一覧画面」「登録／更新画面」の2画面構成となっています。
プロジェクトマスタメンテナンス(期間化マスタ)では、次の様な画面構成となります。

- 一覧画面
マスタデータの検索、一覧表示を行う画面です。
次の画面は検索結果、および検索オプションを表示した状態の画面です。

No	プロジェクトコード	プロジェクト名	上位プロジェクト名	ステータス
1	000001	情報システム株式会社 ワークフローシステム再構築プロジェクト		●
2	000002	社内環境整備プロジェクト		●
3	000003	※社向け顧客・定費申請・承認ワークフローシステム(SAPフロント)導入		●
4	000004	△社 システム保守(2013年度)		●
5	000005	システム更新対応(社内)		●
6	000006	※社 ワークフローシステム保守		●
7	000007	H25年度通常運用(問合せ等)		●
8	000008	※社 システム構築検証支援		●
9	000009	新製品構築プロジェクト		●
10	000010	品質保証プロジェクト		●

- 登録／更新画面
マスタデータの登録、更新、削除を行う画面です。
次の画面は更新時の画面です。

開始日: 1900/01/01, 終了日: 2999/12/31

会社: サンプル会社

プロジェクトコード: 000001

プロジェクト名: 日本語 (情報システム株式会社 ワークフローシステム再構築)

英語: D Company Workflow System Restructuring Project

中国語(中華人民共和国): 信息系統公司 工作流系統重新构建项目

上位プロジェクト: [検索]

更新, 削除, 一覧に戻る

コラム

非期間化マスタの画面構成は「金融機関マスタメンテナンス」を参照してください。

コラム

intra-mart Accel Kaiden!の標準的なマスタメンテナンス機能をプログラミングする場合、
「一覧画面」は設定のみで、実際にプログラミングする必要はありません。
「登録／更新画面」は下図の様に期間一覧、操作ボタン、マスタメンテナンス本体を組み合わせて構成しておりますが、
期間一覧、操作ボタンは共通化されているため、マスタメンテナンス本体部分のみをプログラミングします。

期間一覧 1900/01/01- 2999/12/31 期間分割 期間変更	開始日 - 終了日 1900/01/01 - 2999/12/31	マスタメンテナンス本体 操作ボタン 更新 削除
	会社 サンプル会社	
	プロジェクトコード 000001	
	プロジェクト名 日本語 ○○情報システム株式会社ワークフローシステム再構築 英語 ○○ Information System Co., Ltd. workflow system n 中国語 ○○信息系统有限公司工作流系统改造工程	
	上位プロジェクト 	

- 一覽畫面

チュートリアルマスタメンテナンス

←

期間一覧

開始日 * - 終了日 *

1900/01/01 - 2999/12/31

会社 *

サンプル会社

チュートリアルコード *

チュートリアル名 *

日本語

英語

中国語

登録

一覧に戻る

登録／更新画面

登録 / 更新 / 削除

チュートリアルマスタメンテナンス

新規登録

検索

No	▲ チュートリアルコード	チュートリアル名	ステータス

本チュートリアルの手順は次の通りです。

テーブル作成

- テーブル定義
- DDL
 - Oracle Database
 - PostgreSQL
 - Microsoft SQL Server

テーブル定義

本チュートリアルでは次の様な定義のマスタテーブルを使用します。

- チュートリアルマスタ「k99m_tutorial」

No.	論理名	物理名	型	長さ	必須	PK
1	会社コード	company_cd	文字列	100	Y	○
2	チュートリアルコード	tutorial_cd	文字列	100	Y	○
5	チュートリアル名称	tutorial_name	文字列	1000	Y	
4	ロケールID	locale_id	文字列	50	Y	○
3	期間コード	term_cd	文字列	50	Y	○
6	開始日	start_date	タイムスタンプ		Y	
7	終了日	end_date	タイムスタンプ		Y	
8	削除フラグ	delete_flag	文字列	1	Y	
9	予備項目A	extension_a	文字列	1000		
10	予備項目B	extension_b	文字列	1000		
11	予備項目C	extension_c	文字列	1000		
12	予備項目D	extension_d	文字列	1000		
13	予備項目E	extension_e	文字列	1000		
14	予備項目F	extension_f	文字列	1000		
15	予備項目G	extension_g	文字列	1000		
16	予備項目H	extension_h	文字列	1000		
17	予備項目I	extension_i	文字列	1000		
18	予備項目J	extension_j	文字列	1000		
19	登録日時	entry_ts	タイムスタンプ		Y	
20	登録者コード	entry_user_cd	文字列	100	Y	
21	更新カウンタ	renew_cnt	数値	18,0	Y	
22	更新日時	renew_ts	タイムスタンプ		Y	
23	更新者コード	renew_user_cd	文字列	100	Y	



コラム

- 予備項目A～J
intra-mart Accel Kaiden!の各テーブルでは、カスタマイズを考慮し予備項目をあらかじめ定義しています。
実際のプログラミング時には予備項目の設定は不要です。



注意

- 登録日時、登録者コード、更新カウンタ、更新日時、更新者コード
intra-mart Accel Kaiden!の各テーブルで共通的に定義している項目です。
本コラムは必ず定義してください。

DDL

利用するデータベースに合わせて、次のいずれかのDDLを実行します。
本チュートリアルでは、Oracle Databaseの使用を推奨します。

Oracle Database


```

create table k99m_tutorial
(
  company_cd          varchar2(100) not null,
  tutorial_cd         varchar2(100) not null,
  tutorial_name       varchar2(1000) not null,
  locale_id          varchar2(50) not null,
  term_cd            varchar2(50) not null,
  start_date         timestamp not null,
  end_date           timestamp not null,
  delete_flag        varchar2(1) not null,
  extension_a        varchar2(1000),
  extension_b        varchar2(1000),
  extension_c        varchar2(1000),
  extension_d        varchar2(1000),
  extension_e        varchar2(1000),
  extension_f        varchar2(1000),
  extension_g        varchar2(1000),
  extension_h        varchar2(1000),
  extension_i        varchar2(1000),
  extension_j        varchar2(1000),
  entry_ts           timestamp not null,
  entry_user_cd      varchar2(100) not null,
  renew_cnt          numeric(18,0) not null,
  renew_ts           timestamp not null,
  renew_user_cd      varchar2(100) not null
);

alter table k99m_tutorial add primary key (company_cd, tutorial_cd, locale_id, term_cd);

```

PostgreSQL

```

create table k99m_tutorial
(
  company_cd          varchar(100) not null,
  tutorial_cd         varchar(100) not null,
  tutorial_name       varchar(1000) not null,
  locale_id          varchar(50) not null,
  term_cd            varchar(50) not null,
  start_date         timestamp without time zone not null,
  end_date           timestamp without time zone not null,
  delete_flag        varchar(1) not null,
  extension_a        varchar(1000),
  extension_b        varchar(1000),
  extension_c        varchar(1000),
  extension_d        varchar(1000),
  extension_e        varchar(1000),
  extension_f        varchar(1000),
  extension_g        varchar(1000),
  extension_h        varchar(1000),
  extension_i        varchar(1000),
  extension_j        varchar(1000),
  entry_ts           timestamp without time zone not null,
  entry_user_cd      varchar(100) not null,
  renew_cnt          numeric(18,0) not null,
  renew_ts           timestamp without time zone not null,
  renew_user_cd      varchar(100) not null
);

alter table k99m_tutorial add primary key (company_cd, tutorial_cd, locale_id, term_cd);

```

Microsoft SQL Server

```

create table k99m_tutorial
(
  company_cd          nvarchar(100) not null,
  tutorial_cd         nvarchar(100) not null,
  tutorial_name       nvarchar(1000) not null,
  locale_id          nvarchar(50) not null,
  term_cd            nvarchar(50) not null,
  start_date         datetime2 not null,
  end_date           datetime2 not null,
  delete_flag        nvarchar(1) not null,
  extension_a         nvarchar(1000),
  extension_b         nvarchar(1000),
  extension_c         nvarchar(1000),
  extension_d         nvarchar(1000),
  extension_e         nvarchar(1000),
  extension_f         nvarchar(1000),
  extension_g         nvarchar(1000),
  extension_h         nvarchar(1000),
  extension_i         nvarchar(1000),
  extension_j         nvarchar(1000),
  entry_ts           datetime2 not null,
  entry_user_cd      nvarchar(100) not null,
  renew_cnt          numeric(18,0) not null,
  renew_ts           datetime2 not null,
  renew_user_cd      nvarchar(100) not null
);

alter table k99m_tutorial
add constraint pk_k99m_tutorial primary key (company_cd, tutorial_cd, locale_id, term_cd);

```

S2JDBC

- [S2JDBC-Genの実行](#)
- [自動生成ソース](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)

S2JDBC-Genの実行

チュートリアルマスタのDDLを実行後、S2JDBC-Genにて次のソースを自動生成します。

- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.K99mTutorial
- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.K99mTutorialNames
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99mTutorialService
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99mTutorialCondition

コラム

S2JDBC-Genにてソースを自動生成した際に、次の個所を確認してください。

- “K99mTutorial”のスーパークラスが「jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity」に設定されていること。
- “K99mTutorial”に登録日時、登録者コード、更新カウンタ、更新日時、更新者コードが定義されていないこと。
- “K99mTutorialService”のスーパークラスが「jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService」に設定されていること。

```

src
├── main
│   └── generated
│       ├── jp
│       │   ├── co
│       │   │   ├── slcs
│       │   │   │   ├── kaiden2
│       │   │   │   │   ├── tutorial
│       │   │   │   │   │   ├── foundation
│       │   │   │   │   │   │   ├── model
│       │   │   │   │   │   │   │   ├── entity
│       │   │   │   │   │   │   │   │   ├── K99mTutorial.java
│       │   │   │   │   │   │   │   │   ├── K99mTutorialNames.java
│       │   │   │   │   │   │   │   └── service
│       │   │   │   │   │   │   │       ├── K99mTutorialCondition.java
│       │   │   │   │   │   │   │       └── K99mTutorialService.java

```

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/entity/K99mTutorial.java

```

package jp.co.slcs.kaiden2.tutorial.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * チュートリアルマスタ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k99m_tutorial")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K99mTutorial extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** チュートリアルコード. */
    @Id
    @Column(name = "tutorial_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String tutorialCd;

    /** チュートリアル名称. */
    @Column(name = "tutorial_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
    public String tutorialName;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** 期間コード. */
    @Id
    @Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String termCd;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp endDate;

    /** 削除フラグ. */
    @Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
    public String deleteFlag;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

- Path
src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/entity/K99mTutorialNames.java

```

package jp.co.slcs.kaiden2.tutorial.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K99mTutorial} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl"})
public final class K99mTutorialNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K99mTutorialNames() {
        super();
    }

    /**
     * companyCd のプロパティ名を返却.
     *
     * @return companyCd のプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * tutorialCd のプロパティ名を返却.
     *
     * @return tutorialCd のプロパティ名
     */
    public static PropertyName<String> tutorialCd() {
        return new PropertyName<String>("tutorialCd");
    }

    /**
     * tutorialName のプロパティ名を返却.
     */

```

```

    * @return tutorialNameのプロパティ名
    */
    public static PropertyName<String> tutorialName() {
        return new PropertyName<String>("tutorialName");
    }

    /**
     * localeIdのプロパティ名を返却。
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * termCdのプロパティ名を返却。
     *
     * @return termCdのプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * startDateのプロパティ名を返却。
     *
     * @return startDateのプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDateのプロパティ名を返却。
     *
     * @return endDateのプロパティ名
     */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }

    /**
     * deleteFlagのプロパティ名を返却。
     *
     * @return deleteFlagのプロパティ名
     */
    public static PropertyName<String> deleteFlag() {
        return new PropertyName<String>("deleteFlag");
    }

    /**
     * extensionAのプロパティ名を返却。
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却。
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却。
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却。
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }
}

```

```

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名

```

```

*/
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K99mTutorialNames extends PropertyName<K99mTutorial> {

    /**
     * インスタンスを構築.
     */
    public _K99mTutorialNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K99mTutorialNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K99mTutorialNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * tutorialCdのプロパティ名を返却.
     *
     * @return tutorialCdのプロパティ名
     */
    public PropertyName<String> tutorialCd() {
        return new PropertyName<String>("tutorialCd");
    }

    /**
     * tutorialNameのプロパティ名を返却.
     *
     * @return tutorialNameのプロパティ名
     */
    public PropertyName<String> tutorialName() {
        return new PropertyName<String>("tutorialName");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public PropertyName<String> termCd() {

```

```

    return new PropertyName<String>(this, "termCd");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>(this, "deleteFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *

```



```

    * @return extensionGのプロパティ名
    */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

- Path
src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99mTutorialService.java

```

package jp.co.slcs.kaiden2.tutorial.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.tutorial.foundation.model.entity.K99mTutorial;

/**
 * {@link K99mTutorial}のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K99mTutorialService extends GenerateService<K99mTutorial> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99mTutorialCondition.java

```

package jp.co.slcs.kaiden2.tutorial.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K99mTutorial}の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K99mTutorialCondition extends
    AbstractEntityCondition<K99mTutorialCondition> {

    /**
     * インスタンスを構築
     */
    public K99mTutorialCondition() {
    }

    /**
     * インスタンスを構築
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K99mTutorialCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** companyCdの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> companyCd =
        new NotNulltableStringCondition<K99mTutorialCondition>("companyCd", this);

    /** tutorialCdの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> tutorialCd =
        new NotNulltableStringCondition<K99mTutorialCondition>("tutorialCd", this);

    /** tutorialNameの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> tutorialName =
        new NotNulltableStringCondition<K99mTutorialCondition>("tutorialName", this);

    /** localeIdの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> localeId =
        new NotNulltableStringCondition<K99mTutorialCondition>("localeId", this);

    /** termCdの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> termCd =
        new NotNulltableStringCondition<K99mTutorialCondition>("termCd", this);

    /** startDateの条件. */
    public NotNulltableCondition<K99mTutorialCondition, Timestamp> startDate =
        new NotNulltableCondition<K99mTutorialCondition, Timestamp>("startDate", this);

    /** endDateの条件. */
    public NotNulltableCondition<K99mTutorialCondition, Timestamp> endDate =
        new NotNulltableCondition<K99mTutorialCondition, Timestamp>("endDate", this);

    /** deleteFlagの条件. */
    public NotNulltableStringCondition<K99mTutorialCondition> deleteFlag =

```

```

public NotNullableStringCondition<K99mTutorialCondition> deleteFlag =
    new NotNullableStringCondition<K99mTutorialCondition>("deleteFlag", this);

/** extensionAの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionA =
    new NullableStringCondition<K99mTutorialCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionB =
    new NullableStringCondition<K99mTutorialCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionC =
    new NullableStringCondition<K99mTutorialCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionD =
    new NullableStringCondition<K99mTutorialCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionE =
    new NullableStringCondition<K99mTutorialCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionF =
    new NullableStringCondition<K99mTutorialCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionG =
    new NullableStringCondition<K99mTutorialCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionH =
    new NullableStringCondition<K99mTutorialCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionI =
    new NullableStringCondition<K99mTutorialCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K99mTutorialCondition> extensionJ =
    new NullableStringCondition<K99mTutorialCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<K99mTutorialCondition, Timestamp> entryTs =
    new NotNullableCondition<K99mTutorialCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<K99mTutorialCondition> entryUserCd =
    new NotNullableStringCondition<K99mTutorialCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<K99mTutorialCondition, Long> renewCnt =
    new NotNullableCondition<K99mTutorialCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullableCondition<K99mTutorialCondition, Timestamp> renewTs =
    new NotNullableCondition<K99mTutorialCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<K99mTutorialCondition> renewUserCd =
    new NotNullableStringCondition<K99mTutorialCondition>("renewUserCd", this);
}

```

Manager

- [Managerの作成](#)
- [Manager実装クラスの作成](#)
- [ManagerDtoの作成](#)
- [SQLファイルの作成](#)

Managerの作成

jp.co.slcs.kaiden2.base.foundation.master.TermMasterManager を継承してMasterTutorialManagerインターフェースを作成します。

```
package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial;

import java.util.List;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;

/**
 * チュートリアルマスタマネージャクラスのinterface.
 */
public interface MasterTutorialManager extends TermMasterManager<MasterTutorialManagerDto> {

    /**
     * 新規登録用の検証定義リストを返却します.
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

    /**
     * 更新登録用の検証定義リストを返却します.
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

}
```

Manager実装クラスの作成

jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager を継承してMasterTutorialManagerImplクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.MasterTutorialManager;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.MasterTutorialManagerDto;
import jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99mTutorialService;

public class MasterTutorialManagerImpl extends AbstractTermMasterManager<MasterTutorialManagerDto> implements
    MasterTutorialManager {

    /** サービスクラス */
    @Resource
    protected K99mTutorialService k99mTutorialService;

    /** 主キー検索 */
    @Override
    public MasterTutorialManagerDto select(MasterTutorialManagerDto criteria, AuthorityCriteriaDto authCriteria) {
        return super.select(k99mTutorialService, criteria, authCriteria);
    }

    /** 主キー検索(認可設定無視) */
    @Override
    public MasterTutorialManagerDto selectIgnoreAuthCompany(MasterTutorialManagerDto criteria) {
        return super.selectIgnoreAuthCompany(k99mTutorialService, criteria);
    }

    /** 汎用検索 */
    @Override
```

```

@Override
public List<MasterTutorialManagerDto> selectList(MasterTutorialManagerDto criteria,
        AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectList(k99mTutorialService, criteria, authCriteria, order);
}

/** 汎用検索(認可設定無視) */
@Override
public List<MasterTutorialManagerDto> selectListIgnoreAuthCompany(MasterTutorialManagerDto criteria,
        List<OrderDto> order) {
    return super.selectListIgnoreAuthority(k99mTutorialService, criteria, order);
}

/** ロケール検索 */
@Override
public List<MasterTutorialManagerDto> selectLocales(MasterTutorialManagerDto criteria,
        AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k99mTutorialService, criteria, authCriteria);
}

/** ロケール検索(認可設定無視) */
@Override
public List<MasterTutorialManagerDto> selectLocalesIgnoreAuthCompany(MasterTutorialManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k99mTutorialService, criteria);
}

/** 参照検索 */
@Override
public List<MasterTutorialManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
        AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {"tutorialName", "extensionA", "extensionB", "extensionC", "extensionD", "extensionE",
        "extensionF", "extensionG", "extensionH", "extensionI", "extensionJ", "companyName"};
    return super.selectBySql(k99mTutorialService, "selectView.sql", criteria, order, authCriteria, likeTarget);
}

/** 参照検索(認可設定無視) */
@Override
public List<MasterTutorialManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria,
        List<OrderDto> order) {
    String[] likeTarget = {"tutorialName", "extensionA", "extensionB", "extensionC", "extensionD", "extensionE",
        "extensionF", "extensionG", "extensionH", "extensionI", "extensionJ", "companyName"};
    return super.selectBySqlIgnoreAuthority(k99mTutorialService, "selectView.sql", criteria, order, likeTarget);
}

/** キーワード検索 */
@Override
public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {"tutorialName", "extensionA", "extensionB", "extensionC", "extensionD", "extensionE",
        "extensionF", "extensionG", "extensionH", "extensionI", "extensionJ", "companyName"};
    return super.selectListByKeywords(k99mTutorialService, "selectView.sql", criteria, order, keywordsCriteria,
        authCriteria, likeTarget);
}

/** キーワード検索(認可設定無視) */
@Override
public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria) {
    String[] likeTarget = {"tutorialName", "extensionA", "extensionB", "extensionC", "extensionD", "extensionE",
        "extensionF", "extensionG", "extensionH", "extensionI", "extensionJ", "companyName"};
    return super.selectListByKeywordsIgnoreAuthority(k99mTutorialService, "selectView.sql", criteria, order,
        keywordsCriteria, likeTarget);
}

/** 有効化 */
@Override
public void toEnabled(MasterTutorialManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toEnabled(k99mTutorialService, criteria, authCriteria);
}

/** 有効化(認可設定無視) */
@Override
public void toEnabledIgnoreAuthCompany(MasterTutorialManagerDto criteria) throws KaidenOptimisticLockException {
    super.toEnabledIgnoreAuthCompany(k99mTutorialService, criteria);
}

/** 無効化 */
@Override
public void toDisabled(MasterTutorialManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toDisabled(k99mTutorialService, criteria, authCriteria);
}

/** 無効化(認可設定無視) */
@Override

```

```

@Override
public void toDisabledIgnoreAuthCompany(MasterTutorialManagerDto criteria) throws KaidenOptimisticLockException {
    super.toDisabledIgnoreAuthCompany(k99mTutorialService, criteria);
}

/** 登録 */
@Override
public int entry(List<MasterTutorialManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
    return super.entry(k99mTutorialService, dataList, authCriteria);
}

/** 登録(認可設定無視) */
@Override
public int entryIgnoreAuthCompany(List<MasterTutorialManagerDto> dataList) throws KaidenDuplicationEntryException {
    return super.entryIgnoreAuthCompany(k99mTutorialService, dataList);
}

/** 削除 */
@Override
public int remove(MasterTutorialManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException {
    return super.remove(k99mTutorialService, criteria, authCriteria);
}

/** 削除(認可設定無視) */
@Override
public int removeIgnoreAuthCompany(MasterTutorialManagerDto criteria) {
    return super.removeIgnoreAuthCompany(k99mTutorialService, criteria);
}

/** 更新 */
@Override
public int renew(List<MasterTutorialManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    return super.renew(k99mTutorialService, dataList, authCriteria);
}

/** 更新(認可設定無視) */
@Override
public int renewIgnoreAuthCompany(List<MasterTutorialManagerDto> dataList) throws KaidenOptimisticLockException {
    return super.renewIgnoreAuthCompany(k99mTutorialService, dataList);
}

/** 期間検索 */
@Override
public List<MasterTutorialManagerDto> selectTerms(MasterTutorialManagerDto criteria,
    AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k99mTutorialService, criteria, authCriteria);
}

/** 期間検索(認可設定無視) */
@Override
public List<MasterTutorialManagerDto> selectTermsIgnoreAuthCompany(MasterTutorialManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k99mTutorialService, criteria);
}

/** 基準日指定検索 */
@Override
public List<MasterTutorialManagerDto> selectListIncludesSearchDate(MasterTutorialManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k99mTutorialService, criteria, authCriteria, order);
}

/** 基準日指定検索(認可設定無視) */
@Override
public List<MasterTutorialManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(
    MasterTutorialManagerDto criteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k99mTutorialService, criteria, order);
}

/** 期間分割 */
@Override
public void divideTerm(MasterTutorialManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
    super.divideTerm(k99mTutorialService, criteria, divideDate, authCriteria);
}

/** 期間分割(認可設定無視) */
@Override
public void divideTermIgnoreAuthCompany(MasterTutorialManagerDto criteria, Date divideDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.divideTermIgnoreAuthCompany(k99mTutorialService, criteria, divideDate);
}

/** 期間変更 */

```

```

@Override
public void moveTerm(MasterTutorialManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k99mTutorialService, criteria, newStartDate, newEndDate, authCriteria);
}

/** 期間変更(認可設定無視) */
@Override
public void moveTermIgnoreAuthCompany(MasterTutorialManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k99mTutorialService, criteria, newStartDate, newEndDate);
}

/** DTOのクラスを返却. */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/** 重複チェック確認フィールドを返却. */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {"companyCd", "tutorialCd"});
}

/** 登録時入力チェック. */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/** 更新時入力チェック. */
@Override
public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/** 登録時入力チェック. */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    result.add(new ValidateDefsDto("companyCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("companyCd", VALID_COMPANY_EXISTS, "companyCd", "", "true"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialCd", VALID_MAX_LENGTH, "100"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("localeId", VALID_REQUIRED));
    result.add(new ValidateDefsDto("localeId", VALID_LOCALE));
    result.add(new ValidateDefsDto("tutorialName", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialName", VALID_MAX_LENGTH, "250"));
    result.add(new ValidateDefsDto("startDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("startDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("endDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_DATE_MIN, "startDate"));
    result.add(new ValidateDefsDto("deleteFlag", VALID_FLAG));
    return result;
}

/** 更新時入力チェック. */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    result.add(new ValidateDefsDto("companyCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("companyCd", VALID_COMPANY_EXISTS, "companyCd", "", "true"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialCd", VALID_MAX_LENGTH, "100"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("termCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("termCd", VALID_MAX_LENGTH, "50"));
    result.add(new ValidateDefsDto("termCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("localeId", VALID_REQUIRED));
    result.add(new ValidateDefsDto("localeId", VALID_LOCALE));
    result.add(new ValidateDefsDto("tutorialName", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialName", VALID_MAX_LENGTH, "250"));
    result.add(new ValidateDefsDto("startDate", VALID_REQUIRED));

```

```

result.add(new ValidateDefsDto("startDate", VALID_DATE));
result.add(new ValidateDefsDto("endDate", VALID_REQUIRED));
result.add(new ValidateDefsDto("endDate", VALID_DATE));
result.add(new ValidateDefsDto("endDate", VALID_DATE_MIN, "startDate"));
return result;
}
}

```

コラム

Managerクラスのメソッドにはそれぞれ「会社認可設定を含める」「会社認可設定を無視する」の2種類存在します。
チュートリアルマスタメンテナンス機能では「会社認可設定を含める」メソッドを使用します。

コラム

Actionクラス⇒Logicクラス⇒Managerクラスが処理の流れとなりますが、
ほとんどの処理は intra-mart Accel Kaiden! 内部の基底クラスで行っています。
ActionクラスメソッドとManagerクラスメソッドのマッピングを以下に示します。

Actionクラスメソッド	処理	Managerクラスメソッド
search	検索画面 検索	selectListKeywordFilter
select	メンテナンス画面 表示	selectLocales
entry	メンテナンス画面 登録	entry
toDisabled	メンテナンス画面 無効化	toDisabled
remove	メンテナンス画面 削除	remove
renew	メンテナンス画面 更新	renew
selectTerm	メンテナンス画面 期間データ取得	selectTerms
toEnabled	メンテナンス画面 有効化	toEnabled
divideTerm	メンテナンス画面 期間分割	divideTerm
moveTerm	メンテナンス画面 期間変更	moveTerm

ManagerDtoの作成

jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto を継承してMasterTutorialManagerDtoクラスを作成します。
対象テーブルの列名と、同時に取得する列名がある場合は定義します。


```

package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

public class MasterTutorialManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;
    /** チュートリアルコード. */
    public String tutorialCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールid. */
    public String localeId;
    /** チュートリアル名称. */
    public String tutorialName;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 削除フラグ. */
    public String deleteFlag;
    /** 予備項目A. */
    public String extensionA;
    /** 予備項目B. */
    public String extensionB;
    /** 予備項目C. */
    public String extensionC;
    /** 予備項目D. */
    public String extensionD;
    /** 予備項目E. */
    public String extensionE;
    /** 予備項目F. */
    public String extensionF;
    /** 予備項目G. */
    public String extensionG;
    /** 予備項目H. */
    public String extensionH;
    /** 予備項目I. */
    public String extensionI;
    /** 予備項目J. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 会社名称(他マスタ名称). */
    public String companyName;
}

```



コラム

ManagerDtoクラスは検索条件の設定、取得結果の保持で使います。

SQLファイルの作成

Managerのキーワード検索で使用するSQLファイルを作成します。

配置場所は「WEB-INF/classes/META-INF/sql/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99mTutorialService」です。

ファイル名はselectView.sqlとします。(上記マネージャで使用しているファイル名です)

```

select
    tutorial.company_cd
, tutorial.tutorial_cd
, tutorial.term_cd
, tutorial.locale_id
, tutorial.tutorial_name
, tutorial.start_date
, tutorial.end_date
, tutorial.delete_flag
, tutorial.extension_a
, tutorial.extension_b
, tutorial.extension_c
, tutorial.extension_d
, tutorial.extension_e
, tutorial.extension_f

```

```

, tutorial.extension_g
, tutorial.extension_h
, tutorial.extension_i
, tutorial.extension_j
, tutorial.entry_ts
, tutorial.entry_user_cd
, tutorial.renew_cnt
, tutorial.renew_ts
, tutorial.renew_user_cd
, dept.department_name as company_name
from k99m_tutorial tutorial
left join ( select imm_department.company_cd
, imm_department.department_set_cd
, imm_department.department_cd
, imm_department.locale_id
, imm_department.start_date
, imm_department.end_date
, imm_department.department_name
from imm_department) dept
on tutorial.company_cd = dept.company_cd
and tutorial.company_cd = dept.department_set_cd
and tutorial.company_cd = dept.department_cd
and tutorial.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/
and dept.end_date > /*searchCriteriaDate*/
--ELSE and dept.start_date <= tutorial.start_date
--ELSE and dept.end_date > tutorial.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
tutorial.company_cd in /*companyCd*/(
/*END*/
/*IF tutorialCd != null*/
and tutorial.tutorial_cd = /*tutorialCd*/
/*END*/
/*IF termCd != null*/
and tutorial.term_cd = /*termCd*/
/*END*/
/*IF localeId != null*/
and tutorial.locale_id = /*localeId*/
/*END*/
/*IF tutorialName != null*/
and tutorial.tutorial_name like /*tutorialName*/%' ESCAPE '$'
/*END*/
/*IF searchCriteriaDate != null*/
and tutorial.start_date <= /*searchCriteriaDate*/
and tutorial.end_date > /*searchCriteriaDate*/
/*END*/
/*IF deleteFlag != null*/
and tutorial.delete_flag = /*deleteFlag*/
/*END*/
/*IF extensionA != null*/
and tutorial.extension_a like /*extensionA*/%' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and tutorial.extension_b like /*extensionB*/%' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and tutorial.extension_c like /*extensionC*/%' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and tutorial.extension_d like /*extensionD*/%' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and tutorial.extension_e like /*extensionE*/%' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and tutorial.extension_f like /*extensionF*/%' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
and tutorial.extension_g like /*extensionG*/%' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
and tutorial.extension_h like /*extensionH*/%' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
and tutorial.extension_i like /*extensionI*/%' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
and tutorial.extension_j like /*extensionJ*/%' ESCAPE '$'
/*END*/
/*IF companyName != null*/
and dept.department_name like /*companyName*/' ESCAPE '$'
/*END*/

```

```
/*END*/  
/*IF orderBy != null*/  
ORDER BY /*$orderBy*/company_cd  
/*END*/
```

diconファイルの作成

項目

- [kaiden_tutorial.diconの作成](#)
- [kaiden.diconの作成](#)
- [convention.propertiesの作成](#)

[kaiden_tutorial.diconの作成](#)

diconファイルを作成します。配置場所は「resources/」です。

プロジェクト名を「kaiden_tutorial」としましたので、ファイル名を「kaiden_tutorial.dicon」とします。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">

<components>

  <!-- manager -->
  <component name="masterTutorialManager" class="jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl.MasterTutorialManagerImpl" instance="prototype"></component>

  <!-- helper -->

  <!-- validator -->

  <!-- service -->
  <component name="k99mTutorialService" class="jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99mTutorialService" instance="prototype"></component>
</components>
```



コラム

チュートリアルマスタメンテナンス機能では、DiconファイルにはManagerクラスとServiceクラスを定義します。

[kaiden.diconの作成](#)

「kaiden.dicon」を作成します。配置場所は「webapp/WEB-INF/classes/」です。

作成後、上記で作成したdiconファイルを追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN" "http://www.seasar.org/dtd/components24.dtd">
<components xmlns:xi="http://www.w3.org/2001/XInclude">
  <include path="s2jdbc.dicon"/>
  <component name="sqlLogRegistry" class="org.seasar.extension.jdbc.SqlLogRegistry">@org.seasar.extension.jdbc.SqlLogRegistryLocator@getInstance()</component>

  <xi:include href="/kaiden_base.dicon" />
  <xi:include href="/kaiden_extension_imw.dicon" />
  <xi:include href="/kaiden_product_workflow.dicon" />
  <!-- ↓を追加します -->
  <xi:include href="/kaiden_tutorial.dicon" />
  <!-- ↑を追加します -->
</components>
```



コラム

「kaiden_tutorial.dicon」の設定後、「app.dicon」に設定を追加します。

[convention.propertiesの作成](#)

「convention.properties」を作成します。配置場所は「resources/META-INF/」です。

```
root.package.name=jp.co.slcs.kaiden2.tutorial.feature
```

サーフェス定義の作成

項目

- プロパティークラスの作成
- サーフェス定義ファイルの作成

プロパティークラスの作成

jp.co.slcs.kaiden2.tutorial.foundation.conf.TutorialPropクラスを作成します。
サーフェス定義ファイルから取得する際のキーとなります。

```
package jp.co.slcs.kaiden2.tutorial.foundation.conf;

public final class TutorialProp {

    /** 定数. */
    public static final class C {

    }

    /** メッセージ. */
    public static final class M {

        /** チュートリアルマスタ関連. */
        public static final class Tutorial {

        }

    }

    /** サーフェス. */
    public static final class S {

        /** 会社コード. */
        public static final String COMPANY = "TTRL.company";
        /** 開始日. */
        public static final String START_DATE = "TTRL.startDate";
        /** 終了日. */
        public static final String END_DATE = "TTRL.endDate";
        /** チュートリアル. */
        public static final String TUTORIAL = "TTRL.tutorial";
        /** チュートリアルコード. */
        public static final String TUTORIAL_CD = "TTRL.tutorialCd";
        /** チュートリアル名. */
        public static final String TUTORIAL_NAME = "TTRL.tutorialName";

        /** チュートリアルマスタ関連. */
        public static final class Tutorial {

            /** チュートリアルマスタメンテナンス画面タイトル. */
            public static final String MAINT_TITLE = "TTRL.master.tutorial.title";

        }

    }

}
```

サーフェス定義ファイルの作成

サーフェス定義ファイルを作成します。配置場所は「/WEB-INF/classes/jp/co/slcs/kaiden2/」です。
以下をベースにして「tutorial_surfaces_ja.xml」「tutorial_surfaces_en.xml」「tutorial_surfaces_zh_CN.xml」の3ファイルを作成します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>

<!-- チュートリアル -->
<entry key="TTRL.company">会社コード</entry>
<entry key="TTRL.startDate">開始日</entry>
<entry key="TTRL.endDate">終了日</entry>
<entry key="TTRL.tutorial">チュートリアル</entry>
<entry key="TTRL.tutorialCd">チュートリアルコード</entry>
<entry key="TTRL.tutorialName">チュートリアル名</entry>

<!-- チュートリアルマスタメンテナンス画面 -->
<entry key="TTRL.master.tutorial.title">チュートリアルマスタメンテナンス</entry>

</properties>
```

マスタメンテナンス機能設定ファイルの作成

マスタメンテナンス機能の設定ファイルを作成します。

ファイル名をMasterTutorialMasterMaintenanceSettings.xmlとし、「/WEB-INF/classes/jp/co/slcs/kaiden2/tutorial/feature/action/kaiden/master」に配置します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericMasterMaintenanceSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterMaintenanceSettings">
  <!-- 画面タイトルラベルID -->
  <titleSurfaceKey>TTRL.master.tutorial.title</titleSurfaceKey>
  <!-- 検索画面JSPフルパス -->
  <searchGadgetJsp></searchGadgetJsp><!-- デフォルトを利用 -->
  <!-- メンテナンス画面JSPフルパス -->
  <maintenanceGadgetJsp>/WEB-INF/view/kaiden2/tutorial/master/masterTutorial/masterTutorialMainte.jsp</maintenanceGadgetJsp>
  <!-- AjaxAccessUrl -->
  <ajaxAccessUrl>kaiden/master/masterTutorial</ajaxAccessUrl>
  <!-- 1ページあたりの件数(0以下はページングせずすべて表示) -->
  <limit>20</limit>
  <!-- 検索条件の検索基準日 使用/不使用 -->
  <useSearchCriteriaDate>true</useSearchCriteriaDate>
  <!-- 検索条件の検索基準日 必須/任意 -->
  <requiredSearchCriteriaDate>true</requiredSearchCriteriaDate>
  <!-- 検索条件の検索基準日 デフォルト値(システム日付) 設定/非設定(必須の場合にはこの設定にかかわらずデフォルト値が設定される) -->
  <defaultSearchCriteriaDate>true</defaultSearchCriteriaDate>
  <!-- 検索条件の会社 使用/不使用 -->
  <useSearchCompany>true</useSearchCompany>
  <!-- 検索条件のロケール 使用/不使用 -->
  <useSearchLocale>true</useSearchLocale>
  <!-- 検索条件のステータス無効データも検索。使用/不使用 -->
  <useSearchWithDisabled>true</useSearchWithDisabled>
  <!-- 汎用一覧・検索画面の行番号ラベルID -->
  <rownumberSurfaceKey></rownumberSurfaceKey><!-- デフォルトを利用 -->
  <!-- 汎用一覧・検索画面のカラム設定 -->
  <listCols>
    <listCol>
      <!-- 対象マスタのフィールドキー -->
      <fieldKey>tutorialCd</fieldKey>
      <!-- キーワード検索対象とするか否か -->
      <keywordSearchTarget>true</keywordSearchTarget>
      <!-- 表示幅 -->
      <width>150px</width>
      <!-- 表示位置 -->
      <align></align>
      <!-- ヘッダラベルID -->
      <headerSurfaceKey>TTRL.tutorialCd</headerSurfaceKey>
      <!-- ソート可能か否か -->
      <sortable>true</sortable>
    </listCol>
    <listCol>
      <fieldKey>tutorialName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width></width>
      <align></align>
      <headerSurfaceKey>TTRL.tutorialName</headerSurfaceKey>
      <sortable>true</sortable>
    </listCol>
    <listCol>
      <fieldKey>deleteFlag</fieldKey>
      <keywordSearchTarget>false</keywordSearchTarget>
      <width>100px</width>
      <align>center</align>
      <headerSurfaceKey>BASE.status</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
  </listCols>
  <!-- 検索・一覧画面のソート順(汎用マスター一覧は、第二ソートキーまで有効)-->
  <orders>
    <order>
      <fieldKey>tutorialCd</fieldKey>
      <type>asc</type>
    </order>
    <order>
      <fieldKey>tutorialName</fieldKey>
      <type>asc</type>
    </order>
  </orders>
  <!-- 対象マスタの主キーフィールド -->
  <pkFields>
    <pkField>companyCd</pkField>
    <pkField>tutorialCd</pkField>
    <pkField>termCd</pkField>
    <pkField>localeId</pkField>
  </pkFields>
</genericMasterMaintenanceSettings>
```



コラム

「アクションクラスの同一パッケージ」で「クラス名の先頭（Actionを除去） + "MasterMaintenanceSettings.xml"」がマスタメンテナンス機能設定ファイルとして認識されます。



コラム

マスタメンテナンス機能設定ファイルで設定する内容の一部を以下に挙げます。

- 画面出力項目の定義
- ソート順
- 対象マスタの主キーフィールド

その他、上記のサンプルのコメントを元に設定内容を確認ください。

Formクラスの作成

jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm を継承してMasterTutorialFormクラスを作成します。
intra-mart Accel Kaiden! 標準のマスタメンテナンス機能のみ使用する場合、空実装で構いません。

```
package jp.co.slcs.kaiden2.tutorial.feature.form.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm;

public class MasterTutorialForm extends AbstractTermMasterMaintenanceForm {

}
```



コラム

基底クラスでは主に一覧画面の設定情報を保持しています。

Actionクラスの作成

jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceAction を継承してMasterTutorialActionクラスを作成します。
基本的な処理はこの基底クラスに実装されています。
intra-mart Accel Kaiden! 標準のマスタメンテナンス機能が使用できればよい場合、各処理のメソッドは基底クラスの同名メソッドを呼び出すだけです。

```
package jp.co.slcs.kaiden2.tutorial.feature.action.kaiden.master;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceDto;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceAction;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.tutorial.feature.dto.kaiden.master.MasterTutorialDto;
import jp.co.slcs.kaiden2.tutorial.feature.form.kaiden.master.MasterTutorialForm;
import jp.co.slcs.kaiden2.tutorial.feature.logic.kaiden.master.MasterTutorialLogic;
import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;

public class MasterTutorialAction extends AbstractTermMasterMaintenanceAction {

    /** Dto. */
    @Resource
    public MasterTutorialDto masterTutorialDto;

    /** Form. */
    @ActionForm
    @Resource
    public MasterTutorialForm masterTutorialForm;

    /** Logic. */
    @Resource
    public MasterTutorialLogic masterTutorialLogic;

    /** 入力ページパスを返却します. */
    @Override
    @Execute(validator = false)
    // @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
    public String index() throws KaidenSystemException {
        // プロパティファイルからマスタメンテナンス機能の設定を読み込み
        MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
        masterTutorialDto.settings = new MasterMaintenanceSettingsDto(reader.read(MasterTutorialAction.class));

        // 入力ページパスを返却します
        return super.index();
    }
}
```

```

}

/** 検索. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String search() {
    // 汎用検索画面の検索処理を行います
    return super.search();
}

/** メンテナンス画面表示. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String select() {
    // メンテナンス画面表示処理を行います
    return super.select();
}

/** 登録. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String entry() {
    // メンテナンス画面登録処理を行います
    return super.entry();
}

/** 無効化. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String toDisabled() {
    // メンテナンス画面無効化処理を行います
    return super.toDisabled();
}

/** 削除. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String remove() {
    // メンテナンス画面削除処理を行います
    return super.remove();
}

/** 更新. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String renew() {
    // メンテナンス画面更新処理を行います
    return super.renew();
}

/** 期間データ取得. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String selectTerm() {
    // メンテナンス画面期間データ取得処理を行います
    return super.selectTerm();
}

/** 有効化. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String toEnabled() {
    // メンテナンス画面有効化処理を行います
    return super.toEnabled();
}

/** 期間分割. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
public String divideTerm() {
    // メンテナンス画面期間分割処理を行います
    return super.divideTerm();
}

/** 期間変更. */
@Override
@Execute validator = false
// @Authz(uri = "service://kaiden/master/tutorial", action = "execute")

```



```
// @Override - ServiceMasterMaintenanceDto, action = execute /
public String moveTerm() {
    // メンテナンス画面期間変更処理を行います
    return super.moveTerm();
}

/** Dtoを取得. */
@Override
protected AbstractMasterMaintenanceDto getDto() {
    // 業務固有のDtoを取得します。基底クラスより呼ばれます。
    return masterTutorialDto;
}

/** Formを取得. */
@Override
protected AbstractMasterMaintenanceForm getForm() {
    // 業務固有のFormを取得します。基底クラスより呼ばれます。
    return masterTutorialForm;
}

/** Logicを取得. */
@Override
protected AbstractMasterMaintenanceLogic getLogic() {
    // 業務固有のLogicを取得します。基底クラスより呼ばれます。
    return masterTutorialLogic;
}
}
```



コラム

マスタメンテナンスのActionクラスの詳細は「[マスタメンテナンスActionクラスプログラミング](#)」を参照ください。



コラム

@Authz アノテーションにて実行メソッドと認可を紐づけますが、本チュートリアルではコメントアウトしています。
認可設定についての詳細は intra-mart Accel Platform のドキュメントを参照ください。

Dtoクラスの作成

jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceDto を継承してMasterTutorialDtoクラスを作成します。
intra-mart Accel Kaiden! 標準のマスタメンテナンス機能のみ使用する場合、空実装で構いません。

```
package jp.co.slcs.kaiden2.tutorial.feature.dto.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceDto;

public class MasterTutorialDto extends AbstractTermMasterMaintenanceDto {

}
```



コラム

基底クラスでは主に一覧画面の設定情報を保持しています。

Logicクラスの作成

jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceLogic を継承してMasterTutorialLogicクラスを作成します。

```

package jp.co.slcs.kaiden2.tutorial.feature.logic.kaiden.master;

import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.base.foundation.master.MasterManager;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.tutorial.feature.action.kaiden.master.MasterTutorialAction;
import jp.co.slcs.kaiden2.tutorial.foundation.conf.TutorialProp;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.MasterTutorialManager;

public class MasterTutorialLogic extends AbstractTermMasterMaintenanceLogic {

    /** Manager. */
    @Resource
    public MasterTutorialManager masterTutorialManager; // 業務固有のManagerをDILします

    /** Managerを取得. */
    @Override
    protected MasterManager<?> getManager() {
        // 業務固有のManagerを取得します。基底クラスより呼ばれます。
        return masterTutorialManager;
    }

    /** キーワード検索対象カラムを返却. */
    @Override
    protected String[] getKeywordSearchTargetCol() {
        // プロパティファイルからマスタメンテナンス機能の設定を読み込み
        MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
        MasterMaintenanceSettingsDto settings;
        try {
            settings = new MasterMaintenanceSettingsDto(reader.read(MasterTutorialAction.class));
        } catch (KaidenSystemException e) {
            throw KaidenRuntimeException.makeInstance(e);
        }
        // キーワード検索対象カラムをサブクラスより返却します
        return settings.searchTargetCol.toArray(new String[] {});
    }

    /** 国際化対応するフィールドを返却. */
    @Override
    protected List<String> get118nTargetField() {
        // 国際化対応するフィールドをサブクラスより返却します
        return Arrays.asList("tutorialName");
    }

    /** 登録・更新用の入力データをサブクラスより返却. */
    @Override
    protected Map<String, Object> getInputData(AbstractMasterMaintenanceForm form) {
        // 登録・更新用の入力データをサブクラスより返却します
        return JsonUtil.getMap(form.updateDataJson, Arrays.asList("inputArea"));
    }

    /** サーフェス定義マップを返却. */
    @Override
    protected Map<String, String> getSurfaces() {
        // サーフェス定義
        // 入力チェックエラー発生時のエラーメッセージラベルに使用します。
        Map<String, String> result = new HashMap<String, String>();
        result.put("companyCd", propertyHelper.getSurface(TutorialProp.S.COMPANY));
        result.put("tutorialCd", propertyHelper.getSurface(TutorialProp.S.TUTORIAL_CD));
        result.put("startDate", propertyHelper.getSurface(TutorialProp.S.START_DATE));
        result.put("endDate", propertyHelper.getSurface(TutorialProp.S.END_DATE));
        result.put("tutorialName", propertyHelper.getSurface(TutorialProp.S.TUTORIAL_NAME));
        return result;
    }
}

```

コラム

マスタメンテナンスのLogicクラスの詳細は「[サーバサイド](#)」を参照ください。

JSPの作成

/WEB-INF/view/kaiden2/tutorial/master/masterTutorial/masterTutorialMainte.jspを作成します。

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_base" uri="http://kaiden.slcs.co.jp/taglib/listdata/base" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.tutorial.foundation.conf.TutorialProp" %>

<%-- product_workflow 共通インクルード --%>
<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<%-- 汎用マスター一覧・検索 --%>
<%-- 設定 --%>
<c:set var="gadgetClass">tutorialMainte</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="gadgetTitle">${maintenanceDto.settings.title}</c:set>
<c:set var="accessUrl">${maintenanceDto.settings.ajaxAccessUrl}</c:set>

<c:set var="surfaceStartDate"><%= BaseProp.S.START_DATE %></c:set>
<c:set var="surfaceConcatChar"><%= BaseProp.S.CONCAT_CHAR %></c:set>
<c:set var="surfaceEndDate"><%= BaseProp.S.END_DATE %></c:set>
<c:set var="surfaceCompany"><%= BaseProp.S.COMPANY %></c:set>
<c:set var="surfaceTutorialCd"><%= TutorialProp.S.TUTORIAL_CD %></c:set>
<c:set var="surfaceTutorialName"><%= TutorialProp.S.TUTORIAL_NAME %></c:set>

<%-- バリデーション --%>
<c:set var="validStartDate_def">{k_required:[], k_date:[]}</c:set>
<c:set var="validEndDate_def">{k_required:[], k_date:[], k_dateMin:['startDate']}</c:set>
<c:set var="validCompanyCd_def">{k_required:[], k_companyExists:['companyCd', '', 'true']}</c:set>
<c:set var="validTutorialCd_def">{k_required:[], k_kaidenId:[], k_maxLength:['100']}</c:set>
<c:set var="validTutorialName_def">{k_required:[], k_maxLength:['250']}</c:set>

<%-- 期間一覧ガジェット --%>
<div class="kaiden-ui-master-term-list">
  <jsp:include page="/WEB-INF/view/kaiden2/base/foundation/gadget/genericMasterTermList.jsp" flush="true">
    <jsp:param name="gadget_instance" value="0" />
  </jsp:include>
</div>
<div class="kaiden-ui-master-maintenance-area">
  <k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${gadgetTitle}"
    gadgetId="${gadgetId}"
    hideGadgetTitle="true"
    class="kaiden-ui-master-maintenance-input-area">

    <k:blockContainer id="inputArea">
      <form id="inputForm" onsubmit="return false;">
        <table class="mui-form" style="table-layout:fixed">
          <colgroup>
            <col style="width:170px;" />
            <col />
          </colgroup>
          <tr style="height:30px;">
            <th>
              <k:surface surfaceKey="${surfaceStartDate}" style="display:inline-block" requiredMark="true"/>
              <k:surface surfaceKey="${surfaceConcatChar}" style="display:inline-block" />
              <k:surface surfaceKey="${surfaceEndDate}" style="display:inline-block" requiredMark="true"/>
            </th>
            <td>
              <k:date name="startDate"
                mode="0"
                surfaceKey="${surfaceStartDate}"
                validationDefault="${validStartDate_def}"
                />
              <k:surface surfaceKey="${surfaceConcatChar}" style="display:inline-block;margin-left:1em;margin-right:1em;" />
              <k:date name="endDate"
                mode="0"
                surfaceKey="${surfaceEndDate}"
                validationDefault="${validEndDate_def}"
                />
            </td>
          </tr>
          <tr>
            <th>
              <k:surface surfaceKey="${surfaceCompany}" requiredMark="true"/>
            </th>
            <td>
              <k:surface surfaceKey="${surfaceTutorialCd}" requiredMark="true"/>
            </td>
          </tr>
        </table>
      </form>
    </k:gadgetContainer>
  </div>

```

```

</td>
<td>
    <klist_base:listDataCompany name="readParmitCompany" type="readParmitCompany" blank="false"/>
    <k:select name="companyCd"
        mode="0"
        validationDefault="{validCompanyCd_def}"
        surfaceKey="{surfaceCompany}"
        dataName="readParmitCompany"
        style="width:300px;"
        includeDisabled="true"/>
</td>
</tr>
<tr>
<th>
    <k:surface surfaceKey="{surfaceTutorialCd}" requiredMark="true"/>
</th>
<td>
    <k:codeText name="tutorialCd"
        mode="0"
        style="width:300px;"
        surfaceKey="{surfaceTutorialCd}"
        validationDefault="{validTutorialCd_def}"
        maxLength="100"
    />
</td>
</tr>
<tr>
<th>
    <k:surface surfaceKey="{surfaceTutorialName}" requiredMark="true"/>
</th>
<td>
    <k:text-locales name="tutorialName"
        mode="0"
        style="width:300px;"
        surfaceKey="{surfaceTutorialName}"
        validationDefault="{validTutorialName_def}"
        imeMode="on"
        maxLength="250"
    />
</td>
</tr>
<tr>
    <k:hidden name="renewCnt" mode="0"/>
    <k:hidden name="termCd" mode="0"/>
    <k:hidden name="deleteFlag" mode="0"/>
    <k:hidden name="pk_json" mode="0"/>
</tr>
</table>
</form>
</k:blockContainer>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {

var eventMan = KAIDEN.getGadget("${gadgetId}").eventManager
, inputAreaAccessor = KAIDEN.getGadget("${gadgetId}").blockAccessors["inputArea"];

//trunsNewEntry(新規入力画面表示) イベントリスナの登録
eventMan.registPushListener("trunsNewEntry", function(**Object*/arg) {
    //入力欄を一括クリア
    KAIDEN.getGadget("${gadgetId}").clearInput("inputArea");

    //開始日・終了日にシステム開始日・システム終了日を設定
    inputAreaAccessor("startDate").setValue("${k:getSystemStartDate()}");
    inputAreaAccessor("endDate").setValue("${k:getSystemEndDateDecrement()}");

    //開始日・終了日、会社、チュートリアルコードをdisabled
    inputAreaAccessor("startDate").removeDisabled();
    inputAreaAccessor("endDate").removeDisabled();
    inputAreaAccessor("companyCd").removeDisabled();
    inputAreaAccessor("tutorialCd").removeDisabled();

    if (inputAreaAccessor("companyCd").elem().children().length > 0) {
        inputAreaAccessor("companyCd").elem().get(0).selectedIndex = 0;
    }

    //会社をフォーカス
    inputAreaAccessor("companyCd").elem().focus();

});

//loadData(表示対象データのロード) イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
    var ajax = KAIDEN.Ajax()
    , criteria = {}
    , stat = false;

    criteria.criteriaJson = arg;

```

```

ajax.postSyncAuto("${accessUrl}/select", criteria
, function(returnData) {
    //正常系
    //メンテナンス画面にデータセット
    for (var key in returnData.mainteTarget) {
        inputAreaAccessor(key).setValueIgnore(returnData.mainteTarget[key]);
    }
    inputAreaAccessor("pk_json").setValue(arg); //主キーを保存
    stat = true;
}
, function(returnData) {
    stat = false;
}
);
return stat;
});

//trunsRenewEntry (更新画面表示) イベントリスナの登録
eventMan.registPushListener("trunsRenewEntry", function(**Object*/arg) {
    //開始日・終了日、会社、チュートリアルコードをdisabled
    inputAreaAccessor("startDate").disabled();
    inputAreaAccessor("endDate").disabled();
    inputAreaAccessor("companyCd").disabled();
    inputAreaAccessor("tutorialCd").disabled();

    $("div[data-kaiden-gadget-id=${gadgetId}"] input[data-kaiden-org-name^='tutorialName']").eq(0).focus();

});

//currentPrimaryInfo (現在表示中データの主キーを要求) イベントリスナの登録
eventMan.registPullListener("claimCurrentDataPk", function(**Object*/arg) {
    return inputAreaAccessor("pk_json").getValue(); //主キーを返却
});

});
</script>
<!-- マスタメンテナンス汎用ボタンガジェット --%>
<div style="clear:both;">
<jsp:include page="/WEB-INF/view/kaiden2/base/foundation/gadget/genericMasterOperationButton.jsp" flush="true">
    <jsp:param name="gadget_instance" value="0" />
    <jsp:param name="maintenanceGadgetId" value="${gadgetId}" />
</jsp:include>
</div>
</div>

```

コラム

ファイルパスは「マスタメンテナンス機能設定ファイル」で設定します。
[マスタメンテナンス機能設定ファイルの作成](#) を参照ください。

サーバサイド

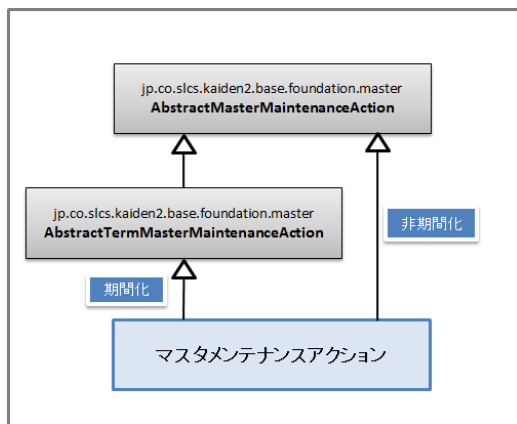
本項では、マスタメンテナンスのサーバサイドのプログラミング方法について解説します。

- マスタメンテナンスActionクラスプログラミング
 - 構成
 - マスタメンテナンス初期表示処理
 - 各種処理
- マスタメンテナンスLogicクラスプログラミング
 - 構成
 - キーワード検索対象カラム返却
 - 国際化対応フィールド返却
 - 入力データ返却
 - サーフェス定義マップ返却
- マスタメンテナンスManagerクラスプログラミング
 - 構成
 - 各種API
 - 重複チェックフィールド返却
 - バリデート処理

マスタメンテナンスActionクラスプログラミング

マスタメンテナンスActionクラスのサーバサイドプログラミングについて解説します。

構成



マスタメンテナンス初期表示処理

マスタメンテナンス機能の初期表示処理をアクションクラスのindexメソッドに実装します。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのアクションはこちら](#))

■ プログラミング例

```
/**
 * チュートリアルマスタメンテナンス (Action).
 */
public class MasterTutorialAction extends AbstractTermMasterMaintenanceAction {

    // 中略...

    /** 入力ページパスを返却します. */
    @Override
    @Execute(validator = false)
    // @Authz(uri = "service://kaiden/master/tutorial", action = "execute")
    public String index() throws KaidenSystemException {
        // プロパティファイルからマスタメンテナンス機能の設定を読み込み
        MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
        // 設定読み込み処理にActionクラスを引数に渡します
        masterTutorialDto.settings = new MasterMaintenanceSettingsDto(reader.read(MasterTutorialAction.class));

        // 入力ページパスを返却します
        return super.index();
    }

    // 中略...
}
```

■ 解説

マスタメンテナンス機能設定ファイルを読み込み、MasterTutorialDto.settingsにセットします。
設定読み込み処理にチュートリアルマスタメンテナンスActionクラスを渡すことで、
intra-mart Accel Kaiden!基底処理でチュートリアルマスタのマスタメンテナンス機能設定ファイルを特定し、読み込んでいます。

**コラム**

「アクションクラスの同一パッケージ」で「クラス名の先頭（Actionを除去） + "MasterMaintenanceSettings.xml"」がマスタメンテナンス機能設定ファイルとして認識されます。

**コラム**

マスタメンテナンス機能設定ファイルの内容は「[マスタメンテナンス機能設定ファイル](#)」を参照ください。

**コラム**

return super.index();は intra-mart Accel Kaiden!基底クラスが、intra-mart Accel Kaiden!共通のマスタメンテナンス画面JSPのパスを返却しています。

各種処理

マスタメンテナンス機能の検索や登録に対応するメソッドに実装します。

チュートリアルプログラムは intra-mart Accel Kaiden!基底クラスのメソッドを呼び出しているだけなので、ここでは基底クラスの処理を紹介します。登録処理のみをピックアップします。

- プログラミング例

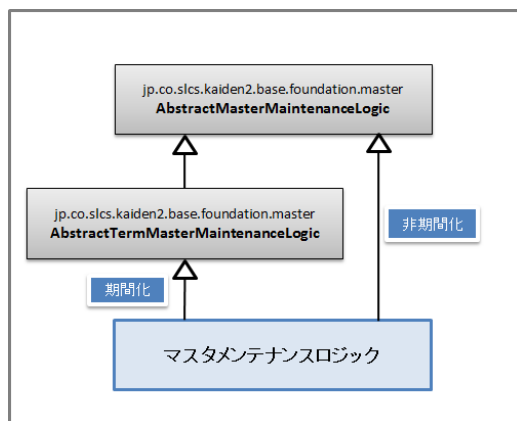
```
public String entry() {
    LogicResultDto result;
    try {
        result = getLogic().entryAjax(getForm());
    } catch (KaidenHaventAuthorityToCompanyException e) {
        result = LogicResultDto.makeErrorResult(e);
    } catch (KaidenDuplicationEntryException e) {
        result = LogicResultDto.makeErrorResult(e);
    }
    ResponseWriter.writeJson(JSON.encode(result));
    return null;
}
```

- 解説

マスタメンテナンスLogicクラスのメソッドを実行し、結果をJSON形式でレスポンスに出力しています。
ExceptionのキャッチはAction処理で行っています。
エラー結果をレスポンスに出力することで、クライアント処理はそれを解析し画面にエラーメッセージを表示します。

マスタメンテナンスLogicクラスプログラミング

マスタメンテナンスLogicクラスのサーバサイドプログラミングについて解説します。

構成**キーワード検索対象カラム返却**

マスタメンテナンスLogicクラスにキーワード検索対象カラム返却メソッドに実装します。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのロジックはこちら](#))

- プログラミング例

```

/** キーワード検索対象カラムを返却. */
@Override
protected String[] getKeywordSearchTargetCol() {
    // プロパティファイルからマスタメンテナンス機能の設定を読み込み
    MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
    MasterMaintenanceSettingsDto settings;
    try {
        settings = new MasterMaintenanceSettingsDto(reader.read(MasterTutorialAction.class));
    } catch (KaidenSystemException e) {
        throw KaidenRuntimeException.makeInstance(e);
    }
    // キーワード検索対象カラムをサブクラスより返却します
    return settings.searchTargetCol().toArray(new String[] {});
}

```

■ 解説

「マスタメンテナンス機能設定ファイル」で設定したファイルを読み込み、キーワード検索対象カラムの配列を返却しています。設定ファイルはActionクラスの初期表示処理で読み込んだファイルと同じものです。



コラム

チュートリアルマスタメンテナンスの設定ファイルでは「チュートリアルコード」、「チュートリアル名称」が返却されます。

国際化対応フィールド返却

マスタメンテナンスLogicクラスに国際化対応フィールド返却メソッドに実装します。

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのロジックはこちら](#))

■ プログラミング例

```

/** 国際化対応するフィールドを返却. */
@Override
protected List<String> getI18nTargetField() {
    // 国際化対応するフィールドをサブクラスより返却します
    return Arrays.asList("tutorialName");
}

```



コラム

チュートリアルマスタメンテナンスでは「チュートリアル名称」が国際化対応フィールドです。

入力データ返却

マスタメンテナンスLogicクラスに入力データ返却メソッドに実装します。

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのロジックはこちら](#))

■ プログラミング例

```

/** 登録・更新用の入力データをサブクラスより返却. */
@Override
protected Map<String, Object> getInputData(AbstractMasterMaintenanceForm form) {
    // 登録・更新用の入力データをサブクラスより返却します
    return JsonUtil.getMap(form.updateDataJson, Arrays.asList("inputArea"));
}

```



コラム

“inputArea”はJSPに記載した入力データエリアのキーです。

サーフェス定義マップ返却

マスタメンテナンスLogicクラスにサーフェス定義マップ返却メソッドに実装します。

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのロジックはこちら](#))

■ プログラミング例


```
/** サーフェス定義マップを返却. */
@Override
protected Map<String, String> getSurfaces() {
    // サーフェス定義
    // 入力チェックエラー発生時のエラーメッセージラベルに使用します。
    Map<String, String> result = new HashMap<String, String>();
    result.put("companyCd", propertyHelper.getSurface(TutorialProp.S.COMPANY));
    result.put("tutorialCd", propertyHelper.getSurface(TutorialProp.S.TUTORIAL_CD));
    result.put("startDate", propertyHelper.getSurface(TutorialProp.S.START_DATE));
    result.put("endDate", propertyHelper.getSurface(TutorialProp.S.END_DATE));
    result.put("tutorialName", propertyHelper.getSurface(TutorialProp.S.TUTORIAL_NAME));
    return result;
}
```

- 解説
サーフェス定義のMapを生成し、返却します。
ここで生成したMapはManagerクラスのバリデートメソッドに渡され、発生したエラーメッセージのラベルに使用されます。

マスタメンテナンスManagerクラスプログラミング

マスタメンテナンスManagerクラスのサーバサイドプログラミングについて解説します。

構成

構成については [Manager](#) を参照してください。

各種API

マスタメンテナンスManagerクラスに各種APIを実装します。
intra-mart Accel Kaiden!標準版のマスタマネージャは下記APIを提供しています。

- 機能一覧(非期間化マスタ、期間化マスタ共通)

機能概要	メソッド
主キー検索	select
主キー検索 (認可設定無視)	selectIgnoreAuthCompany
汎用検索	selectList
汎用検索 (認可設定無視)	selectListIgnoreAuthCompany
ロケール検索	selectLocales
ロケール検索 (認可設定無視)	selectLocalesIgnoreAuthCompany
参照検索	selectListWithReference
参照検索 (認可設定無視)	selectListWithReferenceIgnoreAuthCompany
キーワード検索	selectListKeywordFilter
キーワード検索 (認可設定無視)	selectListKeywordFilterIgnoreAuthCompany
有効化	toEnabled
有効化 (認可設定無視)	toEnabledIgnoreAuthCompany
無効化	toDisabled
無効化 (認可設定無視)	toDisabledIgnoreAuthCompany
登録	entry
登録 (認可設定無視)	entryIgnoreAuthCompany
削除	remove
削除 (認可設定無視)	removeIgnoreAuthCompany
更新	renew
更新 (認可設定無視)	renewIgnoreAuthCompany

- 機能一覧(期間化マスタ専用)

機能概要	メソッド
期間検索	selectTerms
期間検索 (認可設定無視)	selectTermsIgnoreAuthCompany
基準日指定検索	selectListIncludesSearchDate
基準日指定検索 (認可設定無視)	selectListIncludesSearchDateIgnoreAuthCompany
期間分割	divideTerm
期間分割 (認可設定無視)	divideTermIgnoreAuthCompany

機能概要	メソッド
期間変更	moveTerm
期間変更（認可設定無視）	moveTermIgnoreAuthCompany



コラム

intra-mart Accel Kaiden! マスタは「期間化マスタ」と「非期間化マスタ」の2種類が存在します。

開始日終了日を保持しているかどうかではありません。

- ・データ新規登録時、システムの開始日から終了日までの全期間のデータが、重複や間断なく連続して登録される。
- ・削除フラグを保持し、有効無効の切り替えを行うことができる

に該当すれば「期間化マスタ」、該当しなければ「非期間化マスタ」です。



コラム

各種APIの挙動については、それぞれのマスタマネージャの [intra-mart Accel Kaiden! JavaDoc](#) を参照してください。

重複チェックフィールド返却

マスタメンテナンスManagerクラスに重複チェックフィールド返却メソッドに実装します。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのマネージャはこちら](#))

■ プログラミング例

```
/** 重複チェック確認フィールドを返却. */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {"companyCd", "tutorialCd"});
}
```

■ 解説

会社コードとチュートリアル名称を返却しています。

Managerクラスの新規登録処理時、ここで返却するフィールドキーの重複チェック処理を intra-mart Accel Kaiden! 内部の基底クラスにて行っています。

バリデート処理

マスタメンテナンスManagerクラスにバリデート処理メソッドに実装します。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルマスタのマネージャはこちら](#))

■ プログラミング例

```

/** 登録時入力チェック. */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/** 更新時入力チェック. */
@Override
public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/** 登録時入力チェック. */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    result.add(new ValidateDefsDto("companyCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("companyCd", VALID_COMPANY_EXISTS, "companyCd", "", "true"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialCd", VALID_MAX_LENGTH, "100"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("localeId", VALID_REQUIRED));
    result.add(new ValidateDefsDto("localeId", VALID_LOCALE));
    result.add(new ValidateDefsDto("tutorialName", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialName", VALID_MAX_LENGTH, "250"));
    result.add(new ValidateDefsDto("startDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("startDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("endDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_DATE_MIN, "startDate"));
    result.add(new ValidateDefsDto("deleteFlag", VALID_FLAG));
    return result;
}

/** 更新時入力チェック. */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    result.add(new ValidateDefsDto("companyCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("companyCd", VALID_COMPANY_EXISTS, "companyCd", "", "true"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialCd", VALID_MAX_LENGTH, "100"));
    result.add(new ValidateDefsDto("tutorialCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("termCd", VALID_REQUIRED));
    result.add(new ValidateDefsDto("termCd", VALID_MAX_LENGTH, "50"));
    result.add(new ValidateDefsDto("termCd", VALID_KAIDEN_ID));
    result.add(new ValidateDefsDto("localeId", VALID_REQUIRED));
    result.add(new ValidateDefsDto("localeId", VALID_LOCALE));
    result.add(new ValidateDefsDto("tutorialName", VALID_REQUIRED));
    result.add(new ValidateDefsDto("tutorialName", VALID_MAX_LENGTH, "250"));
    result.add(new ValidateDefsDto("startDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("startDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_REQUIRED));
    result.add(new ValidateDefsDto("endDate", VALID_DATE));
    result.add(new ValidateDefsDto("endDate", VALID_DATE_MIN, "startDate"));
    return result;
}

```

■ 解説

新規登録前にvalidate4Entryメソッド、更新前にvalidate4Renewメソッドが実行されることを想定しています。
チュートリアルでは intra-mart Accel Kaiden! 基底クラスにて新規登録時と更新時に呼び出されています。(そのためチュートリアルのLogicには記載がありません。)

new ValidateDefsDto(“フィールドキー”, “バリデーションID”, 引数...)とすることで、バリデーションの定義ができます。
バリデーション定義をリストにセットしていき、makeValidateExecuter(dtoList, surfaces, list)にてバリデーションが実施されます。



コラム

バリデーション定義については [バリデーション](#) を参照してください。

マスタ検索

本項では、マスタ検索機能の概要を解説します。

機能説明

本項では、プロジェクトマスタ検索を参考に、検索機能の概要や構成を解説します。

概要

各機能で使用するマスタデータの検索を行う機能です。

intra-mart Accel Kaiden!のマスタメンテナンス機能をプログラミングする際には、次のモジュールが必要となります。

- intra-mart Accel Kaiden! 基盤モジュール

マスタ検索機能は大きく分けて「マスタ検索」「タグライブラリ」の2機能があり、
「マスタ検索」は、Ajax通信やサーバサイドの処理などで汎用的に利用されるマスタデータの検索機能、
「タグライブラリ」はJSPで簡単に「マスタ検索」を利用するための機能と定義しています。

構成

intra-mart Accel Kaiden!の検索機能は「タグライブラリ」から使用することができます。

プロジェクト検索では、次の様な画面構成となります。

- 検索時**
マスタデータの検索時の画面です。
テキストボックス左の検索アイコンをクリックするか、テキストボックス内でENTERキーを押すと、検索結果を表示します。
また、テキストボックス内にキーワードを入力した状態で検索を行うことで、検索結果の絞り込みが行えます。



No	プロジェクトコード	プロジェクト名	上位プロジェクト...	上位プロジェクト名
1	000001	〇〇情報システム株式会社 ...		
2	000002	社内環境整備プロジェクト		
3	000003	××社向け勤怠・経費申請...		
4	000004	△社 システム保守(2013...		
5	000005	システム更新対応(社内)		
6	000006	〇〇社 ワークフローシステム...		
7	000007	H25年度通常運用(問合...		
8	000008	■社 システム構築検討支援		

- 選択後**
マスタデータ選択後の画面です。



補足

検索機能(タグライブラリ)の検索結果、タグライブラリの属性について解説します。

- コーディング例**
検索機能(タグライブラリ)は次の様にコーディングします。

```
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>

<kmsw:project
  name      = "targetProjectName"
  mode      = "0"
  resultValue = "projectName"
  resultChain = "{targetCompanyCd:'companyCd', targetProjectCd:'projectCd', targetParentCompanyCd:'parentCompanyCd', targetParentProjectCd:'parentProjectCd', targetE
  searchCriteriaCompany = "comp_sample_01"
  searchCriteriaDate    = ""
  searchCriteriaProjectCd = "parentProjectCd"
  primaryTarget        = "parentCompanyCd,parentProjectCd"
  validationDefault    = "{k_required:['']}"
/>
<k:hidden name="targetCompanyCd" mode="0"/>
<k:hidden name="targetProjectCd" mode="0"/>
<k:hidden name="targetParentCompanyCd" mode="0"/>
<k:hidden name="targetParentProjectCd" mode="0"/>
<k:hidden name="targetDeleteFlag" mode="0"/>
```

- 検索結果**
多くの画面では、検索結果として名称のみを表示しておりますが、検索結果には名称以外にも多くの情報を返却しています。
各マスタ検索機能で返却される情報は、「[タグライブラリ - マスタ検索](#)」を参照してください。

コーディング例の様にプロジェクトマスタ検索機能を利用した場合は、マスタデータ選択時に次の情報が返却されます。
(コーディング例で使用している情報は太字で表しています。)

- **companyCd**(会社コード)
 - companyName(会社名)
 - **projectCd** (プロジェクトコード)
 - **projectName** (プロジェクト名)
 - **parentCompanyCd** (会社コード(上位))
 - **parentProjectCd** (プロジェクトコード(上位))
 - parentProjectName (プロジェクト名(上位))
 - localeId(ロケールID)
 - startDate(開始日)
 - endDate(終了日)
 - termCd(期間コード)
 - **deleteFlag** (削除フラグ)
 - extensionA(予備項目A)
 - extensionB(予備項目B)
 - extensionC(予備項目C)
 - extensionD(予備項目D)
 - extensionE(予備項目E)
 - extensionF(予備項目F)
 - extensionG(予備項目G)
 - extensionH(予備項目H)
 - extensionI(予備項目I)
 - extensionJ(予備項目J)
 - entryTs(登録日)
 - entryUserCd(登録者)
 - renewCnt(更新回数)
 - renewTs(更新日)
 - renewUserCd(更新者)
 - **属性値**
「resultValue」や「resultChain」に値を設定することで、検索結果を複数のエレメントに設定することが可能です。
 - resultValue
検索結果から特定のプロジェクトを選択した際に、どのマスタ情報を表示するかを設定します。
コーディング例では、「projectName」(プロジェクト名)を設定しています。
 - resultChain
検索結果から特定のプロジェクトを選択した際に、他のエレメントに設定するマスタ情報を設定します。
コーディング例では、次の様に設定しています。
- テキストボックス (Hidden) (name="targetCompanyCd") = "companyCd" (会社コード)
 テキストボックス (Hidden) (name="targetProjectCd") = "projectCd " (プロジェクトコード)
 テキストボックス (Hidden) (name="targetParentCompanyCd") = "parentCompanyCd " (会社コード(上位))
 テキストボックス (Hidden) (name="targetParentProjectCd") = "parentProjectCd " (プロジェクトコード(上位))
 テキストボックス (Hidden) (name="targetDeleteFlag") = "deleteFlag" (削除フラグ)

ガジェット(経費旅費)

本項では、ガジェット(経費旅費)機能の概要、チュートリアル、クライアントサイド、サーバサイドを解説します。

機能説明

本項では、経費明細ガジェットを参考に、ガジェット（経費旅費）機能の概要や構成を解説します。

- 概要
 - バリエーション違いの互換性
 - 申請書ガジェットインスタンスの採番方法
- 構成
 - 画面構成
- 申請書ガジェットマスタデータについて
 - 申請書ガジェットマスタ
 - 申請書ガジェット概要マスタ
 - 申請書ガジェットタグマスタ
 - 申請書ガジェットサーフェスマスタ
 - 申請書ガジェットブロックマスタ
 - 申請書ガジェットフィールドグループマスタ
 - 申請書ガジェットフィールドマスタ

概要

申請書（IM-Workflow）で利用されることを目的に作られたガジェットを申請書ガジェットと呼んでいます。
各申請書ガジェットデータのデータ登録や更新、画面表示処理を行う機能です。
intra-mart Accel Kaiden!の申請書ガジェット機能をプログラミングする際には、次のモジュールが必要となります。

- intra-mart Accel Kaiden! 基盤モジュール
- intra-mart Accel Kaiden! IM-Workflow連携モジュール

バリエーション違いの互換性

intra-mart Accel Kaiden!にて提供している申請書ガジェットは、
ガジェットクラスが同じであればガジェットバリエーションが異なる場合もデータの互換性を担保しています。
実際にはテーブルは同一のものを利用しており、
バリエーションが異なるものは画面レイアウト（や若干の入力項目）の差異があると位置づけています。

申請書ガジェットインスタンスの採番方法

前述の通り、ガジェットクラス・ガジェットバリエーションはそのガジェットの種類などを表します。
ガジェットインスタンスは、同一ガジェットの同時利用を行えるようにするための連番です。

ただし、申請書機能では、ガジェットインスタンスはガジェットクラスごとの連番で割り振られます。

- 採番ルール

ガジェットクラス	ガジェットバリエーション	ガジェットインスタンス
gadgetA	v01	0
gadgetA	v02	1
gadgetB	v02	0
gadgetB	v01	1
gadgetA	v02	2

構成

intra-mart Accel Kaiden!にて提供している申請書ガジェットは、申請書（IM-Workflow）で利用できます。
利用するガジェットはintra-mart Accel Kaiden!の申請書マスタにて設定します。

i コラム

申請書マスタの操作については、intra-mart Accel Kaiden! 経費旅費 / 管理者操作ガイドを参照してください。

画面構成

経費明細ガジェットでは、次のような画面構成となります。

- 申請画面表示時
経費明細ガジェット画面です。
経費明細ガジェットに値を入力する場合は行を挿入する必要があります。

経費明細

発生日 *

経費内容 *

☐

経費区分 *

勘定科目

税区分

部門

プロジェクト

支払分類 *

支払先

金額 *

1

立替

申請書ガジェットマスタデータについて

申請書ガジェットの情報はデータベース上で保持します。
情報は以下のテーブルで保持し、以下すべてのテーブルデータを総称して「申請書ガジェットマスタデータ」と呼びます。

- 申請書ガジェットマスタ
- 申請書ガジェット概要マスタ
- 申請書ガジェットタグマスタ
- 申請書ガジェットサーフェスマスタ
- 申請書ガジェットブロックマスタ
- 申請書ガジェットフィールドグループマスタ
- 申請書ガジェットフィールドマスタ

i コラム

申請書ガジェット作成時には申請書ガジェットマスタデータを検討の上、各テーブルにデータを投入する必要があります。

申請書ガジェットマスタ

申請書ガジェットの根幹となる情報を保持するマスタです。

ガジェットクラス	ガジェットの種別をあらわします。 intra-mart Accel Kaiden!標準では日当明細ガジェットは「allowanceDetail」、宿泊明細ガジェットは「lodgeDetail」となります。
ガジェットバリエーション	同一ガジェット種別で画面レイアウト違いを識別します。 intra-mart Accel Kaiden!標準では日当明細ガジェット(国内向け)は「v01」、日当明細ガジェット(海外向け)は「v02」となります。
JSPパス	JSPファイルのパスを設定します。
JAVAクラスパス	JAVAクラスパスを設定します。
サーフェスキー	ガジェットのタイトルとなるサーフェスキーを設定します。 後述するガジェットサーフェスマスタのサーフェスキーと紐づけます。
ソートキー	ソートキーを設定します。

申請書ガジェット概要マスタ

申請書ガジェットの概要情報を保持するマスタです。

ガジェット名	ガジェット名を設定します。 使用するロケール分のレコードデータを定義する必要があります。
ガジェット概要	ガジェット概要を設定します。 ガジェットマスタメンテナンス画面のツールチップとなります。 使用するロケール分のレコードデータを定義する必要があります。

申請書ガジェットタグマスタ

ガジェットの属性を保持するマスタです。

タグ	タグを設定します。

i コラム

下記のタグが存在し、設定することでガジェットに属性が付与され、ガジェット属性ごとの振る舞いを定義できます。
intra-mart Accel Kaiden!標準のガジェットには下記以外のタグも設定されていますが、将来拡張用のタグです。

タグ分類	タグ名	説明
制約	constraintDuplicationSelf	申請書内で自身と重複を禁止します。
制約	constraintDuplicationClass	申請書内で同ガジェットクラスと重複を禁止します。
対象申請書タイプ	appAdvance	対象申請書タイプ「事前申請」を設定します。 申請書マスタメンテナンスで申請書タイプ「事前申請書」を選択した場合、選択できるガジェットになります。
対象申請書タイプ	appSettle	対象申請書タイプ「精算申請」を設定します。 申請書マスタメンテナンスで申請書タイプ「精算申請書」を選択した場合、選択できるガジェットになります。
対象申請書タイプ	appNormal	対象申請書タイプ「一般」を設定します。 申請書マスタメンテナンスで申請書タイプ「一般」を選択した場合、選択できるガジェットになります。
対象申請書タイプ	appBill	対象申請書タイプ「請求書払申請」を設定します。 申請書マスタメンテナンスで申請書タイプ「請求書払申請書」を選択した場合、選択できるガジェットになります。

申請書ガジェットサーフェスマスタ

ガジェットのサーフェス情報を保持するマスタです。
サーフェスキーに対して、システムで使用するロケール分のレコードデータを定義する必要があります。

申請書ガジェットブロックマスタ

ガジェットのブロック情報を保持するマスタです。
ブロックIDに対して、ブロックタイプ、ソートキー、サーフェスキーを設定します。

ブロックID	ブロックIDを設定します。
ブロックタイプ	「normal」「detail」のいずれかを設定します。 明細行を保持しない場合は「normal」を設定します。 明細行を保持する場合は「detail」を設定します。
ソートキー	ソートキーを設定します。将来拡張用の項目です。
サーフェスキー	ガジェットのタイトルとなるサーフェスのキーを設定します。

i コラム

ブロックIDは、ガジェット入力データを保持するトランザクションテーブル1つに対応して定義します。
例えば intra-mart Accel Kaiden標準の「交際明細ガジェット」の場合、ブロックIDを3つ定義しています。

ブロックID	ブロックタイプ	対応するトランザクションテーブル
entInfTable	normal	交際明細テーブル
entInfOthDtlTable	detail	交際明細先方明細テーブル
entInfOurDtlTable	detail	交際明細当社明細テーブル

申請書ガジェットフィールドグループマスタ

ガジェットのフィールドグループ情報を保持するマスタです。
役割としては以下になります。

- 「ガジェットフィールド」のグルーピング
- 「ガジェットブロック」との紐づけ
- 必須タイプの設定

ブロックID	ガジェットブロックマスタのブロックIDに紐づけて設定します。
フィールドグループID	フィールドグループIDを設定します。
サーフェスキー	ガジェットサーフェスマスタのサーフェスキーに紐づけて設定します。 ここで紐づけたサーフェスは申請書マスタメンテナンスのガジェット項目の任意/必須切り替え画面のラベルとなります。
必須タイプ	項目の任意/必須を設定します。 設定値は以下の通りとなります。 「0:変更可能(デフォルト任意)」 「1:変更可能(デフォルト必須)」 「a:常時任意」 「b:常時必須」 から選択します。
入力タイプ	項目の任意/必須を設定します。 設定値は以下の通りとなります。 「0:設定変更可」 「a:ラベル(変更不可)」 「b:テキスト(変更不可)」 「c:従来通り」
ソートキー	ソートキーを設定します。将来拡張用の項目です。

i コラム

必須タイプが「変更可能」である場合は申請書マスタメンテナンスで項目の任意/必須を切り替えることができます。

i コラム

入力タイプが「設定変更可」である場合は申請書マスタメンテナンスで項目のテキスト/ラベルを切り替えることができます。
画面種別が「申請」「一時保存」「申請書(起票案件)」の場合は変更できません。

! 注意

項目の任意/必須は申請書マスタメンテナンスで切り替えられるようにするため、
ガジェットの入力データを保持するトランザクションテーブルには基本的にNotNull制約を付与しないでください。

申請書ガジェットフィールドマスタ

ガジェットのフィールド情報を保持するマスタです。

ブロックID	ガジェットブロックマスタのブロックIDに紐づけて設定します。
フィールドキー	フィールドキーを設定します。
フィールドグループID	ガジェットフィールドマスタのフィールドグループIDに紐づけて設定します。
サーフェスキー	ガジェットサーフェスマスタのサーフェスキーに紐づけて設定します。 ここで紐づけたサーフェスは申請書ガジェットのラベルとなります。
必須タイプ	項目の任意/必須を設定します。 設定値は以下の通りとなります。 「0:変更可能(デフォルト任意)」 「1:変更可能(デフォルト必須)」 「a:常時任意」「b:常時必須」 「z:申請書ガジェットフィールドグループマスタの設定を引き継ぐ」
入力タイプ	項目の任意/必須を設定します。 設定値は以下の通りとなります。 「0:設定変更可」将来拡張用の項目です。 「a:ラベル(変更不可)」将来拡張用の項目です。 「b:テキスト(変更不可)」 「c:従来通り」 「z:申請書ガジェットフィールドグループマスタの設定を引き継ぐ」
ソートキー	ソートキーを設定します。将来拡張用の項目です。

チュートリアル

申請書マスタメンテナンス機能にて、チュートリアルガジェットを含んだ申請書（IM-Workflowコンテンツ定義）を作成し、IM-Workflowのルート定義、フロー定義の設定を行います。
ガジェット（経費旅費）機能の解説は「[機能説明](#)」を参照してください。

■ 申請画面

チュートリアル申請書

← 申請 一時保存 申請書コピー

2013/08/01

申請会社	サンプル会社	申請組織	
申請者(代理申請者)	上田辰男		

チュートリアルガジェット

テキスト	
金額*	

申請 一時保存

■ 承認画面

チュートリアル申請書

← 処理 申請履歴

0000000025

チュートリアル申請書 2013/08/01

申請会社	サンプル会社	申請組織	サンプル課12
申請者(代理申請者)	上田辰男		

チュートリアルガジェット

テキスト	チュートリアルガジェットのテキストです。
金額	19,800

処理

本チュートリアルの手順は次の通りです。

チュートリアルガジェットテーブル

項目

- チュートリアルガジェットテーブル定義
 - DDL
- Oracle Database
- PostgreSQL
- Microsoft SQL Server

チュートリアルガジェットテーブル定義

チュートリアルガジェットテーブル「k99t_tutorial」のレイアウトは下記の通りです。

No.	論理名	物理名	型	長さ	必須	PK
1	システム案件ID	system_matter_id	varchar2	20	Y	1
2	ユーザデータID	user_data_id	varchar2	20	Y	2
3	ガジェットクラス	gadget_class	varchar2	100	Y	3
4	ガジェットインスタンス	gadget_instance	number	18,0	Y	4
5	ガジェットID	gadget_id	varchar2	220	Y	
6	ガジェットバリエーション	gadget_variation	varchar2	100	Y	
7	テキスト	tutorial_text	varchar2	1000		
8	通貨コード	tutorial_currency_cd	varchar2	100		
9	金額	tutorial_amount	number	28,10		
10	予備項目A	extension_a	varchar2	1000		
11	予備項目B	extension_b	varchar2	1000		
12	予備項目C	extension_c	varchar2	1000		
13	予備項目D	extension_d	varchar2	1000		
14	予備項目E	extension_e	varchar2	1000		
15	予備項目F	extension_f	varchar2	1000		
16	予備項目G	extension_g	varchar2	1000		
17	予備項目H	extension_h	varchar2	1000		
18	予備項目I	extension_i	varchar2	1000		
19	予備項目J	extension_j	varchar2	1000		
20	登録日時	entry_ts	timestamp		Y	
21	登録者コード	entry_user_cd	varchar2	100	Y	
22	更新カウンタ	renew_cnt	number	18,0	Y	
23	更新日時	renew_ts	timestamp		Y	
24	更新者コード	renew_user_cd	varchar2	100	Y	



コラム

- 予備項目A～J
intra-mart Accel Kaiden!の各テーブルでは、カスタマイズを考慮し予備項目をあらかじめ定義しています。
実際のプログラミング時には予備項目の設定は不要です。



注意

- 登録日時、登録者コード、更新カウンタ、更新日時、更新者コード
intra-mart Accel Kaiden!の各テーブルで共通的に定義している項目です。
本カラムは必ず定義してください。

DDL

利用するデータベースに合わせて、次のいずれかのDDLを実行します。
本チュートリアルでは、Oracle Databaseの使用を推奨します。



コラム

本チュートリアルでは、「テキスト」を入力任意項目、「金額」を入力必須項目としますが、「金額」にNotNull制約は設定しません。
申請書ガジェットテーブルの入力項目については「[機能説明](#)」を参照ください。

```

create table k99t_tutorial
(
  system_matter_id      varchar2(20) not null,
  user_data_id          varchar2(20) not null,
  gadget_class          varchar2(100) not null,
  gadget_instance       numeric(18,0) not null,
  gadget_id             varchar2(220) not null,
  gadget_variation      varchar2(100) not null,
  tutorial_text         varchar2(1000),
  tutorial_currency_cd  varchar2(100),
  tutorial_amount       numeric(28,10),
  extension_a          varchar2(1000),
  extension_b          varchar2(1000),
  extension_c          varchar2(1000),
  extension_d          varchar2(1000),
  extension_e          varchar2(1000),
  extension_f          varchar2(1000),
  extension_g          varchar2(1000),
  extension_h          varchar2(1000),
  extension_i          varchar2(1000),
  extension_j          varchar2(1000),
  entry_ts             timestamp not null,
  entry_user_cd        varchar2(100) not null,
  renew_cnt            numeric(18,0) not null,
  renew_ts             timestamp not null,
  renew_user_cd        varchar2(100) not null
);

alter table k99t_tutorial add primary key (system_matter_id, user_data_id, gadget_class, gadget_instance);

```

PostgreSQL

```

create table k99t_tutorial
(
  system_matter_id      varchar(20) not null,
  user_data_id          varchar(20) not null,
  gadget_class          varchar(100) not null,
  gadget_instance       numeric(18,0) not null,
  gadget_id             varchar(220) not null,
  gadget_variation      varchar(100) not null,
  tutorial_text         varchar(1000),
  tutorial_currency_cd  varchar(100),
  tutorial_amount       numeric(28,10),
  extension_a          varchar(1000),
  extension_b          varchar(1000),
  extension_c          varchar(1000),
  extension_d          varchar(1000),
  extension_e          varchar(1000),
  extension_f          varchar(1000),
  extension_g          varchar(1000),
  extension_h          varchar(1000),
  extension_i          varchar(1000),
  extension_j          varchar(1000),
  entry_ts             timestamp without time zone not null,
  entry_user_cd        varchar(100) not null,
  renew_cnt            numeric(18,0) not null,
  renew_ts             timestamp without time zone not null,
  renew_user_cd        varchar(100) not null
);

alter table k99t_tutorial add primary key (system_matter_id, user_data_id, gadget_class, gadget_instance);

```

Microsoft SQL Server

```

create table k99t_tutorial
(
  system_matter_id      nvarchar(20) not null,
  user_data_id          nvarchar(20) not null,
  gadget_class          nvarchar(100) not null,
  gadget_instance       numeric(18,0) not null,
  gadget_id             nvarchar(220) not null,
  gadget_variation      nvarchar(100) not null,
  tutorial_text         nvarchar(1000),
  tutorial_currency_cd  nvarchar(100),
  tutorial_amount       numeric(28,10),
  extension_a          nvarchar(1000),
  extension_b          nvarchar(1000),
  extension_c          nvarchar(1000),
  extension_d          nvarchar(1000),
  extension_e          nvarchar(1000),
  extension_f          nvarchar(1000),
  extension_g          nvarchar(1000),
  extension_h          nvarchar(1000),
  extension_i          nvarchar(1000),
  extension_j          nvarchar(1000),
  entry_ts             datetime2 not null,
  entry_user_cd        nvarchar(100) not null,
  renew_cnt            numeric(18,0) not null,
  renew_ts             datetime2 not null,
  renew_user_cd        nvarchar(100) not null
);

alter table k99t_tutorial
add constraint pk_k99t_tutorial primary key (system_matter_id, user_data_id, gadget_class, gadget_instance);

```

ガジェットマスタデータ作成

項目
■ 申請書ガジェットマスタデータの作成 ■ Oracle Database/PostgreSQL ■ Microsoft SQL Server

申請書ガジェットマスタデータの作成

申請書ガジェットマスタデータを作成します。
利用するデータベースに合わせて、次のいずれかのInsert文を実行します。
このデータを登録することで、申請書マスタメンテナンス画面でこれから作成する「チュートリアルガジェット」を申請書にセットすることができます

 **コラム**

詳細は「[申請書ガジェットマスタデータについて](#)」を参照してください。

Oracle Database/PostgreSQL

```
-- 申請書ガジェット
insert into k10m_gadget(gadget_class, gadget_variation, jsp_path, java_path, sort_key, surface_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'appAdvance', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 1, 'surface1', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェット概要
insert into k10m_gadget_description(gadget_class, gadget_variation, locale_id, description, gadget_name, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'ja', 'チュートリアルガジェット概要', 'チュートリアルガジェット概要', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_description(gadget_class, gadget_variation, locale_id, description, gadget_name, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'en', 'Tutorial Gadget Overview', 'Tutorial Gadget Overview', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェットタグ
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'appAdvance', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'appNormal', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'appSettle', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェットサーフェス
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface1', 'ja', 'チュートリアルガジェットサーフェス1', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface1', 'en', 'Tutorial Gadget Surface 1', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface2', 'ja', 'チュートリアルガジェットサーフェス2', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface2', 'en', 'Tutorial Gadget Surface 2', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface3', 'ja', 'チュートリアルガジェットサーフェス3', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface3', 'en', 'Tutorial Gadget Surface 3', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface4', 'ja', 'チュートリアルガジェットサーフェス4', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'surface4', 'en', 'Tutorial Gadget Surface 4', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェットブロック
insert into k10m_gadget_block(gadget_class, gadget_variation, gadget_block, gadget_block_type, sort_key, surface_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'block1', 1, 'surface1', to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェットフィールドグループ
insert into k10m_gadget_fld_grp(gadget_class, gadget_variation, gadget_block, field_group_id, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp1', 'surface1', 'required', 'text', 1, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_fld_grp(gadget_class, gadget_variation, gadget_block, field_group_id, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp2', 'surface1', 'required', 'text', 2, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
-- 申請書ガジェットフィールド
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp1', 'field1', 'surface1', 'required', 'text', 1, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp1', 'field2', 'surface1', 'required', 'text', 2, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp2', 'field1', 'surface1', 'required', 'text', 1, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', 'v01', 'block1', 'fld_grp2', 'field2', 'surface1', 'required', 'text', 2, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1', 0, to_timestamp('2011-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'user1')
```

Microsoft SQL Server

```
-- 申請書ガジェット挿入
insert into k10m_gadget(gadget_class, gadget_variation, jsp_path, java_path, sort_key, surface_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'k10m', 'application/javascript', 'application/java', 1, 'surface', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェット概要
insert into k10m_gadget_description(gadget_class, gadget_variation, locale_id, description, gadget_name, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'k10m', 'ja', '申請書ガジェットの概要', '申請書ガジェットの概要', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_description(gadget_class, gadget_variation, locale_id, description, gadget_name, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'k10m', 'en', 'Application Gadget Overview', 'Application Gadget Overview', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェットタグ
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'appAdvance', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'appNormal', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_tags(gadget_class, gadget_variation, gadget_tag, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'appSettle', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェットサーフェス
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface1', 'ja', '申請書ガジェットのサーフェス1', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface1', 'en', 'Application Gadget Surface 1', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface2', 'ja', '申請書ガジェットのサーフェス2', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface2', 'en', 'Application Gadget Surface 2', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface3', 'ja', '申請書ガジェットのサーフェス3', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface3', 'en', 'Application Gadget Surface 3', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface4', 'ja', '申請書ガジェットのサーフェス4', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_surface(gadget_class, gadget_variation, surface_key, locale_id, surface, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'surface4', 'en', 'Application Gadget Surface 4', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェットブロック
insert into k10m_gadget_block(gadget_class, gadget_variation, gadget_block, gadget_block_type, sort_key, surface_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'blockType1', 1, 'surface1', '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェットフィールドグループ
insert into k10m_gadget fld_grp(gadget_class, gadget_variation, gadget_block, field_group_id, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'fieldGroup1', 'surface1', N'text', N'text', 1, '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget fld_grp(gadget_class, gadget_variation, gadget_block, field_group_id, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'fieldGroup2', 'surface1', N'text', N'text', 2, '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
-- 申請書ガジェットフィールド
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'fieldGroup1', N'fieldKey1', 'surface1', N'text', N'text', 1, '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'fieldGroup1', N'fieldKey2', 'surface1', N'text', N'text', 2, '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
insert into k10m_gadget_field(gadget_class, gadget_variation, gadget_block, field_group_id, field_key, surface_key, required_type, input_type, sort_key, entry_ts, entry_user_cd, renew_cnt, renew_ts, renew_user_cd) values ('tutorial', N'v01', N'block1', N'fieldGroup2', N'fieldKey1', 'surface1', N'text', N'text', 1, '2015-01-01 00:00:00', '1000', 0, '2015-01-01 00:00:00', '1000')
```

S2JDBC-Gen

- S2JDBC-Genの実行
- 自動生成ソース
 - Entity
 - Names
 - Service
 - Condition

S2JDBC-Genの実行

チュートリアルガジェットテーブルを作成後、S2JDBC-Gen にて以下のソースファイルを自動生成します。

- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.K99tTutorial
- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.K99tTutorialNames
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99tTutorialService
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99tTutorialCondition



コラム

S2JDBC-Genにてソースを自動生成した際に、次の個所を確認してください。

- “K99tTutorial”のスーパークラスが「jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity」に設定されていること。
- “K99tTutorial”に登録日時、登録者コード、更新カウンタ、更新日時、更新者コードが定義されていないこと。
- “K99tTutorialService”のスーパークラスが「jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService」に設定されていること。

自動生成ソース

```
src
├── main
│   └── generated
│       ├── jp
│       │   ├── co
│       │   │   ├── slcs
│       │   │   │   ├── kaiden2
│       │   │   │   │   ├── tutorial
│       │   │   │   │   │   ├── foundation
│       │   │   │   │   │   │   ├── model
│       │   │   │   │   │   │   │   ├── entity
│       │   │   │   │   │   │   │   │   ├── K99tTutorial.java
│       │   │   │   │   │   │   │   │   ├── K99tTutorialNames.java
│       │   │   │   │   │   │   │   └── service
│       │   │   │   │   │   │   │       ├── K99tTutorialCondition.java
│       │   │   │   │   │   │   │       └── K99tTutorialService.java
```

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/entity/K99tTutorial.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * チュートリアルガジェット.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k99t_tutorial")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K99tTutorial extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件ID. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザデータID. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;
```

```

/** ガジェットID. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** テキスト. */
@Column(name = "tutorial_text", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String tutorialText;

/** 通貨コード. */
@Column(name = "tutorial_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String tutorialCurrencyCd;

/** 金額. */
@Column(name = "tutorial_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal tutorialAmount;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

- **Path**
src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/entity/K99tTutorialNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K99tTutorial} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K99tTutorialNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K99tTutorialNames() {

```

```

    super();
}

/**
 * systemMatterIdのプロパティ名を返却.
 *
 * @return systemMatterIdのプロパティ名
 */
public static PropertyName<String> systemMatterId() {
    return new PropertyName<String>("systemMatterId");
}

/**
 * userDataIdのプロパティ名を返却.
 *
 * @return userDataIdのプロパティ名
 */
public static PropertyName<String> userDataId() {
    return new PropertyName<String>("userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public static PropertyName<String> gadgetClass() {
    return new PropertyName<String>("gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * tutorialTextのプロパティ名を返却.
 *
 * @return tutorialTextのプロパティ名
 */
public static PropertyName<String> tutorialText() {
    return new PropertyName<String>("tutorialText");
}

/**
 * tutorialCurrencyCdのプロパティ名を返却.
 *
 * @return tutorialCurrencyCdのプロパティ名
 */
public static PropertyName<String> tutorialCurrencyCd() {
    return new PropertyName<String>("tutorialCurrencyCd");
}

/**
 * tutorialAmountのプロパティ名を返却.
 *
 * @return tutorialAmountのプロパティ名
 */
public static PropertyName<BigDecimal> tutorialAmount() {
    return new PropertyName<BigDecimal>("tutorialAmount");
}

/**
 * extensionAのプロパティ名を返却.
 *

```

```

    * @return extensionAのプロパティ名
    */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }
}

```

```

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K99tTutorialNames extends PropertyName<K99tTutorial> {

    /**
     * インスタンスを構築.
     */
    public _K99tTutorialNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K99tTutorialNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K99tTutorialNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }
}

```

```

* userDataIdのプロパティ名を返却.
*
* @return userDataIdのプロパティ名
*/
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
* gadgetClassのプロパティ名を返却.
*
* @return gadgetClassのプロパティ名
*/
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
* gadgetInstanceのプロパティ名を返却.
*
* @return gadgetInstanceのプロパティ名
*/
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
* gadgetIdのプロパティ名を返却.
*
* @return gadgetIdのプロパティ名
*/
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
* gadgetVariationのプロパティ名を返却.
*
* @return gadgetVariationのプロパティ名
*/
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
* tutorialTextのプロパティ名を返却.
*
* @return tutorialTextのプロパティ名
*/
public PropertyName<String> tutorialText() {
    return new PropertyName<String>(this, "tutorialText");
}

/**
* tutorialCurrencyCdのプロパティ名を返却.
*
* @return tutorialCurrencyCdのプロパティ名
*/
public PropertyName<String> tutorialCurrencyCd() {
    return new PropertyName<String>(this, "tutorialCurrencyCd");
}

/**
* tutorialAmountのプロパティ名を返却.
*
* @return tutorialAmountのプロパティ名
*/
public PropertyName<BigDecimal> tutorialAmount() {
    return new PropertyName<BigDecimal>(this, "tutorialAmount");
}

/**
* extensionAのプロパティ名を返却.
*
* @return extensionAのプロパティ名
*/
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
* extensionBのプロパティ名を返却.
*
* @return extensionBのプロパティ名
*/
public PropertyName<String> extensionB() {

```

```

    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。

```

```

*
* @return entryUserCdのプロパティ名
*/
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
* renewCntのプロパティ名を返却.
*
* @return renewCntのプロパティ名
*/
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
* renewTsのプロパティ名を返却.
*
* @return renewTsのプロパティ名
*/
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
* renewUserCdのプロパティ名を返却.
*
* @return renewUserCdのプロパティ名
*/
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99tTutorialService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K99tTutorial;

/**
* {@link K99tTutorial}のサービスクラス.
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K99tTutorialService extends GenerateService<K99tTutorial> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99tTutorialCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
* {@link K99tTutorial}の条件クラス.
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K99tTutorialCondition extends
    AbstractEntityCondition<K99tTutorialCondition> {

    /**
    * インスタンスを構築.
    */
}

```



```

public K99tTutorialCondition() {
}

/**
 * インスタンスを構築.
 *
 * @param prefix プレフィックス
 * @param where 検索条件
 */
public K99tTutorialCondition(String prefix, ComplexWhere where) {
    super(prefix, where);
}

/** systemMatterIdの条件. */
public NullableStringCondition<K99tTutorialCondition> systemMatterId =
    new NullableStringCondition<K99tTutorialCondition>("systemMatterId", this);

/** userDataIdの条件. */
public NullableStringCondition<K99tTutorialCondition> userDataId =
    new NullableStringCondition<K99tTutorialCondition>("userDataId", this);

/** gadgetClassの条件. */
public NullableStringCondition<K99tTutorialCondition> gadgetClass =
    new NullableStringCondition<K99tTutorialCondition>("gadgetClass", this);

/** gadgetInstanceの条件. */
public NullableCondition<K99tTutorialCondition, Long> gadgetInstance =
    new NullableCondition<K99tTutorialCondition, Long>("gadgetInstance", this);

/** gadgetIdの条件. */
public NullableStringCondition<K99tTutorialCondition> gadgetId =
    new NullableStringCondition<K99tTutorialCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NullableStringCondition<K99tTutorialCondition> gadgetVariation =
    new NullableStringCondition<K99tTutorialCondition>("gadgetVariation", this);

/** tutorialTextの条件. */
public NullableStringCondition<K99tTutorialCondition> tutorialText =
    new NullableStringCondition<K99tTutorialCondition>("tutorialText", this);

/** tutorialCurrencyCdの条件. */
public NullableStringCondition<K99tTutorialCondition> tutorialCurrencyCd =
    new NullableStringCondition<K99tTutorialCondition>("tutorialCurrencyCd", this);

/** tutorialAmountの条件. */
public NullableCondition<K99tTutorialCondition, BigDecimal> tutorialAmount =
    new NullableCondition<K99tTutorialCondition, BigDecimal>("tutorialAmount", this);

/** extensionAの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionA =
    new NullableStringCondition<K99tTutorialCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionB =
    new NullableStringCondition<K99tTutorialCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionC =
    new NullableStringCondition<K99tTutorialCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionD =
    new NullableStringCondition<K99tTutorialCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionE =
    new NullableStringCondition<K99tTutorialCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionF =
    new NullableStringCondition<K99tTutorialCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionG =
    new NullableStringCondition<K99tTutorialCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionH =
    new NullableStringCondition<K99tTutorialCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K99tTutorialCondition> extensionI =
    new NullableStringCondition<K99tTutorialCondition>("extensionI", this);

/** extensionJの条件. */

```

```

public NullableStringCondition<K99tTutorialCondition> extensionJ =
    new NullableStringCondition<K99tTutorialCondition>("extensionJ", this);

/** entryTsの条件. */
public NullableCondition<K99tTutorialCondition, Timestamp> entryTs =
    new NullableCondition<K99tTutorialCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NullableStringCondition<K99tTutorialCondition> entryUserCd =
    new NullableStringCondition<K99tTutorialCondition>("entryUserCd", this);

/** renewCntの条件. */
public NullableCondition<K99tTutorialCondition, Long> renewCnt =
    new NullableCondition<K99tTutorialCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NullableCondition<K99tTutorialCondition, Timestamp> renewTs =
    new NullableCondition<K99tTutorialCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NullableStringCondition<K99tTutorialCondition> renewUserCd =
    new NullableStringCondition<K99tTutorialCondition>("renewUserCd", this);
}

```

Managerクラスの作成

- Managerの作成
- Manager実装クラスの作成
- ManagerDtoの作成
- SQLファイルの作成



コラム

申請書ガジェットのManagerクラスの詳細は「[申請書ガジェットManagerクラスプログラミング](#)」を参照ください。

Managerの作成

jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManagerを継承してGadgetTutorialManagerインターフェースを作成します。
 申請書ガジェットに必要な最低限のDB操作メソッドはPluginGadgetManagerインターフェースで提供しています。
 マネージャ固有のメソッドを実装する場合はGadgetTutorialManagerインターフェースに追加します。

```
package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial;

import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;

/**
 * チュートリアルガジェットテーブルマネージャクラスのinterface.
 */
public interface GadgetTutorialManager extends PluginGadgetManager<GadgetTutorialManagerDto> {

    /** ガジェット表示用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** 申請書コピー用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectView.sql";
}
```

Manager実装クラスの作成

GadgetTutorialManagerImplクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl;

import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManager;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManagerDto;
import jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99tTutorialService;

/**
 * チュートリアルガジェットテーブルへの各種操作を提供するAPI.
 */
public class GadgetTutorialManagerImpl extends AbstractWfManager<GadgetTutorialManagerDto> implements
    GadgetTutorialManager {

    /** サービスクラス. */
    @Resource
    protected K99tTutorialService k99tTutorialService;

    /** 主キー検索. */
    @Override
    public GadgetTutorialManagerDto selectByPk(GadgetTutorialManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k99tTutorialService, criteria);
    }

    /** リスト検索. */
    @Override
    public List<GadgetTutorialManagerDto> selectList(GadgetTutorialManagerDto criteria) {
        return super.selectPkUsedUserDataId(k99tTutorialService, criteria);
    }

    /** 表示条件検索. */
    @Override
    public List<GadgetTutorialManagerDto> selectView(CriteriaForSqlDto criteria) {
        return super.selectBySqlForView(k99tTutorialService, SELECT_SQLFILE_NM, criteria);
    }

    /** 申請書コピー検索. */
    @Override
    public List<GadgetTutorialManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {
        return super.selectBySqlForViewForCopy(k99tTutorialService, SELECT_SQLFILE_NM_FOR_COPY, criteria);
    }

    /** 過去案件検索. */
    @Override
    public List<GadgetTutorialManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        return null;
    }

    /** 一括挿入. */
    @Override
    public int entry(List<GadgetTutorialManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.insertBatch(k99tTutorialService, dataList));
    }
}
```

```

/** 一括更新. */
@Override
public int renew(List<GadgetTutorialManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.renewBatch(k99tTutorialService, dataList));
}

/** 削除. */
@Override
public int remove(GadgetTutorialManagerDto data) {
    return super.deleteUsedUserDataId(k99tTutorialService, data);
}

/** 一括削除. */
@Override
public int removeAll(GadgetTutorialManagerDto data) {
    return super.deleteNoUsedUserDataId(k99tTutorialService, data);
}

/** 過去案件登録. */
@Override
public int matterArchiveRegister(CriteriaForSqlDto criteria, List<GadgetTutorialManagerDto> dataList) {
    return 0;
}

/** 過去案件処理時削除. */
@Override
public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
    return super.deleteNoUsedUserDataId(k99tTutorialService, criteria);
}

/** 過去案件一括削除. */
@Override
public int removeAllArchive(CriteriaForSqlDto criteria) {
    return 0;
}
}

```

ManagerDtoの作成

GadgetTutorialManagerDtoクラスを作成します。

対象テーブルのカラムと、同時に取得するカラムがある場合は定義します。

```

package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * チュートリアルガジェットテーブルDTO.
 */
public class GadgetTutorialManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstance;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバージョン. */
    public String gadgetVariation;
    /** テキスト. */
    public String tutorialText;
    /** 通貨コード. */
    public String tutorialCurrencyCd;
    /** 金額. */
    public BigDecimal tutorialAmount;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 通貨名称. */
    public String tutorialCurrencyName;
}

```



コラム

ManagerDtoクラスは検索条件の設定、取得結果の保持で使用します。

SQLファイルの作成

selectView.sqlの名前でSQLファイルを作成します。ファイル名はManagerインターフェースで宣言したものです。

配置場所は「WEB-INF/classes/META-INF/sql/jp/co/slcs/kaiden2/tutorial/foundation/model/service/K99tTutorialService」です。

```

select
  tu.system_matter_id,
  tu.user_data_id,
  tu.gadget_class,
  tu.gadget_instance,
  tu.gadget_id,
  tu.gadget_variation,
  tu.tutorial_text,
  tu.tutorial_currency_cd,
  tu.tutorial_amount,
  tu.extension_a,
  tu.extension_b,
  tu.extension_c,
  tu.extension_d,
  tu.extension_e,
  tu.extension_f,
  tu.extension_g,
  tu.extension_h,
  tu.extension_i,
  tu.extension_j,
  tu.entry_ts,
  tu.entry_user_cd,
  tu.renew_cnt,
  tu.renew_ts,
  tu.renew_user_cd,
  cu.currency_name as tutorial_currency_name
from
  k99t_tutorial tu
  inner join
    k20c_matter_control c1
  on
    tu.system_matter_id = c1.system_matter_id
  and
    tu.user_data_id = c1.user_data_id
  left outer join imm_currency cu
  on
    c1.search_criteria_company = cu.company_cd
  and
    tu.tutorial_currency_cd = cu.currency_cd
  and
    /*localeId*/ja' = cu.locale_id
  and
    cu.delete_flag = '0'
where
  tu.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  tu.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
  tu.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  tu.gadget_instance = /*gadgetInstance*/0
/*END*/
/*IF gadgetId != null*/
and
  tu.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

diconファイルの作成

項目

- kaiden_tutorial.diconの作成

kaiden_tutorial.diconの作成

diconファイルを作成します。配置場所は「resources/」です。

プロジェクト名を「kaiden_tutorial」としましたので、ファイル名を「kaiden_tutorial.dicon」とします。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">

<components>

  <!-- manager -->
  <component name="masterTutorialManager" class="jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl.MasterTutorialManagerImpl" instance="prototype"></component>
  <!-- ↓ ↓ 追加します -->
  <component name="gadgetTutorialManager" class="jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl.GadgetTutorialManagerImpl" instance="prototype"></component>
  <!-- ↑ ↑ 追加します -->

  <!-- helper -->

  <!-- validator -->

  <!-- service -->
  <component name="k99mTutorialService" class="jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99mTutorialService" instance="prototype"></component>
  <!-- ↓ ↓ 追加します -->
  <component name="k99tTutorialService" class="jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99tTutorialService" instance="prototype"></component>
  <!-- ↑ ↑ 追加します -->
</components>
```



コラム

チュートリアルマスタメンテナンス機能では、DiconファイルにはManagerクラスとServiceクラスを定義します。



コラム

上記ファイルは「チュートリアル」にて作成したdiconファイルを修正する記事となっています。

Formクラスの作成

jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageFormを継承してFormクラスを作成します。

クラス名は「業務名」+「ガジェットバリエーション」+「Form」としてください。

本チュートリアルでは、GadgetTutorialV01Formクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * チュートリアルガジェット (Form).
 */
public class GadgetTutorialV01Form extends AbstractImwPageForm {

}
```

Actionクラスの作成

項目

- 申請書ガジェットのActionクラスについて
 - 業務基底アクションクラス
 - 業務アクションクラス



コラム

申請書ガジェットのActionクラスの詳細は「[申請書ガジェットActionクラスプログラミング](#)」を参照ください。

申請書ガジェットのActionクラスについて

申請書ガジェットのActionクラスはガジェットバリエーションを問わない共通した処理を実装する「業務基底アクションクラス」と、ガジェットバリエーションごとに異なる処理を実装する「業務アクションクラス」を作成することを推奨します。

業務基底アクションクラス

業務基底アクションクラスには、ガジェットバリエーションを問わないガジェットクラス共通の処理を実装します。

クラス名は「Abstract」 + 「業務名」 + 「ActionBase」としてください。

本チュートリアルでは、AbstractGadgetTutorialActionBaseクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial.feature.action.kaiden.gadget;

import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arxn.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * チュートリアルガジェット基底クラス (Action).
 */
public abstract class AbstractGadgetTutorialActionBase implements GadgetActionIF {

    /** 申請画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /** 再申請画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /** 一時保存画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /** 処理画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /** 確認画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadConfirm() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理詳細画面のデータロード.
     */
}
```



```

*
* @return Ajaxアクセスのため、常にnullを返却
*/
@Override
@Execute validator = false
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/** 画面表示用データを返却. */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), new ConvertForJsonSource());
}

/** 参照詳細画面のデータロード. */
@Override
@Execute validator = false
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/** 確認詳細画面のデータロード. */
@Override
@Execute validator = false
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/** 申請書コピーのデータロード. */
@Override
@Execute validator = false
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    return null;
}

/** 過去案件詳細画面のデータロード. */
@Override
@Execute validator = false
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConvertor())));
    return null;
}

/** 書式変換設定を返却. */
protected TutorialConvertor getConvertor() {
    return new TutorialConvertor(getForm());
}

/** チュートリアルガジェットの書式変換. */
private class TutorialConvertor extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private final AbstractImwPageForm frm;

    /** NumberHelper. */
    private final NumberHelper numberHelper;

    /** コンストラクタ. */
    public TutorialConvertor(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /** 入力項目用書式変換. */
    @Override
    protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

        Map<String, Object> result = new HashMap<String, Object>(map);

        // 金額の書式変換
        if (result.containsKey("tutorialAmount")) {
            BigDecimal amount = (BigDecimal) result.get("tutorialAmount");

            Object currencyCd = result.get("tutorialCurrencyCd");
            result.put("tutorialAmount",
                numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
        }

        return result;
    }
}

```

```

/** 表示項目用書式変換. */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 金額の書式変換
    if (map.containsKey("tutorialAmount")) {
        BigDecimal amount = (BigDecimal) map.get("tutorialAmount");
        Object currencyCd = map.get("tutorialCurrencyCd");
        result.put("tutorialAmount",
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    return result;
}
}
}

```

業務アクションクラス

ガジェットActionクラスには、ガジェットバリエーションごとに異なる処理を実装します。

クラス名は「業務名」+「ガジェットバリエーション」+「Action」としてください。

本チュートリアルでは、GadgetTutorialV01Actionクラスを作成します。

```

package jp.co.slcs.kaiden2.tutorial.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.tutorial.feature.form.kaiden.gadget.GadgetTutorialV01Form;
import jp.co.slcs.kaiden2.tutorial.feature.logic.kaiden.gadget.tutorial.GadgetTutorialV01Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * チュートリアルv01ガジェット (Action).
 */
public class GadgetTutorialV01Action extends AbstractGadgetTutorialActionBase {

    /** Logic. */
    @Resource
    public GadgetTutorialV01Logic gadgetTutorialV01Logic;

    /** Form. */
    @ActionForm
    @Resource
    public GadgetTutorialV01Form gadgetTutorialV01Form;

    /** Logicを取得. */
    @Override
    public PluginLogicIF getLogic() {
        return gadgetTutorialV01Logic;
    }

    /** Formを取得. */
    @Override
    public AbstractImwPageForm getForm() {
        return gadgetTutorialV01Form;
    }
}

```

Logicクラスの作成

項目

- 申請書ガジェットのLogicクラスについて
 - 業務基底ロジッククラス
 - 業務ロジッククラス



コラム

申請書ガジェットのLogicクラスの詳細は「[申請書ガジェットLogicクラスプログラミング](#)」を参照ください。

申請書ガジェットのLogicクラスについて

「業務基底ロジッククラス」と「業務ロジッククラス」を作成することを推奨します。

業務基底ロジッククラス

jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogicを継承して業務基底ロジッククラスを作成します。

業務基底ロジッククラスには、ガジェットバリエーションを問わないガジェットクラス共通の処理を実装します。

クラス名は「"Abstract" + "業務名" + "LogicBase"」としてください。

本チュートリアルでは、AbstractGadgetTutorialLogicBaseクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial.feature.logic.kaiden.gadget.tutorial;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManager;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManagerDto;
import org.seasar.framework.beans.util.Beans;

/**
 * チュートリアルガジェット基底クラス (Logic).
 */
public abstract class AbstractGadgetTutorialLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: チュートリアルガジェット. */
    protected static final String BLOCK_KEY_TUTORIAL_TABLE = "tutorialTable";

    /** GadgetTutorialManager. */
    @Resource
    protected GadgetTutorialManager gadgetTutorialManager;

    /** CurrencyHelper. */
    @Resource
    protected CurrencyHelper currencyHelper;

    /** コンストラクタ. */
    public AbstractGadgetTutorialLogicBase() {
        super();
    }

    /** 申請処理時に実行されるメソッド. */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /** 一時保存案件の申請処理時に実行されるメソッド. */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
```

```

        deleteTable(true);
        return insertTable();
    }

    /** 未申請案件の申請処理時に実行されるメソッド: */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /** 承認処理時に実行されるメソッド: */
    @Override
    public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /** 承認終了処理時に実行されるメソッド: */
    @Override
    public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /** 否認処理時に実行されるメソッド: */
    @Override
    public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 取止め処理時に実行されるメソッド: */
    @Override
    public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 案件操作処理時に実行されるメソッド: */
    @Override
    public int matterHandle(KaidenDto dto) {
        return 0;
    }

    /** 引戻し処理時に実行されるメソッド: */
    @Override
    public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 再申請処理時に実行されるメソッド: */
    @Override
    public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /** 保留処理時に実行されるメソッド: */
    @Override
    public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 保留解除処理時に実行されるメソッド: */
    @Override
    public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 差戻し処理時に実行されるメソッド: */
    @Override
    public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 差戻し後の引戻し処理時に実行されるメソッド: */
    @Override
    public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /** 一時保存(新規登録)処理時に実行されるメソッド: */

```

```

@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/** 一時保存(削除)処理時に実行されるメソッド. */
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/** 一時保存(更新)処理時に実行されるメソッド. */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/** 案件開始処理時に実行されるメソッド. */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 案件終了処理時に実行されるメソッド. */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 到達処理時に実行されるメソッド. */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 未完了案件削除処理時に実行されるメソッド. */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/** 過去案件削除処理時に実行されるメソッド. */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    // アーカイブは考慮しない
    return 0;
}

/** 完了案件削除処理時に実行されるメソッド. */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/** データ取得メソッド. */
@Override
protected LinkedHashMap<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    GadgetTutorialManagerDto dto = Beans.createAndCopy(GadgetTutorialManagerDto.class, conditionDto).execute();
    List<GadgetTutorialManagerDto> listSelect = gadgetTutorialManager.selectList(dto);
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
    Map<String, Object> dtoMap = new HashMap<String, Object>();
    if (!listSelect.isEmpty()) {
        dtoMap = BeanUtil.toMap(listSelect.get(0), true);
    }
    rv.put(BLOCK_KEY_TUTORIAL_TABLE, dtoMap);
    return rv;
}

/** データ取得メソッド. */
@Override
protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
    throws KaidenSystemException {
    return null;
}

/** データ登録処理. */
protected int insertTable() throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = JsonUtil.getDto(GadgetTutorialManagerDto.class,
        super.getJsonDataNormalBlock(BLOCK_KEY_TUTORIAL_TABLE));
    return gadgetTutorialManager.entry(makeSingleInsertList(mngDto));
}

/** データ削除処理. */
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = new GadgetTutorialManagerDto();

```

```

mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
mngDto.userDataId = super.getGadgetDto().userDataId;
mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
if (tempFlg) {
    mngDto.systemMatterId = "temporary";
}
return gadgetTutorialManager.remove(mngDto);
}

/** データ削除処理(履歴を含む). */
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = new GadgetTutorialManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    return gadgetTutorialManager.removeAll(mngDto);
}

/** 申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 一時保存案件の申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 未申請案件の申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 承認アクション時バリデート定義処理. */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 承認終了アクション時バリデート定義処理. */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 再申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 一時保存(新規登録)アクション時バリデート定義処理. */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/** 一時保存(更新)アクション時バリデート定義処理. */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/** フィールドバリデート定義. */
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {
    // 本機能ではサブクラスで実装します
}

/** ガジェット内個別バリデート処理. */
@Override
protected List<KaidenValidateResult> postSingleValidate(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {

```

```

    return new ArrayList<KaidenValidateResult>();
}

/** 大域バリデート処理. */
@Override
protected List<KaidenValidateResult> globalValidateMain(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {
    return new ArrayList<KaidenValidateResult>();
}

/** 初期表示用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        // 新規申請の場合
        result.put(BLOCK_KEY_TUTORIAL_TABLE, new ArrayList<GadgetTutorialManagerDto>());
    } else {
        // 新規申請でない場合
        result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectView(criteria));
    }
    return result;
}

/** 申請書コピー用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectViewForCopy(criteria));
    return result;
}

/** 過去案件用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    // アーカイブは考慮しない
    return new HashMap<String, Object>();
}

/** 検索結果のマスタ整合性チェック. */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_TUTORIAL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<GadgetTutorialManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), GadgetTutorialManagerDto.class);
            for (GadgetTutorialManagerDto sqlResult : sqlResultList) {
                if (isEmpty(sqlResult.tutorialCurrencyCd) && isEmpty(sqlResult.tutorialCurrencyName)) {
                    String[] fieldId = getArgs("tutorialCurrencyCd", "tutorialCurrencyName");
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(sqlResult.tutorialCurrencyCd, sqlResult.tutorialCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_TUTORIAL_TABLE, fieldId, null, masterName, values,
                        message));
                }
            }
        }
    }
    return result;
}

/** データ補完処理. */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    // サブクラスで実装するため基底では何も行わない
    return;
}

/** 案件退避処理時の登録処理. */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    // アーカイブは考慮しない
    return 0;
}

/** 退避済完了案件削除対象マネージャリストを返却. */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    // アーカイブは考慮しない
    return new ArrayList();
}

```

```

/** JSON出力用の形式に変換し同時に書式も設定. */
@Override
protected Map<String, Object> mapConvert(Map<String, Object> dataMap, ConvertForJsonSource converter,
    ProcessType processType) {

    @SuppressWarnings("unchecked")
    List<GadgetTutorialManagerDto> listDto = (List<GadgetTutorialManagerDto>) dataMap.get(BLOCK_KEY_TUTORIAL_TABLE);
    if (CollectionUtil.isEmpty(listDto)) {
        dataMap.put(BLOCK_KEY_TUTORIAL_TABLE, BeanUtil.toMap(listDto.get(0), true));
    } else {
        dataMap.put(BLOCK_KEY_TUTORIAL_TABLE, new HashMap<String, Object>());
    }

    return converter.mapConvert(dataMap);
}
}

```

業務ロジッククラス

業務基底ロジッククラスを継承して業務ロジッククラスを作成します。

業務ロジッククラスには、ガジェットバリエーションごとに異なる処理を実装します。

クラス名は「業務名」 + 「ガジェットバリエーション」 + 「Logic」としてください。

本チュートリアルでは、GadgetTutorialV01Logicクラスを作成します。

```

package jp.co.slcs.kaiden2.tutorial.feature.logic.kaiden.gadget.tutorial;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;

/**
 * チュートリアルv01ガジェット (Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class GadgetTutorialV01Logic extends AbstractGadgetTutorialLogicBase {

    /** チュートリアルガジェット.金額の最小値. */
    public static final int TUTORIAL_AMOUNT_NUMERIC_MIN = 0;

    /** チュートリアルガジェット.テキストの最大長. */
    public static final int TUTORIAL_TEXT_LEN = 250;

    /** コンストラクタ. */
    public GadgetTutorialV01Logic() {
        super();
    }

    /** フィールドバリデイト定義. */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        if (fieldDefsMap.containsKey(BLOCK_KEY_TUTORIAL_TABLE)) {
            addValidateDefsRequired(BLOCK_KEY_TUTORIAL_TABLE, tmpFlg);
            addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialText", "maxLength",
                StringUtil.toString(TUTORIAL_TEXT_LEN));
            addValidateDefsMap(BLOCK_KEY_TUTORIAL_TABLE, "tutorialCurrencyCd", "immCurrencyExists",
                "tutorialCurrencyCd", "searchCriteriaCompany");
            addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialAmount", "currencyAmnt",
                "tutorialCurrencyCd", "searchCriteriaCompany");
            addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialAmount", "numericMin",
                StringUtil.toString(TUTORIAL_AMOUNT_NUMERIC_MIN));
        }
    }

    /** データ補完処理. */
    @Override
    protected void complementJsonData(GadgetMediatorParams params) {
        Map<String, Object> blockDat = getJsonDataNormalBlock(BLOCK_KEY_TUTORIAL_TABLE);
        String currencyCd1 = StringUtil.toString(blockDat.get("tutorialCurrencyCd"));

        if (isEmpty(currencyCd1)) {
            blockDat.put("tutorialCurrencyCd",
                currencyHelper.getCompanyCurrencyCd((String) blockDat.get("searchCriteriaCompany")));
        }
    }
}

```


/WEB-INF/view/kaiden2/tutorial/gadget/gadgetTutorial/gadgetTutorialV01.jsp作成します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>

<!-- チュートリアルガジェット -->
<c:set var="gadgetClass">tutorial</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/gadgetTutorialV01</c:set>
<c:set var="gadgetBlockId">tutorialTable</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="nomal" value="${gadget.blocks[gadgetBlockId]}" />

<%-- サーフエス --%>
<%-- タイトル --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<%-- テキスト --%>
<c:set var="surfaceTutorialText">${gadget.surfaces["tutorialText"]}</c:set>
<%-- 金額 --%>
<c:set var="surfaceTutorialAmount">${gadget.surfaces["tutorialAmount"]}</c:set>

<%-- 入力区分 / 必須区分 --%>
<%-- テキスト --%>
<c:set var="modeTutorialText">${nomal.fields["tutorialText"].inputType}</c:set>
<c:set var="reqTutorialText" >${nomal.fields["tutorialText"].requiredFlag}</c:set>
<%-- 金額 --%>
<c:set var="modeTutorialAmount">${nomal.fields["tutorialAmount"].inputType}</c:set>
<c:set var="reqTutorialAmount" >${nomal.fields["tutorialAmount"].requiredFlag}</c:set>

<%-- バリデーション --%>
<%-- テキスト --%>
<c:set var="validTutorialText_def">[k_maxLength:[250]]</c:set>
<c:set var="validTutorialText_apply">${"true" == reqTutorialText ? "k_required:[''] : ''"}</c:set>
<c:set var="validTutorialText_tempsave"></c:set>
<c:set var="validTutorialText_reapply">${"true" == reqTutorialText ? "k_required:[''] : ''"}</c:set>
<c:set var="validTutorialText_process">${"true" == reqTutorialText ? "k_required:[''] : ''"}</c:set>
<%-- 金額 --%>
<c:set var="validTutorialAmount_def">[k_currencyAmnt:[',', 'searchCriteriaCompany'], k_numericMin:[0]]</c:set>
<c:set var="validTutorialAmount_apply">${"true" == reqTutorialAmount ? "k_required:[''] : ''"}</c:set>
<c:set var="validTutorialAmount_tempsave"></c:set>
<c:set var="validTutorialAmount_reapply">${"true" == reqTutorialAmount ? "k_required:[''] : ''"}</c:set>
<c:set var="validTutorialAmount_process">${"true" == reqTutorialAmount ? "k_required:[''] : ''"}</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${surfaceChapterTitle}"
    gadgetId="${gadgetId}"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="caution" style="display:block;">
</k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:blockContainer id="${gadgetBlockId}">
<table class="imui-form" style="table-layout:fixed">
<colgroup>
<col style="width:150px;" />
<col />
</colgroup>
<tr>
<th>
<k:surface
    name="tutorialTextSurface"
    surfaceKey="${surfaceTutorialText}"
    mode="0"
    requiredMark="${reqTutorialText}"
/>
</th>
<td>
<k:text
    name="tutorialText"
    style="width:675px;"
    surfaceKey="${surfaceTutorialText}"
    mode="${modeTutorialText}"
    validationDefault="${validTutorialText_def}"
    validationApply="${validTutorialText_apply}"
    validationTempsave="${validTutorialText_tempsave}"
    validationReapply="${validTutorialText_reapply}"
    validationProcess="${validTutorialText_process}"
    maxlength="250"
/>
</td>
</tr>
</table>
</k:blockContainer>
</form>
</k:gadgetContainer>
```

```

</td>
</tr>
<tr>
<th>
<k:surface
  name="tutorialAmountSurface"
  surfaceKey="{surfaceTutorialAmount}"
  mode="0"
  requiredMark="{reqTutorialAmount}"
/>
</th>
<td>
<k:numericText
  name="tutorialAmount"
  style="width:100px;"
  class="kaiden-ui-numericText-label-nowrap"
  surfaceKey="{surfaceTutorialAmount}"
  mode="{modeTutorialAmount}"
  validationDefault="{validTutorialAmount_def}"
  validationApply="{validTutorialAmount_apply}"
  validationTempsave="{validTutorialAmount_tempsave}"
  validationReapply="{validTutorialAmount_reapply}"
  validationProcess="{validTutorialAmount_process}"
  formatted="view"
/>
</td>
</tr>
</table>
</k:blockContainer>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){

var eventMan = KAIDEN.getGadget("${gadgetId}").eventManager
, blockAccessor = KAIDEN.getGadget("${gadgetId}").blockAccessors["${gadgetBlockId}"];

/*****
 * 内部イベント
 *****/

/** 入力項目のクリア */
KAIDEN.getGadget("${gadgetId}").variable.clearData = function() {
  blockAccessor("tutorialText").setValue("");
  blockAccessor("tutorialAmount").setValue("");
};

/*****
 * 受信イベント (push)
 *****/

//loadData データ読み込み イベントリスナの登録
eventMan.registerPushListener("loadData", function(**Object*/arg) {
  KAIDEN.getGadget("${gadgetId}").variable.clearData();
  return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registerPushListener("loadData4Copy", function(**Object*/arg) {
  blockAccessor("tutorialText").setValue("");
  blockAccessor("tutorialAmount").setValue("");
  return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg, {"${gadgetBlockId}":{"tutorialAmount":true}});
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registerPushListener("loadData4Settle", function(**Object*/arg) {
  blockAccessor("tutorialText").setValue("");
  blockAccessor("tutorialAmount").setValue("");
  return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg, {"${gadgetBlockId}":{"tutorialAmount":true}});
});
});
</script>

```



コラム

スマートフォン画面を作成する場合、スマートフォン用にJSPを作成する必要があります。
 詳細は「[スマートフォン用のJSP作成](#)」を参照ください。

クライアントサイド

本項では、ガジェット(経費旅費)クライアントサイドのプログラミング方法について解説します。

- 申請書ガジェットの設定内容取得
- スマートフォン用のJSP作成
 - データの取得と画面表示

申請書ガジェットの設定内容取得

申請書ガジェットでは、サーフェースの内容や「申請書メンテナンス」で設定された内容をスクリプトレット(EL式)にて取り出せます。

「jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageDto」をappDtoという名前で取り出せます。

詳細は『intra-mart Accel Kaiden! JavaDoc』を参照してください。

- コーディング例

```
<c:set var="gadgetClass">expenseInfo</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/expenseInfoV01</c:set>
<c:set var="gadgetBlockId">expenseInfoTable</c:set>

<!-- 設定内容の取り出し -->
<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="block" value="${gadget.blocks[gadgetBlockId]}" />

<!--サーフェスの取り出し-->
<%-- タイトル --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<%-- テキスト --%>
<c:set var="surfaceTutorialText">${gadget.surfaces["summary1"]}</c:set>

<!-- 入力区分 / 必須区分の取り出し -->
<%-- テキスト --%>
<c:set var="modeTutorialText">${nomal.fields["summary1"].inputType}</c:set>
<c:set var="reqTutorialText" >${nomal.fields["summary1"].requiredFlag}</c:set>
```

スマートフォン用のJSP作成

PC版のJSPではjavascript(jQuery)などを利用して、データの表示などを行っていましたが、スマートフォン版のJSPでは、javascriptは利用せずに、JSPのみで作成しています。

※一部jQueryモバイルの機能を利用しています。

i コラム

データの表示がないJSPでも「ファイル名 + "_sp.jsp"」を用意しておく必要があります。

データの取得と画面表示

以下のようにEL式(function)の「kimw:load4View」にて、通常ajaxにてデータを取得しているところを、スクリプトレット(EL式)にてデータを取得できます。

- コーディング例

```
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>

<c:set var="gadgetClass">ガジェットクラス</c:set>
<c:set var="gadgetVariation">ガジェットバリエーション</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">コンポーネント名</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>

<%-- ガジェットデータの読み込み --%>
<c:set var="gadgetResponse" value="${kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>

<%-- ガジェットデータを画面に表示 --%>
<c:forEach items="${gadgetResponse.data[TMテーブルID]}" var="detailData">
  <c:out value="${detailData.item1}">
  <c:out value="${detailData.item2}">
  <c:out value="${detailData.item3}">
</c:forEach>
```

i コラム

コンポーネント名は、ajax通信するActionクラスの名前を指定します。

指定された名前でDIコンテナから取り出して処理します。

サーバサイド

本項では、ガジェット(経費旅費)サーバサイドのプログラミング方法について解説します。

- [申請書ガジェットActionクラスプログラミング](#)
 - [申請書ガジェットActionクラスの構成](#)
- [申請書ガジェットLogicクラスプログラミング](#)
 - [申請書ガジェットLogicクラスの構成](#)
- [IM-Workflow案件番号採番機能の作成](#)
 - [案件番号採番処理](#)
 - [案件番号採番処理の設定ファイル](#)
- [申請書ガジェットManagerクラスプログラミング](#)
 - [申請書ガジェットManagerクラスの構成](#)

[申請書ガジェットActionクラスプログラミング](#)

申請書ガジェットActionクラスのサーバサイドプログラミングについて解説します。

ロジッククラスの呼び出し

項目

- チュートリアルプログラム
- 画面表示データ取得メソッド(view)
- 申請書コピーデータ取得メソッド(viewForCopy)
- 過去案件詳細画面表示データ取得メソッド(viewArchive)

アクションクラスにてロジッククラスのメソッドを呼び出します。
3つのパターンがありますので、それぞれについて解説します。

チュートリアルプログラム

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのアクションはこちら](#))

```
/**
 * チュートリアルガジェット基底クラス (Action).
 */
public abstract class AbstractGadgetTutorialActionBase implements GadgetActionIF {

    // 中略...

    /** 画面表示用データを返却. */
    @Override
    public LogicResultDto getViewData() {
        return getLogic().view(getForm(), new ConvertForJsonSource());
    }

    // 中略...

    /** 申請書コピーのデータロード. */
    @Override
    @Execute(validator = false)
    public String loadCopy() {
        ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
        return null;
    }

    // 中略...

    /** 過去案件詳細画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadArchiveMatterDetail() {
        ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConvertor())));
        return null;
    }

    // 中略...
}
```



コラム

後述する申請書ガジェットマネージャクラスでview,viewForCopy,viewArchiveに対応するメソッドがあります。
下記の違いを意識した取得方法を実装してください。

画面表示データ取得メソッド(view)

対象の申請書自身の情報を取得する際に使用します。

申請書コピーデータ取得メソッド(viewForCopy)

申請書コピー機能使用時に使用します。
画面表示データ取得メソッド(view)とはマスタ参照時の基準日を切り替える必要がある点で異なるため、viewとは別メソッドとなっています。

- view・・・対象の申請書自身の基準日
- viewForCopy・・・コピー先申請書の基準日

過去案件詳細画面表示データ取得メソッド(viewArchive)

intra-martのアーカイブ機能によって過去案件となった申請書情報を取得する際に使用します。
画面表示データ取得メソッド(view)とは取得対象テーブルが異なるため、viewとは別メソッドとなっています。

表示値の書式変換処理

ガジェットに表示する値の書式変換処理は、Actionクラスに書式変換クラスを実装し、Logic呼び出し時の引数に指定しています。

チュートリアルで作成したチュートリアルガジェットでは、「金額」項目に対して書式変換しています。

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのアクションはこちら](#))

```
/**
 * チュートリアルガジェット基底クラス (Action).
 */
public abstract class AbstractGadgetTutorialActionBase implements GadgetActionIF {

    /** 処理画面のデータロード. */
    @Override
    @Execute(validator = false)
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /** 画面表示用データを返却. */
    @Override
    public LogicResultDto getViewData() {
        return getLogic().view(getForm(), getConvertor());
    }

    /** 書式変換設定を返却. */
    protected TutorialConvertor getConvertor() {
        return new TutorialConvertor(getForm());
    }

    /** チュートリアルガジェットの書式変換. */
    private class TutorialConvertor extends ConvertForJsonSource {

        /** リクエストパラメータ. */
        private final AbstractImwPageForm frm;

        /** NumberHelper. */
        private final NumberHelper numberHelper;

        /** コンストラクタ. */
        public TutorialConvertor(AbstractImwPageForm frm) {
            this.frm = frm;
            numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
        }

        /** 入力項目用書式変換. */
        @Override
        protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

            Map<String, Object> result = new HashMap<String, Object>(map);

            // 金額の書式変換
            if (result.containsKey(TUTORIAL_AMOUNT)) {
                BigDecimal amount = (BigDecimal) result.get(TUTORIAL_AMOUNT);

                Object currencyCd = result.get(TUTORIAL_CURRENCY_CD);
                result.put(TUTORIAL_AMOUNT,
                    numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
            }

            return result;
        }

        /** 表示項目用書式変換. */
        @Override
        protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
            Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

            // 金額の書式変換
            if (map.containsKey(TUTORIAL_AMOUNT)) {
                BigDecimal amount = (BigDecimal) map.get(TUTORIAL_AMOUNT);
                Object currencyCd = map.get(TUTORIAL_CURRENCY_CD);
                result.put(TUTORIAL_AMOUNT,
                    numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
            }

            return result;
        }
    }
}
```

解説

チュートリアルではTutorialConvertorという書式変換クラスをインナークラスで作成しています。

書式変換クラスには、入力項目用書式変換を行うformatPlainValueメソッドと、表示項目用書式変換を行うformatFormattedValueメソッドを実装します。
このインナークラスは、Logic呼び出し時に引数に渡し、Logicで書式変換処理を実行します。

i コラム

業務固有の書式変換処理が不要である場合は、jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSourceクラスを渡してください。

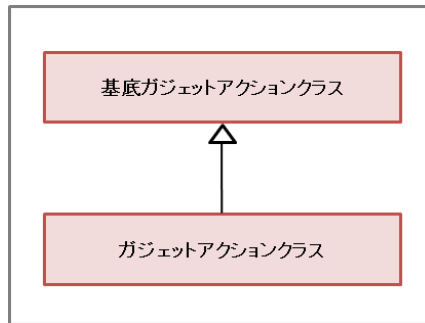
i コラム

申請書ガジェットActionクラスは画面表示処理の際に呼ばれますが、申請、承認などワークフローアクション処理時には呼ばれません。

i コラム

ガジェットバリエーション違いで異なる実装をすべきかどうかを検討し、「業務基底アクションクラス」「業務アクションクラス」の実装を行ってください。

申請書ガジェットActionクラスの構成



申請書ガジェットLogicクラスプログラミング

申請書ガジェットLogicクラスのサーバサイドプログラミングについて解説します。

画面表示処理

申請書ガジェットLogicクラスの画面表示処理について解説します。

基底クラスで実行される処理の流れに沿って、業務ロジックで実装する処理を呼び出される順番で解説していきます。

- データの取得
- 検索データのマスタ整合性チェック
- 書式変換処理

データの取得

データ取得処理メソッドを実装します。

業務ロジックにて実装することが強制されています。

メソッド名は、画面表示データの取得は「selectBlockDatasForView」、申請書コピーデータの取得は「selectBlockDatasForCopy」、

過去案件詳細画面表示データの取得は「selectBlockDatasForArchive」であり、それぞれ実装する必要があります。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

```
/** 初期表示用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        // 新規申請の場合
        result.put(BLOCK_KEY_TUTORIAL_TABLE, new ArrayList<GadgetTutorialManagerDto>());
    } else {
        // 新規申請でない場合
        result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectView(criteria));
    }
    return result;
}

/** 申請書コピー用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectViewForCopy(criteria));
    return result;
}

/** 過去案件用検索詳細処理. */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    // アーカイブは考慮しない
    return new HashMap<String, Object>();
}
```

- 解説
 - selectBlockDatasForView
画面表示データの検索処理を実装します。
システム案件IDとユーザデータIDの有無で新規申請かどうかを判定しています。
新規申請でない場合は登録済みのトランザクションデータを取得し、ガジェットブロックIDをマップキーにして返却します。



注意

新規申請の場合はnullを返さないように注意してください。

- selectBlockDatasForCopy
申請書コピーデータの検索処理を実装します。
検索結果をガジェットブロックIDをマップキーにして返却します。
selectBlockDatasForViewとは異なるマネージャAPIを使用している点に注意してください。
- selectBlockDatasForArchive
過去案件詳細画面表示データの検索処理を実装します。
検索結果をガジェットブロックIDをマップキーにして返却します。
[アーカイブ](#)については別途解説しますのでここでは割愛します。

検索データのマスタ整合性チェック

取得したデータのマスタ存在チェック処理を実装します。

業務ロジックにて実装することが強制されています。

メソッド名は、「checkSelectedData」とします。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

```

/** 検索結果のマスタ整合性チェック. */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_TUTORIAL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<GadgetTutorialManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), GadgetTutorialManagerDto.class);
            for (GadgetTutorialManagerDto sqlResult : sqlResultList) {
                if (isNotEmpty(sqlResult.tutorialCurrencyCd) && isEmpty(sqlResult.tutorialCurrencyName)) {
                    String[] fieldId = getArgs("tutorialCurrencyCd", "tutorialCurrencyName");
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(sqlResult.tutorialCurrencyCd, sqlResult.tutorialCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_TUTORIAL_TABLE, fieldId, null, masterName, values,
                        message));
                }
            }
        }
    }
    return result;
}

```

■ 解説

検索結果のマスタ整合性チェック処理を実装します。

この処理は「画面表示データ検索」「申請書コピーデータ検索」「過去案件詳細画面表示データ検索」で取得したデータに対して実施されます。チュートリアルでは、トランザクションテーブルの通貨コードに紐づくIM-共通マスタ通貨マスタの整合性をチェックしています。

マスタ不整合が発生していると判定した場合、「フィールドキー」「マスタ名」「フィールド値」「エラーメッセージ」を、エラー情報保持用のオブジェクトにセットして、返却します。



コラム

マスタ不整合情報として「フィールドキー」「マスタ名」「フィールド値」「エラーメッセージ」をセットして返却していますが、intra-mart Accel Kaiden! 標準では画面に表示していません。(マスタ不整合が発生しているかどうかのみ) 詳細な不整合情報を画面に表示するための拡張向けの情報となります。

書式変換処理

取得データの書式変換処理を実装します。

アクションクラスから渡された書式変換クラスを使用します。

また、業務ロジックでの実装は必須ではありませんが、ガジェットブロックタイプが「normal」の場合、ここでList形式からMap形式に変換する必要があります。メソッド名は「mapConvert」とします。

```

/** JSON出力用の形式に変換し同時に書式も設定. */
@Override
protected Map<String, Object> mapConvert(Map<String, Object> dataMap, ConvertForJsonSource converter,
    ProcessType processType) {

    // チュートリアルガジェットはガジェットブロックタイプが「normal」ですので、ListからMapに変換しています
    @SuppressWarnings("unchecked")
    List<GadgetTutorialManagerDto> listDto = (List<GadgetTutorialManagerDto>) dataMap.get(BLOCK_KEY_TUTORIAL_TABLE);
    if (CollectionUtil.isEmpty(listDto)) {
        dataMap.put(BLOCK_KEY_TUTORIAL_TABLE, BeanUtil.toMap(listDto.get(0), true));
    } else {
        dataMap.put(BLOCK_KEY_TUTORIAL_TABLE, new HashMap<String, Object>());
    }

    // コンバータクラスを使用して書式変換処理を行っています
    return converter.mapConvert(dataMap);
}

```

■ 解説

書式変換処理に使用するコンバータクラスはActionからLogic呼び出し時に渡した書式変換クラスになります。

もしmapConvertを業務ロジックで実装しない場合、intra-mart Accel Kaiden!内部の基底ロジックでreturn converter.mapConvert(dataMap);が実行されます。

ワークフローアクション処理

申請書ガジェットLogicクラスのワークフローアクション処理について解説します。

項目

- [処理の流れ](#)
- [前処理](#)
- [ガジェット内バリデート処理](#)
- [ガジェット間バリデート処理](#)
- [処理](#)

[処理の流れ](#)

intra-mart Accel Kaiden! のワークフローアクション処理はjp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorがつかさどります。

以降、これをガジェットメディエータと呼びます。

ガジェットメディエータは以下の処理を行っています。

- [前処理](#)
- [ガジェット内バリデート処理](#)
- [ガジェット間バリデート処理](#)
- [処理](#)

複数ガジェットが存在する場合、全ガジェットに対して上記の処理を実行します。

処理順は、例えばAガジェットとBガジェットが存在する場合、以下の順序となります。

1. 「Aガジェット前処理」
2. 「Bガジェット前処理」
3. 「Aガジェット内バリデート処理」
4. 「Bガジェット内バリデート処理」
5. 「Aガジェット間バリデート処理」
6. 「Bガジェット間バリデート処理」
7. 「Aガジェット処理」
8. 「Bガジェット処理」

各工程でどのような処理を行っているかを、以下に簡単に記載します。詳細は個別のページを参照ください。

業務ロジックで実装不要のものは個別ページはありません。

[前処理](#)

- [ブロックごとのサーフェスマップ生成](#)
- [バリデート定義](#)
- [ガジェットデータ取得](#)
- [ガジェットデータ補完](#)

[ガジェット内バリデート処理](#)

- [バリデート定義による入力チェック](#)
- [バリデート定義以外の入力チェック](#)

[ガジェット間バリデート処理](#)

- [ガジェット間バリデート処理](#)

[処理](#)

- [メイン処理](#)

バリデート定義

入力値チェックを業務ロジックにて定義します。

このタイミングでは入力チェックは実施されず、ガジェット内バリデート処理にて実施されます。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

AbstractGadgetTutorialLogicBase.java

```

/** 申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 一時保存案件の申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 未申請案件の申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 承認アクション時バリデート定義処理. */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 承認終了アクション時バリデート定義処理. */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 再申請アクション時バリデート定義処理. */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/** 一時保存(新規登録)アクション時バリデート定義処理. */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/** 一時保存(更新)アクション時バリデート定義処理. */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/** フィールドバリデート定義. */
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {
    // 本機能ではサブクラスで実装します
}

```

GadgetTutorialV01Logic.java

```

/** フィールドバリデート定義. */
@Override
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    if (fieldDefsMap.containsKey(BLOCK_KEY_TUTORIAL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_TUTORIAL_TABLE, tmpFlg);
        addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialText", "maxLength",
            StringUtil.toString(TUTORIAL_TEXT_LEN));
        addValidateDefsMap(BLOCK_KEY_TUTORIAL_TABLE, "tutorialCurrencyCd", "immCurrencyExists",
            "tutorialCurrencyCd", "searchCriteriaCompany");
        addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialAmount", "currencyAmnt",
            "tutorialCurrencyCd", "searchCriteriaCompany");
        addValidateDefsMapOnlyInput(BLOCK_KEY_TUTORIAL_TABLE, "tutorialAmount", "numericMin",
            StringUtil.toString(TUTORIAL_AMOUNT_NUMERIC_MIN));
    }
}

```

解説

GadgetTutorialV01Logicでチュートリアルガジェットの入力値チェックを定義しています。

addValidateDefsRequired(BLOCK_KEY_TUTORIAL_TABLE, tmpFlg);にて、申請書ガジェットマスターデータを元に必須チェックを定義しています。
第一引数にガジェットブロックID、第二引数に一時保存フラグを指定することで、一時保存でない場合に必須チェックを実施させることができます。

その他の行で各項目に対して必須チェック以外の入力チェックを定義しています。
addValidateDefsMapOnlyInputは画面項目の表示モードが"0"の場合のみチェックを実施し、
addValidateDefsMapは画面項目の表示モードに関わりなくチェックを実施します。

コラム

表示モードについては「[Mode&Formated](#)」も参照ください。

ガジェットデータ取得

ガジェットデータを取得します。

処理の大部分は基底ロジックで行っています。

ガジェット情報は、表示モードが"0"の場合はリクエストデータから取得できますが、表示モードが"1"の場合はリクエストデータから取得することができません。そのためトランザクションテーブルからガジェットデータを取得し、これをリクエストデータとマージしてアクション処理で扱う「ガジェットデータ」としています。

上記のうち、トランザクションテーブルからガジェットデータを取得する処理は業務ロジックで実装する必要があります。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

```
/** データ取得メソッド */
@Override
protected LinkedHashMap<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    GadgetTutorialManagerDto dto = Beans.createAndCopy(GadgetTutorialManagerDto.class, conditionDto).execute();
    List<GadgetTutorialManagerDto> listSelect = gadgetTutorialManager.selectList(dto);
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
    Map<String, Object> dtoMap = new HashMap<String, Object>();
    if (!listSelect.isEmpty()) {
        dtoMap = BeanUtil.toMap(listSelect.get(0), true);
    }
    rv.put(BLOCK_KEY_TUTORIAL_TABLE, dtoMap);
    return rv;
}
```

- 解説

ManagerのAPIを使用して、チュートリアルガジェットテーブルから該当の申請書のデータを取得、結果をMapにセットして返却しています。

ガジェットデータ補完

ガジェットデータを補完します。

補完処理は基底ロジックで行っている処理と、業務ロジックで実装することができる処理があります。

基底ロジックでは、ガジェットデータに対して「検索基準日」「検索基準会社」をガジェットデータに補完しています。

業務ロジックには、要件に合わせた補完処理を実装することになります。

例えば、画面に入力項目は用意しないが、ガジェットデータとして登録したい値がある場合などでは、この補完処理にて実装することができます。

チュートリアルガジェットでは、金額に対する通貨コードを補完しています。(画面に入力項目はありません。)

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

AbstractGadgetTutorialLogicBase.java

```
/** データ補完処理. */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    // サブクラスで実装するため基底では何も行わない
    return;
}
```

GadgetTutorialV01Logic.java

```
/** データ補完処理. */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    Map<String, Object> blockDat = getJsonDataNormalBlock(BLOCK_KEY_TUTORIAL_TABLE);
    String currencyCd1 = StringUtil.toString(blockDat.get("tutorialCurrencyCd"));

    if (isEmpty(currencyCd1)) {
        blockDat.put("tutorialCurrencyCd",
            currencyHelper.getCompanyCurrencyCd((String) blockDat.get("searchCriteriaCompany")));
    }
}
```

■ 解説

チュートリアルガジェットテーブルに、会社通貨コードを登録するために補完しています。



コラム

currencyHelper.getCompanyCurrencyCd("会社コード");で、会社通貨の通貨コードを取得できます。
会社通貨は「WEB-INF/conf/kaiden2/BaseSettings.xml」で設定されています。



コラム

ガジェットブロックタイプが「normal」の場合、getJsonDataNormalBlock("ガジェットブロックID");でガジェットデータを取得できます。
対して「detail」の場合はgetJsonDataDetailBlock("ガジェットブロックID");で取得できます。
ガジェットデータはすべてString型で格納されています。

バリデート定義以外の入力チェック

ガジェット内バリデート処理には「バリデート定義による入力チェック」「バリデート定義以外を入力チェック」があります。
「バリデート定義による入力チェック」は [バリデート定義](#) で定義した通りに基底クラスが入力チェックを実施しますので特に解説しません。

「バリデート定義以外を入力チェック」は必要であれば業務ロジックで実装します。
例えば、バリデータクラスでは対応しきれないような、明細行を跨いだ入力チェックなどは、ここで実装します。

チュートリアルガジェットではバリデート定義以外を入力チェックは実装していません。

ガジェット間バリデート処理

ガジェット間バリデート処理は、必要であれば業務ロジックで実装します。

ガジェット間バリデートは、ガジェットメタデータによって全ガジェットのガジェット内バリデートを終えた後に実施されます。
必須チェックや型チェックのような、単項目チェックレベルのバリデートは完了している前提でガジェット間バリデートを実装できます。

[ガジェット通信](#)によって他のガジェットの値を取得し、自身のガジェットの値と比較する、といったチェックを実装します。

チュートリアルガジェットではガジェット間バリデート処理の入力チェックは実装していません。

メイン処理

ワークフローアクションのメイン処理を実装します。
業務ロジックにて実装することを強制されています。

チュートリアルでは、申請や一時保存のタイミングでトランザクションテーブルにデータを登録しています。

チュートリアルのプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのロジックはこちら](#))

```

/** 申請処理時に実行されるメソッド: */
@Override
public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/** 一時保存案件の申請処理時に実行されるメソッド: */
@Override
public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/** 未申請案件の申請処理時に実行されるメソッド: */
@Override
public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/** 承認処理時に実行されるメソッド: */
@Override
public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/** 承認終了処理時に実行されるメソッド: */
@Override
public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/** 否認処理時に実行されるメソッド: */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 取止め処理時に実行されるメソッド: */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 案件操作処理時に実行されるメソッド: */
@Override
public int matterHandle(KaidenDto dto) {
    return 0;
}

/** 引戻し処理時に実行されるメソッド: */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 再申請処理時に実行されるメソッド: */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/** 保留処理時に実行されるメソッド: */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

```

```

/** 保留解除処理時に実行されるメソッド. */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 差戻し処理時に実行されるメソッド. */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 差戻し後の引戻し処理時に実行されるメソッド. */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 一時保存(新規登録)処理時に実行されるメソッド. */
@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/** 一時保存(削除)処理時に実行されるメソッド. */
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/** 一時保存(更新)処理時に実行されるメソッド. */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/** 案件開始処理時に実行されるメソッド. */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 案件終了処理時に実行されるメソッド. */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 到達処理時に実行されるメソッド. */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/** 未完了案件削除処理時に実行されるメソッド. */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/** 過去案件削除処理時に実行されるメソッド. */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    // アーカイブは考慮しない
    return 0;
}

/** 完了案件削除処理時に実行されるメソッド. */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/** データ登録処理. */
protected int insertTable() throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = JsonUtil.getDto(GadgetTutorialManagerDto.class,
        super.getJsonDataNormalBlock(BLOCK_KEY_TUTORIAL_TABLE));
    return gadgetTutorialManager.entry(makeSingleInsertList(mngDto));
}

/** データ削除処理. */
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = new GadgetTutorialManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userId = super.getGadgetDto().userId;

```

```

mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
if (tempFlg) {
    mngDto.systemMatterId = "temporary";
}
return gadgetTutorialManager.remove(mngDto);
}

/** データ削除処理(履歴を含む). */
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    GadgetTutorialManagerDto mngDto = new GadgetTutorialManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    return gadgetTutorialManager.removeAll(mngDto);
}

```

解説

各ワークフローアクション処理に対応するメソッド名は、上記サンプルプログラムを参照ください。

- ワークフローアクション処理の引数について
KaidenDtoクラスのオブジェクトが渡されますが、実態はjp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTOクラスになります。
詳細はJavaDocを参照ください。
- approve、approveEndメソッド ユーザーデータIDの切り替えについて
ユーザーデータIDは一時保存のときには「temporary」が振られ、申請時や再申請時に新たに一意になるIDが振られます。
このように新たにユーザーデータIDが振られている場合、直前のユーザーデータIDはsuper.getGadgetDto().beforeUserDataIdで取得することができます。
- insertTableメソッド ガジェットデータの取得について
super.getJsonDataNormalBlock("ガジェットブロックID")で、ガジェットブロックタイプが「normal」のデータを取得することができます。
対して、ガジェットブロックタイプが「detail」のガジェットデータはsuper.getJsonDataDetailBlock("ガジェットブロックID")で取得できます。
詳細はJavaDocを参照ください。

項目

- [ガジェット間通信保持保証用インタフェース](#)
- [イベント受信側Logic](#)
- [イベント要求側Logic](#)

ガジェット間通信の実装方法について解説します。

同一申請書内に異なるガジェットの値を参照したい場合、ガジェット間通信を実装する必要があります。
本項では、Bガジェットの保持するXXX金額を、Aガジェットが取得するケースのサンプルプログラムを下記に掲載します。

[ガジェット間通信保持保証用インタフェース](#)

ガジェット間通信イベントを受信し返却するためには、ガジェット間通信用インターフェースを作成します。

```
package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common;

import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.gadget.GadgetCouplingIF;

/**
 * ガジェット間通信 XXX金額取得メソッド保持保証用インタフェース.
 */
public interface ClaimXxxAmountIF extends GadgetCouplingIF {

    /**
     * メソッド名:XXX金額取得.
     */
    public static final String METHOD_NM_GET_CLAIM_XXX_AMOUNT = "getClaimXxxAmount";
    /** 戻り値キー:XXX金額. */
    public static final String RTN_KEY_CLAIM_XXX_AMOUNT = "claimXxxAmount";

    /**
     * XXX金額を返却.
     *
     * 戻り値に格納する値の型は、BigDecimal型にする必要があります。
     */
    public Map<String, Object> getClaimXxxAmount(Map<String, Object> params);
}
```

■ 解説

jp.co.slcs.kaiden2.base.foundation.gadget.GadgetCouplingIFを継承します。
ガジェット間通信の受信イベントを持たせるガジェットに当インターフェースを実装させ、getClaimXxxAmountメソッドを保持することを保証します。

intra-mart Accel Kaiden! 内部のガジェット間通信基底クラスではメソッドの戻り値をObject型でしか扱えないため、実態の型をJavaDocに記載するようにしてください。

[イベント受信側Logic](#)

ガジェット間通信イベントを受信する側のLogicには以下のように実装します。

```

/**
 * Bガジェット用ロジッククラス.
 */
public class BGadgetV01Logic extends AbstractBGadgetLogicBase implements ClaimXxxAmountIF {

    // 中略...

    /**
     * XXX金額の合計を返却.
     */
    public Map<String, Object> getClaimXxxAmount(Map<String, Object> params) {
        List<Map<String, Object>> blocjData = getJsonDataDetailBlock("bGadgetTable");
        Map<String, Object> retMap = new HashMap<String, Object>();
        BigDecimal sum = BigDecimal.ZERO;

        for (Map<String, Object> j : blocjData) {
            String strAmnt = StringUtil.toString(j.get("companyAmount"));
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }

        retMap.put(ClaimXxxAmountIF.RTN_KEY_CLAIM_XXX_AMOUNT, sum);

        return retMap;
    }

    // 中略...
}

```

■ 解説

Bガジェットが保持するXXX金額の合計を算出し、値をインターフェースで定義したキーに紐づけてマップにセットします。戻り値の型はインターフェースのJavaDocで指定された型にするように注意してください。

イベント要求側Logic

ガジェット間通信イベントを要求する側のLogicには以下のように実装します。
取得値の型はインターフェースのJavaDocで指定された型にするように注意してください。

戻り値が一つの場合

```

BigDecimal appCost = super.getSingleValue(BigDecimal.class, args, ClaimXxxAmountIF.class,
    ClaimXxxAmountIF.METHOD_NM_GET_CLAIM_XXX_AMOUNT,
    ClaimXxxAmountIF.RTN_KEY_CLAIM_XXX_AMOUNT);

```

戻り値が複数の場合

```

List<BigDecimal> appCost = super.getMultiValue(BigDecimal.class, args, ClaimXxxAmountIF.class,
    ClaimXxxAmountIF.METHOD_NM_GET_CLAIM_XXX_AMOUNT,
    ClaimXxxAmountIF.RTN_KEY_CLAIM_XXX_AMOUNT);

```

項目

- [アーカイブ用のテーブルを用意](#)
- [S2JDBC-Gen自動生成ソース](#)
- [Logicクラスの実装](#)
- [Managerクラスの実装](#)

アーカイブ処理の実装方法について解説します。
IM-Workflow のアーカイブ処理で、案件の情報が過去案件として退避されますが、intra-mart Accel Kaiden! 側で合わせて退避する必要がある場合は実装することになります。
サンプルとして、チュートリアルで作成したチュートリアルガジェットにアーカイブ処理を追加していきます。



コラム

IM-Workflow のアーカイブ処理では「テーブル名 + "yyyyMM"」のテーブルに退避しており年月ごとのテーブルが作成されますが、intra-mart Accel Kaiden! 標準では退避テーブルは一つで、年月のカラムを保持します。

アーカイブ用のテーブルを用意

(過去案件)チュートリアルガジェットテーブル「ka99t_tutorial」のレイアウトは下記の通りです。

No.	論理名	物理名	型	長さ	必須	PK
1	システム案件ID	system_matter_id	varchar2	20	Y	1
2	ユーザデータID	user_data_id	varchar2	20	Y	2
3	ガジェットクラス	gadget_class	varchar2	100	Y	3
4	ガジェットインスタンス	gadget_instance	number	18,0	Y	4
5	ロケールid	locale_id	varchar2	50	Y	4
6	アーカイブ年月	archive_month	varchar2	6	Y	4
7	ガジェットID	gadget_id	varchar2	220	Y	
8	ガジェットバリエーション	gadget_variation	varchar2	100	Y	
9	テキスト	tutorial_text	varchar2	1000		
10	通貨コード	tutorial_currency_cd	varchar2	100		
11	通貨名	tutorial_currency_name	varchar2	1000		
12	金額	tutorial_amount	number	28,10		
13	予備項目A	extension_a	varchar2	1000		
14	予備項目B	extension_b	varchar2	1000		
15	予備項目C	extension_c	varchar2	1000		
16	予備項目D	extension_d	varchar2	1000		
17	予備項目E	extension_e	varchar2	1000		
18	予備項目F	extension_f	varchar2	1000		
19	予備項目G	extension_g	varchar2	1000		
20	予備項目H	extension_h	varchar2	1000		
21	予備項目I	extension_i	varchar2	1000		
22	予備項目J	extension_j	varchar2	1000		
23	登録日時	entry_ts	timestamp		Y	
24	登録者コード	entry_user_cd	varchar2	100	Y	
25	更新カウンタ	renew_cnt	number	18,0	Y	
26	更新日時	renew_ts	timestamp		Y	
27	更新者コード	renew_user_cd	varchar2	100	Y	

■ 解説

[チュートリアルガジェットテーブル](#)と比較すると、「ロケールID」「アーカイブ年月」「通貨名」が追加されています。

- ロケールID
アーカイブ処理時にシステムロケール分のレコードを作成します。
- アーカイブ年月
アーカイブ処理を実施した年月を「YYYYMM」形式で格納します。
- 通貨名
アーカイブ時点のマスタ情報を格納します。
マスタ情報変更の過去案件情報に影響及ぼさないようにします。

S2JDBC-Gen自動生成ソース

(過去案件)チュートリアルマスタテーブルを作成後、S2JDBC-Gen にて以下のソースファイルを自動生成します。

- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.Ka99tTutorial
- jp.co.slcs.kaiden2.tutorial.foundation.model.entity.Ka99tTutorialNames
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.Ka99tTutorialCondition
- jp.co.slcs.kaiden2.tutorial.foundation.model.service.Ka99tTutorialService

Logicクラスの実装

チュートリアルガジェットのLogicクラスを修正します。

変更箇所のみを記載しています。([チュートリアルガジェットのロジックはこちら](#))

```
/**
 * チュートリアルガジェット基底クラス (Logic).
 */
public abstract class AbstractGadgetTutorialLogicBase extends AbstractExpensePluginGadgetLogic {

    // 中略...

    /** 過去案件削除処理時に実行されるメソッド. */
    @Override
    public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = gadgetTutorialManager
            .removeAllArchive(super.convCriteriaForSqlDtoToForArchive(super.getGadgetDto(), imwPluginParameterDto));
        return rv;
    }

    // 中略...

    /** 初期表示用検索詳細処理. */
    @Override
    protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
        Map<String, Object> result = new HashMap<String, Object>();

        if (StringUtil.isEmpty(criteria.systemMatterId)
            || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
            // 新規申請の場合
            result.put(BLOCK_KEY_TUTORIAL_TABLE, new ArrayList<GadgetTutorialManagerDto>());
        } else {
            // 新規申請でない場合
            result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectView(criteria));
        }
        return result;
    }

    // 中略...

    /** 過去案件用検索詳細処理. */
    @Override
    protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
        Map<String, Object> result = new HashMap<String, Object>();
        result.put(BLOCK_KEY_TUTORIAL_TABLE, gadgetTutorialManager.selectViewArchive(criteria));
        return result;
    }

    // 中略...

    /** 案件退避処理時の登録処理. */
    @Override
    protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
        @SuppressWarnings("unchecked")
        List<GadgetTutorialManagerDto> list = (List<GadgetTutorialManagerDto>) dataMap.get(BLOCK_KEY_TUTORIAL_TABLE);
        int rv = gadgetTutorialManager.matterArchiveRegister(criteria, list);
        return rv;
    }

    /** 退避済完了案件削除対象マネージャリストを返却. */
    @Override
    protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
        List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
        list.add(gadgetTutorialManager);
        return list;
    }

    // 中略...
}
```

解説

アーカイブに関連するメソッドを実装します。

- selectBlockDatasForArchive
過去案件詳細画面表示時に実行されます。
intra-mart Accel Kaiden! 基底クラスからcriteriaが渡されており、これがそのままManagerの検索条件となります。
criteriaにセットされている情報は下記の通りです。
 - システム案件ID
 - ユーザーデータID

- 検索基準会社
- ログインユーザのロケールID
- ガジェットクラス
- ガジェットインスタンス
- 検索基準日
- matterArchiveRegister
アーカイブ処理時に実行されます。
intra-mart Accel Kaiden! システムロケール数分繰り返し実行されます。
intra-mart Accel Kaiden! 基底クラスから渡されているcriteriaの情報は下記の通りです。
 - システム案件ID
 - ユーザデータID
 - ガジェットクラス
 - ガジェットインスタンス
 - ロケールID
 - アーカイブ年月

dataMapには、過去案件用テーブルに登録するデータがセットされています。

intra-mart Accel Kaiden! 基底クラスにて「システム案件ID」「ユーザデータID」を検索条件としてデータを取得した結果をセットしています。

取得のメソッドはselectBlockDatasForViewを使用しています。



コラム

「WEB-INF/classes/jp/co/slcs/kaiden2/extension_imw_constants.xml」にて、アーカイブ処理時に履歴データ(ユーザデータID違いのデータ)を退避するかどうかの設定がで履歴データも退避するよう設定されている場合は、上記の過去案件用テーブル登録データの取得条件からユーザデータIDがクリアされます。
デフォルトは「履歴データは退避しない」に設定されています。

- getDeleteTargetsForArchiveManagers
アーカイブ処理時に実行されます。
過去案件用テーブルにデータ登録後、チュートリアルガジェットテーブルのデータを削除するためのマネージャを返却しています。

Managerクラスの実装

チュートリアルガジェットのManagerクラスを修正します。

変更箇所のみを記載しています。([チュートリアルガジェットのマネージャはこちら](#))

```

package jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.impl;

import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManager;
import jp.co.slcs.kaiden2.tutorial.foundation.manager.tutorial.GadgetTutorialManagerDto;
import jp.co.slcs.kaiden2.tutorial.foundation.model.service.K99tTutorialService;

/**
 * チュートリアルガジェットテーブルへの各種操作を提供するAPI.
 */

public class GadgetTutorialManagerImpl extends AbstractWfManager<GadgetTutorialManagerDto> implements
    GadgetTutorialManager {

    // 中略

    /** サービスクラス. */
    @Resource
    protected Ka99tTutorialService ka99tTutorialService;

    // 中略...

    /** 過去案件検索. */
    @Override
    public List<GadgetTutorialManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        return super.selectListViewForArchive(ka99tTutorialService, criteria);
    }

    // 中略...

    /** 過去案件登録. */
    @Override
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<GadgetTutorialManagerDto> dataList) {
        return super.entryArchive(ka99tTutorialService, criteria, dataList);
    }

    /** 過去案件処理時削除. */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataId(k99tTutorialService, criteria);
    }

    /** 過去案件一括削除. */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataIdForArchive(ka99tTutorialService, criteria);
    }
}

```

■ 解説

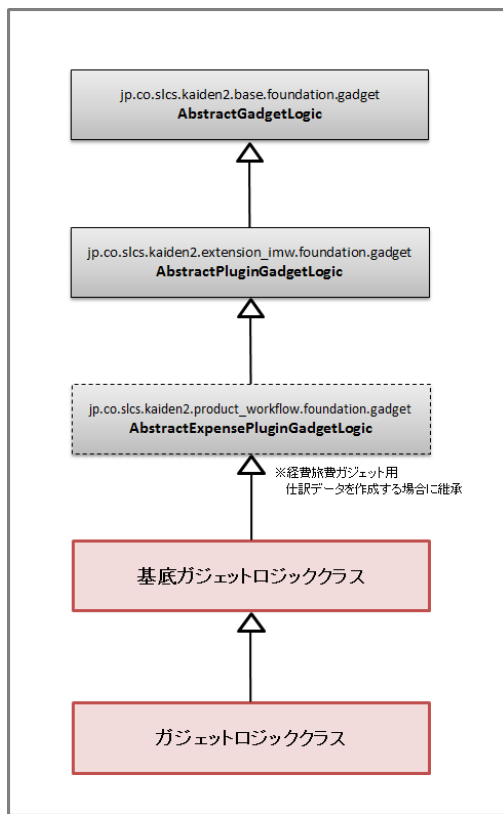
アーカイブに関連するメソッドを実装します。
intra-mart Accel Kaiden! 内部の基底クラスの呼び出しで実装は完了します。
基底マネージャの処理を簡単に解説します。

- selectViewArchive
(過去案件)チュートリアルガジェットテーブルから全アーカイブ年月分のデータを取得し返却します。
検索条件にはLogicクラスから渡されるcriteriaを使用しています。
ただし、検索条件のうちアーカイブ年月は明示的にクリアしています。
- matterArchiveRegister
(過去案件)チュートリアルガジェットテーブルにデータを登録します。
登録の際、dataListの各登録データに対して、criteriaの「ロケールID」「アーカイブ年月」を補完しています。
- matterArchivePostDelete
チュートリアルガジェットテーブルのデータを削除します。
- removeAllArchive
(過去案件)チュートリアルガジェットテーブルのデータを削除します。

i コラム

ガジェットバリエーション違いで異なる実装をすべきかどうかを検討し、「業務基底ロジッククラス」「業務ロジッククラス」の実装を行ってください。

申請書ガジェットLogicクラスの構成



IM-Workflow案件番号採番機能の作成

intra-mart Accel Kaiden!では、IM-Workflowの案件番号を採番している処理を新規で作成し、申請書マスタメンテナンス機能で申請書に紐づけることができます。実装方法を以下に解説していきます。

案件番号採番処理

案件採番処理を作成します。

jp.co.slcs.kaiden2.extension_imw.foundation.util.imw.ImwNumberingインターフェースを実装し、String型の案件番号を返却してください。

下記のサンプルでは、IM-Workflowの標準で採番される案件番号の先頭"p"を付与して返却しています。

```

package jp.co.slcs.kaiden2.tutorial.foundation.util.imw;

import java.util.Map;
import jp.co.intra_mart.foundation.workflow.exception.WorkflowException;
import jp.co.intra_mart.foundation.workflow.util.WorkflowNumberingManager;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.ActionProcessParameterDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.util.imw.ImwNumbering;

/**
 * 案件番号採番拡張クラス サンプル.
 */
public class ImwNumberingExtensionSample implements ImwNumbering {

    /**
     * 案件番号を採番処理を行い、結果を返却.
     */
    @Override
    public String getNewMatterNumber(ActionProcessParameterDto _parameter, Map<String, Object> _userParameter) {
        // IM-Workflow標準で採番される案件番号にプリフィックスを付与する
        String matterNumber = null;
        String prefix = "p";
        try {
            matterNumber = prefix + WorkflowNumberingManager.getNumber();
        } catch (WorkflowException e) {
            throw KaidenRuntimeException.makeInstance(e);
        }
        return matterNumber;
    }
}

```

案件番号採番処理の設定ファイル

案件番号採番処理設定ファイルに設定を追加します。

設定ファイルは「WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/ImwNumberingSettings.xml」です。

idタグにDIコンテナへの登録名、messageIdには申請書マスタメンテナンス画面の案件番号採番クラスドロップダウンに表示されるメッセージIDを指定してください。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<imwNumberingSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/imwNumberingSettings">
  <!-- 案件番号採番クラス -->
  <workflowMatterNumberingClasses>
    <workflowMatterNumberingClass>
      <!-- 案件番号採番クラス (DIコンテナへの登録名) -->
      <id>imwNumbering</id>
      <!-- 説明 (product_workflow_message.xmlのエントリー) -->
      <messageId>PWKF.defaultNumberingClass</messageId>
    </workflowMatterNumberingClass>
    <workflowMatterNumberingClass>
      <!-- 案件番号採番クラス (DIコンテナへの登録名) -->
      <id>imwNumberingExt</id>
      <messageId>TTTL.extensionNumberingClass</messageId>
    </workflowMatterNumberingClass>
  </workflowMatterNumberingClasses>
</imwNumberingSettings>
```



コラム

ここではdiconファイルの編集に関する記述は省略しています。
「imwNumbering」と紐づけて、作成した案件番号採番処理のクラスをdiconに設定してください。



コラム

ここではメッセージファイルの編集に関する記述は省略しています。

申請書ガジェットManagerクラスプログラミング

申請書ガジェットManagerクラスを作成する際には、
jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManagerクラスを継承することを推奨します。
申請書ガジェットマネージャに必要な一通りの機能が揃っており、実装が容易になります。

チュートリアルプログラムを抜粋したものを以下に掲載します。([チュートリアルガジェットのマネージャはこちら](#))

```
/**
 * チュートリアルガジェットテーブルマネージャクラスのinterface.
 */
public interface GadgetTutorialManager extends PluginGadgetManager<GadgetTutorialManagerDto> {

    // 中略...

}
```



コラム

マネージャの各メソッドの仕様については『intra-mart Accel Kaiden! JavaDoc』を参照してください。

申請書ガジェットManagerクラスの構成

[構成はこちら](#)

ガジェット(勤務管理)

本項では、ガジェット(勤務管理)機能の概要、チュートリアルを解説します。

- 概要
- 構成

- intra-mart Accel Kaiden! 基盤モジュール
- intra-mart Accel Kaiden! ワークフローモジュール

- 勤務表画面

 コラム

- 勤務入力画面

2015/01/21(水) 青柳 辰巳

保存

予定申請

実績申請

勤務入力一括登録

勤務状況

点検スタートス

申請スタートス

caution

警告時間

出勤

IN

OUT

勤務時間

休憩時間

09:00

-

17:00

0:00

+

休憩

休憩時間

休憩理由

09:00

12:00 - 13:00

休憩

17:00

勤務情報入力ガジェット

予定労働時間

労働時間

休憩時間

休憩理由

時間外

7:00

7:00

4:0

4:0

1:00

日次集計値表示ガジェット

即時更新

2015/01/21(水) 青柳 辰巳

保存

予定申請

実績申請

勤務入力一括登録

一括登録先日付一覧

2015/01/21

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

一括保存

一括予定申請

一括実績申請

勤務状況

点検スタートス

申請スタートス

caution

警告時間

出勤

IN

OUT

勤務時間

休憩時間

09:00

-

17:00

0:00

+

休憩

休憩時間

休憩理由

09:00

12:00 - 13:00

休憩

17:00

勤務情報一括入力ガジェット

- PDF

[illegible]

 コラム

勤務体系カテゴリ別の集計値を表示します。

チュートリアル

本項では勤務管理の機能拡張および管理者機能のプログラミング方法について解説します。
ガジェット(勤務管理)機能の解説は「[機能説明](#)」を参照してください。

intra-mart Accel Kaiden! 勤務管理モジュール の機能拡張について解説します。

勤務表機能関連の拡張

勤務体系カテゴリ共通集計値表示ガジェット

勤務体系カテゴリ共通集計値表示ガジェットは、intra-mart Accel Kaiden! 勤務管理モジュール の設定でテナント全体または会社ごとに設定されています。
<workReportCommonSummaryPattern>にガジェットクラス+ "_"+ガジェットバリエーションの形式となっており、ガジェットマスタ(k10m_gadget)から処理を取得するキーとなります。

拡張したプログラムを、ガジェットバリエーションを変更して紐づけてください。



コラム

intra-mart Accel Kaiden! 勤務管理モジュール 設定ファイルについては、『intra-mart Accel Kaiden! セットアップガイド』を参照してください。

その他ガジェット(勤務体系カテゴリ共通集計値表示ガジェット以外)

勤務体系カテゴリ共通集計値表示ガジェット以外のガジェットは、勤務管理ガジェットテーブル(k31m_lbr_app_gadget)で勤務体系カテゴリごとに設定されており、ガジェットマスタ(k10m_gadget)から処理を取得するキーとなります。

以下に固定労働時間制のもののみを記載します。

ガジェット名	区分	ガジェットクラス	ガジェットバリエーション
勤務表リスト部ガジェット(入力時)	workReportInput	workReportList	v01
勤務表リスト部ガジェット(承認時)	workReportDrApprove	workReportList	v01
勤務表リスト部ガジェット(点検時)	workReportDrInspect	workReportList	v01
勤務体系カテゴリ別集計値表示ガジェット	wrSummary	wrSummaryNormal	v01
勤務情報入力ガジェット	dailyRecord	lbrDailyRecord	v01
日次集計値表示ガジェット	dailySummary	dailySummaryNormal	v01
PDF集計値表示ガジェット	wrSummaryPdf	wrSummaryPdfNormal	v01

拡張したプログラムを、ガジェットバリエーションを変更して紐づけてください。

集計項目を新たに追加する

集計項目を新たに追加する手順を解説します。

はじめに - 集計項目とは

intra-mart Accel Kaiden! 勤務管理モジュール では 従業員が入力する勤務情報をインプットとして、何らかの計算を行った結果の値(集計値)を保持する項目 のことを集計項目と呼びます。

例えば、休憩開始時刻が12:00、休憩終了時刻が13:00、と従業員が登録したとします。

集計処理は上記時刻をインプットとして計算を行い、「休憩時間」という集計項目に1時間という結果を登録します。

また、集計する対象期間によって集計単位という属性があります。

集計単位は「日次」「週次」「締め期間」「月次」が存在します。

- 日次集計項目

1日単位で集計した結果を保持します。

上記で例として挙げた休憩時間は、1日単位で集計した結果なので集計単位は「日次」となります。

日次集計項目は、「週次」「締め期間」「月次」の元となります。

- 週次集計項目

1週間単位で集計した結果を保持します。

1週間が何曜日から始まるかは、従業員が所属する勤務体系によって決まります。

- 締め期間集計項目

締め期間単位で集計した結果を保持します。

締め期間とは、勤怠を締める期間の単位を指します。

給与システムに連携する給与計算基礎データは、締め期間集計項目を出力して生成する想定になっています。



コラム

締め期間は、標準版では1ヶ月のみ対応しています。

- 月次集計項目

1ヶ月単位で集計した結果を保持します。

次項以降、集計項目を新たに追加する場合の手順をサンプルプログラムをもとに解説していきます。

要件

以下の要件があります。

- 遅刻時間と早退時間を合算した、遅刻早退時間の集計項目を新たに作成したい
- 合算値は日次単位、締め期間単位でそれぞれ保持したい
- 締め期間単位で合算した遅刻早退時間は、30分を超えた場合のみ集計し、30分未満の場合は0分として扱いたい

これに対応した集計項目を、作成していきます。

定義

集計単位を定める

集計単位を定めます。
集計項目の単位には以下があります。

- 日次集計項目
- 週次集計項目
- 締め期間集計項目
- 月次集計項目

今回は「日次集計項目」と「締め期間集計項目」を作成します。

集計キーを定める

集計キーを定めます。
今回は日次集計項目の集計キーを「DA_late_early_time_M」、締め期間集計項目を「PA_late_early_time_M」とします。

i コラム

集計キーの採番ルールは以下のようになっています。
[集計キー接頭辞(集計単位)] + [集計キー接頭辞(由来)] + “_” + [“集計キー語幹” + “_” + [集計キー接尾辞(メジャー)]]

集計キー接頭辞(集計単位)

- D: 日次集計
- W: 週次集計
- P: 締め期間集計
- M: 月次集計

集計キー接頭辞(由来)

- A: 集計API内で固定文字列
- S: 勤務状況マスタ由来
- T: 集計タグマスタ由来
- D: 休憩・控除・時間年休大分類マスタ由来
- M: 休憩・控除・時間年休中分類マスタ由来
- L: 休憩・控除・時間年休マスタ由来

集計キー語幹

- 集計API固有文字列
- 勤務状況コード
- 集計タグコード
- 休憩・控除・時間年休大分類コード
- 休憩・控除・時間年休中分類コード
- 休憩・控除・時間年休コード

集計キー接尾辞(メジャー)

- M: 時間(分)
- D: 日数
- C: 回数

i コラム

集計の定義を行わなくても、マスタにデータが登録されているだけで勤務情報が入力されると自動的に集計されるものがあります。

- 勤務状況マスタ
- 集計タグマスタ(メンテナンス画面なし)
- 休憩・控除・時間年休大分類マスタ(メンテナンス画面なし)
- 休憩・控除・時間年休中分類マスタ(メンテナンス画面なし)
- 休憩・控除・時間年休マスタ

集計定義を登録する

集計定義の登録を行います。

- 集計定義マスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	勤務体系カテゴリコード	ws_category_cd	varchar2	20	Y	PK	
2	会社コード	company_cd	varchar2	100	Y	PK	※未使用、拡張用
3	勤務体系コード	ws_agreement_cd	varchar2	100	Y	PK	※未使用、拡張用
4	集計キーID	sum_key_id	varchar2	100	Y	PK	
5	期間コード	term_cd	varchar2	50	Y	PK	
6	開始日	start_date	timestamp		Y		

No.	論理名	物理名	型	長さ	必須	PK	備考
7	終了日	end_date	timestamp		Y		
8	集計処理	sum_api_id	varchar2	100	Y		
9	処理順序	execute_order	number	15,0			
10	集計区分	sum_div	varchar2	100	Y		
11	パラメータID	parameter_data_id	varchar2	100			

日次集計処理と締め期間集計処理の定義を登録します。



コラム

DB上は「勤務体系カテゴリ」「会社」「勤務体系」ごとに定義できることを想定した構造になっていますが、intra-mart Accel Kaiden! 勤務管理モジュール 標準版は「勤務体系カテゴリ」のみを使用しています。「会社」「勤務体系」については、拡張用の項目になっています。



コラム

「集計処理+"Calc"」がコンポーネントIDとなります。

■ 集計定義パラメータマスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	パラメタID	parameter_data_id	varchar2	100	Y	PK	
2	パラメタSEQ	parameter_seq	number	15,0	Y	PK	
3	枝番号	parameter_seq	number	18,0	Y	PK	
4	パラメタ値	parameter_value	varchar2	4000			

集計処理に渡すパラメータの定義を登録します。

今回の要件の一つである「30分」をここで渡すこととします。

```
-- 集計定義マスタ
```

```
insert into k31m_ws_use_sum_api values ('normal', '#default#', 'default', 'daily_late_early_time_sum_keys', 'default', '1900-01-01 00:00:00', '3000-01-01 00:00:00', 'dailyLateEarlyTime', 1)
insert into k31m_ws_use_sum_api values ('normal', '#default#', 'default', 'period_late_early_time_sum_keys', 'default', '1900-01-01 00:00:00', '3000-01-01 00:00:00', 'periodLateEarlyTime', 1)
```

```
-- 集計定義パラメータマスタ
```

```
insert into k31m_ws_sum_api_prm_val values ('periodLateEarlyTimeParam', 1, 1, 30, null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00')
```



コラム

固定労働時間制の定義のみを追加する例となっています。
必要に応じて他の勤務体系カテゴリの定義を登録します。

実装

集計処理を作成します。

日次集計クラス作成

締め期間集計を行う集計クラス名をDailyLateEarlyTimeCalcとします。

標準で集計される日次遅刻時間と日次早退時間を足して、日次遅刻早退時間の値として保持します。

```
package jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.daily;

import static jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.conf.KIsKey.D;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto.DailySumDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.SumCalcIF;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.model.dto.GenericSumCalcDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.model.dto.SumCalcDefDto;
import jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.daily.DailyLateEarlyTimeCalc.DailyLateEarlyTimeCalcDto;

/**
 * 遅刻早退合計時間算出クラス。
 */
public class DailyLateEarlyTimeCalc implements SumCalcIF<DailySumDto, DailyLateEarlyTimeCalcDto> {

    /**
     * 集計するのに必要な値をDTOクラスで宣言します。
     */
    public static class DailyLateEarlyTimeCalcDto extends GenericSumCalcDto {

        /** 遅刻時間(分)。 */
        public BigDecimal lateMin;
        /** 早退時間(分)。 */
        public BigDecimal earlyMin;
    }

    /**
     * 集計処理基盤が、集計定義を元に対象の集計クラスのcalcForSumメソッドを呼び出します。
     * calcForSumメソッドでは、引数から集計に必要な値を取り出してDtoにセットし、
     * 実処理を行うcalcメソッドを呼び出します。
     * 結果を集計基盤に返却します。
     */
    @Override
    public Map<String, BigDecimal> calcForSum(SumDto sumDto, Map<String, BigDecimal> dailySumData, SumCalcDefDto def,
        DailySumDto dailySumDto) {

        DailyLateEarlyTimeCalcDto calcDto = new DailyLateEarlyTimeCalcDto();
        // 遅刻時間を算出済み日次集計データより取得
        calcDto.lateMin = nullToZero(dailySumData.get(D.M_1001_M));
        // 早退時間を算出済み日次集計データより取得
        calcDto.earlyMin = nullToZero(dailySumData.get(D.M_1002_M));

        return calc(calcDto);
    }

    /**
     * 集計処理をcalcメソッドに実装します。
     * 集計結果を集計キーに紐づけてセットし、返却します。
     */
    @Override
    public Map<String, BigDecimal> calc(DailyLateEarlyTimeCalcDto calcDto, String... sumKeys) {

        Map<String, BigDecimal> resultMap = new HashMap<String, BigDecimal>();

        // 算出結果をセット
        resultMap.put("DA_late_early_time_M", calcDto.lateMin.add(calcDto.earlyMin));

        return resultMap;
    }
}
```

締め期間集計クラス作成

締め期間集計を行う集計クラス名をPeriodLateEarlyTimeCalcとします。

標準で集計される締め期間遅刻時間と締め期間早退時間を足して、締め期間遅刻早退時間の値として保持します。

しきい値となる「30分」は、集計定義のパラメータとして登録し、当集計クラスはそこから受け取ることにします。



コラム

集計定義パラメータについては後述します。

```

package jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.multiperiod.period;

import static jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import static jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil.toBigDecimal;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.product_labormgr.foundation.manager.sum.SummaryManager;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto.MultiPeriodSumDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.SumCalcIF;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.model.dto.GenericSumCalcDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.calc.model.dto.SumCalcDefDto;
import jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.multiperiod.period.PeriodLateEarlyTimeCalc.PeriodLateEarlyTimeCalcDto;

/**
 * 締め期間遅刻早退時間算出クラス。
 */
public class PeriodLateEarlyTimeCalc implements SumCalcIF<MultiPeriodSumDto, PeriodLateEarlyTimeCalcDto> {

    /**
     * 集計定義パラメータを取得するためのパラメータシーケンスを宣言しています。
     */
    private static final long PARAM_SEQ_1 = 1;

    public static class PeriodLateEarlyTimeCalcDto extends GenericSumCalcDto {

        /** 遅刻早退時間(分)。 */
        public BigDecimal lateEarlyMin;
        /** 下限時間。 */
        public BigDecimal lowerLimitMin;
    }

    /** SummaryManager。 */
    @Resource
    protected SummaryManager summaryManager;

    @Override
    public Map<String, BigDecimal> calcForSum(SumDto sumDto, Map<String, BigDecimal> sumData, SumCalcDefDto def,
        MultiPeriodSumDto multiPeriodSumDto) {

        PeriodLateEarlyTimeCalcDto calcDto = new PeriodLateEarlyTimeCalcDto();

        /**
         * 締め期間中の日次遅刻早退時間の合計を取得します。
         */
        // 日次遅刻早退時間の合計を取得
        calcDto.lateEarlyMin = nullToZero(summaryManager.getWrecSumDailyByPeriod(sumDto.splInfo.getUserCd(),
            sumDto.splInfo.getPeriodStartDate(), sumDto.splInfo.getPeriodStartDate(), "DA_late_early_time_M").get(
                "DA_late_early_time_M"));

        /**
         * 集計定義パラメータとして定義されている30分を取得します。
         */
        // 集計定義パラメータより、下限時間を取得
        calcDto.lowerLimitMin = toBigDecimal(def.param.getSingleParam(PARAM_SEQ_1));

        return calc(calcDto);
    }

    @Override
    public Map<String, BigDecimal> calc(PeriodLateEarlyTimeCalcDto calcDto, String... sumKeys) {

        Map<String, BigDecimal> resultMap = new HashMap<String, BigDecimal>();

        // 算出結果をセット
        if (calcDto.lateEarlyMin.compareTo(calcDto.lowerLimitMin) < 0) {
            // 下限時間より少ない
            resultMap.put("PA_late_early_time_M", BigDecimal.ZERO);
        } else {
            // 下限時間以上
            resultMap.put("PA_late_early_time_M", calcDto.lateEarlyMin);
        }

        return resultMap;
    }
}

```

デフォルト締め期間集計のスキップ処理を追加

標準の処理で、日次集計値を締め期間内で合計した値を締め期間集計値として自動的に登録しています。

(これは締め期間でのみです。週次、月次では行っていません。)

今回のように単純な日次集計値の合計でなく、独自に締め期間集計クラスを作成する場合、自動登録処理をスキップする必要があります。

MultiPeriodSumクラスを以下のように拡張します。

```
package jp.co.slcs.kaiden2.tutorial2.foundation.sum.process;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.MultiPeriodSum;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto;

public class MultiPeriodSumExt extends MultiPeriodSum {

    /*
     * 締め期間集計処理の基盤クラスのgetExcludeTargetSumKey(SumDto)メソッドで、
     * 日次集計データの単純な合算をしない項目を定義しています。
     */
    @Override
    protected String[] getExcludeTargetSumKey(SumDto _sumDto) {
        List<String> list = new ArrayList<String> (Arrays.asList(super.getExcludeTargetSumKey(_sumDto)));
        /*
         * 日次集計項目の"DA_late_early_time_M"を追加します。
         * 締め期間集計値の自動計算が行われなくなります。
         */
        list.add("DA_late_early_time_M");
        return list.toArray(new String[list.size()]);
    }
}
```



コラム

日次集計値の単純な合計を締め期間集計値としてよい場合は、締め期間集計クラスを独自に作成する必要はありません。

diconファイル作成

diconファイルを作成します。

ファイル名をkaiden_tutorial2.diconとします。

"lateEarlyPeriodTimeChecker"は、次項のチェック処理のものになりますので、ここでは読み飛ばしてください。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components>
  <component name="dailyLateEarlyTimeCalc" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.daily.DailyLateEarlyTimeCalc" instance="prototype"></component>
  <component name="periodLateEarlyTimeCalc" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.multiperiod.period.PeriodLateEarlyTimeCalc" instance="prototype"></component>
  <component name="multiPeriodSum" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.process.MultiPeriodSumExt" instance="prototype"></component>
  <component name="lateEarlyPeriodTimeChecker" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.check.multiperiod.period.LateEarlyPeriodTimeChecker" instance="prototype"></component>
</components>
```



コラム

kaiden_tutorial2.diconをインクルードしてください。



コラム

コンポーネントIDを「集計定義テーブル.集計処理+"Calc"」とすることで、集計基盤に集計処理クラスとして認識されます。



コラム

kaiden_product_labormgr.diconの"multiPeriodSum"の定義を無効にしてください。

確認

勤務表で入力を行い、想定した結果が日次集計テーブル(k31t_wrec_sum_daily)、締め期間集計テーブル(k31t_wrec_sum_period)に登録されていることを確認してください。

**コラム**

intra-mart Accel Kaiden! 勤務管理モジュール 標準版は、時間数の値を分単位で登録しています。

集計チェックを新たに追加する

集計チェックを新たに追加する手順を解説します。

はじめに - 集計チェックとは

intra-mart Accel Kaiden! 勤務管理モジュール では 集計項目の値に対して何らかのチェックを行い、チェックに引っかかった場合にアラートを上げる処理 のことを集計チェックと呼びます。
単純な入力値に対する必須チェック等は業務ロジックで行うべきもので、集計チェックには含みません。

次項以降、集計項目を新たに追加する場合の手順をサンプルプログラムをもとに解説していきます。

要件

以下の要件があるとして。

- 前項で作成した「遅刻早退時間」に対して、チェックを行いたい。
- 警告レベル1～4ごとにしきい値を設定したい。

これに対応した集計チェックを、作成していきます。

 コラム

標準版では警告レベル1～4までをサポートしており、それぞれの位置づけとして以下を想定しています。

- レベル1:軽度警告
- レベル2:重度警告
- レベル3:会社規定違反
- レベル4:法令違反

ただし、上記を強制するものではなく、自由にレベルの位置づけをして頂いて構いません。

定義

集計チェック定義の登録を行います。
対象テーブルと登録すべきデータを解説します。

■ ガジェットマスタ

No.	論理名	物理名	型	長さ	必須	PK
1	ガジェットクラス	gadget_class	varchar2	100	Y	PK
2	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK
3	JSPパス	jsp_path	varchar2	1000		
4	JAVAクラスパス	java_path	varchar2	1000		
5	ソートキー	sort_key	number	15,0	Y	
6	サーフェスキー	surface_key	varchar2	100	Y	

集計チェック処理のガジェットクラス・ガジェットバリエーションに対して、
集計チェック処理設定ガジェットJSPパスと集計処理実行クラスを定義します。
ここでは、以下のように設定することとします。

ガジェットクラス	klsLateEarlyTime
ガジェットバリエーション	v01
JSPパス	/WEB-INF/view/kaiden2/tutorial2/gadget/klsLateEarlyPeriodTime/klsLateEarlyPeri
Javaパス	lateEarlyPeriodTime



コラム

「Javaパス+「Checker」」がコンポーネントIDとなります。

■ ガジェットタグマスタ

No.	論理名	物理名	型	長さ	必須	PK
1	ガジェットクラス	gadget_class	varchar2	100	Y	PK
2	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK
3	タグ	gadget_tag	varchar2	100	Y	PK

ガジェットのタグとして「sumCheck」(チェック処理であることを示す)と「period」(締め期間集計値に対するチェックであることを示す)を設定します。



コラム

集計チェック処理に対するタグは以下を付与します。

集計期間	タグ
日次	sumCheck,daily
週次	sumCheck,weekly
締め期間	sumCheck,period
月次	sumCheck,monthly
その他	sumCheck,other

■ ガジェットサーフェスマスタ

No.	論理名	物理名	型	長さ	必須	PK
1	ガジェットクラス	gadget_class	varchar2	100	Y	PK
2	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK
3	サーフェスキー	surface_key	varchar2	100	Y	PK
4	ロケールID	locale_id	varchar2	50	Y	PK
5	サーフェス	surface	varchar2	4000	Y	

集計チェック処理設定ガジェットのタイトルとして表示されます。
「遅刻早退時間チェック」を設定します。

■ 集計チェックAPIパラメータマスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	ガジェットクラス	gadget_class	varchar2	100	Y	PK	
2	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK	
3	パラメータキー	parameter_key	varchar2	100	Y	PK	
4	ロケールID	locale_id	varchar2	50	Y	PK	
5	パラメータ種別	parameter_def_pattern_cd	varchar2	100	Y		
6	パラメータ名称	parameter_name	varchar2	1000	Y		※未使用、拡張用
7	備考	note	varchar2	4000			

集計チェック処理が受け取るパラメータの顔ぶれを設定します。

ここでは、「criteriaValue」というパラメータキーに対して、レベル1～4までのチェックしきい値を指定します。



コラム

パラメータ種別には、パラメータの指定方法を指定しています。

1	単一値指定
2	重大度別指定
3	複数値指定
4	その他

■ 集計チェックAPIパラメータ内訳マスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	ガジェットクラス	gadget_class	varchar2	100	Y	PK	
2	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK	
3	パラメータキー	parameter_key	varchar2	100	Y	PK	
4	パラメータSEQ	parameter_seq	number	15,0	Y	PK	
5	パラメータデータタイプ	parameter_data_type	varchar2	100	Y		※未使用、拡張用
6	サーフェスキー	surface_key	varchar2	100			※未使用、拡張用
7	重大度	severe_level	varchar2	100			
8	増減区分	increase_type	varchar2	100			※未使用、拡張用
9	変更可否	allow_upd_flag	varchar2	1	Y		※未使用、拡張用
10	削除可否	allow_del_flag	varchar2	1	Y		※未使用、拡張用

集計チェックAPIパラメータの詳細情報を登録します。



コラム

パラメータデータタイプには、パラメータとして渡される型を指定します。

1	数値
2	文字列
3	真偽値
4	日付
5	時間
6	不等号



コラム

重大度には、集計チェックAPIパラメータマスタパラメータ種別が「重大度指定」の場合のレベルを指定します。

標準では1～4のみをサポートしています。

重大度指定でないパラメータの場合はnullを指定してください。

■ 集計チェックマスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	勤務体系カテゴリコード	ws_category_cd	varchar2	20	Y	PK	

No.	論理名	物理名	型	長さ	必須	PK	備考
2	会社コード	company_cd	varchar2	100	Y	PK	
3	勤務体系コード	ws_agreement_cd	varchar2	100	Y	PK	
4	ガジェットクラス	gadget_class	varchar2	100	Y	PK	
5	ガジェットバリエーション	gadget_variation	varchar2	100	Y	PK	
6	ガジェットインスタンス	gadget_instance	number	18,0	Y	PK	
7	期間コード	term_cd	varchar2	50	Y	PK	
8	パラメータID	parameter_data_id	varchar2	100	Y		
9	処理順序	execute_order	number	15,0			
10	チェック要否	check_required_flag	varchar2	1	Y		

勤務体系ごとの集計チェック処理の定義を設定します。

集計チェック処理作成時のデータは「勤務体系カテゴリ:normal」「会社:#default#」「勤務体系:default」として、勤務体系カテゴリごとの共通設定として登録します。

ここでの登録値が、勤務体系マスタで最初に集計チェック定義のメンテナンスを行う画面のパラメータ初期値となります。



コラム

チェック要否には「0:チェックを行わない」「1:チェックを行う」を設定します。
運用中に集計チェック処理を追加する場合は「0」を登録してください。



コラム

処理順序には処理を行う順番を設定します。
nullを登録すると、処理順序は保証されません。

■ 集計チェックパラメータマスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	パラメータID	parameter_data_id	varchar2	100	Y	PK	
2	パラメータキー	parameter_key	varchar2	100	Y	PK	
3	パラメータSEQ	parameter_seq	number	15,0	Y	PK	
4	パラメータ値	parameter_value	varchar2	4000			

集計チェック処理のパラメータに渡すパラメータを登録します。

ここでの登録値が、勤務体系マスタで最初に集計チェック定義のメンテナンスを行う画面のパラメータ初期値となります。

INSERT文は以下になります。

[illegible]

注意

集計チェックマスタ.ガジェットバリエーションは、DB定義上100byteとなっていますが、50byteを超えて登録しないでください。
SQL Serverの主キー最大長の900byteを超える可能性があります。

集計チェック処理

集計チェッククラス作成

クラス名をLateEarlyPeriodTimeCheckerとします。

```

package jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.check.multiperiod.period;

import static jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import static jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil.toMsgHHMM;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.product_labormgr.foundation.conf.SevereLevel;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.process.model.dto.SumDto.MultiPeriodSumDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.SummaryUtil;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.AbstractSummaryChecker;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.model.dto.GenericBorderCheckDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckDefDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckDto;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckResult;
import jp.co.slcs.kaiden2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckerGenericDto;

public class LateEarlyPeriodTimeChecker extends AbstractSummaryChecker<MultiPeriodSumDto, GenericBorderCheckDto> {

    /* *****
    * パラメータキーを宣言しています。
    * ***** */
    protected static final String CRITERIA_VALUE = "criteriaValue";

    @Override
    public List<SumCheckResult> checkForSum(Map<String, BigDecimal> sumData, SumCheckDefDto sumCheckDef,
        MultiPeriodSumDto multiPeriodSumDto) {

        GenericBorderCheckDto gbcDto = new GenericBorderCheckDto(sumData, sumCheckDef);

        /* *****
        * チェック対象期間の開始日と終了日を取得しています。
        * エラーメッセージに出力します。
        * ***** */
        gbcDto.checkTargetStartDate = BeanUtil.toCast(MultiPeriodSumDto.class, multiPeriodSumDto).getAssignStartDate();
        gbcDto.checkTargetEndDate = BeanUtil.toCast(MultiPeriodSumDto.class, multiPeriodSumDto).getAssignStartDate();

        return check(gbcDto);
    }

    @Override
    public List<SumCheckResult> check(GenericBorderCheckDto btcDto) {

        SumCheckDefDto scDef = btcDto.sumCheckDef;

        List<SumCheckResult> resultList = new ArrayList<SumCheckResult>();

        /* *****
        * 締め期間集計データから、遅刻早退時間の値を取得します。
        * ***** */
        BigDecimal targetValue = nullToZero(btcDto.sumData.get("PA_late_early_time_M"));

        SumCheckResult scResult = null;

        /* *****
        * レベルごとのしきい値と比較します。
        * レベル1から順に比較し、結果を上書きしています。
        * ***** */
        for (SevereLevel level : SevereLevel.values()) {

            SumCheckDto def = scDef.param.getBorderDefs(level).get(CRITERIA_VALUE);

            if (null == def || StringUtil.isEmpty(def.paramValue)) {
                continue;
            }

            BigDecimal criteriaValue = toAnyType(BigDecimal.class, def.paramValue);

            if (null == criteriaValue) {
                continue;
            }

            // 基準値 [%演算子%] 対象値 の場合
            if (targetValue.compareTo(criteriaValue) > 0) {
                /* *****
                * チェック結果メッセージIDの引数を設定しています。
                * ***** */
                scResult = makeSumCheckResult(scDef);
            }
        }
    }

```

```

        scResult = makeSummaryCheckResult(scResult);
        scResult.judgeStatus = true;
        scResult.severeLevel = level.getCode();
        scResult.addMsgArg(btcDto.checkTargetStartDate);
        scResult.addMsgArg(btcDto.checkTargetEndDate);
        scResult.addMsgArg("TTRL.summaryItem.V.PA_late_early_time_M");
        scResult.addMsgArg(toMsgHHMM(targetValue));
        scResult.addMsgArg(toMsgHHMM(criteriaValue));
    }
}

if (null != scResult) {
    resultList.add(scResult);
}

return resultList;
}

@Override
protected String getMessageId() {
    /* *****
     * チェック結果メッセージIDを返します。
     * ***** */
    return "I.TTRL.summaryCheck.0001";
}

@Override
protected Object[] getMsgArgs(SumCheckerGenericDto scgDto, Object... args) {
    return SummaryUtil.makeArgs(scgDto, args);
}
}

```

メッセージ定義ファイル作成

ファイル名をtutorial2_messages.xmlとします。

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment/>
<entry key="I.TTRL.summaryCheck.0001">{[0]}～{[1]}の期間で{2}{[3]}が{4}を超過しています</entry>
</properties>

```

サーフェス定義ファイル作成

ファイル名をtutorial2_messages.xmlとします。

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="TTRL.summaryItem.V.PA_late_early_time_M">遅刻早退時間</entry>
</properties>

```

diconファイル作成

diconファイルを作成します。

ファイル名をkaiden_tutorial2.diconとします。

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components>
<component name="dailyLateEarlyTimeCalc" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.daily.DailyLateEarlyTimeCalc" instance="prototype"></component>
<component name="periodLateEarlyTimeCalc" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.calc.multiperiod.period.PeriodLateEarlyTimeCalc" instance="prototype"></component>
<component name="multiPeriodSum" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.process.MultiPeriodSumExt" instance="prototype"></component>
<component name="lateEarlyPeriodTimeChecker" class="jp.co.slcs.kaiden2.tutorial2.foundation.sum.util.check.multiperiod.period.LateEarlyPeriodTimeChecker" instance="prototype"></component>
</components>

```



コラム

コンポーネントIDを「ガジェットマスタ.Java/バス+「Checker」」とすることで、集計チェック基盤に集計チェッククラスとして認識されます。

集計チェック定義登録ガジェット

Actionクラス作成

アクションクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial2.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.product_labormgr.feature.action.kaiden.gadget.AbstractKlsSettingActionBase;
import jp.co.slcs.kaiden2.product_labormgr.feature.form.kaiden.gadget.AbstractKlsSettingForm;
import jp.co.slcs.kaiden2.product_labormgr.feature.logic.kaiden.gadget.AbstractKlsSettingLogic;
import jp.co.slcs.kaiden2.tutorial2.feature.form.kaiden.gadget.KlsLateEarlyPeriodTimeV01Form;
import jp.co.slcs.kaiden2.tutorial2.feature.logic.kaiden.gadget.KlsLateEarlyPeriodTimeV01Logic;
import org.seasar.struts.annotation.ActionForm;

public class KlsLateEarlyPeriodTimeV01Action extends AbstractKlsSettingActionBase {

    @ActionForm
    @Resource
    public KlsLateEarlyPeriodTimeV01Form klsLateEarlyPeriodTimeV01Form;

    @Resource
    public KlsLateEarlyPeriodTimeV01Logic klsLateEarlyPeriodTimeV01Logic;

    @Override
    protected AbstractKlsSettingForm getForm() {
        return klsLateEarlyPeriodTimeV01Form;
    }

    @Override
    protected AbstractKlsSettingLogic getLogic() {
        return klsLateEarlyPeriodTimeV01Logic;
    }
}
```

Formクラス作成

アクションクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial2.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.product_labormgr.feature.form.kaiden.gadget.AbstractKlsSettingForm;

public class KlsLateEarlyPeriodTimeV01Form extends AbstractKlsSettingForm {
}
```

Logicクラス作成

アクションクラスを作成します。

```
package jp.co.slcs.kaiden2.tutorial2.feature.logic.kaiden.gadget;

import jp.co.slcs.kaiden2.product_labormgr.feature.logic.kaiden.gadget.AbstractKlsSettingLogic;

public class KlsLateEarlyPeriodTimeV01Logic extends AbstractKlsSettingLogic {
}
```

JSP作成

JSPファイル名をklsLateEarlyPeriodTimeV01.jspとします。

「/WEB-INF/view/kaiden2/tutorial2/gadget/klsLateEarlyPeriodTime」に配置します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil"%>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_labormgr.foundation.conf.ProductLabormgrProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.ViewMode" %>

<%-- 遅刻早退時間チェックガジェット --%>
<c:set var="gadgetClass">${param.gadgetClass}</c:set>
<c:set var="gadgetVariation">${param.gadgetVariation}</c:set>
<c:set var="gadgetInstance">${param.gadgetInstance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="gadgetTitle">${param.gadgetName}</c:set>
<c:set var="accessUrl">kaiden/gadget/klsLateEarlyPeriodTimeV01</c:set>
<c:set var="sumCheckBlockId">sumCheckList</c:set>
<c:set var="sumCheckDialogId">sumCheckContent</c:set>

<%-- 設定等 --%>
<c:set var="klsLateEarlyPeriodTimeV01Form">${param.klsLateEarlyPeriodTimeV01Form}</c:set>
```

```

<c:set var="wsCd">${param.wsCd}</c:set>
<c:set var="companyCd">${param.companyCd}</c:set>
<c:set var="wsAgreementCd">${param.wsAgreementCd}</c:set>
<c:set var="openApplyDate">${param.openApplyDate}</c:set>
<c:set var="termCd">${param.termCd}</c:set>
<c:set var="inputMode">${param.inputMode}</c:set>
<c:set var="sumDiv">${param.sumDiv}</c:set>
<c:set var="checkRequiredFlag">${param.checkRequiredFlag}</c:set>

<c:set var="debug">${param.debug}</c:set>

<%-- 画面固有バリデーションID --%>
<c:set var="validTimeMax4Caution1">k_${gadgetId}TimeMax4Caution1</c:set>
<c:set var="validTimeMax4Caution2">k_${gadgetId}TimeMax4Caution2</c:set>

<%-- 画面表示モード --%>
<c:set var="input"><%= ViewMode.INPUT.getCode() %></c:set>
<c:set var="view"><%= ViewMode.VIEW.getCode() %></c:set>

<%-- サーフエス --%>
<c:set var="surfaceCheckRequiredFlag"><%= ProductLabormgrProp.S.WsAgreement.SUM_CHECK_REQUIRED %></c:set>

<c:set var="surfaceCriteriaValue"><%= ProductLabormgrProp.S.WsAgreement.OVER_TIME_LIMIT %></c:set>

<c:set var="surfaceCaution1"><%= ProductLabormgrProp.S.CAUTION_1 %></c:set>
<c:set var="surfaceLevel1"><%= ProductLabormgrProp.S.LEVEL_1 %></c:set>
<c:set var="surfaceCaution2"><%= ProductLabormgrProp.S.CAUTION_2 %></c:set>
<c:set var="surfaceLevel2"><%= ProductLabormgrProp.S.LEVEL_2 %></c:set>
<c:set var="surfaceCaution3"><%= ProductLabormgrProp.S.CAUTION_3 %></c:set>
<c:set var="surfaceLevel3"><%= ProductLabormgrProp.S.LEVEL_3 %></c:set>
<c:set var="surfaceCaution4"><%= ProductLabormgrProp.S.CAUTION_4 %></c:set>
<c:set var="surfaceLevel4"><%= ProductLabormgrProp.S.LEVEL_4 %></c:set>

<%-- 集計チェック設定 --%>
<%-- 保存 --%>
<c:set var="surfaceRefreshSumCheck"><%= BaseProp.S.SAVE %></c:set>
<%-- チェックを有効にします。(文字列化済み) --%>
<c:set var="surfaceCheckRequiredFlagOn_text"><%= PropertyJspUtil.getSurface(ProductLabormgrProp.S.WsAgreement.CHECK_REQUIRED_FLAG_ON) %></c:set>
<%-- チェックを無効にします。(文字列化済み) --%>
<c:set var="surfaceCheckRequiredFlagOff_text"><%= PropertyJspUtil.getSurface(ProductLabormgrProp.S.WsAgreement.CHECK_REQUIRED_FLAG_OFF) %></c:set>
<%-- 集計チェックの内容を表示します。(文字列化済み) --%>
<c:set var="surfaceSumCheckContentInfo_text"><%= PropertyJspUtil.getSurface(ProductLabormgrProp.S.WsAgreement.SUM_CHECK_CONTENT_INFO) %></c:set>
<%-- 有効(文字列化済み) --%>
<c:set var="surfaceEnabled_text"><%= PropertyJspUtil.getSurface(BaseProp.S.ENABLED) %></c:set>
<%-- 無効(文字列化済み) --%>
<c:set var="surfaceDisabled_text"><%= PropertyJspUtil.getSurface(BaseProp.S.DISABLED) %></c:set>

<%-- ダイアログボタン --%>
<%-- 閉じる(文字列化済み) --%>
<c:set var="surfaceReturnlist_text"><%=PropertyJspUtil.getSurface(BaseProp.S.CLOSE)%></c:set>

<%-- メッセージ --%>
<%-- 保存します。 --%>
<c:set var="messageSaveConfirm"><%=PropertyJspUtil.getMessage(BaseProp.M.W0012)%></c:set>
<%-- "{0}"以下で入力してください。 --%>
<c:set var="validMsg_timeMax"><%= PropertyJspUtil.getMessage(BaseProp.M.W0226) %></c:set>

<%-- バリデーション --%>
<c:set var="validCriteriaValue_1_settings">[k_time:['], k_${gadgetId}TimeMax4Caution1:[criteriaValue|2', 'criteriaValue|3', 'criteriaValue|4'], k_timeMax2:[1000:00]]</c:set>
<c:set var="validCriteriaValue_2_settings">[k_time:['], k_${gadgetId}TimeMax4Caution2:[criteriaValue|3', 'criteriaValue|4'], k_timeMax2:[1000:00]]</c:set>
<c:set var="validCriteriaValue_3_settings">[k_time:['], k_timeMax:[criteriaValue|4'], k_timeMax2:[1000:00]]</c:set>
<c:set var="validCriteriaValue_4_settings">[k_time:['], k_timeMax2:[1000:00]]</c:set>

<k:gadgetContainer gadgetId="${gadgetId}"
    gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${gadgetTitle}"
    hideGadgetTitle="true">

<form id="gadget_form" onsubmit="return false;">
<k:blockContainer id="${sumCheckBlockId}" style="padding:0px 20px;">
<table class="imui-table-mixed" style="table-layout:fixed;margin-bottom:0.5em;">
<colgroup>
<col style="width:100%;"/>
</colgroup>
<tr>
<th>
<k:surface name="checkRequiredFlagOn" surfaceKey="" class="kaiden-ui-wsA_sumCheck kaiden-ui-icon-enable16 mr-5 hidden kaiden-ui-wsA_sumCheck-label" title="${surfaceCh
<k:surface name="checkRequiredFlagOff" surfaceKey="" class="kaiden-ui-wsA_sumCheck kaiden-ui-icon-disable16 mr-5 hidden kaiden-ui-wsA_sumCheck-label" title="${surfaceCh
<k:hidden name="checkRequiredFlag" mode="${inputMode}"/>
<k:surface surfaceKey="${gadgetTitle}"/>
<k:a name="checkContentLink" class="im-ui-icon-common-16-question mr-5 float-R" mode="0" title="${surfaceSumCheckContentInfo_text}"/>
<k:hidden name="parameterDataId" mode="${inputMode}"/>
</th>
</tr>
</table>
<div class="sumCheckParams" style="display:none">

```

```

<tr class= sumCheckParams style= display:none, >
<td>
<div style="margin-left:50px">
<table style="width:100%">
<colgroup>
<col />
<col style="width:150px;"/>
</colgroup>
<tr>
<td>
<table class="imui-table" style="margin-bottom:0.5em;">
<colgroup>
<col style="width:200px"/>
<col style="" />
</colgroup>
<thead>
<tr>
<th></th>
<th><span class="kaiden-ui-icon-warn-l16 mr-5 valign-M"></span><k:surface surfaceKey="${surfaceLevel1}" class="valign-M"/></th>
<th><span class="kaiden-ui-icon-warn-l16 mr-5 valign-M"></span><k:surface surfaceKey="${surfaceLevel2}" class="valign-M"/></th>
<th><span class="kaiden-ui-icon-warn-l16 mr-5 valign-M"></span><k:surface surfaceKey="${surfaceLevel3}" class="valign-M"/></th>
<th><span class="kaiden-ui-icon-warn-l16 mr-5 valign-M"></span><k:surface surfaceKey="${surfaceLevel4}" class="valign-M"/></th>
</tr>
</thead>
<tbody>
<tr>
<th><k:surface surfaceKey="${surfaceCriteriaValue}"/></th>
<td>
<k:time name="criteriaValue|1"
mode="${inputMode}"
surfaceKey="${surfaceCaution1}"
validationSettingsKlsDaily36LimitV01="${validCriteriaValue_1_settings}"
/>
</td>
<td>
<k:time name="criteriaValue|2"
mode="${inputMode}"
surfaceKey="${surfaceCaution2}"
validationSettingsKlsDaily36LimitV01="${validCriteriaValue_2_settings}"
/>
</td>
<td>
<k:time name="criteriaValue|3"
mode="${inputMode}"
surfaceKey="${surfaceCaution3}"
validationSettingsKlsDaily36LimitV01="${validCriteriaValue_3_settings}"
/>
</td>
<td>
<k:time name="criteriaValue|4"
mode="${inputMode}"
surfaceKey="${surfaceCaution4}"
validationSettingsKlsDaily36LimitV01="${validCriteriaValue_4_settings}"
/>
</td>
</tr>
</tbody>
</table>
</td>
<td class="align-C valign-M">
<k:button mode="${inputMode}" class="imui-small-button" value="${surfaceRefreshSumCheck}" name="settings"/>
</td>
</tr>
</table>
</div>
</td>
</tr>
</table>
</k:blockContainer>
</form>
<k:dialog title="${gadgetTitle}" id="${sumCheckDialogId}">
<div class="imui-box-supplementation">
<span class="im-ui-icon-common-24-question mr-5" ></span>
<k:surface name="checkContent" surfaceKey="" style="display:inline-block;vertical-align:top;"/>
</div>
</k:dialog>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {
var eventMan = KAIDEN.getGadget("${gadgetId}").eventManager;
var blockA = KAIDEN.getGadget("${gadgetId}").blockAccessors["${sumCheckBlockId}"]
, dialog = KAIDEN.getGadget("${gadgetId}").dialogs["${sumCheckDialogId}"]
, dialogA = KAIDEN.getGadget("${gadgetId}").blockAccessors["${sumCheckDialogId}"];

//ガジェット初期設定
KAIDEN.getGadget("${gadgetId}").variable.loadSumCheckSetting = function() {
var aiax = KAIDEN.Aiax()

```

```

var ajax = new XMLHttpRequest();
, criteria = {}
criteria.wsCategoryId = "${wsCategoryId}";
criteria.companyCd = "${companyCd}";
criteria.wsAgreementCd = "${wsAgreementCd}";
criteria.openApplyDate = "${openApplyDate}";
criteria.termCd = "${termCd}";
criteria.gadgetClass = "${gadgetClass}";
criteria.gadgetVariation = "${gadgetVariation}";
criteria.gadgetInstance = "${gadgetInstance}";

ajax.postSyncAuto("${accessUrl}/loadSumCheck", criteria, function(returnData) {

var sumCheckData = returnData.$[sumDiv][0];

//チェック利用の設定
blockA("checkRequiredFlag").setValue(sumCheckData.checkRequiredFlag);
if ("1" === sumCheckData.checkRequiredFlag) {
    blockA().elem().find("tr.sumCheckParams").show("fast");
    blockA("checkRequiredFlagOn").removeClass("hidden");
    blockA("checkRequiredFlagOff").addClass("hidden");
} else {
    blockA().elem().find("tr.sumCheckParams").hide("fast");
    blockA("checkRequiredFlagOn").addClass("hidden");
    blockA("checkRequiredFlagOff").removeClass("hidden");
}
blockA("parameterDataId").setValue(sumCheckData.paramDataId);
dialogA("checkContent").setValue(KAIDEN.Util.escapeHTML(sumCheckData.checkContent));

//重要度ごとのパラメータ値
var mildParam = sumCheckData.param.borders.MILD
, severeParam = sumCheckData.param.borders.SEVERE
, wsParam = sumCheckData.param.borders.WS
, violationParam = sumCheckData.param.borders.VIOLATION

//Lv.1
for (var key in mildParam) {
    switch(blockA(key + "|1").getType()) {
        case "time":
            blockA(key + "|1").setMinutes(mildParam[key].paramValue);
            break;
        case "timelabel":
            blockA(key + "|1").setMinutes(mildParam[key].paramValue);
            break;
        default:
            blockA(key + "|1").setValueIgnore(mildParam[key].paramValue);
            break;
    }
}
//Lv.2
for (var key in severeParam) {
    switch(blockA(key + "|2").getType()) {
        case "time":
            blockA(key + "|2").setMinutes(severeParam[key].paramValue);
            break;
        case "timelabel":
            blockA(key + "|2").setMinutes(severeParam[key].paramValue);
            break;
        default:
            blockA(key + "|2").setValueIgnore(severeParam[key].paramValue);
            break;
    }
}
//Lv.3
for (var key in wsParam) {
    switch(blockA(key + "|3").getType()) {
        case "time":
            blockA(key + "|3").setMinutes(wsParam[key].paramValue);
            break;
        case "timelabel":
            blockA(key + "|3").setMinutes(wsParam[key].paramValue);
            break;
        default:
            blockA(key + "|3").setValueIgnore(wsParam[key].paramValue);
            break;
    }
}
//Lv.4
for (var key in violationParam) {
    switch(blockA(key + "|4").getType()) {
        case "time":
            blockA(key + "|4").setMinutes(violationParam[key].paramValue);
            break;
        case "timelabel":
            blockA(key + "|4").setMinutes(violationParam[key].paramValue);
            break;
        default:

```



```

        blockA(key + "|4").setValueIgnore(violationParam[key].paramValue);
        break;
    }
}
});
};

// .....
// 内部イベント
// .....

if ("${input}" === "${inputMode}") {
    //チェックボックスON/OFFイベント
    blockA().elem(true).find(".kaiden-ui-wsA_sumCheck").on("click", function(e) {
        if ("1" === blockA("checkRequiredFlag").getValue()) {
            blockA("checkRequiredFlag").setValue("0");
        } else {
            blockA("checkRequiredFlag").setValue("1");
        }
    });
    var ajax = KAIDEN.Ajax()
        , data = {}
        , wsUseSumCheck = {}
        , updateData = {};
    wsUseSumCheck.wsCategoryCd = "${wsCategoryCd}";
    wsUseSumCheck.companyCd = "${companyCd}";
    wsUseSumCheck.wsAgreementCd = "${wsAgreementCd}";
    wsUseSumCheck.searchCriteriaDate = "${openApplyDate}";
    wsUseSumCheck.gadgetClass = "${gadgetClass}";
    wsUseSumCheck.gadgetVariation = "${gadgetVariation}";
    wsUseSumCheck.gadgetInstance = "${gadgetInstance}";
    wsUseSumCheck.checkRequiredFlag = blockA("checkRequiredFlag").getValue();
    wsUseSumCheck.parameterDataId = blockA("parameterDataId").getValue();

    updateData = KAIDEN.gadgetMan.getDataSet("${gadgetId}");
    updateData.wsUseSumCheck = wsUseSumCheck

    data.updateDataJson = KAIDEN.Util.stringifyJSON(updateData);
    ajax.postSyncAuto("${accessUrl}/renewCheckFlag", data, function(returnData) {
        KAIDEN.getGadget("${gadgetId").variable.loadSumCheckSetting(); //設定内容を検索
    });

    e.stopPropagation(); //hrefの動作を抑止
});
} else {
    blockA("checkRequiredFlagOn").elem(true).attr("title", "${surfaceEnabled_text}");
    blockA("checkRequiredFlagOff").elem(true).attr("title", "${surfaceDisabled_text}");
}

//チェック内容リンク押下イベント
blockA("checkContentLink").elem(true).on("click", function(e) {
    //閉じるボタン
    dialog.addButton("${surfaceReturnlist_text}", function() {
        dialog.close();
    });
    dialog.open();
    e.stopPropagation(); //hrefの動作を抑止
});

//チェック設定リンク押下イベント
blockA("settings").elem(true).on("click", function(e) {
    kaidenErrorDisplay("hide");
    var ajax = KAIDEN.Ajax()
        , data = {}
        , wsUseSumCheck = {}
        , updateData = {};

    if (!KAIDEN.gadgetMan.preSingleValidate()) {return}; //単項目チェック前
    if ("${debug}" || KAIDEN.gadgetMan.validate("${settingsKlsdaily36limitv01", false)) { //バリデーション
        imuiConfirm("${messageSaveConfirm}", "", function() {
            wsUseSumCheck.wsCategoryCd = "${wsCategoryCd}";
            wsUseSumCheck.companyCd = "${companyCd}";
            wsUseSumCheck.wsAgreementCd = "${wsAgreementCd}";
            wsUseSumCheck.searchCriteriaDate = "${openApplyDate}";
            wsUseSumCheck.gadgetClass = "${gadgetClass}";
            wsUseSumCheck.gadgetVariation = "${gadgetVariation}";
            wsUseSumCheck.gadgetInstance = "${gadgetInstance}";
            wsUseSumCheck.checkRequiredFlag = blockA("checkRequiredFlag").getValue();

            updateData = KAIDEN.gadgetMan.getDataSet("${gadgetId}");
            updateData.wsUseSumCheck = wsUseSumCheck

            data.updateDataJson = KAIDEN.Util.stringifyJSON(updateData);
            ajax.postSyncAuto("${accessUrl}/renewParamValues", data, function(returnData) {
                KAIDEN.getGadget("${gadgetId").variable.loadSumCheckSetting(); //設定内容を検索
            });
        });
    }
});

```

```

    } else {
        kaidenErrorDisplay("show");
    }
});

KAIDEN.gadget("$gadgetId").variable.loadSumCheckSetting(); //設定内容を検索

// .....
// その他チェック
// .....
/**
 * 注意レベル1最大時刻チェック.
 * <p>値 < 注意レベル2 < 注意レベル3 < 注意レベル4 を許容</p>
 * @param value 値
 * @param element 対象エレメント
 * @param param パラメータ (配列)
 */
KAIDEN.validator.addValidateMethod("${validTimeMax4Caution1}", function(**String*/value, /**HtmlElement*/element, /**array*/param) {
    var level2 = KAIDEN.gadgetMan.getNeighboringAccessor(element, param[0]).getValue();
    var level3 = KAIDEN.gadgetMan.getNeighboringAccessor(element, param[1]).getValue();
    var level4 = KAIDEN.gadgetMan.getNeighboringAccessor(element, param[2]).getValue();

    if (level2) {
        if (level2 <= value) {
            var arg = [level2];
            return KAIDEN.validator.validateFunction("k_timeMax")(value, element, arg);
        }
    }

    if (level3) {
        if (level3 <= value) {
            var arg = [level3];
            return KAIDEN.validator.validateFunction("k_timeMax")(value, element, arg);
        }
    }

    if (level4) {
        if (level4 <= value) {
            var arg = [level4];
            return KAIDEN.validator.validateFunction("k_timeMax")(value, element, arg);
        }
    }
    return true;
});
KAIDEN.validator.setInvalidMessage("${validTimeMax4Caution1}",function(arg, element) {
    var msgParam = []
    $.each(arg, function(index, value) {
        if (msgParam [0]) {
            msgParam [0] = msgParam [0] + " " + KAIDEN.gadgetMan.getNeighboringAccessor(element, arg[index]).elem().attr("data-kaiden-surface")
        } else {
            msgParam [0] = KAIDEN.gadgetMan.getNeighboringAccessor(element, arg[index]).elem().attr("data-kaiden-surface")
        }
    });
    return KAIDEN.validator.formatMessage("${validMsg_timeMax}", element, msgParam);
});

/**
 * 注意レベル2最大時刻チェック.
 * <p>値 < 注意レベル3 < 注意レベル4 を許容</p>
 * @param value 値
 * @param element 対象エレメント
 * @param param パラメータ (配列)
 */
KAIDEN.validator.addValidateMethod("${validTimeMax4Caution2}", function(**String*/value, /**HtmlElement*/element, /**array*/param) {
    var level3 = KAIDEN.gadgetMan.getNeighboringAccessor(element, param[0]).getValue();
    var level4 = KAIDEN.gadgetMan.getNeighboringAccessor(element, param[1]).getValue();

    if (level3) {
        if (level3 <= value) {
            var arg = [level3];
            return KAIDEN.validator.validateFunction("k_timeMax")(value, element, arg);
        }
    }

    if (level4) {
        if (level4 <= value) {
            var arg = [level4];
            return KAIDEN.validator.validateFunction("k_timeMax")(value, element, arg);
        }
    }
    return true;
});
KAIDEN.validator.setInvalidMessage("${validTimeMax4Caution2}",function(arg, element) {
    var msgParam = []
    $.each(arg, function(index, value) {
        if (msgParam [0]) {

```

```
msgParam [0] = msgParam [0] + " " + KaidEN.gadgetMan.getNeighboringAccessor(element, arg[index]).elem().attr("data-kaiden-surface")
} else {
  msgParam [0] = KaidEN.gadgetMan.getNeighboringAccessor(element, arg[index]).elem().attr("data-kaiden-surface")
}
});
return KaidEN.validator.formatMessage("${validMsg_timeMax}", element, msgParam);
});
});
</script>
```

確認

勤務体系マスタで、警告レベルごとのしきい値を設定してください。

(設定方法は『[intra-mart Accel Kaiden! 勤務管理 / 管理者操作ガイド](#)』を参照してください)

遅刻早退時間が上記で設定したしきい値を超えた場合、レベルに応じたチェック結果が違反管理テーブル(k31t_violation)に登録されていることを確認してください。

 コラム

k31t_violation.message_idにはメッセージ定義ファイルのメッセージIDが登録されており、
k31t_violation.message_argsにはメッセージの引数がカンマ区切りで登録されています。

intra-mart Accel Kaiden! 勤務管理モジュール の **管理者** 機能拡張について解説します。

「管理者」とは

intra-mart Accel Kaiden! 勤務管理モジュール には「管理者」という概念があります。
組織管理者マスタメンテナンス機能で、組織に対して管理者を設定されることで「管理者」となります。

i コラム

組織管理者マスタメンテナンス機能にて、IM-共通マスタ で設定した組織に対して、管理者となるユーザを設定します。
詳細は『intra-mart Accel Kaiden! 勤務管理 / ユーザ操作ガイド』を参照してください。

i コラム

本書では以下のように呼称します。

- 組織の管理者に設定されたユーザ・・・『管理者』
- 管理者として使用することができる機能・・・『管理者機能』
- 管理者から見て管理する対象の組織・・・『管理組織』
- 管理者の管理組織に所属するユーザ・・・『管理ユーザ』

管理者は管理者機能を使用することで、管理組織に対して処理を行ったり、管理ユーザの情報を閲覧、訂正、及び設定変更を行うことができます。

intra-mart Accel Kaiden! 勤務管理モジュール 標準版での管理者機能は以下になります。

機能	備考
勤務表照会機能	管理ユーザの勤務表を照会できます。また、勤務表の訂正を行うことができます。
代理設定機能	管理ユーザの代理設定を行うことができます。
承認者設定機能	管理ユーザの承認者設定を行うことができます。
日次承認機能	管理ユーザの申請に対して日次承認を行うことができます。
締め処理機能	管理組織の組織締めを行うことができます。管理組織が会社である場合、会社締めを行うことができます。

i コラム

管理者の設定をする際、権限の範囲としての権限セットを指定します。
権限セットとは、個別権限をカテゴライズしたものです。
権限セットは「権限セットマスタ(k31m_authorityset)」個別権限は「権限セット明細マスタ(k31m_authorityset_dtl)」で定義されており、intra-mart Accel Kaiden! 勤務管理モジュールメンテナンス機能はありません。
標準版での組織管理者の設定仕様は以下の通りです。

- 権限セットは「人事権限」「管理者権限」のいずれかを選択可能
- 人事権限は会社締め後であっても勤務表の訂正を行うことができるのに対して、
管理者権限は組織締めされるまでに限って勤務表の訂正を行うことができます。
- 設定した組織とその配下組織すべての管理者に設定されたことになる
- 会社に対して管理者を設定することで、会社全体の管理者となる

締め処理機能で会社締めを行うためには、会社全体の管理者である必要があります。

管理者機能作成時のポイント

管理者機能は、管理ユーザを特定しそのユーザに紐づくデータを表示または処理する必要があります。

管理ユーザと判定される条件は以下の通りです。

- 組織管理者マスタメンテナンスで、組織に対して対象の機能の権限が付与されていること(管理組織が設定されていること)
- 管理組織が属する会社に対して、認可が設定されていること(参照権または編集権)
- 管理組織に IM-共通マスタ 上で所属しているユーザであること



コラム

「所属する」について、主所属か副所属かを intra-mart Accel Kaiden! 勤務管理モジュール 設定ファイルで設定することができます。

intra-mart Accel Kaiden! 勤務管理モジュール 設定ファイルについては、『[intra-mart Accel Kaiden! セットアップガイド](#)』を参照してください。

管理者機能を作成する際のポイントは、**ログインユーザ(管理者)からみた管理ユーザを取得する** ところです。

intra-mart Accel Kaiden! 勤務管理モジュール の基盤処理にて、管理ユーザを取得する仕組みを提供しています。

1つめとして、**アクセス権フィルタ** を使用する方法を、マスタメンテナンス機能の作成を例にして解説します。

アクセス権フィルタ

intra-mart Accel Kaiden! 勤務管理モジュール では、アクセスできる相手を制御するためのフィルタークラスを提供しています。

アクセス可能なユーザを判定し、ユーザリストを生成するフィルタ

- 管理者向けアクセス可能ユーザフィルタ

jp.co.slcs.kaiden2.product_labormgr.foundation.manager.access_data_filter.LaborEmpAuthorizedAccessUserFilter

管理組織に設定されていて且つ管理組織が属する会社に認可(参照権 or 編集権、パラメータで指定)があるかを判定し、その組織に所属するユーザのリストを生成します。

フィルタを生成するサンプルコードを以下に記載します。

```
// 認可条件
AuthorityCriteriaDto authCriteria = new AuthorityCriteriaDto();
authCriteria.action = CompanyAuthzActionType.READER;
authCriteria.locale = contextHelper.getLoginUserLocale();
authCriteria.searchCriteriaDate = dateHelper.getNowTimestamp();
authCriteria.userCd = contextHelper.getLoginUserCd();

// 会社コード(省略可)
String companyCd = "xxx";

// 管理組織に所属するユーザを検索する基準日
Timestamp a2CriteriaDate = dateHelper.getNowTimestamp();

// 権限コード
String authorityCd = "xxx";

// ログインユーザ除外フラグ(true:除外する)
boolean isExcludePerson = false;

// フィルタ生成
LaborSupAuthorizedAccessUserFilter filter = new LaborSupAuthorizedAccessUserFilter(authCriteria,
companyCd, a2CriteriaDate, authorityCd, isExcludePerson);
```

- 本人向けアクセス可能ユーザフィルタ

jp.co.slcs.kaiden2.product_labormgr.foundation.manager.access_data_filter.LaborSupAuthorizedAccessUserFilter

本人が所属する会社に認可(参照権 or 編集権、パラメータで指定)があるかを判定し、認可ありの場合は本人のユーザコードをリストにセットします。

フィルタを生成するサンプルコードを以下に記載します。

```
// 認可条件
AuthorityCriteriaDto authCriteria = new AuthorityCriteriaDto();
authCriteria.action = CompanyAuthzActionType.READER;
authCriteria.locale = contextHelper.getLoginUserLocale();
authCriteria.searchCriteriaDate = dateHelper.getNowTimestamp();
authCriteria.userCd = contextHelper.getLoginUserCd();

// フィルタ生成
LaborEmpAuthorizedAccessUserFilter filter = new LaborEmpAuthorizedAccessUserFilter(authCriteria);
```

i コラム

上記は intra-mart Accel Kaiden! 勤務管理モジュール で提供しているフィルタクラスですが、intra-mart Accel Kaiden! 基盤モジュール でもアクセス権フィルタクラスを提供しています。

- ユーザフィルタ

jp.co.slcs.kaiden2.base.foundation.manager.access_data_filter.StandardAuthorizedAccessUserFilter
認可がある会社に所属しているユーザを取得し、ユーザコードリストを生成します。

- 会社フィルタ

jp.co.slcs.kaiden2.base.foundation.manager.access_data_filter.StandardAuthorizedAccessCompanyFilter
認可がある会社の会社コードリストを生成します。

- 組織フィルタ

jp.co.slcs.kaiden2.base.foundation.manager.access_data_filter.StandardAuthorizedAccessDepartmentFilter
認可がある会社に所属している組織を取得し、会社コード・組織セットコード・組織コードオブジェクトのリストを生成します。

マスタメンテナンス機能 Action

Actionクラスでのポイントは以下の通りです。

- 継承するクラス

期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterA2MaintenanceLogicクラスを継承し、
非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスを継承します。

■ 権限コードの宣言

対象の機能の権限コードを宣言します。

ここで宣言するコードは項目マスタに登録されていることが前提となります。

■ 実装するフィルタメソッド

READERアクセス権フィルタ取得メソッドとWRITERアクセス権フィルタ取得メソッドをそれぞれ記述します。

■ Logic取得メソッド

Logicにフィルタ情報をセットして返却します。

```
/* *****
 * ここでは期間化マスタの例として
 * jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterA2MaintenanceActionクラスを継承します。
 * 非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceActionクラスを継承します。
 * ***** */
public class DeptAdminFuncAction extends AbstractTermMasterA2MaintenanceAction {

    @Resource
    public DeptAdminFuncLogic deptAdminFuncLogic;

    /* *****
     * 管理組織に所属するユーザ、を判定する基準日。
     * ***** */
    protected Timestamp a2CriteriaDate;

    /* *****
     * 権限コードを宣言します。
     * 権限コードは項目マスタに登録します。
     * ***** */
    protected static final String TARGET_AUTHORITY_ID_CD = "310004-XXX";

    /* *****
     * データ読み取り時(READER)のアクセス権フィルタ取得メソッドを記述します。
     * ***** */
    @Override
    protected AuthorizedAccessUserFilterIF[] getReadAuthorizedAccessUserFilter() {

        AuthorityCriteriaDto authCriteria = new AuthorityCriteriaDto();
        authCriteria.action = CompanyAuthzActionType.READER;
        authCriteria.locale = contextHelper.getLoginUserLocale();
        /* *****
         * 管理組織を取得する際の基準日を設定します。
         * ***** */
        authCriteria.searchCriteriaDate = dateHelper.getNowTimestamp();
        authCriteria.userCd = contextHelper.getLoginUserCd();
        LaborSupAuthorizedAccessUserFilter filter = new LaborSupAuthorizedAccessUserFilter(authCriteria,
            "%会社コード%", a2CriteriaDate, TARGET_AUTHORITY_ID_CD, false);

        return new AuthorizedAccessUserFilterIF[] {filter};
    }

    /* *****
     * データ書き込み時(WRITER)のアクセス権フィルタ取得メソッドを記述します。
     * ***** */
    @Override
    protected AuthorizedAccessUserFilterIF[] getWriteAuthorizedAccessUserFilter() {
        AuthorityCriteriaDto authCriteria = new AuthorityCriteriaDto();
        authCriteria.action = CompanyAuthzActionType.WRITER;
        authCriteria.locale = contextHelper.getLoginUserLocale();
        authCriteria.searchCriteriaDate = dateHelper.getNowTimestamp();
        authCriteria.userCd = contextHelper.getLoginUserCd();
        LaborSupAuthorizedAccessUserFilter filter = new LaborSupAuthorizedAccessUserFilter(authCriteria,
            "%会社コード%", a2CriteriaDate, TARGET_AUTHORITY_ID_CD, false);

        return new AuthorizedAccessUserFilterIF[] {filter};
    }

    /* *****
     * Logic取得メソッド。AbstractTermMasterA2MaintenanceActionが利用します。
     * Logicにフィルタ情報をセットして返却します。
     * ***** */
    @Override
    protected AbstractMasterA2MaintenanceLogic getLogic() {
        deptAdminFuncLogic.setReadA2Filter(getReadAuthorizedAccessUserFilter());
        deptAdminFuncLogic.setWriteA2Filter(getWriteAuthorizedAccessUserFilter());
        deptAdminFuncLogic.setCallerAction(this.getClass());
        return deptAdminFuncLogic;
    }

    /* *****
     * 検索画面で検索ボタンを押下したときのアクション。
     * ***** */
}
```



```

    検索画面に検索ボタンを押す時にこのメソッドを。
    * 管理組織に所属するユーザ、を判定する基準日をセットします。
    * 非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスを継承します。
    */

@Override
public String search() {
    /*
     * 管理組織に所属するユーザ、を判定する基準日をセットします。
     * 非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスを継承します。
     */

    a2CriteriaDate = dateHelper.toTimestamp("%基準日%");
    return super.search();
}
// ・
// ・
// ・
// ・
// 以降、イベントに対応するメソッドを記述
}

```



コラム

マスタメンテナンスActionクラスプログラミングと重複する箇所は記載を省略しています。

マスタメンテナンス機能 Logic

Logicクラスでのポイントは以下の通りです。

- 継承するクラス

期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterA2MaintenanceLogicクラスを継承し、
非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスを継承します。

```

/*
 * ここでは期間化マスタの例として
 * jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterA2MaintenanceLogicクラスを継承します。
 * 非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスを継承します。
 */

public class DeptAdminFuncLogic extends AbstractTermMasterA2MaintenanceLogic {

    // ・
    // ・
    // 記載を省略します。
    // ・
    // ・
}

```



コラム

マスタメンテナンスLogicクラスプログラミングと重複する箇所は記載を省略しています。



コラム

Managerを個別に呼び出す必要がある場合、パラメータでアクセス権フィルタクラスを渡します。
読み取り権の場合はreadA2Filter、書き込み権の場合はwriteA2Filterになります。
詳細はjp.co.slcs.kaiden2.base.foundation.master.AbstractMasterA2MaintenanceLogicクラスのAPIリストを参照ください。

マスタメンテナンス機能 Manager

Managerクラスでのポイントは以下の通りです。

- 継承するクラス

期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterA2Manager<DTO, F>クラスを継承し、
非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.manager.AbstractMasterA2Manager<DTO, F>クラスを継承します。

- 権限を考慮するメソッド、権限を考慮しないメソッドを実装

権限を考慮するメソッドと権限を考慮しないメソッドを実装します。
権限を考慮するメソッドにアクセス権フィルタを渡すことで、管理対象ユーザかどうかの判定を行い、結果をフィルタリングして返却します。

```

public interface DeptAdminFuncManager extends
    TermMasterA2Manager<DeptAdminFuncManagerDto, AuthorizedAccessUserFilterIF> {

    // ・
    // ・
    // 記載を省略します。
    // ・
    // ・
}

```

```
/* *****
 * ここでは期間化マスタの例として
 * jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterA2Manager<DTO, F>クラスを継承します。
 * 非期間化マスタの場合はjp.co.slcs.kaiden2.base.foundation.manager.AbstractMasterA2Manager<DTO, F>クラスを継承します。
 * ***** */

public class DeptAdminFuncManagerImpl extends
    AbstractTermMasterA2Manager<DeptAdminFuncManagerDto, AuthorizedAccessUserFilterIF> implements
    DeptAdminFuncManager {

    // ・
    // ・
    // 記載を省略します。
    // ・
    // ・

    /* *****
     * 権限を考慮した主キー検索メソッド。
     * Actionで定義したFilterクラスをパラメータで渡します。
     * ***** */

    @Override
    public DeptAdminFuncManagerDto select(DeptAdminFuncManagerDto criteria,
        AuthorizedAccessUserFilterIF[] a2Filter) {
        return super.select(k31mXxxService, criteria, a2Filter);
    }

    /* *****
     * 権限を考慮しない主キー検索メソッド。
     * 会社認可、及びアクセス権による絞り込みを行わないで結果を返却します。
     * ***** */

    @Override
    public DeptAdminFuncManagerDto selectIgnoreAuth(DeptAdminFuncManagerDto criteria) {
        return super.selectIgnoreAuth(k31mXxxService, criteria);
    }

    // ・
    // ・
    // 記載を省略します。
    // ・
    // ・
}
```



コラム
マスタメンテナンスManagerクラスプログラミングと重複する個所は記載を省略しています。

2つめとして、アクセス権フィルタを使用せずに **DBファンクション** を直接使用するケースを解説します。

DBファンクション

intra-mart Accel Kaiden! 勤務管理モジュール では、管理組織及び管理ユーザを取得する手段として、DBファンクションを提供しています。
前項で紹介したマスタメンテナンス機能のアクセス権フィルタクラスの実態も、以下のDBファンクションを使用しています。

管理組織取得DBファンクション - kimm_has_authority_dept

引数

No.	論理名	型
1	ユーザコード(管理者)	varchar2
2	管理ユーザ所属判定基準日	timestamp
3	管理者権限判定基準日	timestamp
4	会社コード	varchar2
5	組織セットコード	varchar2
6	権限コード	varchar2
7	組織コード	varchar2

戻り値

No.	論理名	型
1	会社コード	varchar2
2	組織セットコード	varchar2
3	組織コード	varchar2

管理ユーザ取得DBファンクション - klbr_has_authority_user

引数

No.	論理名	型
1	ユーザコード(管理者)	varchar2
2	管理ユーザ所属判定基準日	timestamp
3	管理者権限判定基準日	timestamp
4	会社コード	varchar2
5	組織セットコード	varchar2
6	権限コード	varchar2
7	組織コード	varchar2
8	主所属フラグ	varchar2

戻り値

No.	論理名	型
1	ユーザコード(管理ユーザ)	varchar2

その他

本項では、他の項で解説していないその他機能の概要を解説します。

機能説明

本項では、intra-mart Accel Kaiden!XMLについて解説します。

- constants
 - 基盤モジュール
 - IM-Workflow連携モジュール
 - ワークフローモジュール
 - 勤務管理モジュール

constants

intra-mart Accel Kaiden!では各モジュールごとに定数ファイルを用意しています。
各モジュールで定義している定数は以下の通りです。

基盤モジュール

- Path
src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/base_constants.xml

Key	初期値	項目名	説明
BASE.version	2.0.2	基盤モジュールバージョン	モジュールのバージョンを定義しています。
BASE.currency.scaleInt	9	通貨整数部桁数	intra-mart Accel Kaiden!で使用する通貨整数部桁数を定義しています。
BASE.currency.scaleDec	4	通貨小数部桁数	intra-mart Accel Kaiden!で使用する通貨小数部桁数を定義しています。
BASE.timeFormat.nextDaySymbol	+	24時間制時刻表記 翌日記号	時刻テキストボックスで24時間を超えた場合に表示する記号
BASE.pdf.ja.font	HeiseiKakuGo-W5	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のjaフォント(itextのフォントを指定)を定義しています。
BASE.pdf.ja.enCoding	UniJIS-UCS2-H	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のjaフォント(itextのフォントを指定)を定義しています。
BASE.pdf.en.font	HeiseiKakuGo-W5	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のenフォント(itextのフォントを指定)を定義しています。
BASE.pdf.en.enCoding	UniJIS-UCS2-H	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のenフォント(itextのフォントを指定)を定義しています。
BASE.pdf.zh_CN.font	HeiseiKakuGo-W5	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のzh_CNフォント(itextのフォントを指定)を定義しています。
BASE.pdf.zh_CN.enCoding	UniJIS-UCS2-H	PDF出力ロケールごとのフォント(itextのフォントを指定)	PDF出力のzh_CNフォント(itextのフォントを指定)を定義しています。
BASE.filtering.conditionMode	0	フィルタリング条件競合時の条件適用モード	以下の定義によって条件を適用します。 0:論理積(and) 1:論理和(or)

IM-Workflow連携モジュール

- Path
src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/extension_imw_constants.xml

Key	初期値	項目名	説明
kaiden_extension_imw	2.0.2	IM-Workflow連携モジュールバージョン	モジュールのバージョンを定義しています。
EIMW.allow_cpimatterdelete	0	完了案件削除を可否	以下の定義によって削除可否を判定します。 1:許容 それ以外:許容しない
EIMW.archivemode	0	アーカイブ処理時のデータ保存範囲の定義	以下の定義によって保存範囲を判定します。 0:履歴含まず 1:履歴含む
EIMW.jsp_path.select_company	/kaiden2/extension_imw/workflow /select_company /select_company.jsp	会社選択画面JSPパス	会社選択画面のJSPを定義しています。
EIMW.jsp_path.apply	/kaiden2/extension_imw /workflow/apply /apply.jsp	申請画面JSPパス	申請画面のJSPを定義しています。
EIMW.jsp_path.apply_sp	なし	申請(sp)画面JSPパス	申請(sp)画面のJSPを定義しています。
EIMW.jsp_path.temporary_save	/kaiden2/extension_imw /workflow/temporary_save /temporary_save.jsp	一時保存画面JSPパス	一時保存画面のJSPを定義しています。
EIMW.jsp_path.temporary_save_sp	なし	一時保存(sp)画面JSPパス	一時保存(sp)画面のJSPを定義しています。
EIMW.jsp_path.unapply	/kaiden2/extension_imw /workflow/unapply /unapply.jsp	申請(起票案件)画面JSPパス	申請(起票案件)画面のJSPを定義しています。

Key	初期値	項目名	説明
EIMW.jsp_path.unapply_sp	なし	申請(起票案件)(sp)画面JSPパス	申請(起票案件)(sp)画面のJSPを定義しています。
EIMW.jsp_path.reapply	/kaiden2/extension_imw /workflow/reapply /reapply.jsp	再申請画面JSPパス	再申請画面のJSPを定義しています。
EIMW.jsp_path.reapply_sp	なし	再申請(sp)画面JSPパス	再申請(sp)画面のJSPを定義しています。
EIMW.jsp_path.process	/kaiden2/extension_imw /workflow/process /process.jsp	処理画面JSPパス	処理画面のJSPを定義しています。
EIMW.jsp_path.process_sp	/kaiden2/extension_imw /workflow/process /process_sp.jsp	処理(sp)画面JSPパス	処理(sp)画面のJSPを定義しています。
EIMW.jsp_path.confirm	/kaiden2/extension_imw /workflow/confirm /confirm.jsp	確認画面JSPパス	確認画面のJSPを定義しています。
EIMW.jsp_path.confirm_sp	/kaiden2/extension_imw /workflow/confirm /confirm_sp.jsp	確認(sp)画面JSPパス	確認(sp)画面のJSPを定義しています。
EIMW.jsp_path.process_detail	/kaiden2/extension_imw /workflow/process_detail /process_detail.jsp	処理詳細画面JSPパス	処理詳細画面のJSPを定義しています。
EIMW.jsp_path.process_detail_sp	/kaiden2/extension_imw /workflow/process_detail /process_detail_sp.jsp	処理詳細(sp)画面JSPパス	処理詳細(sp)画面のJSPを定義しています。
EIMW.jsp_path.reference_detail	/kaiden2/extension_imw /workflow/reference_detail /reference_detail.jsp	参照詳細画面JSPパス	参照詳細画面のJSPを定義しています。
EIMW.jsp_path.reference_detail_sp	/kaiden2/extension_imw /workflow/reference_detail /reference_detail_sp.jsp	参照詳細(sp)画面JSPパス	参照詳細(sp)画面のJSPを定義しています。
EIMW.jsp_path.confirm_detail	/kaiden2/extension_imw /workflow/confirm_detailx /confirm_detail.jsp	確認詳細画面JSPパス	確認詳細画面のJSPを定義しています。
EIMW.jsp_path.confirm_detail_sp	/kaiden2/extension_imw /workflow/confirm_detail /confirm_detail_sp.jsp	確認詳細(sp)画面JSPパス	確認詳細(sp)画面のJSPを定義しています。
EIMW.jsp_path.archive_matter_detail	/kaiden2/extension_imw /workflow/archive_matter_detail /archive_matter_detail.jsp	過去詳細画面JSPパス	過去詳細画面のJSPを定義しています。
EIMW.jsp_path.archive_matter_detail_sp	/kaiden2/extension_imw /workflow/archive_matter_detail /archive_matter_detail_sp.jsp	過去詳細(sp)画面JSPパス	過去詳細(sp)画面のJSPを定義しています。
EIMW.jsp_path.sp_nosupport	/kaiden2/extension_imw /workflow/sp_nosupport /sp_nosupport.jsp	モバイルサポート外画面JSPパス	モバイルサポート外画面のJSPを定義しています。
EIMW.jsp_path.history	/kaiden2/extension_imw /workflow/history /history.jsp	履歴画面JSPパス	履歴画面のJSPを定義しています。
EIMW.jsp_path.print	/kaiden2/extension_imw /workflow/print /print.jsp	印刷画面JSPパス	印刷画面のJSPを定義しています。
EIMW.jsp_path.matter_search	/kaiden2/extension_imw /workflow/matter_search /matter_search.jsp	申請書検索画面JSPパス	申請書検索画面のJSPを定義しています。
EIMW.sendback_regist	1	差戻し時、処理画面の入力値登録有無	以下の条件で入力値登録有無を判定します。 0:登録しない、及びサーバサイド入力チェックを! 1:登録する、及びサーバサイド入力チェックを実行
EIMW.reserve_regist	1	保留時、処理画面の入力値登録有無	以下の条件で入力値登録有無を判定します。 0:登録しない、及びサーバサイド入力チェックを! 1:登録する、及びサーバサイド入力チェックを実行
EIMW.reserve_cancel_regist	1	保留解除時、処理画面の入力値登録有無	以下の条件で入力値登録有無を判定します。 0:登録しない、及びサーバサイド入力チェックを! 1:登録する、及びサーバサイド入力チェックを実行
EIMW.jsp_include_path.matter_info	/WEB-INF/view/kaiden2/extension_imw /workflow/matter_info /matter_info.jsp	案件情報JSPパス	案件情報のJSPを定義しています。
EIMW.jsp_include_path.matter_info_print	/WEB-INF/view/kaiden2/extension_imw /workflow/matter_info /matter_info.jsp	案件情報(印刷用)パス	案件情報(印刷用)のJSPを定義しています。

Key	初期値	項目名	説明
EIMW.jsp_path.print	/kaiden2/extension_imw /workflow/print /print.jsp	印刷画面JSPパス	印刷画面のJSPを定義しています。
EIMW.jsp_include_path.history_info	/WEB-INF/view//kaiden2/extension_imw /workflow/history_info /history_info.jsp	案件履歴JSPパス	案件履歴のJSPを定義しています。

ワークフローモジュール

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow_constants.xml

Key	初期値	項目名	説明
PWKF.version	2.0.2	ワークフローモジュールバージョン	モジュールのバージョンを定義しています。
PWKF.master.app.dispPagePathFlag	0	申請書マスタのページパス入力欄の表示	以下の定義によって入力有無を切り替えます。 0:表示しない(標準) 1:表示する(カスタマイズ向け)
PWKF.master.app.processInputSp	0	申請書マスタの承認入力時のsp画面利用可否	以下の定義によってsp画面利用有無を切り替えます。 0:利用不可(標準) 1:利用可(カスタマイズ向け)

勤務管理モジュール

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_labormgr_constants.xml

Key	初期値	項目名	説明
PLBR.version	2.0.2	勤務管理モジュールバージョン	モジュールのバージョンを定義しています。
PLBR.workTimeInputMax	48	勤務時間入力欄最大入力可能時刻	入力可能時刻を切り替えます。 ex. 48 = 当日の0:00～翌日の23:59の間で入力可能 48を超える設定はサポート対象外です

サンプル

ガジェット(経費旅費)

経費明細

経費明細のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   ├── K30tExpenseDetail.java
│   │   │   │   │   │   │   │   │   ├── K30tExpenseDetailNames.java
│   │   │   │   │   │   │   │   │   ├── Ka30tExpenseDetail.java
│   │   │   │   │   │   │   │   │   ├── Ka30tExpenseDetailNames.java
│   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │       ├── K30tExpenseDetailCondition.java
│   │   │   │   │   │   │   │       ├── K30tExpenseDetailService.java
│   │   │   │   │   │   │   │       ├── Ka30tExpenseDetailCondition.java
│   │   │   │   │   │   │   │       └── Ka30tExpenseDetailService.java
│   │   ├── java
│   │   │   ├── jp
│   │   │   │   ├── co
│   │   │   │   │   ├── slcs
│   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── AbstractExpenseDetailActionBase.java
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV01Action.java
│   │   │   │   │   │   │   │   │   ├── conf
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailFieldName.java
│   │   │   │   │   │   │   │   │   ├── form
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV01Form.java
│   │   │   │   │   │   │   │   │   ├── logic
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   ├── AbstractExpenseDetailLogicBase.java
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV01Logic.java
│   │   │   │   │   │   │   │   └── foundation
│   │   │   │   │   │   │   │       ├── manager
│   │   │   │   │   │   │   │       │   ├── expensedetail
│   │   │   │   │   │   │   │       │   │   ├── ExpenseDetailManager.java
│   │   │   │   │   │   │   │       │   │   ├── ExpenseDetailManagerDto.java
│   │   │   │   │   │   │   │       │   └── impl
│   │   │   │   │   │   │   │       │       └── ExpenseDetailManagerImpl.java
│   │   ├── resources
│   │   │   ├── META-INF
│   │   │   │   ├── sql
│   │   │   │   │   ├── jp
│   │   │   │   │   │   ├── co
│   │   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── impl
│   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── ExpenseDetailManagerImpl
│   │   │   │   │   │   │   │   │   │   │   │   │   │   └── selectForJournalSourceDtl.sql
│   │   │   │   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── K30tExpenseDetailService
│   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── selectView.sql
│   │   │   │   │   │   │   │   │   │   │   │   │   │   └── selectViewForCopy.sql
│   │   ├── webapp
│   │   │   ├── WEB-INF
│   │   │   │   ├── view
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   ├── expenseDetail
│   │   │   │   │   │   │   │   │   ├── expenseDetailV01.jsp
│   │   │   │   │   │   │   │   │   └── expenseDetailV01_sp.jsp

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractExpenseDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 経費明細がジェット 基底Actionの抽象クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractExpenseDetailActionBase implements GadgetActionIF {

    /**
     * 申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }
}

```

```

/**
 * 確認画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConvertor());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    return null;
}

```

```

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #ExpenseDetailConverter 経費明細(標準)ガジェット}の書式変換を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected ConvertForJsonSource getConverter() {
    return new ExpenseDetailConverter(getForm());
}

/**
 * 経費明細ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class ExpenseDetailConverter extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;
    /**
     * NumberHelper.
     */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public ExpenseDetailConverter(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     * <tr>
     * <td>換算額</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * </table>
     * </p>
     */
    @Override
    protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

        Map<String, Object> result = new HashMap<String, Object>(map);

        // 取引通貨額の書式変換
        if (result.containsKey(EXPENSE_DETAIL_TRANS_AMOUNT)) {
            BigDecimal amount = (BigDecimal) result.get(EXPENSE_DETAIL_TRANS_AMOUNT);
            Object currencyCd = result.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD);
            result.put(EXPENSE_DETAIL_TRANS_AMOUNT,

```

```

        numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (result.containsKey(EXPENSE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(EXPENSE_DETAIL_COMPANY_RATE);
        result.put(EXPENSE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }

    // 換算額の書式変換
    if (result.containsKey(EXPENSE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(EXPENSE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = result.get(EXPENSE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>換算額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fidNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fidNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(EXPENSE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(EXPENSE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(EXPENSE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(EXPENSE_DETAIL_COMPANY_RATE);
        result.put(EXPENSE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }

    // 換算額の書式変換
    if (map.containsKey(EXPENSE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(EXPENSE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = map.get(EXPENSE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}
}
}

```

Action

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/ExpenseDetailV01Action.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.ExpenseDetailV01Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail.ExpenseDetailV01Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 経費明細(標準)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailV01Action extends AbstractExpenseDetailActionBase {

    /**
     * ExpenseDetailV01Logic.
     */
    @Resource
    public ExpenseDetailV01Logic expenseDetailV01Logic;

    /**
     * ExpenseDetailV01Form.
     */
    @ActionForm
    @Resource
    public ExpenseDetailV01Form expenseDetailV01Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return expenseDetailV01Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return expenseDetailV01Form;
    }
}

```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/ExpenseDetailV01Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 経費明細(標準)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailV01Form extends AbstractImwPageForm {

}

```

Logic(基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/expensedetail/AbstractExpenseDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_OCCUR_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_SUMMARY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;

```



```

import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細ガジェット用Logicの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractExpenseDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 経費明細ガジェット. */
    protected static final String BLOCK_KEY_EXPENSE_DETAIL_TABLE = "expenseDetailTable";

    /** 経費内容. */
    public static final int SUMMARY_LEN = 250;
    /** 税率(整数部). */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部). */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部). */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部). */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先. */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部). */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部). */
    public static final int QUANTITY_DEC_LEN = 10;

    /** ExpenseDetailManager. */
    @Resource
    protected ExpenseDetailManager expenseDetailManager;

    /** JournalSourceDtlManager. */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper. */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** コンストラクタ. */
    public AbstractExpenseDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 経費明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる経費明細テーブルを削除後に、経費明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 経費明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }
}

```

```

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
 * </p>
 */
@Override
public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
 * </p>
 */
@Override
public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 経費明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}

```

```

    * {@inheritDoc}
    * <p>
    * 本機能では何も処理を行わず、常に0を返却します。
    * </p>
    */
    @Override
    public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 経費明細を登録します。
    * </p>
    */
    @Override
    public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
    * {@inheritDoc}
    * <p>
    * 経費明細テーブルの対象データを削除します。<br>
    * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
    * </p>
    */
    @Override
    public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTable(true);
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる経費明細テーブルを削除後に、経費明細テーブルを登録します。<br>
    * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
    * </p>
    */
    @Override
    public int tempSaveInsert(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

```

```

public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了],[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * </p>
 */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.inmwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 経費明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userDataId = pluginGadgetDTO.userDataId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 経費明細データを取得
            List<BeanMap> beanMapList = expenseDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる経費明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる(過去案件)経費明細テーブルを一括削除します。
 * </p>
 */
@Override

```

```

    public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = expenseDetailManager
            .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto()).imwPluginParameterDto));
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる経費明細の履歴を含めた全てのデータを削除します。
     * </p>
     */
    @Override
    public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
     * {@inheritDoc}
     *
     * @see #viewMapFormatted
     */
    @Override
    protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
        return viewMapFormatted(conditionDto, null);
    }

    /**
     * {@inheritDoc}
     * <p>
     * データの取得は、{@link ExpenseDetailManager#selectList(ExpenseDetailManagerDto)}を利用します。
     * </p>
     */
    @Override
    protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
        throws KaidenSystemException {
        ExpenseDetailManagerDto dto = Beans.createAndCopy(ExpenseDetailManagerDto.class, conditionDto).execute();
        List<ExpenseDetailManagerDto> listSelect = expenseDetailManager.selectList(dto);

        LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
        List<Map<String, Object>> lstDtoMap = new ArrayList<Map<String, Object>>();
        for (ExpenseDetailManagerDto src : listSelect) {
            if (converter == null) {
                lstDtoMap.add(BeanUtil.toMap(src, true));
            } else {
                lstDtoMap.add(converter.mapConvert(BeanUtil.toMap(src, true)));
            }
        }

        rv.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, lstDtoMap);
        return rv;
    }

    /**
     * 経費明細テーブルデータ登録処理。
     *
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int insertTable() throws KaidenSystemException {
        // 経費明細ガジェットデータをMap形式で取得
        List<ExpenseDetailManagerDto> dtos = JsonUtil.getDtoList(ExpenseDetailManagerDto.class,
            getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE));

        // 経費明細テーブル登録
        return expenseDetailManager.entry(dtos);
    }

    /**
     * 仕訳連携用明細テーブルデータ登録処理。
     *
     * @param srcList 経費明細テーブルデータリスト
     * @param matterEndProcessDto 案件終了処理用DTO
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
        MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
        for (JournalSourceDtlManagerDto srcDto : srcList) {
            // 仕訳データ区分をセット
            srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
        }

        // 仕訳連携用明細テーブル登録

```

```

    return journalSourceDtlManager.entry(srcList);
}

/**
 * 経費明細テーブルデータ削除処理.
 *
 * @param tempFlg 一時保存データフラグ
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    ExpenseDetailManagerDto mngDto = new ExpenseDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = super.getGadgetDto().userDataId;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    return expenseDetailManager.remove(mngDto);
}

/**
 * 経費明細テーブルデータ削除処理(履歴を含む).
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    ExpenseDetailManagerDto mngDto = new ExpenseDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    return expenseDetailManager.removeAll(mngDto);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */

```

```

@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>

```

```

* </tr>
* <tr>
* <td>発生日</td>
* <td>{@link DateValidator 日付型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>摘要</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>経費区分コード</td>
* <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>取引通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>取引通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td rowspan="2">会社通貨換算レート</td>
* <td>{@link DigitsValidator 小数量型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数量型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>

```



```

* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
KaidenApplicationException {

    if (fieldDefsMap.containsKey(BLOCK_KEY_EXPENSE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_EXPENSE_DETAIL_TABLE, tmpFlg);
        // 発生日
        addValidateOnlyInput(EXPENSE_DETAIL_OCCUR_DATE, VALID_DATE);
        // 摘要
        addValidateOnlyInput(EXPENSE_DETAIL_SUMMARY, VALID_MAX_LENGTH, getArgs(SUMMARY_LEN));
        // 経費区分コード
        addValidate(EXPENSE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, EXPENSE_DETAIL_EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidate(EXPENSE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, EXPENSE_DETAIL_ACCOUNT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidate(EXPENSE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, EXPENSE_DETAIL_TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 取引通貨コード
        addValidate(EXPENSE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, EXPENSE_DETAIL_TRANS_CURRENCY_CD,
            getGadgetDto().searchCriteriaCampany);
        // 会社通貨換算レート
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_RATE, VALID_DIGITS,
            getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨コード
        addValidate(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS,
            EXPENSE_DETAIL_COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 支払分類コード
        addValidate(EXPENSE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, EXPENSE_DETAIL_PAY_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidate(EXPENSE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, EXPENSE_DETAIL_PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払先名
        addValidateOnlyInput(EXPENSE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
        // 組織コード
        addValidate(EXPENSE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCampany,
            EXPENSE_DETAIL_DEPARTMENT_SET_CD, EXPENSE_DETAIL_DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidate(EXPENSE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, EXPENSE_DETAIL_PROJECT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 数量
        addValidateOnlyInput(EXPENSE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
        // 会社通貨額合計
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
    }
}

/**
 * ブロックの共通シングルバリデート定義MAPにエンタリを追加.
 *
 * @param fieldKey フィールドキー
 * @param validationId バリデートキー
 * @param args バリデート実行時アーギュメント
 */
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * ブロックの共通シングルバリデート定義MAPにエンタリを追加.
 *
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 *

```

```

* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
* {@inheritDoc}
* <p>
* システム案件idが未設定の場合(初回表示時など)は、空のリストを返却します。<br>
* 設定されている場合は、{@link ExpenseDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, new ArrayList<ExpenseDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectView(criteria));
    }
    return result;
}

/**
* {@inheritDoc}
* <p>
* {@link ExpenseDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
* {@inheritDoc}
* <p>
* {@link ExpenseDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectViewArchive(criteria));
    return result;
}

/**
* {@inheritDoc}
* <p>
* 取得した経費明細のデータに対して、以下マスタの整合性チェックを行います。
* <ul>
* <li>[*1]通貨マスタ(imm_currency)</li>
* <li>組織マスタ(imm_depaertment)</li>
* <li>プロジェクトマスタ</li>
* <li>支払分類マスタ</li>
* <li>支払方法マスタ</li>
* <li>経費区分マスタ</li>
* <li>勘定科目マスタ</li>
* <li>税区分マスタ</li>
* </ul>
* *1: 取引通貨と会社通貨
* </p>
*/
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {

    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_EXPENSE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<ExpenseDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), ExpenseDetailManagerDto.class);

            for (ExpenseDetailManagerDto j : sqlResultList) {
                // 取引通貨チェック
                if (!isEmpty(j.transCurrencyCd) && isEmpty(j.transCurrencyName)) {
                    String[] fieldId = getArgs(EXPENSE_DETAIL_TRANS_CURRENCY_CD, EXPENSE_DETAIL_TRANS_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),

```

```

        BaseProp.S.CURRENCY);
String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
    masterName, values, message));
}

// 会社通貨チェック
if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD,
        EXPENSE_DETAIL_COMPANY_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 組織チェック
if (isNotEmpty(j.departmentSetCd) && isNotEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_DEPARTMENT_SET_CD, EXPENSE_DETAIL_DEPARTMENT_CD,
        EXPENSE_DETAIL_DEPARTMENT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.DEPARTMENT);
    String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// プロジェクトチェック
if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PROJECT_CD, EXPENSE_DETAIL_PROJECT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PROJECT);
    String[] values = getArgs(j.projectCd, j.projectName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 支払分類マスタチェック
if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PAY_TYPE_CD, EXPENSE_DETAIL_PAY_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_TYPE);
    String[] values = getArgs(j.payTypeCd, j.payTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 支払方法マスタチェック
if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PAY_METHOD_CD, EXPENSE_DETAIL_PAY_METHOD_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_METHOD);
    String[] values = getArgs(j.payMethodCd, j.payMethodName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 経費区分マスタチェック
if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_EXP_TYPE_CD, EXPENSE_DETAIL_EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(j.expTypeCd, j.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        j.accountCd = null;
        j.accountName = null;
        j.taxTypeCd = null;
        j.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

```

```

    }

    // 勘定科目マスタチェック
    if (isEmpty(j.accountCd) && isEmpty(j.accountName)) {
        String[] fieldId = getArgs(EXPENSE_DETAIL_ACCOUNT_CD, EXPENSE_DETAIL_ACCOUNT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.ACCOUNT);
        String[] values = getArgs(j.accountCd, j.accountName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
            masterName, values, message));
    }

    // 税区分マスタチェック
    if (isEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
        String[] fieldId = getArgs(EXPENSE_DETAIL_TAX_TYPE_CD, EXPENSE_DETAIL_TAX_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.TAX_TYPE);
        String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
            // 税率、税抜金額、税額をクリア
            j.taxRate = null;
            j.companyAmountExcTax = null;
            j.companyAmountTax = null;
        }

        result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
            masterName, values, message));
    }
}

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Deprecated
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();
    list.add(SYSTEM_MATTER_ID);
    list.add(USER_DATA_ID);
    list.add(GADGET_CLASS);
    list.add(GADGET_INSTANCE);
    return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccCountIF#getClaimExpenccCount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccCount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccCountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.

```

```

    の値を設定。
    *
    * @param params アーギュメント
    * @return 会社支払額
    */

public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(EXPENSE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */

public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(EXPENSE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 費用明細
 */

public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            list.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 費用合計
 */

public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

```

```

BigDecimal sum = BigDecimal.ZERO;

for (Map<String, Object> j : blocjData) {
    String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
    if (StringUtil.isEmpty(strAmnt)) {
        sum = sum.add(new BigDecimal(strAmnt));
    }
}

retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 経費明細テーブルの対象データを過去案件退避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    @SuppressWarnings("unchecked")
    List<ExpenseDetailManagerDto> list = (List<ExpenseDetailManagerDto>) dataMap
        .get(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    int rv = expenseDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(expenseDetailManager);
    return list;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/expensedetail/ExpenseDetailV01Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

```

```

/
* 経費明細(標準)ガジェット (Logic).
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

public class ExpenseDetailV01Logic extends AbstractExpenseDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenccountIF {

    /** コンストラクタ. */
    public ExpenseDetailV01Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します。<br>
     * <br>
     * <p>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力/表示</td>
     * </tr>
     * </table>
     * <p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        super.defineValidateForNormal(params, tmpFlg);

        // 取引通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

        if (!tmpFlg) {
            // 組織コード
            addValidateOnlyInput(EXPENSE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, EXPENSE_DETAIL_DEPARTMENT_CD,
                EXPENSE_DETAIL_PROJECT_CD);
            // プロジェクトコード
            addValidateOnlyInput(EXPENSE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, EXPENSE_DETAIL_DEPARTMENT_CD,
                EXPENSE_DETAIL_PROJECT_CD);
        }
    }

    /**
     * {@inheritDoc}
     * <p>
     * 下記補完を行います。
     * <p>
     * <ul>
     * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
     * <li>取引通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
     * <li>会社通貨額: 取引通貨額の値をコピー</li>
     * <li>会社通貨換算レート: 固定値: 1を設定</li>
     * <li>支払方法: ガジェット間連携項目. 支払方法コードを設定</li>
     * <li>数量: 固定値: 1を設定</li>
     * <li>立替フラグ: 支払分類コードに紐づく立替フラグを支払分類マスクから取得して設定</li>
     * <li>会社通貨額: 取引通貨額 * 通貨レートの計算結果</li>
     * <li>会社通貨額合計: 会社通貨額 * 数量の計算結果</li>
     * </ul>
     */
    @Override
    protected void complementIsbnData(GadgetMediatorParams params) {

```

```

protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 取引通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(EXPENSE_DETAIL_TRANS_CURRENCY_CD, companyCurrencyCd);

            // 会社通貨換算レート
            j.put(EXPENSE_DETAIL_COMPANY_RATE, "1");

            // 数量
            j.put(EXPENSE_DETAIL_QUANTITY, "1");

            // 立替フラグ
            if (StringUtil.isNotEmpty((String) j.get(EXPENSE_DETAIL_PAY_TYPE_CD))) {
                PayTypeManagerDto criteria = new PayTypeManagerDto();
                criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
                criteria.payTypeCd = (String) j.get(EXPENSE_DETAIL_PAY_TYPE_CD);
                criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                    .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
                List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                    criteria, new ArrayList<OrderDto>());
                if (CollectionUtil.isNotEmpty(retList)) {
                    j.put(EXPENSE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
                }
            }
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * </p>
 *
 * @return グローバルデータ補完処理
 */
@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * <p>
 * <ul>
 * <li>会社通貨額: 取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計: 会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

```



```

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(EXPENSE_DETAIL_TRANS_AMOUNT),
                (String) j.get(EXPENSE_DETAIL_COMPANY_RATE)));
            j.put(EXPENSE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(EXPENSE_DETAIL_QUANTITY)));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>支払方法コード:ガジェット間連携項目. 支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード
            if (gcPayMethodCd == null) {
                gcPayMethodCd = getGadgetCouplingPayMethod();
            }
            j.put(EXPENSE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
        }
    }
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenccCount(Map<String, Object> params) {
    return super.putClaimExpenccCount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/expensedetail/ExpenseDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**

```

```

* 経費明細ガジェット用 フィールド名定義クラス。
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public final class ExpenseDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ。
     */
    protected ExpenseDetailFieldName() {
    }

    /** 発生日。 */
    public static final String EXPENSE_DETAIL_OCCUR_DATE = "occurDate";
    /** 摘要。 */
    public static final String EXPENSE_DETAIL_SUMMARY = "summary";
    /** 経費区分コード。 */
    public static final String EXPENSE_DETAIL_EXP_TYPE_CD = "expTypeCd";
    /** 勘定科目コード。 */
    public static final String EXPENSE_DETAIL_ACCOUNT_CD = "accountCd";
    /** 税区分コード。 */
    public static final String EXPENSE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
    /** 税率。 */
    public static final String EXPENSE_DETAIL_TAX_RATE = "taxRate";
    /** 数量。 */
    public static final String EXPENSE_DETAIL_QUANTITY = "quantity";
    /** 取引通貨コード。 */
    public static final String EXPENSE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
    /** 取引通貨額。 */
    public static final String EXPENSE_DETAIL_TRANS_AMOUNT = "transAmount";
    /** 会社通貨コード。 */
    public static final String EXPENSE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
    /** 会社通貨換算レート。 */
    public static final String EXPENSE_DETAIL_COMPANY_RATE = "companyRate";
    /** 会社通貨額。 */
    public static final String EXPENSE_DETAIL_COMPANY_AMOUNT = "companyAmount";
    /** 会社通貨税抜額。 */
    public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
    /** 会社通貨税額。 */
    public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
    /** 会社通貨額合計。 */
    public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
    /** 支払分類コード。 */
    public static final String EXPENSE_DETAIL_PAY_TYPE_CD = "payTypeCd";
    /** 支払方法コード。 */
    public static final String EXPENSE_DETAIL_PAY_METHOD_CD = "payMethodCd";
    /** 支払先コード。 */
    public static final String EXPENSE_DETAIL_PAYEE_CD = "payeeCd";
    /** 支払先名。 */
    public static final String EXPENSE_DETAIL_PAYEE_NAME = "payeeName";
    /** 組織セットコード。 */
    public static final String EXPENSE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
    /** 組織コード。 */
    public static final String EXPENSE_DETAIL_DEPARTMENT_CD = "departmentCd";
    /** プロジェクトコード。 */
    public static final String EXPENSE_DETAIL_PROJECT_CD = "projectCd";

    /** 会社名。 */
    public static final String EXPENSE_DETAIL_COMPANY_NAME = "companyName";
    /** 経費区分名称。 */
    public static final String EXPENSE_DETAIL_EXP_TYPE_NAME = "expTypeName";
    /** 勘定科目名称。 */
    public static final String EXPENSE_DETAIL_ACCOUNT_NAME = "accountName";
    /** 税区分名称。 */
    public static final String EXPENSE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
    /** 取引通貨名称。 */
    public static final String EXPENSE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
    /** 会社通貨名称。 */
    public static final String EXPENSE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
    /** 支払分類名称。 */
    public static final String EXPENSE_DETAIL_PAY_TYPE_NAME = "payTypeName";
    /** 支払方法名称。 */
    public static final String EXPENSE_DETAIL_PAY_METHOD_NAME = "payMethodName";
    /** 組織セット名称。 */
    public static final String EXPENSE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
    /** 組織名称。 */
    public static final String EXPENSE_DETAIL_DEPARTMENT_NAME = "departmentName";
    /** プロジェクト名称。 */
    public static final String EXPENSE_DETAIL_PROJECT_NAME = "projectName";
    /** 立替フラグ。 */
    public static final String EXPENSE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/ExpenseDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細テーブルマネージャークラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ExpenseDetailManager extends PluginGadgetManager<ExpenseDetailManagerDto> {

    /** {@link ExpenseDetailManager#selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link ExpenseDetailManager#selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link ExpenseDetailManager#selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager (実装)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/impl/ExpenseDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tExpenseDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tExpenseDetailService;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link ExpenseDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link ExpenseDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>

```

```

* <td>主キー検索</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>表示案件検索</td>
* <td>selectView</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>コピー案件検索</td>
* <td>selectViewForCopy</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件検索</td>
* <td>selectViewArchive</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括挿入</td>
* <td>entry</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括更新</td>
* <td>renew</td>
* <td>更新</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括削除</td>
* <td>removeAll</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>過去案件登録</td>
* <td>matterArchiveRegister</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件処理時削除</td>
* <td>matterArchivePostDelete</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件一括削除</td>
* <td>removeAllArchive</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能別DTOクラスのI/Oマトリクス</b>
* <table border="1">
* <tr>
* <td colspan="2" rowspan="2" align="center"><b>DTO</b></td>
* <td colspan="2" align="center"><b>主キー</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>リスト</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>表示</b><br>
* 案件</td>
* <td colspan="2" align="center"><b>コピー</b><br>
* 案件</td>
* <td colspan="2" align="center"><b>過去</b><br>
* 案件</td>
* <td colspan="2" align="center"><b>一括</b><br>
* 挿入</td>
* <td colspan="2" align="center"><b>一括</b><br>

```

[illegible]

```

* <td align="center">-</td>
* <td align="center">O</td>
* <td align="center">-</td>
* <td align="center">O</td>
* <td align="center">-</td>
* <td align="center">O</td>
* <td align="center">-</td>
* <td align="center">O</td>
* <td align="center">-</td>
* <td align="center">O</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

```

```

public class ExpenseDetailManagerImpl extends AbstractWfManager<ExpenseDetailManagerDto> implements
    ExpenseDetailManager {

    /** K30tExpenseDetailService. */
    @Resource
    protected K30tExpenseDetailService k30tExpenseDetailService;

    /** Ka30tExpenseDetailService. */
    @Resource
    protected Ka30tExpenseDetailService ka30tExpenseDetailService;

    /**
     * {@inheritDoc}
     */
    @Override
    public ExpenseDetailManagerDto selectByPk(ExpenseDetailManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k30tExpenseDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectList(ExpenseDetailManagerDto criteria) {
        return super.selectPkUsedUserDataId(k30tExpenseDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectView(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForView(k30tExpenseDetailService, SELECT_SQLFILE_NM, criteria, order);
    }

    /** {@inheritDoc} */
    @Override
    public List<ExpenseDetailManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForViewForCopy(k30tExpenseDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectListForViewForArchive(ka30tExpenseDetailService, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<ExpenseDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.insertBatch(k30tExpenseDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<ExpenseDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.renewBatch(k30tExpenseDetailService, dataList));
    }
}

```

```

/**
 * {@inheritDoc}
 */
@Override
public int remove(ExpenseDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tExpenseDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(ExpenseDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tExpenseDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(CriteriaForSqlDto criteria, List<ExpenseDetailManagerDto> dataList) {
    return super.entryArchive(ka30tExpenseDetailService, criteria, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
    return super.deleteNoUsedUserDataId(k30tExpenseDetailService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAllArchive(CriteriaForSqlDto criteria) {
    return super.deleteNoUsedUserDataIdForArchive(ka30tExpenseDetailService, criteria);
}
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/ExpenseDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 経費明細テーブルマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザーデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstance;
    /** タブルid. */
    public Long tupleId;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバージョン. */

```

```

public String gadgetVariation;
/** 行番号. */
public Long rowNumber;

/** 発生日. */
public Timestamp occurDate;
/** 摘要. */
public String summary;
/** 経費区分コード. */
public String expTypeCd;
/** 勘定科目コード. */
public String accountCd;
/** 税区分コード. */
public String taxTypeCd;
/** 税率. */
public BigDecimal taxRate;
/** 数量. */
public BigDecimal quantity;
/** 取引通貨コード. */
public String transCurrencyCd;
/** 取引通貨額. */
public BigDecimal transAmount;
/** 会社通貨コード. */
public String companyCurrencyCd;
/** 会社通貨換算レート. */
public BigDecimal companyRate;
/** 会社通貨額. */
public BigDecimal companyAmount;
/** 会社通貨税抜額. */
public BigDecimal companyAmountExcTax;
/** 会社通貨税額. */
public BigDecimal companyAmountTax;
/** 会社通貨額合計. */
public BigDecimal companyAmountSum;
/** 支払分類コード. */
public String payTypeCd;
/** 支払方法コード. */
public String payMethodCd;
/** 支払先コード. */
public String payeeCd;
/** 支払先名. */
public String payeeName;
/** 組織セットコード. */
public String departmentSetCd;
/** 組織コード. */
public String departmentCd;
/** プロジェクトコード. */
public String projectCd;

/** 予備項目a. */
public String extensionA;
/** 予備項目b. */
public String extensionB;
/** 予備項目c. */
public String extensionC;
/** 予備項目d. */
public String extensionD;
/** 予備項目e. */
public String extensionE;
/** 予備項目f. */
public String extensionF;
/** 予備項目g. */
public String extensionG;
/** 予備項目h. */
public String extensionH;
/** 予備項目i. */
public String extensionI;
/** 予備項目j. */
public String extensionJ;
/** 登録日時. */
public Timestamp entryTs;
/** 登録者コード. */
public String entryUserCd;
/** 更新カウンタ. */
public Long renewCnt;
/** 更新日時. */
public Timestamp renewTs;
/** 更新者コード. */
public String renewUserCd;

/** 検索基準会社コード. */
public String searchCriteriaCompany;
/** 取引通貨名称. */
public String transCurrencyName;
/** 会社通貨名称. */
public String companyCurrencyName;

```



```

/** 組織名称. */
public String departmentName;
/** プロジェクト名称. */
public String projectName;
/** 経費区分名称. */
public String expTypeName;
/** 勘定科目名称. */
public String accountName;
/** 税区分名称. */
public String taxTypeName;
/** 支払分類名称. */
public String payTypeName;
/** 支払方法名称. */
public String payMethodName;
/** 立替フラグ. */
public String paymentFlag;
}

```

Java (S2JDBC)

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tExpenseDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity; implements Serializable {

/**
 * 経費明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_expense_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tExpenseDetail extends GenerateEntity implements Serializable {

/** シリアルバージョンUID. */
private static final long serialVersionUID = 1L;

/** システム案件id. */
@Id
@Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
public String systemMatterId;

/** ユーザデータid. */
@Id
@Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
public String userDataId;

/** ガジェットクラス. */
@Id
@Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetClass;

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

```

```

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 発生日. */
@Column(name = "occur_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp occurDate;

/** 摘要. */
@Column(name = "summary", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String summary;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

```

```

public String departmentCd;

/** プロジェクトコード。 */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 予備項目a。 */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b。 */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c。 */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d。 */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e。 */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f。 */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g。 */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h。 */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i。 */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j。 */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tExpenseDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)経費明細テーブル。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_expense_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl"})
public final class Ka30tExpenseDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID。 */
    private static final long serialVersionUID = 1L;

    /** システム案件id。 */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid。 */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス。 */

```

```

@Id
@Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetClass;

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** アーカイブ年月. */
@Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
public String archiveMonth;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 発生日. */
@Column(name = "occur_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp occurDate;

/** 摘要. */
@Column(name = "summary", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String summary;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

```

```

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */

```

```

@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tExpenseDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tExpenseDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tExpenseDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tExpenseDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }
}

```

```

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumのプロパティ名を返却.
 *
 * @return rowNumのプロパティ名
 */
public static PropertyName<Long> rowNum() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * occurDateのプロパティ名を返却.
 *
 * @return occurDateのプロパティ名
 */
public static PropertyName<Timestamp> occurDate() {
    return new PropertyName<Timestamp>("occurDate");
}

/**
 * summaryのプロパティ名を返却.
 *
 * @return summaryのプロパティ名
 */
public static PropertyName<String> summary() {
    return new PropertyName<String>("summary");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名

```

```

*/
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

```



```

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */

```

```

public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * extensionKのプロパティ名を返却.

```

```

    entryUserCdのプロパティ名を返却.
    *
    * @return entryUserCdのプロパティ名
    */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30tExpenseDetailNames extends PropertyName<K30tExpenseDetail> {

        /**
         * インスタンスを構築.
         */
        public _K30tExpenseDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30tExpenseDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30tExpenseDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *
         * @return systemMatterIdのプロパティ名
         */
        public PropertyName<String> systemMatterId() {
            return new PropertyName<String>(this, "systemMatterId");
        }

        /**
         * userDataIdのプロパティ名を返却.
         *
         * @return userDataIdのプロパティ名
         */
        public PropertyName<String> userDataId() {
            return new PropertyName<String>(this, "userDataId");
        }

        /**
         * gadgetClassのプロパティ名を返却.
         *
         * @return gadgetClassのプロパティ名

```

```

    @return gadgetClassのプロパティ名
    */
    public PropertyName<String> gadgetClass() {
        return new PropertyName<String>(<this>, "gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
    public PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>(<this>, "gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public PropertyName<Long> tupleId() {
        return new PropertyName<Long>(<this>, "tupleId");
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public PropertyName<String> gadgetId() {
        return new PropertyName<String>(<this>, "gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public PropertyName<String> gadgetVariation() {
        return new PropertyName<String>(<this>, "gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却。
     *
     * @return rowNumのプロパティ名
     */
    public PropertyName<Long> rowNum() {
        return new PropertyName<Long>(<this>, "rowNumber");
    }

    /**
     * occurDateのプロパティ名を返却。
     *
     * @return occurDateのプロパティ名
     */
    public PropertyName<Timestamp> occurDate() {
        return new PropertyName<Timestamp>(<this>, "occurDate");
    }

    /**
     * summaryのプロパティ名を返却。
     *
     * @return summaryのプロパティ名
     */
    public PropertyName<String> summary() {
        return new PropertyName<String>(<this>, "summary");
    }

    /**
     * expTypeCdのプロパティ名を返却。
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(<this>, "expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却。
     *
     * @return accountCdのプロパティ名
     */
    public PropertyName<String> accountCd() {
        return new PropertyName<String>(<this>, "accountCd");
    }
}

```

```

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */

```

```

public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**

```

```

* extensionBのプロパティ名を返却.
*
* @return extensionBのプロパティ名
*/
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
* extensionCのプロパティ名を返却.
*
* @return extensionCのプロパティ名
*/
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
* extensionDのプロパティ名を返却.
*
* @return extensionDのプロパティ名
*/
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
* extensionEのプロパティ名を返却.
*
* @return extensionEのプロパティ名
*/
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
* extensionFのプロパティ名を返却.
*
* @return extensionFのプロパティ名
*/
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
* extensionGのプロパティ名を返却.
*
* @return extensionGのプロパティ名
*/
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
* extensionHのプロパティ名を返却.
*
* @return extensionHのプロパティ名
*/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
* extensionIのプロパティ名を返却.
*
* @return extensionIのプロパティ名
*/
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public PropertyName<Timestamp> entryTs() {

```

```

        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tExpenseDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tExpenseDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tExpenseDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tExpenseDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }
}

```



```

    * gadgetClassのプロパティ名を返却.
    *
    * @return gadgetClassのプロパティ名
    */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却.
     *
     * @return gadgetInstanceのプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却.
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonthのプロパティ名を返却.
     *
     * @return archiveMonthのプロパティ名
     */
    public static PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }

    /**
     * gadgetIdのプロパティ名を返却.
     *
     * @return gadgetIdのプロパティ名
     */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却.
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却.
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * occurDateのプロパティ名を返却.
     *
     * @return occurDateのプロパティ名
     */
    public static PropertyName<Timestamp> occurDate() {
        return new PropertyName<Timestamp>("occurDate");
    }

    /**
     * summaryのプロパティ名を返却.
     *
     * @return summaryのプロパティ名
     */
    public static PropertyName<String> summary() {
        return new PropertyName<String>("summary");
    }

```

```

    return new PropertyName<String>( summary );
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *

```

```

    * @return companyAmountのプロパティ名
    */
    public static PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>("companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>("companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }
}

```

```

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public static PropertyName<String> expTypeName() {
    return new PropertyName<String>("expTypeName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public static PropertyName<String> accountName() {
    return new PropertyName<String>("accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public static PropertyName<String> taxTypeName() {
    return new PropertyName<String>("taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public static PropertyName<String> transCurrencyName() {
    return new PropertyName<String>("transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public static PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>("companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public static PropertyName<String> payTypeName() {
    return new PropertyName<String>("payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public static PropertyName<String> paymentFlag() {
    return new PropertyName<String>("paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public static PropertyName<String> payMethodName() {
    return new PropertyName<String>("payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名

```

```

*/
public static PropertyName<String> departmentName() {
    return new PropertyName<String>("departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public static PropertyName<String> projectName() {
    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}
}

```

```

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _Ka30tExpenseDetailNames extends PropertyName<Ka30tExpenseDetail> {

    /**
     * インスタンスを構築.
     */
    public _Ka30tExpenseDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _Ka30tExpenseDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     */

```

```

    * @param parent 親
    * @param name 名前
    */
    public _Ka30tExpenseDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却。
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>((this, "systemMatterId"));
    }

    /**
     * userDataIdのプロパティ名を返却。
     *
     * @return userDataIdのプロパティ名
     */
    public PropertyName<String> userDataId() {
        return new PropertyName<String>((this, "userDataId"));
    }

    /**
     * gadgetClassのプロパティ名を返却。
     *
     * @return gadgetClassのプロパティ名
     */
    public PropertyName<String> gadgetClass() {
        return new PropertyName<String>((this, "gadgetClass"));
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
    public PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>((this, "gadgetInstance"));
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public PropertyName<Long> tupleId() {
        return new PropertyName<Long>((this, "tupleId"));
    }

    /**
     * localeIdのプロパティ名を返却。
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>((this, "localeId"));
    }

    /**
     * archiveMonthのプロパティ名を返却。
     *
     * @return archiveMonthのプロパティ名
     */
    public PropertyName<String> archiveMonth() {
        return new PropertyName<String>((this, "archiveMonth"));
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public PropertyName<String> gadgetId() {
        return new PropertyName<String>((this, "gadgetId"));
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public PropertyName<String> gadgetVariation() {
        return new PropertyName<String>((this, "gadgetVariation"));
    }

```

```

}

/**
 * rowNumのプロパティ名を返却.
 *
 * @return rowNumのプロパティ名
 */
public PropertyName<Long> rowNum() {
    return new PropertyName<Long>(this, "rowNum");
}

/**
 * occurDateのプロパティ名を返却.
 *
 * @return occurDateのプロパティ名
 */
public PropertyName<Timestamp> occurDate() {
    return new PropertyName<Timestamp>(this, "occurDate");
}

/**
 * summaryのプロパティ名を返却.
 *
 * @return summaryのプロパティ名
 */
public PropertyName<String> summary() {
    return new PropertyName<String>(this, "summary");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *

```



```

    * @return transAmountのプロパティ名
    */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>(this, "payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>(this, "payeeCd");
    }
}

```

```

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(<this>, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(<this>, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(<this>, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(<this>, "projectCd");
}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public PropertyName<String> expTypeName() {
    return new PropertyName<String>(<this>, "expTypeName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public PropertyName<String> accountName() {
    return new PropertyName<String>(<this>, "accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public PropertyName<String> taxTypeName() {
    return new PropertyName<String>(<this>, "taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public PropertyName<String> transCurrencyName() {
    return new PropertyName<String>(<this>, "transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>(<this>, "companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */

```

```

/
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**

```

```

    extensionFのプロパティ名を返却.
    *
    @return extensionFのプロパティ名
    */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {

```

```

    private PropertyNameSetting renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tExpenseDetail;

/**
 * {@link K30tExpenseDetail} のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tExpenseDetailService extends GenerateService<K30tExpenseDetail> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tExpenseDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tExpenseDetail;

/**
 * {@link Ka30tExpenseDetail} のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tExpenseDetailService extends GenerateService<Ka30tExpenseDetail> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30tExpenseDetail} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tExpenseDetailCondition extends
    AbstractEntityCondition<K30tExpenseDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public K30tExpenseDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
}

```

```

public K30tExpenseDetailCondition(String prefix, ComplexWhere where) {
    super(prefix, where);
}

/** systemMatterIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> systemMatterId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("systemMatterId", this);

/** userDataIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> userDataId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("userDataId", this);

/** gadgetClassの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetClass =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetClass", this);

/** gadgetInstanceの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> gadgetInstance =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("gadgetInstance", this);

/** tupleIdの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> tupleId =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("tupleId", this);

/** gadgetIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetVariation =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> rowNumber =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("rowNumber", this);

/** occurDateの条件. */
public NullableCondition<K30tExpenseDetailCondition, Timestamp> occurDate =
    new NullableCondition<K30tExpenseDetailCondition, Timestamp>("occurDate", this);

/** summaryの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> summary =
    new NullableStringCondition<K30tExpenseDetailCondition>("summary", this);

/** expTypeCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> expTypeCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> accountCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> taxTypeCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountExcTax =

```

```

    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountExcTax", this);

    /** companyAmountTaxの条件. */
    public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountTax =
        new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountTax", this);

    /** companyAmountSumの条件. */
    public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountSum =
        new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountSum", this);

    /** payTypeCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payTypeCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payTypeCd", this);

    /** payMethodCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payMethodCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payMethodCd", this);

    /** payeeCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payeeCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payeeCd", this);

    /** payeeNameの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payeeName =
        new NullableStringCondition<K30tExpenseDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> departmentSetCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("departmentSetCd", this);

    /** departmentCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> departmentCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("departmentCd", this);

    /** projectCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> projectCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("projectCd", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionA =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionB =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionC =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionD =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionE =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionF =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionG =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionH =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionI =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionJ =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tExpenseDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tExpenseDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tExpenseDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tExpenseDetailCondition>("entryUserCd", this);

```

```

/** renewCntの条件. */
public NotNullCondition<K30tExpenseDetailCondition, Long> renewCnt =
    new NotNullCondition<K30tExpenseDetailCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30tExpenseDetailCondition, Timestamp> renewTs =
    new NotNullCondition<K30tExpenseDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30tExpenseDetailCondition> renewUserCd =
    new NotNullStringCondition<K30tExpenseDetailCondition>("renewUserCd", this);
}

```

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tExpenseDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullCondition;
import org.seasar.extension.jdbc.where.condition.NotNullStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tExpenseDetail} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tExpenseDetailCondition extends
    AbstractEntityCondition<Ka30tExpenseDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tExpenseDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tExpenseDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> systemMatterId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("systemMatterId", this);

    /** userDataIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> userDataId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("userDataId", this);

    /** gadgetClassの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> gadgetClass =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Long> gadgetInstance =
        new NotNullCondition<Ka30tExpenseDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Long> tupleId =
        new NotNullCondition<Ka30tExpenseDetailCondition, Long>("tupleId", this);

    /** localeIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> localeId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("localeId", this);

    /** archiveMonthの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> archiveMonth =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("archiveMonth", this);

    /** gadgetIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> gadgetId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("gadgetId", this);

    /** gadgetLocationの条件. */

```



```

    /** gadgetVariationの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> gadgetVariation =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, Long> rowNumber =
        new NullableCondition<Ka30tExpenseDetailCondition, Long>("rowNumber", this);

    /** occurDateの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, Timestamp> occurDate =
        new NullableCondition<Ka30tExpenseDetailCondition, Timestamp>("occurDate", this);

    /** summaryの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> summary =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("summary", this);

    /** expTypeCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> expTypeCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("expTypeCd", this);

    /** accountCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> accountCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("accountCd", this);

    /** taxTypeCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> taxTypeCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("taxTypeCd", this);

    /** taxRateの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> taxRate =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("taxRate", this);

    /** quantityの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> quantity =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("quantity", this);

    /** transCurrencyCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> transCurrencyCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("transCurrencyCd", this);

    /** transAmountの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> transAmount =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("transAmount", this);

    /** companyCurrencyCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> companyCurrencyCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("companyCurrencyCd", this);

    /** companyRateの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyRate =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyRate", this);

    /** companyAmountの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmount =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmount", this);

    /** companyAmountExcTaxの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountExcTax =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountExcTax", this);

    /** companyAmountTaxの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountTax =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountTax", this);

    /** companyAmountSumの条件. */
    public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountSum =
        new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountSum", this);

    /** payTypeCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> payTypeCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("payTypeCd", this);

    /** payMethodCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> payMethodCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("payMethodCd", this);

    /** payeeCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> payeeCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("payeeCd", this);

    /** payeeNameの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> payeeName =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<Ka30tExpenseDetailCondition> departmentSetCd =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentSetCd", this);

```

```

new NullableStringCondition<Ka30tExpenseDetailCondition>( departmentCd, this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> projectCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> accountName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> departmentName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentName", this);

/** projectNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> projectName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionA =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionB =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionC =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionD =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionE =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionF =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionG =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionH =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionI =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionI", this);

/** extension.Iの条件. */

```

```

    public NullableStringCondition<Ka30tExpenseDetailCondition> extensionJ =
        new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Timestamp> entryTs =
        new NotNullCondition<Ka30tExpenseDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> entryUserCd =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Long> renewCnt =
        new NotNullCondition<Ka30tExpenseDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Timestamp> renewTs =
        new NotNullCondition<Ka30tExpenseDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> renewUserCd =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.occur_date,
    t.summary,
    t.exp_type_cd,
    t.account_cd,
    t.tax_type_cd,
    t.tax_rate,
    t.quantity,
    t.trans_currency_cd,
    t.trans_amount,
    t.company_currency_cd,
    t.company_rate,
    t.company_amount,
    t.company_amount_exc_tax,
    t.company_amount_tax,
    t.company_amount_sum,
    t.pay_type_cd,
    t.pay_method_cd,
    t.payee_cd,
    t.payee_name,
    t.department_set_cd,
    t.department_cd,
    t.project_cd,
    t.extension_a,
    t.extension_b,
    t.extension_c,
    t.extension_d,
    t.extension_e,
    t.extension_f,
    t.extension_g,
    t.extension_h,
    t.extension_i,
    t.extension_j,
    t.entry_ts,
    t.entry_user_cd,
    t.renew_cnt,
    t.renew_ts,
    t.renew_user_cd,
    c1.search_criteria_company,
    m1.currency_name as trans_currency_name,
    m2.department_name,
    m3.project_name,
    m4.pay_type_name,
    m5.pay_method_name,
    m6.currency_name as company_currency_name,

```

```

m7.exp_type_name,
m8.account_name,
m9.tax_type_name,
m4.payment_flag
from
k30t_expense_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeld*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeld*/ja' = m2.locale_id
and
c1.search_criteria_date >= m2.start_date
and
c1.search_criteria_date < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
c1.search_criteria_date >= m3.start_date
and
c1.search_criteria_date < m3.end_date
and
/*localeld*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeld*/ja' = m4.locale_id
and
c1.search_criteria_date >= m4.start_date
and
c1.search_criteria_date < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeld*/ja' = m5.locale_id
and
c1.search_criteria_date >= m5.start_date
and
c1.search_criteria_date < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6
on
c1.search_criteria_company = m6.company_cd
and
t.company_currency_cd = m6.currency_cd
and
/*localeld*/ja' = m6.locale_id

```

```

and
  m6.delete_flag = '0'
left outer join
  k30m_exp_type m7
on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeId*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeId*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeId*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date
and
  c1.search_criteria_date < m9.end_date
and
  m9.delete_flag = '0'
where
  t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

コピー

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailService/selectViewForCopy.sql

```

select
  t.system_matter_id,
  t.user_data_id,
  t.gadget_class,
  t.gadget_instance,
  t.tuple_id,
  t.gadget_id,
  t.gadget_variation,

```

```

t.row_number,
t.occure_date,
t.summary,
t.exp_type_cd,
t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_currency_cd,
t.company_rate,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
ma.exp_type_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> "" then m8.account_name else mb.account_name end as account_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> "" then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
m4.payment_flag
from
k30t_expense_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
/*searchCriteriaDate*/" >= m2.start_date
and
/*searchCriteriaDate*/" < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd

```

```

and
  t.project_cd = m3.project_cd
and
  /*searchCriteriaDate*/" >= m3.start_date
and
  /*searchCriteriaDate*/" < m3.end_date
and
  /*localeId*/ja' = m3.locale_id
and
  m3.delete_flag = '0'
left outer join
  k30m_pay_type m4
on
  c1.search_criteria_company = m4.company_cd
and
  t.pay_type_cd = m4.pay_type_cd
and
  /*localeId*/ja' = m4.locale_id
and
  /*searchCriteriaDate*/" >= m4.start_date
and
  /*searchCriteriaDate*/" < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeId*/ja' = m5.locale_id
and
  /*searchCriteriaDate*/" >= m5.start_date
and
  /*searchCriteriaDate*/" < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeId*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeId*/ja' = m8.locale_id
and
  /*searchCriteriaDate*/" >= m8.start_date
and
  /*searchCriteriaDate*/" < m8.end_date
and
  m8.delete_flag = '0'
left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeId*/ja' = m9.locale_id
and
  /*searchCriteriaDate*/" >= m9.start_date
and
  /*searchCriteriaDate*/" < m9.end_date
and
  m9.delete_flag = '0'

left outer join
  k30m_exp_type ma
on
  c1.search_criteria_company = ma.company_cd
and
  t.exp_type_cd = ma.exp_type_cd
and
  /*localeId*/ja' = ma.locale_id
and
  /*searchCriteriaDate*/" >= ma.start_date

```

```

and
  /*searchCriteriaDate*/ < ma.end_date
and
  ma.delete_flag = '0'
left outer join
  k30m_account mb
on
  ma.company_cd = mb.company_cd
and
  ma.account_cd = mb.account_cd
and
  /*localeId*/ja' = mb.locale_id
and
  /*searchCriteriaDate*/ >= mb.start_date
and
  /*searchCriteriaDate*/ < mb.end_date
and
  m8.delete_flag = '0'
left outer join
  k30m_tax_type mc
on
  ma.company_cd = mc.company_cd
and
  ma.tax_type_cd = mc.tax_type_cd
and
  /*localeId*/ja' = mc.locale_id
and
  /*searchCriteriaDate*/ >= mc.start_date
and
  /*searchCriteriaDate*/ < mc.end_date
and
  mc.delete_flag = '0'
where
  t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

- **Path**
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/impl/ExpenseDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
  t.system_matter_id as system_matter_id,
  t.user_data_id as user_data_id,
  t.gadget_class as gadget_class,
  t.gadget_instance as gadget_instance,
  t.tuple_id as tuple_id,
  t.gadget_id as gadget_id,
  t.gadget_variation as gadget_variation,
  i1.contents_id as contents_id,
  i1.contents_version_id as contents_version_id,
  null as journal_data_type_cd,
  t.row_number as row_number,
  t.exp_type_cd as exp_type_cd,
  t.account_cd as account_cd,
  t.tax_type_cd as tax_type_cd,
  mt.tax_rate as tax_rate,
  t.quantity as quantity,
  t.trans_currency_cd as trans_currency_cd,
  t.trans_amount as trans_amount,
  t.company_currency_cd as company_currency_cd,
  t.company_rate as company_rate,
  t.company_amount as company_amount,

```



```

t.company_amount_exc_tax as company_amount_exc_tax,
t.company_amount_tax as company_amount_tax,
t.company_amount_sum as company_amount_sum,
t.payee_cd as customer_cd,
t.payee_name as customer_name,
t.department_set_cd as department_set_cd,
t.department_cd as department_cd,
t.project_cd as project_cd,
t.pay_method_cd as pay_method_cd,
t.pay_type_cd as pay_type_cd,
null as adv_matter_number,
null as adv_matter_name,
t.occur_date as detail_date1,
null as detail_date2,
t.summary as detail_summary1,
null as detail_summary2,
null as detail_summary3,
null as detail_summary4,
null as detail_summary5,
null as detail_note,
t.extension_a as detail_extension_a,
t.extension_b as detail_extension_b,
t.extension_c as detail_extension_c,
t.extension_d as detail_extension_d,
t.extension_e as detail_extension_e,
t.extension_f as detail_extension_f,
t.extension_g as detail_extension_g,
t.extension_h as detail_extension_h,
t.extension_i as detail_extension_i,
t.extension_j as detail_extension_j,
'0' as journal_status,
null as wish_date,
c1.search_criteria_company,
c1.search_criteria_date

from
k30t_expense_detail t
inner join
k20c_matter_info i1
on
t.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join
k30m_tax_type mt
on
c1.search_criteria_company = mt.company_cd
and
t.tax_type_cd = mt.tax_type_cd
and
c1.search_criteria_date >= mt.start_date
and
c1.search_criteria_date < mt.end_date
and
i1.locale_id_apply = mt.locale_id
and
mt.delete_flag = '0'

where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.row_number
/*END*/

```

JSP

PC

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/expenseDetail/expenseDetailV01.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wkf" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<%-- expenseDetail v01 --%>
<c:set var="gadgetClass">expenseDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/expenseDetailV01</c:set>
<c:set var="tmtableId">expenseDetailTable</c:set>
<c:set var="tmtableControllerBlockId">tmtableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks['expenseDetailTable']}" />

<%-- tmtable --%>
<c:set var="allowAddDetail" value="{1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="{1' == detail.allowDelFlag}" />

<%-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceOccurDate">${gadget.surfaces["occurDate"]}</c:set>
<c:set var="surfaceSummary">${gadget.surfaces["summary"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>

<%-- Mode --%>
<c:set var="modeOccurDate">${detail.fields["occurDate"].inputType}</c:set>
<c:set var="modeSummary">${detail.fields["summary"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>

<%-- RequiredMark --%>
<c:set var="reqOccurDate" >${detail.fields["occurDate"].requiredFlag}</c:set>
<c:set var="reqSummary" >${detail.fields["summary"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validOccurDate_def">{k_date:[""]}</c:set>
<c:set var="validOccurDate_apply">{"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>
<c:set var="validOccurDate_tempsave"></c:set>
<c:set var="validOccurDate_reapply">{"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>
<c:set var="validOccurDate_process">{"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>

```

```

<c:set var="validSummary_def">{k_maxLength:['250']}</c:set>
<c:set var="validSummary_apply">{{"true" == reqSummary ? "k_required:[']' : ''}}</c:set>
<c:set var="validSummary_tempsave"></c:set>
<c:set var="validSummary_reapply">{{"true" == reqSummary ? "k_required:[']' : ''}}</c:set>
<c:set var="validSummary_process">{{"true" == reqSummary ? "k_required:[']' : ''}}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:['expTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validExpTypeCd_apply">{{"true" == reqExpTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">{{"true" == reqExpTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validExpTypeCd_process">{{"true" == reqExpTypeCd ? "k_required:[']' : ''}}</c:set>

<c:set var="validAccountCd_def">{k_accountExists:['accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">{{"true" == reqAccountCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">{{"true" == reqAccountCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validAccountCd_process">{{"true" == reqAccountCd ? "k_required:[']' : ''}}</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:['taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">{{"true" == reqTaxTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{{"true" == reqTaxTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validTaxType_process">{{"true" == reqTaxTypeCd ? "k_required:[']' : ''}}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:['payTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">{{"true" == reqPayTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPayTypeCd_reapply">{{"true" == reqPayTypeCd ? "k_required:[']' : ''}}</c:set>
<c:set var="validPayTypeCd_process">{{"true" == reqPayTypeCd ? "k_required:[']' : ''}}</c:set>

<c:set var="validPayeeName_def">{k_maxLength:['250']}</c:set>
<c:set var="validPayeeName_apply">{{"true" == reqPayeeName ? "k_required:[']' : ''}}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">{{"true" == reqPayeeName ? "k_required:[']' : ''}}</c:set>
<c:set var="validPayeeName_process">{{"true" == reqPayeeName ? "k_required:[']' : ''}}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:['searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:['projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<k:gadgetContainer gadgetClass="{gadgetClass}"
    gadgetVariation="{gadgetVariation}"
    gadgetInstance="{gadgetInstance}"
    gadgetTitle="{surfaceChapterTitle}"
    gadgetId="{gadgetId}"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:ttable id="{ttableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumberTarget="rowNumber">
    <colgroup>
        <col style="width:42px;overflow:hidden">
        <col style="width:230px;overflow:hidden">
        <col style="width:185px;overflow:hidden">
        <col style="width:185px;overflow:hidden">
        <col style="width:185px;overflow:hidden">
    </colgroup>
    <thead>
        <tr>
            <td rowspan="3" style="vertical-align:middle;">
                <k:ttableOperationPanel
                    position="header"
                    allowAdd="{allowAddDetail}"
                    allowDel="{allowDelDetail}"
                />
            </td>
            <th>
                <k:surface
                    name="occurDateSurface"
                    surfaceKey="{surfaceOccurDate}"
                    requiredMark="{reqOccurDate}"
                />
            </th>
            <th colspan="3">
                <k:surface
                    name="summarySurface"
                    surfaceKey="{surfaceSummary}"
                    requiredMark="{reqSummary}"
                />
            </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td colspan="4">
                <k:surface
                    name="summarySurface"
                    surfaceKey="{surfaceSummary}"
                    requiredMark="{reqSummary}"
                />
            </td>
        </tr>
    </tbody>
</table>

```

```

</th>
</tr>
<tr>
<th colspan="2">
<k:surface
  name="expTypeSurface"
  surfaceKey="{surfaceExpType}"
  requiredMark="{reqExpTypeCd}"
  style="display:inline;"
/>
<k:surface
  name="accountSurface"
  surfaceKey="{surfaceAccount}"
  style="display:inline;"
/>
<k:surface
  name="taxTypeSurface"
  surfaceKey="{surfaceTaxType}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="departmentSurface"
  surfaceKey="{surfaceDepartment}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="projectSurface"
  surfaceKey="{surfaceProject}"
/>
</th>
</tr>
<tr>
<th>
<k:surface
  name="payTypeSurface"
  surfaceKey="{surfacePayType}"
  requiredMark="{reqPayTypeCd}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="payeeSurface"
  surfaceKey="{surfacePayee}"
  requiredMark="{reqPayeeName}"
  style="display:inline;"
/>
</th>
<th colspan="2">
<k:surface
  name="transAmountSurface"
  surfaceKey="{surfaceTransAmount}"
  requiredMark="{reqTransAmount}"
/>
</th>
</tr>
</thead>
<tbody>
<tr>
<td rowspan="3">
<k:tmtableOperationPanel
  position="detail"
  allowAdd="{allowAddDetail}"
  allowDel="{allowDelDetail}"
  name="rowNumber"
/>
</td>
<td>
<k:date
  name="occurDate"
  mode="{modeOccurDate}"
  surfaceKey="{surfaceOccurDate}"
  validationDefault="{validOccurDate_def}"
  validationApply="{validOccurDate_apply}"
  validationTempsave="{validOccurDate_tempsave}"
  validationReapply="{validOccurDate_reapply}"
  validationProcess="{validOccurDate_process}"
/>
</td>
<td colspan="3">
<k:text
  name="summary"
  mode="{modeSummary}"
/>

```

```

mode="{modeSummary}"
imeMode="on"
surfaceKey="{surfaceSummary}"
style="width:545px;"
maxlength="250"
validationDefault="{validSummary_def}"
validationApply="{validSummary_apply}"
validationTempsave="{validSummary_tempsave}"
validationReapply="{validSummary_reapply}"
validationProcess="{validSummary_process}"
/>
</td>
<tr>
<td colspan="2">
<k:hidden
name="expTypeCd"
mode="{modeExpTypeCd}"
/>
<kmsw:expType
name="expTypeName"
mode="{modeExpTypeCd}"
imeMode="on"
surfaceKey="{surfaceExpType}"
resultValue="expTypeName"
resultChain="{expTypeCd:expTypeCd, accountCd:accountCd, accountName:accountName, taxTypeCd:taxTypeCd, taxTypeName:taxTypeName}"
primaryTarget="expTypeCd"
searchCriteriaDate="searchCriteriaDate"
searchCriteriaCompany="searchCriteriaCompany"
searchCriteriaExpTypeCd="expTypeCd"
style="width:150px;"
validationDefault="{validExpTypeCd_def}"
validationApply="{validExpTypeCd_apply}"
validationTempsave="{validExpTypeCd_tempsave}"
validationReapply="{validExpTypeCd_reapply}"
validationProcess="{validExpTypeCd_process}"
/>
<k:hidden
name="accountCd"
mode="{modeAccountCd}"
surfaceKey="{surfaceAccount}"
validationDefault="{validAccountCd_def}"
validationApply="{validAccountCd_apply}"
validationTempsave="{validAccountCd_tempsave}"
validationReapply="{validAccountCd_reapply}"
validationProcess="{validAccountCd_process}"
/>
<k:text
name="accountName"
mode="1"
class="kaiden-ui-mst-search-chain-item"
style="width:115px;"
/>
<k:hidden
name="taxTypeCd"
mode="{modeTaxTypeCd}"
surfaceKey="{surfaceTaxType}"
validationDefault="{validTaxType_def}"
validationApply="{validTaxType_apply}"
validationTempsave="{validTaxType_tempsave}"
validationReapply="{validTaxType_reapply}"
validationProcess="{validTaxType_process}"
/>
<k:text
name="taxTypeName"
mode="1"
class="kaiden-ui-mst-search-chain-item"
style="width:115px;"
/>
</td>
<td>
<k:hidden
name="departmentSetCd"
mode="{modeDepartmentCd}"
/>
<k:hidden
name="departmentCd"
mode="{modeDepartmentCd}"
surfaceKey="{surfaceDepartment}"
/>
<kmsb:department
name="departmentName"
mode="{modeDepartmentCd}"
imeMode="on"
surfaceKey="{surfaceDepartment}"
resultValue="departmentName"
resultChain="{departmentSetCd:departmentSetCd, departmentCd:departmentCd}"
primaryTarget="departmentCd:departmentSetCd"

```

```

        primaryTarget="departmentCd,departmentSetCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaDepartmentCd="departmentCd"
        searchCriteriaDepartmentSetCd="departmentSetCd"
        style="width:150px;"
        validationDefault="{validDepartmentCd_def}"
        validationApply="{validDepartmentCd_apply}"
        validationTempsave="{validDepartmentCd_tempsave}"
        validationReapply="{validDepartmentCd_reapply}"
        validationProcess="{validDepartmentCd_process}"
    />
</td>
<td>
    <k:hidden
        name="projectCd"
        mode="{modeProjectCd}"
        surfaceKey="{surfaceProject}"
    />
    <kmsw:project
        name="projectName"
        mode="{modeProjectCd}"
        imeMode="on"
        surfaceKey="{surfaceProject}"
        resultValue="projectName"
        resultChain="{projectCd:'projectCd'}"
        primaryTarget="projectCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaProjectCd="projectCd"
        style="width:150px;"
        validationDefault="{validProjectCd_def}"
        validationApply="{validProjectCd_apply}"
        validationTempsave="{validProjectCd_tempsave}"
        validationReapply="{validProjectCd_reapply}"
        validationProcess="{validProjectCd_process}"
    />
</td>
</tr>
<tr>
<td>
    <klist_wkf:listDataPayType
        blank="false"
        name="payType"
        companyCd="{param.searchCriteriaCompany}"
        searchCriteriaDate="{param.searchCriteriaDate}"
    />
    <k:select
        name="payTypeCd"
        mode="{modePayTypeCd}"
        surfaceKey="{surfacePayType}"
        viewModeName="payTypeName"
        dataName="payType"
        selectChain="{paymentFlag:'paymentFlag'}"
        style="width:230px;"
        validationDefault="{validPayTypeCd_def}"
        validationApply="{validPayTypeCd_apply}"
        validationTempsave="{validPayTypeCd_tempsave}"
        validationReapply="{validPayTypeCd_reapply}"
        validationProcess="{validPayTypeCd_process}"
    />
    <k:hidden
        name="paymentFlag"
        mode="{modePayTypeCd}"
    />
</td>
<td>
    <k:text
        name="payeeName"
        mode="{modePayeeName}"
        imeMode="on"
        surfaceKey="{surfacePayee}"
        style="width:160px;"
        maxlength="250"
        validationDefault="{validPayeeName_def}"
        validationApply="{validPayeeName_apply}"
        validationTempsave="{validPayeeName_tempsave}"
        validationReapply="{validPayeeName_reapply}"
        validationProcess="{validPayeeName_process}"
    />
</td>
<td colspan="2">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericAmountInputParts.jsp">
    <jsp:param name="gadgetId" value="{gadgetId}" />
    <jsp:param name="tmtableId" value="{tmtableId}" />
    <jsp:param name="searchCriteriaCompany" value="{searchCriteriaCompany}" />
    </jsp:include>

```

```

<jsp:param name="modeTransAmnt" value="\${modeTransAmount}" />
<jsp:param name="surfaceTransAmnt" value="\${surfaceTransAmount}" />
<jsp:param name="requiredTransAmntApply" value="\${reqTransAmount}" />
<jsp:param name="requiredTransAmntReapply" value="\${reqTransAmount}" />
<jsp:param name="requiredTransAmntProcess" value="\${reqTransAmount}" />
</jsp:include>
</td>
</tr>
</tbody>
</k:tmtable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {

var eventMan = KAIDEN.getGadget("\${gadgetId}").eventManager
, tmt = KAIDEN.getGadget("\${gadgetId}").tmtables["expenseDetailTable"]
, tma = tmt.tmtableAccessor;

/*****
 * 内部イベント
 *****/

//明細行:行挿入(データ設定前)
tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

//デフォルト部門を要求
var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "\${gadgetId}");
defaultDept = defaultDept[0].result; //先頭行のみ利用
tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
tma("departmentName", tupleId).setValue(defaultDept.departmentName);

//デフォルトプロジェクトを要求
var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "\${gadgetId}");
defaultPrj = defaultPrj[0].result; //先頭行のみ利用
tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
tma("projectName", tupleId).setValue(defaultPrj.projectName);

});

//明細行:行挿入(データ設定後)
tmt.addEventPostNewRow(function(tupleId, tma, data) {
//マスタ検索部品の整合性確認。
KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

//初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
if (tma("payTypeCd", tupleId).elem(true).length > 0) {
KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
}

//事前申請書取込では、会社通貨額が設定されているため、取引通貨額に会社通貨額を設定する
//※外貨版から標準版にコピーする際、会社通貨額(レート換算後)の金額を設定する
var compAmnt = tma("companyAmount", tupleId).getValue();
if (compAmnt) {
tma("transAmount", tupleId).setValue(compAmnt);
tma("companyAmount", tupleId).setValue("");
}
});

/*****
 * 受信イベント(push)
 *****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
tmt.deleteAll();
return KAIDEN.ImwModerator.genericLoadDataEvent("\${gadgetId}", "\${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
return KAIDEN.ImwModerator.genericLoadData4CopyEvent("\${gadgetId}", "\${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
return KAIDEN.ImwModerator.genericLoadData4SettleEvent("\${gadgetId}", "\${accessUrl}", arg);
});
});

```

```

/*****
 * 受信イベント (pull)
 *****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
  return tmt.getColumnValues(["occurDate", "occurDate"]);
});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["transAmount"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
  return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["transAmount", "paymentFlag"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value[0]) && value[1] == "1") {
      result = result.add(new BigDecimal(value[0]));
    }
  });
  return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["transAmount", "paymentFlag"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value[0]) && value[1] != "1") {
      result = result.add(new BigDecimal(value[0]));
    }
  });
  return result.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
  return tmt.getColumnValues(["transAmount"]);
});

//claimExpenccCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenccCount", function(**Object*/arg) {
  return tmt.getCount();
});
});
</script>

```

SP

- Path
src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/expenseDetail/expenseDetailV01_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- expenseDetail v01 -->
<c:set var="gadgetClass">expenseDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">expenseDetailV01Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserMatterId">${param.imwUserMatterId}</c:set>

```



```

<:set var="imwUserData" <param:userData>/></set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">expenseDetailTable</c:set>

<c:set var="gadgetResponse" value="{kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>
<c:set var="gadget" value="{appDto.gadgets[gadgetId]}" />

<!-- Surface -->
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceOccurDate">${gadget.surfaces["occurDate"]}</c:set>
<c:set var="surfaceSummary">${gadget.surfaces["summary"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconp
<h3>
  <span>${surfaceChapterTitle}</span>
  <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="{fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>

<ul data-role="listview" data-divider-theme="b">
  <c:forEach items="{gadgetResponse.data[tmtableId]}" var="detailData">
    <li>
      <span class="ui-li-heading"><c:out value="{detailData.occurDate}"></c:out></span>
      <span class="ui-li-aside"><c:out value="{detailData.companyAmount[1]}"></c:out></span>
      <div class="indent">
        <p class="kaiden-ui-overview-contents"><c:out value="{detailData.summary}"></c:out></p>
        <p class="kaiden-ui-overview-contents-sub"><c:out value="{detailData.payTypeName}"></c:out></p>
      </div>
      <li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="{surfaceBack}"></c:out></a></li>
      <li>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceOccurDate}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.occurDate}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceSummary}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.summary}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceTransAmount}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><strong><c:out value="{detailData.transAmount[1]}"></c:out></strong></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceExpType}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.expTypeName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceAccount}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.accountName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceTaxType}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.taxTypeName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfacePayType}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.payTypeName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfacePayee}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="{detailData.payeeName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceDepartment}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.departmentName}"></c:out></label>
        </div>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceProject}"></c:out></p>
        <div class="indent">
          <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.projectName}"></c:out></label>
        </div>
      </li>
    </ul>
  </li>
</c:forEach>
</ul>
</div>

```


経費明細 (複数通貨)

経費明細 (複数通貨) のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   │   ├── K30tExpenseDetail.java
│   │   │   │   │   │   │   │   │   │   ├── K30tExpenseDetailNames.java
│   │   │   │   │   │   │   │   │   │   ├── Ka30tExpenseDetail.java
│   │   │   │   │   │   │   │   │   │   ├── Ka30tExpenseDetailNames.java
│   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │       ├── K30tExpenseDetailCondition.java
│   │   │   │   │   │   │   │       ├── K30tExpenseDetailService.java
│   │   │   │   │   │   │   │       ├── Ka30tExpenseDetailCondition.java
│   │   │   │   │   │   │   │       └── Ka30tExpenseDetailService.java
│   │   ├── java
│   │   │   ├── jp
│   │   │   │   ├── co
│   │   │   │   │   ├── slcs
│   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── AbstractExpenseDetailActionBase.java
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV02Action.java
│   │   │   │   │   │   │   │   │   ├── conf
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailFieldName.java
│   │   │   │   │   │   │   │   │   ├── form
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV02Form.java
│   │   │   │   │   │   │   │   │   ├── logic
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   ├── AbstractExpenseDetailLogicBase.java
│   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailV02Logic.java
│   │   │   │   │   │   │   │   └── foundation
│   │   │   │   │   │   │   │       ├── manager
│   │   │   │   │   │   │   │       │   ├── expensedetail
│   │   │   │   │   │   │   │       │   │   ├── ExpenseDetailManager.java
│   │   │   │   │   │   │   │       │   │   ├── ExpenseDetailManagerDto.java
│   │   │   │   │   │   │   │       │   └── impl
│   │   │   │   │   │   │   │       │       └── ExpenseDetailManagerImpl.java
│   ├── resources
│   │   ├── META-INF
│   │   │   ├── sql
│   │   │   │   ├── jp
│   │   │   │   │   ├── co
│   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │   │   │   │   │   ├── expensedetail
│   │   │   │   │   │   │   │   │   │   │   │   ├── impl
│   │   │   │   │   │   │   │   │   │   │   │   └── ExpenseDetailManagerImpl
│   │   │   │   │   │   │   │   │   │   │   │       selectForJournalSourceDtl.sql
│   │   │   │   │   │   │   │   │   │   └── model
│   │   │   │   │   │   │   │   │   │       ├── service
│   │   │   │   │   │   │   │   │   │       │   ├── K30tExpenseDetailService
│   │   │   │   │   │   │   │   │   │       │   │   ├── selectView.sql
│   │   │   │   │   │   │   │   │   │       │   │   └── selectViewForCopy.sql
│   ├── webapp
│   │   ├── WEB-INF
│   │   │   ├── view
│   │   │   │   ├── kaiden2
│   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │   ├── expenseDetail
│   │   │   │   │   │   │   │   ├── expenseDetailV02.jsp
│   │   │   │   │   │   │   │   └── expenseDetailV02_sp.jsp

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractExpenseDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 経費明細がジェット 基底Actionの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractExpenseDetailActionBase implements GadgetActionIF {

    /**
     * 申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }
}

```

```

/**
 * 確認画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConvertor());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    return null;
}

```

```

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #ExpenseDetailConverter 経費明細(標準)ガジェット}の書式変換を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected ConvertForJsonSource getConverter() {
    return new ExpenseDetailConverter(getForm());
}

/**
 * 経費明細ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class ExpenseDetailConverter extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;
    /**
     * NumberHelper.
     */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public ExpenseDetailConverter(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     * <tr>
     * <td>換算額</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * </table>
     * </p>
     */
    @Override
    protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

        Map<String, Object> result = new HashMap<String, Object>(map);

        // 取引通貨額の書式変換
        if (result.containsKey(EXPENSE_DETAIL_TRANS_AMOUNT)) {
            BigDecimal amount = (BigDecimal) result.get(EXPENSE_DETAIL_TRANS_AMOUNT);
            Object currencyCd = result.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD);
            result.put(EXPENSE_DETAIL_TRANS_AMOUNT,

```

```

        numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (result.containsKey(EXPENSE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(EXPENSE_DETAIL_COMPANY_RATE);
        result.put(EXPENSE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }

    // 換算額の書式変換
    if (result.containsKey(EXPENSE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(EXPENSE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = result.get(EXPENSE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>換算額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fidNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fidNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(EXPENSE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(EXPENSE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(EXPENSE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(EXPENSE_DETAIL_COMPANY_RATE);
        result.put(EXPENSE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }

    // 換算額の書式変換
    if (map.containsKey(EXPENSE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(EXPENSE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = map.get(EXPENSE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(EXPENSE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}
}
}

```

Action

Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/ExpenseDetailV02Action.java


```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.ExpenseDetailV02Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail.ExpenseDetailV02Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 経費明細(標準・外貨)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailV02Action extends AbstractExpenseDetailActionBase {

    /**
     * ExpenseDetailV02Logic.
     */
    @Resource
    public ExpenseDetailV02Logic expenseDetailV02Logic;

    /**
     * ExpenseDetailV02Form.
     */
    @ActionForm
    @Resource
    public ExpenseDetailV02Form expenseDetailV02Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return expenseDetailV02Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return expenseDetailV02Form;
    }
}

```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/ExpenseDetailV02Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 経費明細(標準・外貨)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailV02Form extends AbstractImwPageForm {

}

```

Logic(基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/expensedetail/AbstractExpenseDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_OCCUR_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_SUMMARY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;

```

```

import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細ガジェット用Logicの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractExpenseDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 経費明細ガジェット。 */
    protected static final String BLOCK_KEY_EXPENSE_DETAIL_TABLE = "expenseDetailTable";

    /** 経費内容。 */
    public static final int SUMMARY_LEN = 250;
    /** 税率(整数部)。 */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部)。 */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部)。 */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部)。 */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先。 */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部)。 */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部)。 */
    public static final int QUANTITY_DEC_LEN = 10;

    /** ExpenseDetailManager。 */
    @Resource
    protected ExpenseDetailManager expenseDetailManager;

    /** JournalSourceDtlManager。 */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper。 */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** コンストラクタ。 */
    public AbstractExpenseDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 経費明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる経費明細テーブルを削除後に、経費明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 経費明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }
}

```

```

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
 * </p>
 */
@Override
public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
 * </p>
 */
@Override
public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 経費明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}

```

```

    * {@inheritDoc}
    * <p>
    * 本機能では何も処理を行わず、常に0を返却します。
    * </p>
    */
    @Override
    public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 直前のユーザデータIDが設定されている場合に、経費明細テーブルを登録します。
    * </p>
    */
    @Override
    public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 経費明細を登録します。
    * </p>
    */
    @Override
    public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
    * {@inheritDoc}
    * <p>
    * 経費明細テーブルの対象データを削除します。<br>
    * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
    * </p>
    */
    @Override
    public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTable(true);
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる経費明細テーブルを削除後に、経費明細テーブルを登録します。<br>
    * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
    * </p>
    */
    @Override
    public int tempSaveInsert(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

```

```

public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了],[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * </p>
 */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.inwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 経費明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userDataId = pluginGadgetDTO.userDataId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 経費明細データを取得
            List<BeanMap> beanMapList = expenseDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる経費明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる(過去案件)経費明細テーブルを一括削除します。
 * </p>
 */
@Override

```

```

    public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = expenseDetailManager
            .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto()).imwPluginParameterDto));
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる経費明細の履歴を含めた全てのデータを削除します。
     * </p>
     */
    @Override
    public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
     * {@inheritDoc}
     *
     * @see #viewMapFormatted
     */
    @Override
    protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
        return viewMapFormatted(conditionDto, null);
    }

    /**
     * {@inheritDoc}
     * <p>
     * データの取得は、{@link ExpenseDetailManager#selectList(ExpenseDetailManagerDto)}を利用します。
     * </p>
     */
    @Override
    protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
        throws KaidenSystemException {
        ExpenseDetailManagerDto dto = Beans.createAndCopy(ExpenseDetailManagerDto.class, conditionDto).execute();
        List<ExpenseDetailManagerDto> listSelect = expenseDetailManager.selectList(dto);

        LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
        List<Map<String, Object>> lstDtoMap = new ArrayList<Map<String, Object>>();
        for (ExpenseDetailManagerDto src : listSelect) {
            if (converter == null) {
                lstDtoMap.add(BeanUtil.toMap(src, true));
            } else {
                lstDtoMap.add(converter.mapConvert(BeanUtil.toMap(src, true)));
            }
        }

        rv.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, lstDtoMap);
        return rv;
    }

    /**
     * 経費明細テーブルデータ登録処理。
     *
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int insertTable() throws KaidenSystemException {
        // 経費明細ガジェットデータをMap形式で取得
        List<ExpenseDetailManagerDto> dtos = JsonUtil.getDtos(ExpenseDetailManagerDto.class,
            getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE));

        // 経費明細テーブル登録
        return expenseDetailManager.entry(dtos);
    }

    /**
     * 仕訳連携用明細テーブルデータ登録処理。
     *
     * @param srcList 経費明細テーブルデータリスト
     * @param matterEndProcessDto 案件終了処理用DTO
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
        MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
        for (JournalSourceDtlManagerDto srcDto : srcList) {
            // 仕訳データ区分をセット
            srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
        }

        // 仕訳連携用明細テーブル登録

```

```

    return journalSourceDtlManager.entry(srcList);
}

/**
 * 経費明細テーブルデータ削除処理.
 *
 * @param tempFlg 一時保存データフラグ
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    ExpenseDetailManagerDto mngDto = new ExpenseDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = super.getGadgetDto().userDataId;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    return expenseDetailManager.remove(mngDto);
}

/**
 * 経費明細テーブルデータ削除処理(履歴を含む).
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    ExpenseDetailManagerDto mngDto = new ExpenseDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    return expenseDetailManager.removeAll(mngDto);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */

```



```

@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>

```

```

* </tr>
* <tr>
* <td>発生日</td>
* <td>{@link DateValidator 日付型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>摘要</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>経費区分コード</td>
* <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>取引通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>取引通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td rowspan="2">会社通貨換算レート</td>
* <td>{@link DigitsValidator 小数量型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数量型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>

```

```

* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
KaidenApplicationException {

    if (fieldDefsMap.containsKey(BLOCK_KEY_EXPENSE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_EXPENSE_DETAIL_TABLE, tmpFlg);
        // 発生日
        addValidateOnlyInput(EXPENSE_DETAIL_OCCUR_DATE, VALID_DATE);
        // 摘要
        addValidateOnlyInput(EXPENSE_DETAIL_SUMMARY, VALID_MAX_LENGTH, getArgs(SUMMARY_LEN));
        // 経費区分コード
        addValidate(EXPENSE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, EXPENSE_DETAIL_EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidate(EXPENSE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, EXPENSE_DETAIL_ACCOUNT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidate(EXPENSE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, EXPENSE_DETAIL_TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 取引通貨コード
        addValidate(EXPENSE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, EXPENSE_DETAIL_TRANS_CURRENCY_CD,
            getGadgetDto().searchCriteriaCampany);
        // 会社通貨換算レート
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_RATE, VALID_DIGITS,
            getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨コード
        addValidate(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS,
            EXPENSE_DETAIL_COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 支払分類コード
        addValidate(EXPENSE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, EXPENSE_DETAIL_PAY_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidate(EXPENSE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, EXPENSE_DETAIL_PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払先名
        addValidateOnlyInput(EXPENSE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
        // 組織コード
        addValidate(EXPENSE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCampany,
            EXPENSE_DETAIL_DEPARTMENT_SET_CD, EXPENSE_DETAIL_DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidate(EXPENSE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, EXPENSE_DETAIL_PROJECT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 数量
        addValidateOnlyInput(EXPENSE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
        // 会社通貨額合計
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
    }
}

/**
 * ブロックの共通シングルバリデート定義MAPにエンリを追加.
 *
 * @param fieldKey フィールドキー
 * @param validationId バリデートキー
 * @param args バリデート実行時アーギュメント
 */
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * ブロックの共通シングルバリデート定義MAPにエンリを追加.
 *
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 *

```

```

* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
* {@inheritDoc}
* <p>
* システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
* 設定されている場合は、{@link ExpenseDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, new ArrayList<ExpenseDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectView(criteria));
    }
    return result;
}

/**
* {@inheritDoc}
* <p>
* {@link ExpenseDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
* {@inheritDoc}
* <p>
* {@link ExpenseDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
* </p>
*/
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_EXPENSE_DETAIL_TABLE, expenseDetailManager.selectViewArchive(criteria));
    return result;
}

/**
* {@inheritDoc}
* <p>
* 取得した経費明細のデータに対して、以下マスタの整合性チェックを行います。
* <ul>
* <li>[*1]通貨マスタ(imm_currency)</li>
* <li>組織マスタ(imm_depaertment)</li>
* <li>プロジェクトマスタ</li>
* <li>支払分類マスタ</li>
* <li>支払方法マスタ</li>
* <li>経費区分マスタ</li>
* <li>勘定科目マスタ</li>
* <li>税区分マスタ</li>
* </ul>
* *1: 取引通貨と会社通貨
* </p>
*/
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {

    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_EXPENSE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<ExpenseDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), ExpenseDetailManagerDto.class);

            for (ExpenseDetailManagerDto j : sqlResultList) {
                // 取引通貨チェック
                if (!isEmpty(j.transCurrencyCd) && isEmpty(j.transCurrencyName)) {
                    String[] fieldId = getArgs(EXPENSE_DETAIL_TRANS_CURRENCY_CD, EXPENSE_DETAIL_TRANS_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),

```

```

        BaseProp.S.CURRENCY);
String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
    masterName, values, message));
}

// 会社通貨チェック
if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_COMPANY_CURRENCY_CD,
        EXPENSE_DETAIL_COMPANY_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 組織チェック
if (isNotEmpty(j.departmentSetCd) && isNotEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_DEPARTMENT_SET_CD, EXPENSE_DETAIL_DEPARTMENT_CD,
        EXPENSE_DETAIL_DEPARTMENT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.DEPARTMENT);
    String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// プロジェクトチェック
if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PROJECT_CD, EXPENSE_DETAIL_PROJECT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PROJECT);
    String[] values = getArgs(j.projectCd, j.projectName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 支払分類マスタチェック
if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PAY_TYPE_CD, EXPENSE_DETAIL_PAY_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_TYPE);
    String[] values = getArgs(j.payTypeCd, j.payTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 支払方法マスタチェック
if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_PAY_METHOD_CD, EXPENSE_DETAIL_PAY_METHOD_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_METHOD);
    String[] values = getArgs(j.payMethodCd, j.payMethodName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

// 経費区分マスタチェック
if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
    String[] fieldId = getArgs(EXPENSE_DETAIL_EXP_TYPE_CD, EXPENSE_DETAIL_EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(j.expTypeCd, j.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        j.accountCd = null;
        j.accountName = null;
        j.taxTypeCd = null;
        j.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }
    result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
        masterName, values, message));
}

```

```

    }

    // 勘定科目マスタチェック
    if (isEmpty(j.accountCd) && isEmpty(j.accountName)) {
        String[] fieldId = getArgs(EXPENSE_DETAIL_ACCOUNT_CD, EXPENSE_DETAIL_ACCOUNT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.ACCOUNT);
        String[] values = getArgs(j.accountCd, j.accountName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
            masterName, values, message));
    }

    // 税区分マスタチェック
    if (isEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
        String[] fieldId = getArgs(EXPENSE_DETAIL_TAX_TYPE_CD, EXPENSE_DETAIL_TAX_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.TAX_TYPE);
        String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
            // 税率、税抜金額、税額をクリア
            j.taxRate = null;
            j.companyAmountExcTax = null;
            j.companyAmountTax = null;
        }

        result.add(new ErrorDetailDto(BLOCK_KEY_EXPENSE_DETAIL_TABLE, fieldId, j.tupleId.toString(),
            masterName, values, message));
    }
}

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Deprecated
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();
    list.add(SYSTEM_MATTER_ID);
    list.add(USER_DATA_ID);
    list.add(GADGET_CLASS);
    list.add(GADGET_INSTANCE);
    return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccCountIF#getClaimExpenccCount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccCount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccCountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.

```

```

    の値を設定。
    *
    * @param params アーギュメント
    * @return 会社支払額
    */

public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(EXPENSE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */

public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(EXPENSE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 費用明細
 */

public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            list.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定。
 *
 * @param params アーギュメント
 * @return 費用合計
 */

public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

```

```

BigDecimal sum = BigDecimal.ZERO;

for (Map<String, Object> j : blocjData) {
    String strAmnt = StringUtil.toString(j.get(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM));
    if (StringUtil.isEmpty(strAmnt)) {
        sum = sum.add(new BigDecimal(strAmnt));
    }
}

retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 経費明細テーブルの対象データを過去案件退避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    @SuppressWarnings("unchecked")
    List<ExpenseDetailManagerDto> list = (List<ExpenseDetailManagerDto>) dataMap
        .get(BLOCK_KEY_EXPENSE_DETAIL_TABLE);
    int rv = expenseDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(expenseDetailManager);
    return list;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/expensedetail/ExpenseDetailV02Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.expensedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail.ExpenseDetailFieldName.EXPENSE_DETAIL_TRANS_CURRENCY_CD;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

```



```

/*
 * 経費明細(標準・外貨)ガジェット(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */

public class ExpenseDetailV02Logic extends AbstractExpenseDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenCountIF {

    /** コンストラクタ. */
    public ExpenseDetailV02Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します。<br>
     * <br>
     * <p>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力/表示</td>
     * </tr>
     * </table>
     * <p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        super.defineValidateForNormal(params, tmpFlg);

        // 取引通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨額
        addValidateOnlyInput(EXPENSE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

        if (!tmpFlg) {
            // 組織コード
            addValidateOnlyInput(EXPENSE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, EXPENSE_DETAIL_DEPARTMENT_CD,
                EXPENSE_DETAIL_PROJECT_CD);
            // プロジェクトコード
            addValidateOnlyInput(EXPENSE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, EXPENSE_DETAIL_DEPARTMENT_CD,
                EXPENSE_DETAIL_PROJECT_CD);
        }
    }

    /**
     * {@inheritDoc}
     * <p>
     * 下記補完を行います。
     * <p>
     * <ul>
     * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
     * <li>数量: 固定で1を設定</li>
     * <li>立替フラグ: 支払分類コードに紐づく立替フラグを支払分類マスタから取得して設定</li>
     * </ul>
     */
    @Override
    protected void complementJsonData(GadgetMediatorParams params) {
        List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

        String companyCurrencyCd = null;

        if (CollectionUtil.isEmpty(blockDat)) {
            for (Map<String, Object> i : blockDat) {

```

```

    for (Map<String, Object> j : blockDat) {
        // 会社通貨コード
        if (companyCurrencyCd == null) {
            companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
        }
        j.put(EXPENSE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

        // 数量
        j.put(EXPENSE_DETAIL_QUANTITY, "1");

        // 立替フラグ
        if (StringUtil.isEmpty((String) j.get(EXPENSE_DETAIL_PAY_TYPE_CD))) {
            PayTypeManagerDto criteria = new PayTypeManagerDto();
            criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
            criteria.payTypeCd = (String) j.get(EXPENSE_DETAIL_PAY_TYPE_CD);
            criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
            List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                criteria, new ArrayList<OrderDto>());
            if (CollectionUtil.isEmpty(retList)) {
                j.put(EXPENSE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
            }
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * <p>
 *
 * @return グローバルデータ補完処理
 */
@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(EXPENSE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(EXPENSE_DETAIL_TRANS_AMOUNT),
                (String) j.get(EXPENSE_DETAIL_COMPANY_RATE)));
            j.put(EXPENSE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(EXPENSE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(EXPENSE_DETAIL_QUANTITY)));
        }
    }
}

```

```

    }

    /**
     * グローバルデータ補完処理.
     * <p>
     * 下記補完を行います。
     * </p>
     * <ul>
     * <li>支払方法コード:ガジェット関連連携項目, 支払方法コードを設定</li>
     * </ul>
     *
     * @param params ガジェットメディエータ処理情報
     */
    public void globalComplementJsonData(GadgetMediatorParams params) {
        List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_EXPENSE_DETAIL_TABLE);

        String gcPayMethodCd = null;

        if (CollectionUtil.isEmpty(blockDat)) {
            for (Map<String, Object> j : blockDat) {
                // 支払方法コード
                if (gcPayMethodCd == null) {
                    gcPayMethodCd = getGadgetCouplingPayMethod();
                }
                j.put(EXPENSE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
            }
        }
    }

    /** {@inheritDoc} */
    @Override
    public Map<String, Object> getClaimExpenccCount(Map<String, Object> params) {
        return super.putClaimExpenccCount(params);
    }

    /** {@inheritDoc} */
    @Override
    public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
        return super.putClaimCompanyPayAmount(params);
    }

    /** {@inheritDoc} */
    @Override
    public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
        return super.putClaimTemporaryPayAmount(params);
    }

    /** {@inheritDoc} */
    @Override
    public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
        return super.putClaimExpenseDetail(params);
    }

    /** {@inheritDoc} */
    @Override
    public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
        return super.putClaimExpenseAmount(params);
    }
}

```

FieldName

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/expensedetail/ExpenseDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.expensedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 経費明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpenseDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected ExpenseDetailFieldName() {

    }

    /** 発生日. */
    public static final String EXPENSE_DETAIL_OCCUR_DATE = "occurDate";
}

```

```

public static final String EXPENSE_DETAIL_ACCOUNT_CD = "accountCd";
/** 摘要. */
public static final String EXPENSE_DETAIL_SUMMARY = "summary";
/** 経費区分コード. */
public static final String EXPENSE_DETAIL_EXP_TYPE_CD = "expTypeCd";
/** 勘定科目コード. */
public static final String EXPENSE_DETAIL_ACCOUNT_CD = "accountCd";
/** 税区分コード. */
public static final String EXPENSE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
/** 税率. */
public static final String EXPENSE_DETAIL_TAX_RATE = "taxRate";
/** 数量. */
public static final String EXPENSE_DETAIL_QUANTITY = "quantity";
/** 取引通貨コード. */
public static final String EXPENSE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
/** 取引通貨額. */
public static final String EXPENSE_DETAIL_TRANS_AMOUNT = "transAmount";
/** 会社通貨コード. */
public static final String EXPENSE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
/** 会社通貨換算レート. */
public static final String EXPENSE_DETAIL_COMPANY_RATE = "companyRate";
/** 会社通貨額. */
public static final String EXPENSE_DETAIL_COMPANY_AMOUNT = "companyAmount";
/** 会社通貨税抜額. */
public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
/** 会社通貨税額. */
public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
/** 会社通貨額合計. */
public static final String EXPENSE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
/** 支払分類コード. */
public static final String EXPENSE_DETAIL_PAY_TYPE_CD = "payTypeCd";
/** 支払方法コード. */
public static final String EXPENSE_DETAIL_PAY_METHOD_CD = "payMethodCd";
/** 支払先コード. */
public static final String EXPENSE_DETAIL_PAYEE_CD = "payeeCd";
/** 支払先名. */
public static final String EXPENSE_DETAIL_PAYEE_NAME = "payeeName";
/** 組織セットコード. */
public static final String EXPENSE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
/** 組織コード. */
public static final String EXPENSE_DETAIL_DEPARTMENT_CD = "departmentCd";
/** プロジェクトコード. */
public static final String EXPENSE_DETAIL_PROJECT_CD = "projectCd";

/** 会社名. */
public static final String EXPENSE_DETAIL_COMPANY_NAME = "companyName";
/** 経費区分名称. */
public static final String EXPENSE_DETAIL_EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String EXPENSE_DETAIL_ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String EXPENSE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
/** 取引通貨名称. */
public static final String EXPENSE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String EXPENSE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 支払分類名称. */
public static final String EXPENSE_DETAIL_PAY_TYPE_NAME = "payTypeName";
/** 支払方法名称. */
public static final String EXPENSE_DETAIL_PAY_METHOD_NAME = "payMethodName";
/** 組織セット名称. */
public static final String EXPENSE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
/** 組織名称. */
public static final String EXPENSE_DETAIL_DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String EXPENSE_DETAIL_PROJECT_NAME = "projectName";
/** 立替フラグ. */
public static final String EXPENSE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/ExpenseDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細テーブルマネージャクラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ExpenseDetailManager extends PluginGadgetManager<ExpenseDetailManagerDto> {

    /** {@link ExpenseDetailManager#selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link ExpenseDetailManager#selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link ExpenseDetailManager#selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager(実装)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/impl/ExpenseDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail.ExpenseDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tExpenseDetailsService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tExpenseDetailsService;
import org.seasar.util.beans.util.BeanMap;

/**
 * 経費明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link ExpenseDetailManager} リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link ExpenseDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b><br></b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>
 * <td>主キー検索</td>
 * <td><b><br></b></td>
 * </tr>
 * <tr>
 * <td>

```

```

* <td>表示案件検索</td>
* <td>selectView</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>コピー案件検索</td>
* <td>selectViewForCopy</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件検索</td>
* <td>selectViewArchive</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括挿入</td>
* <td>entry</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括更新</td>
* <td>renew</td>
* <td>更新</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括削除</td>
* <td>removeAll</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>過去案件登録</td>
* <td>matterArchiveRegister</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件処理時削除</td>
* <td>matterArchivePostDelete</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件一括削除</td>
* <td>removeAllArchive</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能別DTOクラスのI/Oマトリクス</b>
* <table border="1">
* <tr>
* <td colspan="2" rowspan="2" align="center"><b>DTO</b></td>
* <td colspan="2" align="center"><b>主キー</b><br>
* 検索</b></td>
* <td colspan="2" align="center"><b>リスト</b><br>
* 検索</b></td>
* <td colspan="2" align="center"><b>表示</b><br>
* 案件<br>
* 検索</b></td>
* <td colspan="2" align="center"><b>コピー</b><br>
* 案件<br>
* 検索</b></td>
* <td colspan="2" align="center"><b>過去</b><br>
* 案件<br>
* 検索</b></td>
* <td colspan="2" align="center"><b>一括</b><br>
* 挿入</b></td>
* <td colspan="2" align="center"><b>一括</b><br>
* 更新</b></td>
* <td colspan="2" align="center"><b>削除</b></td>
* <td colspan="2" align="center"><b>一括</b><br>
* 削除</b></td>
* <td colspan="2" align="center"><b>過去</b><br>
* 削除</b></td>

```

[illegible]

```

<td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* </tr>
* </table>
* <br>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

public class ExpenseDetailManagerImpl extends AbstractWfManager<ExpenseDetailManagerDto> implements
    ExpenseDetailManager {

    /** K30tExpenseDetailService. */
    @Resource
    protected K30tExpenseDetailService k30tExpenseDetailService;

    /** Ka30tExpenseDetailService. */
    @Resource
    protected Ka30tExpenseDetailService ka30tExpenseDetailService;

    /**
     * {@inheritDoc}
     */
    @Override
    public ExpenseDetailManagerDto selectByPk(ExpenseDetailManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k30tExpenseDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectList(ExpenseDetailManagerDto criteria) {
        return super.selectPkUsedUserDataId(k30tExpenseDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectView(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForView(k30tExpenseDetailService, SELECT_SQLFILE_NM, criteria, order);
    }

    /** {@inheritDoc} */
    @Override
    public List<ExpenseDetailManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForViewForCopy(k30tExpenseDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ExpenseDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectListForViewForArchive(ka30tExpenseDetailService, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<ExpenseDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.insertBatch(k30tExpenseDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<ExpenseDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.renewBatch(k30tExpenseDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override

```



```

@Override
public int remove(ExpenseDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tExpenseDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(ExpenseDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tExpenseDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(CriteriaForSqlDto criteria, List<ExpenseDetailManagerDto> dataList) {
    return super.entryArchive(ka30tExpenseDetailService, criteria, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
    return super.deleteNoUsedUserDataId(k30tExpenseDetailService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAllArchive(CriteriaForSqlDto criteria) {
    return super.deleteNoUsedUserDataIdForArchive(ka30tExpenseDetailService, criteria);
}
}

```

ManagerDto

Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/ExpenseDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.expensedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 経費明細テーブルマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpenseDetailManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザーデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstance;
    /** タブルid. */
    public Long tupleId;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバリエーション. */
    public String gadgetVariation;
    /** 行番号. */
    public Long rowNumber;

    /** 発生日. */
}

```

```

    / 元エロ. */
    public Timestamp occurDate;
    /** 摘要. */
    public String summary;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** 税率. */
    public BigDecimal taxRate;
    /** 数量. */
    public BigDecimal quantity;
    /** 取引通貨コード. */
    public String transCurrencyCd;
    /** 取引通貨額. */
    public BigDecimal transAmount;
    /** 会社通貨コード. */
    public String companyCurrencyCd;
    /** 会社通貨換算レート. */
    public BigDecimal companyRate;
    /** 会社通貨額. */
    public BigDecimal companyAmount;
    /** 会社通貨税抜額. */
    public BigDecimal companyAmountExcTax;
    /** 会社通貨税額. */
    public BigDecimal companyAmountTax;
    /** 会社通貨額合計. */
    public BigDecimal companyAmountSum;
    /** 支払分類コード. */
    public String payTypeCd;
    /** 支払方法コード. */
    public String payMethodCd;
    /** 支払先コード. */
    public String payeeCd;
    /** 支払先名. */
    public String payeeName;
    /** 組織セットコード. */
    public String departmentSetCd;
    /** 組織コード. */
    public String departmentCd;
    /** プロジェクトコード. */
    public String projectCd;

    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 検索基準会社コード. */
    public String searchCriteriaCompany;
    /** 取引通貨名称. */
    public String transCurrencyName;
    /** 会社通貨名称. */
    public String companyCurrencyName;
    /** 組織名称. */
    public String departmentName;
    /** プロジェクト名称. */
    public String projectName;
    /** 経費区分名称. */

```

```

    public String expTypeName;
    /** 勘定科目名称. */
    public String accountName;
    /** 税区分名称. */
    public String taxTypeName;
    /** 支払分類名称. */
    public String payTypeName;
    /** 支払方法名称. */
    public String payMethodName;
    /** 立替フラグ. */
    public String paymentFlag;
}

```

Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tExpenseDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 経費明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_expense_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tExpenseDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;

    /** 行番号. */
    @Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long rowNumber;

    /** 発生日. */

```

```

@Column(name = "occur_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp occurDate;

/** 摘要 */
@Column(name = "summary", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String summary;

/** 経費区分コード */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率 */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量 */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額 */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額 */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額 */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額 */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計 */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名 */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

```

```

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tExpenseDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)経費明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_expense_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tExpenseDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットID. */
    @Id
    @Column(name = "gadget_id", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetId;
}

```

```

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** アーカイブ年月. */
@Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
public String archiveMonth;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 発生日. */
@Column(name = "occur_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp occurDate;

/** 摘要. */
@Column(name = "summary", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String summary;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

```

```

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tExpenseDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tExpenseDetail} の名称クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl"})
public final class K30tExpenseDetailNames {

    /**
     * デフォルトコンストラクタ。
     */
    private K30tExpenseDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却。
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却。
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却。
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
}

```



```

    /*return gadgetInstanceのプロパティ名*/
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却.
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * gadgetIdのプロパティ名を返却.
     *
     * @return gadgetIdのプロパティ名
     */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却.
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却.
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * occurDateのプロパティ名を返却.
     *
     * @return occurDateのプロパティ名
     */
    public static PropertyName<Timestamp> occurDate() {
        return new PropertyName<Timestamp>("occurDate");
    }

    /**
     * summaryのプロパティ名を返却.
     *
     * @return summaryのプロパティ名
     */
    public static PropertyName<String> summary() {
        return new PropertyName<String>("summary");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }

```

```

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */

```

```

    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却。
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却。
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却。
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却。
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却。
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却。
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却。
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * extensionAのプロパティ名を返却。
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却。
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**

```

```

* extensionCのプロパティ名を返却.
*
* @return extensionCのプロパティ名
*/
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
* extensionDのプロパティ名を返却.
*
* @return extensionDのプロパティ名
*/
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
* extensionEのプロパティ名を返却.
*
* @return extensionEのプロパティ名
*/
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
* extensionFのプロパティ名を返却.
*
* @return extensionFのプロパティ名
*/
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
* extensionGのプロパティ名を返却.
*
* @return extensionGのプロパティ名
*/
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
* extensionHのプロパティ名を返却.
*
* @return extensionHのプロパティ名
*/
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
* extensionIのプロパティ名を返却.
*
* @return extensionIのプロパティ名
*/
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
* entryUserCdのプロパティ名を返却.
*
* @return entryUserCdのプロパティ名
*/
public static PropertyName<String> entryUserCd() {

```

```

    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30tExpenseDetailNames extends PropertyName<K30tExpenseDetail> {

    /**
     * インスタンスを構築.
     */
    public _K30tExpenseDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30tExpenseDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30tExpenseDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>(this, "systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public PropertyName<String> userDataId() {
        return new PropertyName<String>(this, "userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public PropertyName<String> gadgetClass() {
        return new PropertyName<String>(this, "gadgetClass");
    }
}

```

```

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * occurDateのプロパティ名を返却.
 *
 * @return occurDateのプロパティ名
 */
public PropertyName<Timestamp> occurDate() {
    return new PropertyName<Timestamp>(this, "occurDate");
}

/**
 * summaryのプロパティ名を返却.
 *
 * @return summaryのプロパティ名
 */
public PropertyName<String> summary() {
    return new PropertyName<String>(this, "summary");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名

```

```

*/
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

```

```

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */

```



```

public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * extensionKのプロパティ名を返却。

```

```

    entryUserCdのプロパティ名を返却。
    *
    * @return entryUserCdのプロパティ名
    */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tExpenseDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tExpenseDetail} の名称クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tExpenseDetailNames {

    /**
     * デフォルトコンストラクタ。
     */
    private Ka30tExpenseDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却。
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataldのプロパティ名を返却。
     *
     * @return userDataldのプロパティ名
     */
    public static PropertyName<String> userDatald() {
        return new PropertyName<String>("userDatald");
    }

    /**
     * gadgetClassのプロパティ名を返却。
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {

```

```

    return new PropertyName<String>("gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public static PropertyName<String> localeId() {
    return new PropertyName<String>("localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public static PropertyName<Long> rowNumber() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * occurDateのプロパティ名を返却.
 *
 * @return occurDateのプロパティ名
 */
public static PropertyName<Timestamp> occurDate() {
    return new PropertyName<Timestamp>("occurDate");
}

/**
 * summaryのプロパティ名を返却.
 *
 * @return summaryのプロパティ名
 */
public static PropertyName<String> summary() {
    return new PropertyName<String>("summary");
}

/**
 * expTypeCdのプロパティ名を返却.

```

```

*
* @return expTypeCdのプロパティ名
*/
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
* accountCdのプロパティ名を返却.
*
* @return accountCdのプロパティ名
*/
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
* taxTypeCdのプロパティ名を返却.
*
* @return taxTypeCdのプロパティ名
*/
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
* taxRateのプロパティ名を返却.
*
* @return taxRateのプロパティ名
*/
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
* quantityのプロパティ名を返却.
*
* @return quantityのプロパティ名
*/
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
* transCurrencyCdのプロパティ名を返却.
*
* @return transCurrencyCdのプロパティ名
*/
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
* transAmountのプロパティ名を返却.
*
* @return transAmountのプロパティ名
*/
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
* companyCurrencyCdのプロパティ名を返却.
*
* @return companyCurrencyCdのプロパティ名
*/
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
* companyRateのプロパティ名を返却.
*
* @return companyRateのプロパティ名
*/
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
* companyAmountのプロパティ名を返却.
*
* @return companyAmountのプロパティ名
*/
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

```

```

}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *

```

```

    * @return projectCdのプロパティ名
    */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却.
     *
     * @return accountNameのプロパティ名
     */
    public static PropertyName<String> accountName() {
        return new PropertyName<String>("accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却.
     *
     * @return taxTypeNameのプロパティ名
     */
    public static PropertyName<String> taxTypeName() {
        return new PropertyName<String>("taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却.
     *
     * @return transCurrencyNameのプロパティ名
     */
    public static PropertyName<String> transCurrencyName() {
        return new PropertyName<String>("transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却.
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public static PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>("companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却.
     *
     * @return payTypeNameのプロパティ名
     */
    public static PropertyName<String> payTypeName() {
        return new PropertyName<String>("payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却.
     *
     * @return paymentFlagのプロパティ名
     */
    public static PropertyName<String> paymentFlag() {
        return new PropertyName<String>("paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却.
     *
     * @return payMethodNameのプロパティ名
     */
    public static PropertyName<String> payMethodName() {
        return new PropertyName<String>("payMethodName");
    }

    /**
     * departmentNameのプロパティ名を返却.
     *
     * @return departmentNameのプロパティ名
     */
    public static PropertyName<String> departmentName() {
        return new PropertyName<String>("departmentName");
    }
}

```

```

/**
 * projectNameのプロパティ名を返却。
 *
 * @return projectNameのプロパティ名
 */
public static PropertyName<String> projectName() {
    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却。
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却。
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */

```

```

/
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却。
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却。
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却。
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _Ka30tExpenseDetailNames extends PropertyName<Ka30tExpenseDetail> {

    /**
     * インスタンスを構築。
     */
    public _Ka30tExpenseDetailNames() {
    }

    /**
     * インスタンスを構築。
     *
     * @param name 名前
     */
    public _Ka30tExpenseDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築。
     *
     * @param parent 親
     * @param name 名前
     */
    public _Ka30tExpenseDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

```



```

    super(parent, name);
}

/**
 * systemMatterIdのプロパティ名を返却.
 *
 * @return systemMatterIdのプロパティ名
 */
public PropertyName<String> systemMatterId() {
    return new PropertyName<String>((this, "systemMatterId"));
}

/**
 * userDataIdのプロパティ名を返却.
 *
 * @return userDataIdのプロパティ名
 */
public PropertyName<String> userDataId() {
    return new PropertyName<String>((this, "userDataId"));
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>((this, "gadgetClass"));
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>((this, "gadgetInstance"));
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>((this, "tupleId"));
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>((this, "localeId"));
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>((this, "archiveMonth"));
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>((this, "gadgetId"));
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>((this, "gadgetVariation"));
}

/**
 * rowNumberのプロパティ名を返却.
 *

```

```

    * @return rowNumのプロパティ名
    */
    public PropertyName<Long> rowNum() {
        return new PropertyName<Long>(this, "rowNum");
    }

    /**
     * occurDateのプロパティ名を返却.
     *
     * @return occurDateのプロパティ名
     */
    public PropertyName<Timestamp> occurDate() {
        return new PropertyName<Timestamp>(this, "occurDate");
    }

    /**
     * summaryのプロパティ名を返却.
     *
     * @return summaryのプロパティ名
     */
    public PropertyName<String> summary() {
        return new PropertyName<String>(this, "summary");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public PropertyName<String> accountCd() {
        return new PropertyName<String>(this, "accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public PropertyName<String> taxTypeCd() {
        return new PropertyName<String>(this, "taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }
}

```

```

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名

```

```

/  

public PropertyName<String> payeeName() {  

    return new PropertyName<String>(this, "payeeName");  

}  

/**  

 * departmentSetCdのプロパティ名を返却.  

 *  

 * @return departmentSetCdのプロパティ名  

 */  

public PropertyName<String> departmentSetCd() {  

    return new PropertyName<String>(this, "departmentSetCd");  

}  

/**  

 * departmentCdのプロパティ名を返却.  

 *  

 * @return departmentCdのプロパティ名  

 */  

public PropertyName<String> departmentCd() {  

    return new PropertyName<String>(this, "departmentCd");  

}  

/**  

 * projectCdのプロパティ名を返却.  

 *  

 * @return projectCdのプロパティ名  

 */  

public PropertyName<String> projectCd() {  

    return new PropertyName<String>(this, "projectCd");  

}  

/**  

 * expTypeNameのプロパティ名を返却.  

 *  

 * @return expTypeNameのプロパティ名  

 */  

public PropertyName<String> expTypeName() {  

    return new PropertyName<String>(this, "expTypeName");  

}  

/**  

 * accountNameのプロパティ名を返却.  

 *  

 * @return accountNameのプロパティ名  

 */  

public PropertyName<String> accountName() {  

    return new PropertyName<String>(this, "accountName");  

}  

/**  

 * taxTypeNameのプロパティ名を返却.  

 *  

 * @return taxTypeNameのプロパティ名  

 */  

public PropertyName<String> taxTypeName() {  

    return new PropertyName<String>(this, "taxTypeName");  

}  

/**  

 * transCurrencyNameのプロパティ名を返却.  

 *  

 * @return transCurrencyNameのプロパティ名  

 */  

public PropertyName<String> transCurrencyName() {  

    return new PropertyName<String>(this, "transCurrencyName");  

}  

/**  

 * companyCurrencyNameのプロパティ名を返却.  

 *  

 * @return companyCurrencyNameのプロパティ名  

 */  

public PropertyName<String> companyCurrencyName() {  

    return new PropertyName<String>(this, "companyCurrencyName");  

}  

/**  

 * payTypeNameのプロパティ名を返却.  

 *  

 * @return payTypeNameのプロパティ名  

 */  

public PropertyName<String> payTypeName() {  

    return new PropertyName<String>(this, "payTypeName");  

}  

}

```

```

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */

```

```

public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailsService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tExpenseDetail;

/**
 * {@link K30tExpenseDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tExpenseDetailsService extends GenerateService<K30tExpenseDetail> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tExpenseDetailsService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tExpenseDetail;

/**
 * {@link Ka30tExpenseDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tExpenseDetailsService extends GenerateService<Ka30tExpenseDetail> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30tExpenseDetail} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tExpenseDetailCondition extends
    AbstractEntityCondition<K30tExpenseDetailCondition> {

    /**
     * インスタンスを構築。
     */
    public K30tExpenseDetailCondition() {
    }

    /**
     * インスタンスを構築。
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tExpenseDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }
}

```

```

/** systemMatterIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> systemMatterId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("systemMatterId", this);

/** userDataIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> userDataId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("userDataId", this);

/** gadgetClassの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetClass =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetClass", this);

/** gadgetInstanceの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> gadgetInstance =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("gadgetInstance", this);

/** tupleIdの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> tupleId =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("tupleId", this);

/** gadgetIdの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetId =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NotNullableStringCondition<K30tExpenseDetailCondition> gadgetVariation =
    new NotNullableStringCondition<K30tExpenseDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullableCondition<K30tExpenseDetailCondition, Long> rowNumber =
    new NotNullableCondition<K30tExpenseDetailCondition, Long>("rowNumber", this);

/** occurDateの条件. */
public NullableCondition<K30tExpenseDetailCondition, Timestamp> occurDate =
    new NullableCondition<K30tExpenseDetailCondition, Timestamp>("occurDate", this);

/** summaryの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> summary =
    new NullableStringCondition<K30tExpenseDetailCondition>("summary", this);

/** expTypeCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> expTypeCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> accountCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> taxTypeCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tExpenseDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tExpenseDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountTax", this);

```



```

    new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountTax", this);

    /** companyAmountSumの条件. */
    public NullableCondition<K30tExpenseDetailCondition, BigDecimal> companyAmountSum =
        new NullableCondition<K30tExpenseDetailCondition, BigDecimal>("companyAmountSum", this);

    /** payTypeCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payTypeCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payTypeCd", this);

    /** payMethodCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payMethodCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payMethodCd", this);

    /** payeeCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payeeCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("payeeCd", this);

    /** payeeNameの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> payeeName =
        new NullableStringCondition<K30tExpenseDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> departmentSetCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("departmentSetCd", this);

    /** departmentCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> departmentCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("departmentCd", this);

    /** projectCdの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> projectCd =
        new NullableStringCondition<K30tExpenseDetailCondition>("projectCd", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionA =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionB =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionC =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionD =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionE =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionF =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionG =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionH =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionI =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tExpenseDetailCondition> extensionJ =
        new NullableStringCondition<K30tExpenseDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tExpenseDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tExpenseDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tExpenseDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tExpenseDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<K30tExpenseDetailCondition, Long> renewCnt =
        new NotNullableCondition<K30tExpenseDetailCondition, Long>("renewCnt", this);

    /** ...の条件. */

```

```

    /** renewTsの条件. */
    public NotNullCondition<K30tExpenseDetailCondition, Timestamp> renewTs =
        new NotNullCondition<K30tExpenseDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullStringCondition<K30tExpenseDetailCondition> renewUserCd =
        new NotNullStringCondition<K30tExpenseDetailCondition>("renewUserCd", this);
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tExpenseDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullCondition;
import org.seasar.extension.jdbc.where.condition.NotNullStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tExpenseDetail} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl"})
public final class Ka30tExpenseDetailCondition extends
    AbstractEntityCondition<Ka30tExpenseDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tExpenseDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tExpenseDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> systemMatterId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("systemMatterId", this);

    /** userDataIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> userDataId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("userDataId", this);

    /** gadgetClassの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> gadgetClass =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Long> gadgetInstance =
        new NotNullCondition<Ka30tExpenseDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NotNullCondition<Ka30tExpenseDetailCondition, Long> tupleId =
        new NotNullCondition<Ka30tExpenseDetailCondition, Long>("tupleId", this);

    /** localeIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> localeId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("localeId", this);

    /** archiveMonthの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> archiveMonth =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("archiveMonth", this);

    /** gadgetIdの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> gadgetId =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("gadgetId", this);

    /** gadgetVariationの条件. */
    public NotNullStringCondition<Ka30tExpenseDetailCondition> gadgetVariation =
        new NotNullStringCondition<Ka30tExpenseDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */

```

```

public NullableCondition<Ka30tExpenseDetailCondition, Long> rowNumber =
    new NullableCondition<Ka30tExpenseDetailCondition, Long>("rowNumber", this);

/** occurDateの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, Timestamp> occurDate =
    new NullableCondition<Ka30tExpenseDetailCondition, Timestamp>("occurDate", this);

/** summaryの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> summary =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("summary", this);

/** expTypeCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> accountCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tExpenseDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tExpenseDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payeeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentCd", this);

```

```

/** projectCdの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> projectCd =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> accountName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> departmentName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("departmentName", this);

/** projectNameの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> projectName =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionA =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionB =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionC =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionD =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionE =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionF =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionG =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionH =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionI =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tExpenseDetailCondition> extensionJ =
    new NullableStringCondition<Ka30tExpenseDetailCondition>("extensionJ", this);

/** entryTsの条件. */

```

```

public NotNullCondition<Ka30tExpenseDetailCondition, Timestamp> entryTs =
    new NotNullCondition<Ka30tExpenseDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullStringCondition<Ka30tExpenseDetailCondition> entryUserCd =
    new NotNullStringCondition<Ka30tExpenseDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<Ka30tExpenseDetailCondition, Long> renewCnt =
    new NotNullCondition<Ka30tExpenseDetailCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<Ka30tExpenseDetailCondition, Timestamp> renewTs =
    new NotNullCondition<Ka30tExpenseDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<Ka30tExpenseDetailCondition> renewUserCd =
    new NotNullStringCondition<Ka30tExpenseDetailCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.occur_date,
    t.summary,
    t.exp_type_cd,
    t.account_cd,
    t.tax_type_cd,
    t.tax_rate,
    t.quantity,
    t.trans_currency_cd,
    t.trans_amount,
    t.company_currency_cd,
    t.company_rate,
    t.company_amount,
    t.company_amount_exc_tax,
    t.company_amount_tax,
    t.company_amount_sum,
    t.pay_type_cd,
    t.pay_method_cd,
    t.payee_cd,
    t.payee_name,
    t.department_set_cd,
    t.department_cd,
    t.project_cd,
    t.extension_a,
    t.extension_b,
    t.extension_c,
    t.extension_d,
    t.extension_e,
    t.extension_f,
    t.extension_g,
    t.extension_h,
    t.extension_i,
    t.extension_j,
    t.entry_ts,
    t.entry_user_cd,
    t.renew_cnt,
    t.renew_ts,
    t.renew_user_cd,
    c1.search_criteria_company,
    m1.currency_name as trans_currency_name,
    m2.department_name,
    m3.project_name,
    m4.pay_type_name,
    m5.pay_method_name,
    m6.currency_name as company_currency_name,
    m7.exp_type_name,
    m8.account_name,
    m9.tax_type_name,
    m4.payment_flag

```

```

from
k30t_expense_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeld*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeld*/ja' = m2.locale_id
and
c1.search_criteria_date >= m2.start_date
and
c1.search_criteria_date < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
c1.search_criteria_date >= m3.start_date
and
c1.search_criteria_date < m3.end_date
and
/*localeld*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeld*/ja' = m4.locale_id
and
c1.search_criteria_date >= m4.start_date
and
c1.search_criteria_date < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeld*/ja' = m5.locale_id
and
c1.search_criteria_date >= m5.start_date
and
c1.search_criteria_date < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6
on
c1.search_criteria_company = m6.company_cd
and
t.company_currency_cd = m6.currency_cd
and
/*localeld*/ja' = m6.locale_id
and
m6.delete_flag = '0'
left outer join
k30m_exp_type m7

```

```

on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeId*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeId*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeId*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date
and
  c1.search_criteria_date < m9.end_date
and
  m9.delete_flag = '0'
where
  t.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
  /*END*/
  /*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
  /*END*/
  /*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
  /*END*/
  /*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
  /*END*/
  /*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
  /*END*/
  /*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
  /*END*/

```

コピー

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tExpenseDetailService/selectViewForCopy.sql

```

select
  t.system_matter_id,
  t.user_data_id,
  t.gadget_class,
  t.gadget_instance,
  t.tuple_id,
  t.gadget_id,
  t.gadget_variation,
  t.row_number,
  t.occur_date,
  t.summary,
  t.exp_type_cd,

```

```

t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_currency_cd,
t.company_rate,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
ma.exp_type_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> " then m8.account_name else mb.account_name end as account_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> " then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
m4.payment_flag
from
k30t_expense_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
/*searchCriteriaDate*/" >= m2.start_date
and
/*searchCriteriaDate*/" < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
/*searchCriteriaDate*/" >= m3.start_date

```



```

and
/*searchCriteriaDate*/" < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id
and
/*searchCriteriaDate*/" >= m4.start_date
and
/*searchCriteriaDate*/" < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeId*/ja' = m5.locale_id
and
/*searchCriteriaDate*/" >= m5.start_date
and
/*searchCriteriaDate*/" < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6
on
c1.search_criteria_company = m6.company_cd
and
t.company_currency_cd = m6.currency_cd
and
/*localeId*/ja' = m6.locale_id
and
m6.delete_flag = '0'
left outer join
k30m_account m8
on
c1.search_criteria_company = m8.company_cd
and
t.account_cd = m8.account_cd
and
/*localeId*/ja' = m8.locale_id
and
/*searchCriteriaDate*/" >= m8.start_date
and
/*searchCriteriaDate*/" < m8.end_date
and
m8.delete_flag = '0'
left outer join
k30m_tax_type m9
on
c1.search_criteria_company = m9.company_cd
and
t.tax_type_cd = m9.tax_type_cd
and
/*localeId*/ja' = m9.locale_id
and
/*searchCriteriaDate*/" >= m9.start_date
and
/*searchCriteriaDate*/" < m9.end_date
and
m9.delete_flag = '0'

left outer join
k30m_exp_type ma
on
c1.search_criteria_company = ma.company_cd
and
t.exp_type_cd = ma.exp_type_cd
and
/*localeId*/ja' = ma.locale_id
and
/*searchCriteriaDate*/" >= ma.start_date
and
/*searchCriteriaDate*/" < ma.end_date
and
ma.delete_flag = '0'
left outer join

```

```

left outer join
  k30m_account mb
on
  ma.company_cd = mb.company_cd
and
  ma.account_cd = mb.account_cd
and
  /*localeId*/ja' = mb.locale_id
and
  /*searchCriteriaDate*/" >= mb.start_date
and
  /*searchCriteriaDate*/" < mb.end_date
and
  m8.delete_flag = '0'
left outer join
  k30m_tax_type mc
on
  ma.company_cd = mc.company_cd
and
  ma.tax_type_cd = mc.tax_type_cd
and
  /*localeId*/ja' = mc.locale_id
and
  /*searchCriteriaDate*/" >= mc.start_date
and
  /*searchCriteriaDate*/" < mc.end_date
and
  mc.delete_flag = '0'
where
  t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/expensedetail/impl/ExpenseDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
  t.system_matter_id as system_matter_id,
  t.user_data_id as user_data_id,
  t.gadget_class as gadget_class,
  t.gadget_instance as gadget_instance,
  t.tuple_id as tuple_id,
  t.gadget_id as gadget_id,
  t.gadget_variation as gadget_variation,
  i1.contents_id as contents_id,
  i1.contents_version_id as contents_version_id,
  null as journal_data_type_cd,
  t.row_number as row_number,
  t.exp_type_cd as exp_type_cd,
  t.account_cd as account_cd,
  t.tax_type_cd as tax_type_cd,
  mt.tax_rate as tax_rate,
  t.quantity as quantity,
  t.trans_currency_cd as trans_currency_cd,
  t.trans_amount as trans_amount,
  t.company_currency_cd as company_currency_cd,
  t.company_rate as company_rate,
  t.company_amount as company_amount,
  t.company_amount_exc_tax as company_amount_exc_tax,
  t.company_amount_tax as company_amount_tax,
  t.company_amount_sum as company_amount_sum,
  t.payee_cd as customer_cd,

```

```

t.payee_name as customer_name,
t.department_set_cd as department_set_cd,
t.department_cd as department_cd,
t.project_cd as project_cd,
t.pay_method_cd as pay_method_cd,
t.pay_type_cd as pay_type_cd,
null as adv_matter_number,
null as adv_matter_name,
t.occur_date as detail_date1,
null as detail_date2,
t.summary as detail_summary1,
null as detail_summary2,
null as detail_summary3,
null as detail_summary4,
null as detail_summary5,
null as detail_note,
t.extension_a as detail_extension_a,
t.extension_b as detail_extension_b,
t.extension_c as detail_extension_c,
t.extension_d as detail_extension_d,
t.extension_e as detail_extension_e,
t.extension_f as detail_extension_f,
t.extension_g as detail_extension_g,
t.extension_h as detail_extension_h,
t.extension_i as detail_extension_i,
t.extension_j as detail_extension_j,
'0' as journal_status,
null as wish_date,
c1.search_criteria_company,
c1.search_criteria_date
from
k30t_expense_detail t
inner join
k20c_matter_info i1
on
t.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join
k30m_tax_type mt
on
c1.search_criteria_company = mt.company_cd
and
t.tax_type_cd = mt.tax_type_cd
and
c1.search_criteria_date >= mt.start_date
and
c1.search_criteria_date < mt.end_date
and
i1.locale_id_apply = mt.locale_id
and
mt.delete_flag = '0'

where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.row_number
/*END*/

```

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/expenseDetail/expenseDetailV02.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wkf" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<%-- routeDetail v02 --%>
<c:set var="gadgetClass">expenseDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/expenseDetailV02</c:set>
<c:set var="tmtableId">expenseDetailTable</c:set>
<c:set var="tmtableControllerBlockId">tmtableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks[expenseDetailTable]}" />

<%-- tmtable --%>
<c:set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<%-- Message --%>
<c:set var="msgRateReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0008) %></c:set>

<%-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</c:set>
<c:set var="surfaceOccurDate">${gadget.surfaces["occurDate"]}</c:set>
<c:set var="surfaceSummary">${gadget.surfaces["summary"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceTransCurrency">${gadget.surfaces["amount_transCurrency"]}</c:set>
<c:set var="surfaceCompanyRate">${gadget.surfaces["amount_transRate"]}</c:set>
<c:set var="surfaceCompanyAmount">${gadget.surfaces["amount_companyAmount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>

<%-- Mode --%>
<c:set var="modeOccurDate">${detail.fields["occurDate"].inputType}</c:set>
<c:set var="modeSummary">${detail.fields["summary"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modeTransCurrency">${detail.fields["transCurrencyCd"].inputType}</c:set>
<c:set var="modeCompanyRate">${detail.fields["companyRate"].inputType}</c:set>
<c:set var="modeCompanyAmount">${detail.fields["companyAmount"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>

<%-- RequiredMark --%>
<c:set var="reqOccurDate" >${detail.fields["occurDate"].requiredFlag}</c:set>
<c:set var="reqSummary" >${detail.fields["summary"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqTransCurrency" >${detail.fields["transCurrencyCd"].requiredFlag}</c:set>
<c:set var="reqCompanyRate" >${detail.fields["companyRate"].requiredFlag}</c:set>
<c:set var="reqCompanyAmount" >${detail.fields["companyAmount"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>

```

```

<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validOccurDate_def">{k_date:[""]}</c:set>
<c:set var="validOccurDate_apply">${"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>
<c:set var="validOccurDate_tempsave"></c:set>
<c:set var="validOccurDate_reapply">${"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>
<c:set var="validOccurDate_process">${"true" == reqOccurDate ? "k_required:[""] : ""}</c:set>

<c:set var="validSummary_def">{k_maxLength:["250"]}</c:set>
<c:set var="validSummary_apply">${"true" == reqSummary ? "k_required:[""] : ""}</c:set>
<c:set var="validSummary_tempsave"></c:set>
<c:set var="validSummary_reapply">${"true" == reqSummary ? "k_required:[""] : ""}</c:set>
<c:set var="validSummary_process">${"true" == reqSummary ? "k_required:[""] : ""}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:["expTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validExpTypeCd_apply">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validExpTypeCd_process">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>

<c:set var="validAccountCd_def">{k_accountExists:["accountCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>
<c:set var="validAccountCd_process">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:["taxTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">${"true" == reqTaxTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">${"true" == reqTaxTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validTaxType_process">${"true" == reqTaxTypeCd ? "k_required:[""] : ""}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:["payTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">${"true" == reqPayTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPayTypeCd_reapply">${"true" == reqPayTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validPayTypeCd_process">${"true" == reqPayTypeCd ? "k_required:[""] : ""}</c:set>

<c:set var="validPayeeName_def">{k_maxLength:["250"]}</c:set>
<c:set var="validPayeeName_apply">${"true" == reqPayeeName ? "k_required:[""] : ""}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">${"true" == reqPayeeName ? "k_required:[""] : ""}</c:set>
<c:set var="validPayeeName_process">${"true" == reqPayeeName ? "k_required:[""] : ""}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:["searchCriteriaCompany", 'departmentSetCd', 'departmentCd', 'searchCriteriaDate']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:["projectCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:["departmentCd", 'projectCd']}</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${surfaceChapterTitle}"
    gadgetId="${gadgetId}"
    hideGadgetTitle="true"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="expenseDetailHeaderTable">
<div class="imui-chapter-title">
<h2 style="display:inline-block;width:50%">
<k:surface
    name="chapterTitle"
    surfaceKey="${surfaceChapterTitle}"
/>
</h2>
<k:a
    class="kaiden-ui-icon-refresh-gray16_nosize"
    name="rateReget"
    mode="${modeTransCurrency}"
    text="${surfaceRateReget}"
    style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem"
/>
</div>
</k:blockContainer>
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:ttable id="${tmtableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumberTarget="rowNumber">
<colgroup>

```

```

<col style="width:40px;overflow:hidden">
<col style="width:235px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
</colgroup>
<thead>
<tr>
<td rowspan="3" style="vertical-align:middle;">
<k:tmtableOperationPanel
  position="header"
  allowAdd="{allowAddDetail}"
  allowDel="{allowDelDetail}"
/>
</td>
<th>
<k:surface
  name="occurDateSurface"
  surfaceKey="{surfaceOccurDate}"
  requiredMark="{reqOccurDate}"
/>
</th>
<th colspan="3">
<k:surface
  name="summarySurface"
  surfaceKey="{surfaceSummary}"
  requiredMark="{reqSummary}"
/>
</th>
</tr>
<tr>
<th colspan="2">
<k:surface
  name="expTypeSurface"
  surfaceKey="{surfaceExpType}"
  requiredMark="{reqExpTypeCd}"
  style="display:inline;"
/>
<k:surface
  name="accountSurface"
  surfaceKey="{surfaceAccount}"
  style="display:inline;"
/>
<k:surface
  name="taxTypeSurface"
  surfaceKey="{surfaceTaxType}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="departmentSurface"
  surfaceKey="{surfaceDepartment}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="projectSurface"
  surfaceKey="{surfaceProject}"
/>
</th>
</tr>
<tr>
<th>
<k:surface
  name="payTypeSurface"
  surfaceKey="{surfacePayType}"
  requiredMark="{reqPayTypeCd}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="payeeSurface"
  surfaceKey="{surfacePayee}"
  requiredMark="{reqPayeeName}"
  style="display:inline;"
/>
</th>
<th colspan="2">
<k:surface
  name="transAmountSurface"
  surfaceKey="{surfaceTransAmount}"
  requiredMark="{reqTransAmount}"
/>

```

```

</th>
</tr>
</thead>
<tbody>
<tr>
<td rowspan="3">
<:tmtableOperationPanel
position="detail"
allowAdd="{allowAddDetail}"
allowDel="{allowDelDetail}"
name="rowNumber"
/>
</td>
<td>
<k:date
name="occurDate"
mode="{modeOccurDate}"
surfaceKey="{surfaceOccurDate}"
validationDefault="{validOccurDate_def}"
validationApply="{validOccurDate_apply}"
validationTempsave="{validOccurDate_tempsave}"
validationReapply="{validOccurDate_reapply}"
validationProcess="{validOccurDate_process}"
/>
</td>
<td colspan="3">
<k:text
name="summary"
mode="{modeSummary}"
imeMode="on"
surfaceKey="{surfaceSummary}"
style="width:545px;"
maxlength="250"
validationDefault="{validSummary_def}"
validationApply="{validSummary_apply}"
validationTempsave="{validSummary_tempsave}"
validationReapply="{validSummary_reapply}"
validationProcess="{validSummary_process}"
/>
</td>
<tr>
<td colspan="2">
<k:hidden
name="expTypeCd"
mode="{modeExpTypeCd}"
/>
<kmsw:expType
name="expTypeName"
mode="{modeExpTypeCd}"
imeMode="on"
surfaceKey="{surfaceExpType}"
resultValue="expTypeName"
resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
primaryTarget="expTypeCd"
searchCriteriaDate="searchCriteriaDate"
searchCriteriaCompany="searchCriteriaCompany"
searchCriteriaExpTypeCd="expTypeCd"
style="width:150px;"
validationDefault="{validExpTypeCd_def}"
validationApply="{validExpTypeCd_apply}"
validationTempsave="{validExpTypeCd_tempsave}"
validationReapply="{validExpTypeCd_reapply}"
validationProcess="{validExpTypeCd_process}"
/>
<k:hidden
name="accountCd"
mode="{modeAccountCd}"
surfaceKey="{surfaceAccount}"
validationDefault="{validAccountCd_def}"
validationApply="{validAccountCd_apply}"
validationTempsave="{validAccountCd_tempsave}"
validationReapply="{validAccountCd_reapply}"
validationProcess="{validAccountCd_process}"
/>
<k:text
name="accountName"
mode="1"
class="kaiden-ui-mst-search-chain-item"
style="width:115px;"
/>
<k:hidden
name="taxTypeCd"
mode="{modeTaxTypeCd}"
surfaceKey="{surfaceTaxType}"
validationDefault="{validTaxType_def}"
validationApply="{validTaxType_apply}"

```

```

validationTempsave="{validTaxType_tempsave}"
validationReapply="{validTaxType_reapply}"
validationProcess="{validTaxType_process}"
/>
<k:text
  name="taxTypeName"
  mode="1"
  class="kaiden-ui-mst-search-chain-item"
  style="width:115px"
/>
</td>
<td>
<k:hidden
  name="departmentSetCd"
  mode="{modeDepartmentCd}"
/>
<k:hidden
  name="departmentCd"
  mode="{modeDepartmentCd}"
  surfaceKey="{surfaceDepartment}"
/>
<kmsb:department
  name="departmentName"
  mode="{modeDepartmentCd}"
  imeMode="on"
  surfaceKey="{surfaceDepartment}"
  resultValue="departmentName"
  resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd'}"
  primaryTarget="departmentCd,departmentSetCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaDepartmentCd="departmentCd"
  searchCriteriaDepartmentSetCd="departmentSetCd"
  style="width:150px;"
  validationDefault="{validDepartmentCd_def}"
  validationApply="{validDepartmentCd_apply}"
  validationTempsave="{validDepartmentCd_tempsave}"
  validationReapply="{validDepartmentCd_reapply}"
  validationProcess="{validDepartmentCd_process}"
/>
</td>
<td>
<k:hidden
  name="projectCd"
  mode="{modeProjectCd}"
  surfaceKey="{surfaceProject}"
/>
<kmsw:project
  name="projectName"
  mode="{modeProjectCd}"
  imeMode="on"
  surfaceKey="{surfaceProject}"
  resultValue="projectName"
  resultChain="{projectCd:'projectCd'}"
  primaryTarget="projectCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaProjectCd="projectCd"
  style="width:150px;"
  validationDefault="{validProjectCd_def}"
  validationApply="{validProjectCd_apply}"
  validationTempsave="{validProjectCd_tempsave}"
  validationReapply="{validProjectCd_reapply}"
  validationProcess="{validProjectCd_process}"
/>
</td>
</tr>
<tr>
<td>
<klist_wkf:listDataPayType
  blank="false"
  name="payType"
  companyCd="{param.searchCriteriaCompany}"
  searchCriteriaDate="{param.searchCriteriaDate}"
/>
<k:select
  name="payTypeCd"
  mode="{modePayTypeCd}"
  surfaceKey="{surfacePayType}"
  viewModeName="payTypeName"
  dataName="payType"
  selectChain="{paymentFlag:paymentFlag}"
  style="width:230px;"
  validationDefault="{validPayTypeCd_def}"
  validationApply="{validPayTypeCd_apply}"
  validationTempsave="{validPayTypeCd_tempsave}"
  validationReapply="{validPayTypeCd_reapply}"
  validationProcess="{validPayTypeCd_process}"
/>

```



```

validationReapply= ${validPayTypeCd_reapply}
validationProcess="${validPayTypeCd_process}"
/>
<k:hidden
  name="paymentFlag"
  mode="${modePayTypeCd}"
/>
</td>
<td>
  <k:text
    name="payeeName"
    mode="${modePayeeName}"
    imeMode="on"
    surfaceKey="${surfacePayee}"
    style="width:160px;"
    maxlength="250"
    validationDefault="${validPayeeName_def}"
    validationApply="${validPayeeName_apply}"
    validationTemsave="${validPayeeName_tempsave}"
    validationReapply="${validPayeeName_reapply}"
    validationProcess="${validPayeeName_process}"
  />
</td>
<td colspan="2">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericCurrencyInputParts.jsp">
    <jsp:param name="gadgetId" value="${gadgetId}" />
    <jsp:param name="tmtableId" value="${tmtableId}" />
    <jsp:param name="searchCriteriaCompany" value="${searchCriteriaCompany}" />
    <jsp:param name="companyCurrencyCd" value="${companyCurrencyCd}" />
    <%-- 金額 --%>
    <jsp:param name="modeTransAmnt" value="${modeTransAmount}" />
    <jsp:param name="surfaceTransAmnt" value="${surfaceTransAmount}" />
    <jsp:param name="requiredTransAmntApply" value="${reqTransAmount}" />
    <jsp:param name="requiredTransAmntReapply" value="${reqTransAmount}" />
    <jsp:param name="requiredTransAmntProcess" value="${reqTransAmount}" />
    <%-- 通貨 --%>
    <jsp:param name="modeTransCryCd" value="${modeTransCurrency}" />
    <jsp:param name="surfaceTransCryCd" value="${surfaceTransCurrency}" />
    <jsp:param name="requiredTransCryCdApply" value="${reqTransCurrency}" />
    <jsp:param name="requiredTransCryCdReapply" value="${reqTransCurrency}" />
    <jsp:param name="requiredTransCryCdProcess" value="${reqTransCurrency}" />
    <%-- レート --%>
    <jsp:param name="modeCmpRate" value="${modeCompanyRate}" />
    <jsp:param name="surfaceCmpRate" value="${surfaceCompanyRate}" />
    <jsp:param name="requiredCmpRateApply" value="${reqTransCurrency}" />
    <jsp:param name="requiredCmpRateReapply" value="${reqTransCurrency}" />
    <jsp:param name="requiredCmpRateProcess" value="${reqTransCurrency}" />
    <%-- 換算額 --%>
    <jsp:param name="modeCmpAmnt" value="${modeCompanyAmount}" />
    <jsp:param name="surfaceCmpAmnt" value="${surfaceCompanyAmount}" />
    <jsp:param name="requiredCmpAmntApply" value="${reqCompanyAmount}" />
    <jsp:param name="requiredCmpAmntReapply" value="${reqCompanyAmount}" />
    <jsp:param name="requiredCmpAmntProcess" value="${reqCompanyAmount}" />
  </jsp:include>
</td>

</tr>
</tbody>
</k:tmtable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
  jQuery(function($){

    var eventMan = KAIDEN.getGadget("${gadgetId").eventManager
    , tmt = KAIDEN.getGadget("${gadgetId").tmtables["expenseDetailTable"]
    , tma = tmt.tmtableAccessor;

    /*****
    * 内部イベント
    *****/

    //明細行:行挿入(データ設定前)
    tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

      //デフォルト部門を要求
      var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "${gadgetId}");
      defaultDept = defaultDept[0].result; //先頭行のみ利用
      tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
      tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
      tma("departmentName", tupleId).setValue(defaultDept.departmentName);

      //デフォルトプロジェクトを要求
      var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "${gadgetId}");
      defaultPrj = defaultPrj[0].result; //先頭行のみ利用
      tma("projectCd", tupleId).setValue(defaultPrj.projectCd);

```

```

tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
tma("projectName", tupleId).setValue(defaultPrj.projectName);

});

//明細行:行挿入(データ設定後)
tmt.addEventPostNewRow(function(tupleId, tma, data) {
  //マスタ検索部品の整合性確認.
  KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
  KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
  KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

  //レートボックスのreadonly化
  if (tma("transCurrencyCd", tupleId).elem(true).length > 0) {
    var currencyCd = tma("transCurrencyCd", tupleId).getValue();
    if (currencyCd == "${companyCurrencyCd}") {
      tma("companyRate", tupleId).readonly();
    } else {
      tma("companyRate", tupleId).removeReadonly();
    }
  }
}

//初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
if (tma("payTypeCd", tupleId).elem(true).length > 0) {
  KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
}
});

//レート再取得ロジック
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll = function() {
  tma("transCurrencyCd", tupleId).elem(true).each(function() {
    tmt.variable.loadRate(this);
  });
}

//レート再取得:ボタン押下
KAIDEN.getGadget("${gadgetId}").blockAccessors["expenseDetailHeaderTable"]("rateReget").elem(true).on("click", function(){
  imuiConfirm("${msgRateReget}", "", function() {
    KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
  }, undefined, false);
});

/*****
* 受信イベント (push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
  tmt.deleteAll();
  return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
  return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
  return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//reloadAllCurrencyRate レート再取得 イベントリスナの登録
eventMan.registPushListener("reloadAllCurrencyRate", function(**Object*/arg) {
  KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
});

/*****
* 受信イベント (pull)
*****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
  return tmt.getColumnValues(["occurDate", "occurDate"]);
});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues("companyAmount");
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
});
return result.toString();

```

```

    claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
    eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
        var amounts = tmt.getColumnValues(["companyAmount", "paymentFlag"]);
        var result = new BigDecimal().ZERO;
        $.each(amounts, function(index, value) {
            if ($.isNumeric(value[0]) && value[1] != "1") {
                result = result.add(new BigDecimal(value[0]));
            }
        });
        return result.toString();
    });

    claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
    eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
        var amounts = tmt.getColumnValues(["companyAmount", "paymentFlag"]);
        var result = new BigDecimal().ZERO;
        $.each(amounts, function(index, value) {
            if ($.isNumeric(value[0]) && value[1] != "1") {
                result = result.add(new BigDecimal(value[0]));
            }
        });
        return result.toString();
    });

    claimExpenseDetail 費用明細を要求 イベントリスナの登録
    eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
        return tmt.getColumnValues("companyAmount");
    });

    claimExpenccount 費用明細行数を要求 イベントリスナの登録
    eventMan.registPullListener("claimExpenccount", function(**Object*/arg) {
        return tmt.getCount();
    });
});
</script>

```

SP

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/expenseDetail/expenseDetailV02_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
    <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- expenseDetail v02 --%>
<c:set var="gadgetClass">expenseDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">expenseDetailV02Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">expenseDetailTable</c:set>

<c:set var="gadgetResponse" value="{kimw.load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>
<c:set var="gadget" value="{appDto.gadgets[gadgetId]}" />

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceOccurDate">${gadget.surfaces["occurDate"]}</c:set>
<c:set var="surfaceSummary">${gadget.surfaces["summary"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>

```

```

<:set var="surfaceBack">{gadget:surfaceBack} project }</:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconp
<h3>
  <span>${surfaceChapterTitle}</span>
  <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="{fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>

<ul data-role="listview" data-divider-theme="b">
  <c:forEach items="{gadgetResponse.data[tmtableId]}" var="detailData">
    <li>
      <span class="ui-li-heading"><c:out value="{detailData.occureDate}"></c:out></span>
      <span class="ui-li-aside"><c:out value="{detailData.companyAmount[1]}"></c:out></span>
      <div class="indent">
        <p class="kaiden-ui-overview-contents"><c:out value="{detailData.summary}"></c:out></p>
        <p class="kaiden-ui-overview-contents-sub"><c:out value="{detailData.payTypeName}"></c:out></p>
      </div>
    </li>
    <li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="{surfaceBack}"></c:out></a></li>
    <li>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceOccureDate}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.occureDate}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceSummary}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.summary}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceTransAmount}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents-sub"><c:out value="{detailData.transAmount[1]}"></c:out></label>
        <label class="kaiden-ui-detail-item-contents-sub" style="margin-left:1em"><c:out value="{detailData.transCurrencyName}"></c:out></label>
      </div>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents-sub kaiden-ui-cmpRate"><c:out value="{detailData.companyRate[1]}"></c:out></label>
      </div>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><strong><c:out value="{detailData.companyAmount[1]}"></c:out></strong></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceExpType}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.expTypeName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceAccount}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.accountName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceTaxType}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.taxTypeName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfacePayType}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.payTypeName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfacePayee}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="{detailData.payeeName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceDepartment}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.departmentName}"></c:out></label>
      </div>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceProject}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.projectName}"></c:out></label>
      </div>
    </li>
  </ul>
</li>
</c:forEach>
</ul>
</div>

```

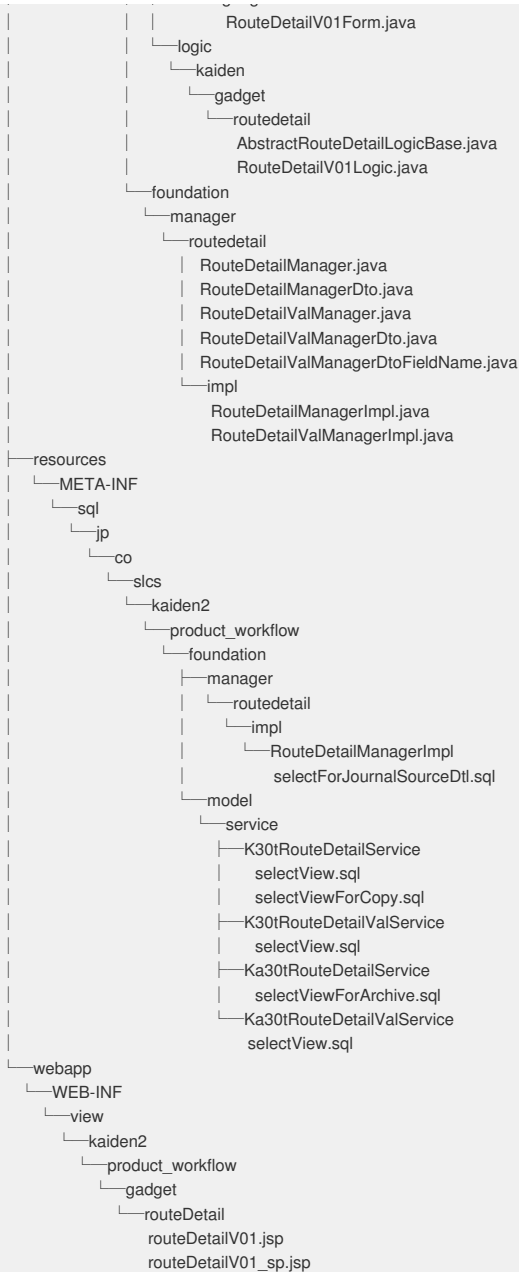
旅程明細

旅程明細のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action \(基底\)](#)
 - [Action](#)
 - [Form](#)
 - [Logic \(基底\)](#)
 - [Logic](#)
 - [FieldName](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
 - [コピー](#)
 - [仕訳連携](#)
 - [アーカイブ](#)
- [JSP](#)
 - [PC](#)
 - [SP](#)

[一覧](#)





Java

Action (基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractRouteDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

```

/**

```

* 旅程明細がジェットActionの抽象クラス。
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public abstract class AbstractRouteDetailActionBase implements GadgetActionIF {

    /**
     * 対象ロジッククラスをサブクラスより返却。
     *
     * @return ロジッククラス
     */
    @Override
    public abstract PluginLogicIF getLogic();

    /**
     * Fromをサブクラスより返却。
     *
     * @return From
     */
    @Override
    public abstract AbstractImwPageForm getForm();

    /**
     * 申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute validator = false
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute validator = false
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false

```

```

public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConvector());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvector())));
    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {

```



```

        ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
        return null;
    }

    /**
     * 書式変換設定を返却.
     * <p>
     * 標準では{@link #RouteDetailConverter 旅程明細 (標準) ガジェット}の書式変換を返却します。<br>
     * 書式を変更する場合は、サブクラスでオーバーライドしてください
     * </p>
     *
     * @return 書式変換後データ
     */
    protected ConvertForJsonSource getConverter() {
        return new RouteDetailConverter(getForm());
    }

    /**
     * 旅程明細ガジェットの書式変換.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    private static class RouteDetailConverter extends ConvertForJsonSource {

        /** リクエストパラメータ。 */
        private AbstractImwPageForm frm;

        /** NumberHelper。 */
        private NumberHelper numberHelper;

        /**
         * コンストラクタ.
         *
         * @param frm リクエストパラメータ
         */
        public RouteDetailConverter(AbstractImwPageForm frm) {
            this.frm = frm;
            numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
        }

        /**
         * {@inheritDoc}
         * <p>
         * 本サブクラスでは、以下の書式変換を行う。<br>
         * <table border="1">
         * <tr>
         * <td>項目名</td>
         * <td>書式</td>
         * </tr>
         * <tr>
         * <td>取引通貨</td>
         * <td>スケールを通貨のスケールに変換</td>
         * </tr>
         * <tr>
         * <td>レート</td>
         * <td>スケールをレートスケールに変換</td>
         * </tr>
         * <tr>
         * <td>換算額</td>
         * <td>スケールを会社通貨のスケールに変換</td>
         * </tr>
         * <tr>
         * <td>会社通貨額合計</td>
         * <td>スケールを会社通貨のスケールに変換</td>
         * </tr>
         * </table>
         * </p>
         */
        @Override
        protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

            Map<String, Object> result = new HashMap<String, Object>(map);

            String currencyCd = (String) result.get(ROUTE_DETAIL_TRANS_CURRENCY_CD);
            String companyCurrencyCd = (String) result.get(ROUTE_DETAIL_COMPANY_CURRENCY_CD);
            // 取引通貨額の書式変換
            if (result.containsKey(ROUTE_DETAIL_TRANS_AMOUNT)) {
                BigDecimal amount = (BigDecimal) result.get(ROUTE_DETAIL_TRANS_AMOUNT);
                result.put(ROUTE_DETAIL_TRANS_AMOUNT,
                    numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, currencyCd));
            }
            // レートの書式変換
            if (result.containsKey(ROUTE_DETAIL_COMPANY_RATE)) {
                BigDecimal rate = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_RATE);
                result.put(ROUTE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
            }
        }
    }

```

```

// 換算額の書式変換
if (result.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT)) {
    BigDecimal companyAmount = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_AMOUNT);
    result.put(ROUTE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
        frm.searchCriteriaCompany, companyCurrencyCd));
}

// 会社通貨額合計の書式変換
if (result.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT_SUM)) {
    BigDecimal companyAmountSum = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_AMOUNT_SUM);
    result.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4Input(companyAmountSum,
        frm.searchCriteriaCompany, companyCurrencyCd));
}

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>換算額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額合計</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * <p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(ROUTE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(ROUTE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(ROUTE_DETAIL_TRANS_CURRENCY_CD);
        result.put(ROUTE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_RATE);
        result.put(ROUTE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }

    Object companyCurrencyCd = map.get(ROUTE_DETAIL_COMPANY_CURRENCY_CD);

    // 換算額の書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_AMOUNT);
        result.put(ROUTE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    // 会社通貨額合計の書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_AMOUNT_SUM);
        result.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4View(companyAmountSum,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}
}

```

Action

■ Path

```
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/RouteDetailV01Action.java
```

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.RouteDetailV01Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail.RouteDetailV01Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 旅程明細(標準)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV01Action extends AbstractRouteDetailActionBase {

    /**
     * RouteDetailV01Logic.
     */
    @Resource
    public RouteDetailV01Logic routeDetailV01Logic;

    /**
     * RouteDetailV01Form.
     */
    @ActionForm
    @Resource
    public RouteDetailV01Form routeDetailV01Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return routeDetailV01Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return routeDetailV01Form;
    }
}

```

Form

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/RouteDetailV01Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 旅程明細(標準)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV01Form extends AbstractImwPageForm {

}

```

Logic(基底)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/routedetail/AbstractRouteDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNameNames.GADGET_CLASS;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ARRIVAL_SPOT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUND_TRIP_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUTE_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_STARTING_SPOT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_SUPPLEMENT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.IdentifierHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.FlagValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;

```

```

import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

/**
 * 旅程ガジェット用ロジックの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractRouteDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 旅程明細(ヘッダー)。 */
    protected static final String BLOCK_KEY_ROUTE_INFO_TABLE = "routeInfoTable";
    /** ガジェットブロックキー: 旅程明細(明細)。 */
    protected static final String BLOCK_KEY_ROUTE_DETAIL_TABLE = "routeDetailTable";

    /** 旅程日付項目長。 */
    public static final int ROUTE_DATE_LEN = 100;
    /** 出発地項目長。 */
    public static final int STARTING_SPOT_LEN = 250;
    /** 到着地項目長。 */
    public static final int ARRIVAL_SPOT_LEN = 250;
    /** 補足項目長。 */
    public static final int SUPPLEMENT_LEN = 1000;
    /** 経路検索id項目長。 */
    public static final int ROUTE_SEARCH_ID_LEN = 100;
    /** 税率(整数部)。 */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部)。 */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部)。 */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部)。 */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先項目長。 */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部)。 */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部)。 */
    public static final int QUANTITY_DEC_LEN = 10;

    /** 旅程明細. 予備項目A~Jの最大長。 */
    public static final int ROUTE_EXTENSION_COMON_LEN = 250;

    /** RouteDetailManager。 */
    @Resource
    protected RouteDetailManager routeDetailManager;

    /** RouteDetailValManager。 */
    @Resource
    protected RouteDetailValManager routeDetailValManager;

    /** JournalSourceDtlManager。 */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper。 */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** IdentifierHelper。 */
    @Resource
    protected IdentifierHelper identifierHelper;

    /** コンストラクタ。 */
    public AbstractRouteDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     *
     * <p>
     * 旅程明細を登録します。
     *
     * </p>
     */
}

```

```

    /
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる旅程明細テーブルを削除後に、旅程明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、旅程明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、旅程明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */

```

```

</p>
*/
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 旅程明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}

```

```

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細を登録します。
     * </p>
     */
    @Override
    public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細テーブルの対象データを削除します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTable(true);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる旅程明細テーブルを削除後に、旅程明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
     * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
     * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
     * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
     * </p>
     */
    @Override
    public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

        PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
        MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
            pluginGadgetDTO.imwPluginParameterDto);

        String status = matterEndProcessDto.lastResultStatus;

        int rv = 0;

        if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

            if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
                // 旅程明細データ取得キー
                CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
                criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
                criteriaDto.userId = pluginGadgetDTO.userId;
                criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
                criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
                // 旅程明細データを取得
                List<BeanMap> beanMapList = routeDetailManager.selectForJournalSourceDtl(criteriaDto);
                List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                    beanMapList);

                // 仕訳連携用明細テーブル登録処理
                rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
            }
        }

        return rv;
    }
}

```



```

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる旅程明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc} *
 * <p>
 * 対象となる(過去案件)旅程明細テーブルを一括削除します。
 * </p>
 */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    CriteriaForSqlDto criteria = super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().imwPluginParameterDto);
    routeDetailValManager.removeAllArchive(criteria);
    int rv = routeDetailManager.removeAllArchive(criteria);
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる旅程明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 *
 * @see #viewMapFormatted
 */
@Override
protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    return viewMapFormatted(conditionDto, null);
}

/**
 * {@inheritDoc}
 * <p>
 * データの取得は、{@link RouteDetailManager#selectList(RouteDetailManagerDto)}を利用します。
 * </p>
 */
@Override
protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
    throws KaidenSystemException {

    RouteDetailManagerDto dto = Beans.createAndCopy(RouteDetailManagerDto.class, conditionDto).execute();
    List<RouteDetailManagerDto> listSelect = routeDetailManager.selectList(dto);
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
    List<Map<String, Object>> lstDtoMap = new ArrayList<Map<String, Object>>();
    for (RouteDetailManagerDto src : listSelect) {
        if (converter == null) {
            lstDtoMap.add(BeanUtil.toMap(src, true));
        } else {
            lstDtoMap.add(converter.mapConvert(BeanUtil.toMap(src, true)));
        }
    }
    rv.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, lstDtoMap);
    return rv;
}

/**
 * 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ登録処理。
 * <D>

```

```

* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は登録対象外です。
* <p>
*
*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int insertTable() throws KaidenSystemException {
    // 旅程明細ガジェットデータをMap形式で取得
    List<RouteDetailManagerDto> dtos = JsonUtil.getDtoList(RouteDetailManagerDto.class,
        getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE));
    for (RouteDetailManagerDto dto : dtos) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {
            dto.routeSearchId = identifierHelper.get();

            // 旅程明細駅すばあと検索テーブルへの登録
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            valTarget.commutationRoute = dto.commutationRoute;
            valTarget.searchCondition = dto.searchCondition;
            valTarget.routeInfo = dto.routeInfo;
            valTarget.assessDate = dto.assessDate;
            valTarget.tranceMeansFlag = dto.tranceMeansFlag;
            valTarget.assessFlag = dto.assessFlag;
            routeDetailValManager.entryIgnoreAuthCompany(valTarget);
        }
    }
    // 旅程明細テーブル登録
    return routeDetailManager.entry(dtos);
}

/**
* 仕訳連携用明細テーブルデータ登録処理.
*
* @param srcList 旅程明細テーブルデータリスト
* @param matterEndProcessDto 案件終了処理用DTO
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
    MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
    for (JournalSourceDtlManagerDto srcDto : srcList) {
        // 仕訳データ区分をセット
        srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
    }

    // 仕訳連携用明細テーブル登録
    return journalSourceDtlManager.entry(srcList);
}

/**
* 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ削除処理.
* <p>
* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は、削除対象外です。
* <p>
*
* @param tempFlg 一時保存データフラグ
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    RouteDetailManagerDto mngDto = new RouteDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userId = super.getGadgetDto().userId;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    List<RouteDetailManagerDto> detailList = routeDetailManager.selectList(mngDto);
    for (RouteDetailManagerDto dto : detailList) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {

            // 旅程明細駅すばあと検索テーブルの削除
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            routeDetailValManager.removeIgnoreAuthCompany(valTarget);
        }
    }
    return routeDetailManager.remove(mngDto);
}

/**
* 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ削除処理(履歴を含む).
* <p>
* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は、削除対象外です。
* <p>

```

```

*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    RouteDetailManagerDto mngDto = new RouteDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    List<RouteDetailManagerDto> detailList = routeDetailManager.selectList(mngDto);
    for (RouteDetailManagerDto dto : detailList) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {

            // 旅程明細駅すばあと検索テーブルの削除
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            routeDetailValManager.removeIgnoreAuthCompany(valTarget);
        }
    }
    return routeDetailManager.removeAll(mngDto);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {

```

```

        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, true);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, true);
    }

    /**
     * フィールドバリデート定義.
     * <p>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>旅程日付</td>
     * <td>{@link DateValidator 日付型チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>出発地</td>
     * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>到着地</td>
     * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
     * </tr>
     * </table>
     *

```

```

* <td align="center">入力</td>
* </tr>
* <tr>
* <td>補足</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>往復フラグ</td>
* <td>{@link FlagValidator フラグチェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>経費区分コード</td>
* <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>取引通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>取引通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td rowspan="2">会社通貨換算レート</td>
* <td>{@link DigitsValidator 小数型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>会社通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数型チェック}</td>
* <td align="center">入力</td>
* </tr>

```

```

* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    if (fieldDefsMap.containsKey(BLOCK_KEY_ROUTE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_ROUTE_DETAIL_TABLE, tmpFlg);
        // 旅程日付
        addValidateOnlyInput(ROUTE_DETAIL_ROUTE_DATE, VALID_DATE);
        // 出発地
        addValidateOnlyInput(ROUTE_DETAIL_STARTING_SPOT, VALID_MAX_LENGTH, getArgs(STARTING_SPOT_LEN));
        // 到着地
        addValidateOnlyInput(ROUTE_DETAIL_ARRIVAL_SPOT, VALID_MAX_LENGTH, getArgs(ARRIVAL_SPOT_LEN));
        // 補足
        addValidateOnlyInput(ROUTE_DETAIL_SUPPLEMENT, VALID_MAX_LENGTH, getArgs(SUPPLEMENT_LEN));
        // 往復フラグ
        addValidateOnlyInput(ROUTE_DETAIL_ROUND_TRIP_FLAG, VALID_FLAG);
        // 経費区分コード
        addValidate(ROUTE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, ROUTE_DETAIL_EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidate(ROUTE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ROUTE_DETAIL_ACCOUNT_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidate(ROUTE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, ROUTE_DETAIL_TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateOnlyInput(ROUTE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 取引通貨コード
        addValidate(ROUTE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, ROUTE_DETAIL_TRANS_CURRENCY_CD,
            getGadgetDto().searchCriteriaCompany);
        // 会社通貨換算レート
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_RATE, VALID_DIGITS,
            getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨コード
        addValidate(ROUTE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, ROUTE_DETAIL_COMPANY_CURRENCY_CD,
            getGadgetDto().searchCriteriaCompany);
        // 会社通貨額
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 支払分類コード
        addValidate(ROUTE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, ROUTE_DETAIL_PAY_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidate(ROUTE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, ROUTE_DETAIL_PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払先名
        addValidateOnlyInput(ROUTE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
        // 組織コード
        addValidate(ROUTE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCompany,
            ROUTE_DETAIL_DEPARTMENT_SET_CD, ROUTE_DETAIL_DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidate(ROUTE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, ROUTE_DETAIL_PROJECT_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 数量
        addValidateOnlyInput(ROUTE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
        // 会社通貨額合計
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
    }
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldKey, validationId, args);
}

```

```

/**
 * ブロックの共通シングルバリデート定義MAPにエンTRIESを追加.
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 *
 * @param fieldKey フィールドキー
 * @param validationId バリデートキー
 * @param args バリデート実行時アーギュメント
 */
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
 * 設定されている場合は、{@link RouteDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, new ArrayList<RouteDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectView(criteria));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link RouteDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link RouteDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した旅程明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_department)</li>
 * <li>プロジェクトマスタ</li>
 * <li>支払分類マスタ</li>
 * <li>支払方法マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * <p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_ROUTE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<RouteDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), RouteDetailManagerDto.class);
        }
    }
}

```

```

for (RouteDetailManagerDto j : sqlResultList) {
    // 取引通貨チェック
    if (isNotEmpty(j.transCurrencyCd) && isEmpty(j.transCurrencyName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_TRANS_CURRENCY_CD, ROUTE_DETAIL_TRANS_CURRENCY_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.CURRENCY);
        String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 会社通貨チェック
    if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, ROUTE_DETAIL_COMPANY_CURRENCY_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.CURRENCY);
        String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 組織チェック
    if (isNotEmpty(j.departmentSetCd) && isEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_DEPARTMENT_SET_CD, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_DEPARTMENT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.DEPARTMENT);
        String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // プロジェクトチェック
    if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PROJECT_CD, ROUTE_DETAIL_PROJECT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PROJECT);
        String[] values = getArgs(j.projectCd, j.projectName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 支払分類マスタチェック
    if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PAY_TYPE_CD, ROUTE_DETAIL_PAY_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PAY_TYPE);
        String[] values = getArgs(j.payTypeCd, j.payTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 支払方法マスタチェック
    if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PAY_METHOD_CD, ROUTE_DETAIL_PAY_METHOD_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PAY_METHOD);
        String[] values = getArgs(j.payMethodCd, j.payMethodName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 経費区分マスタチェック
    if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_EXP_TYPE_CD, ROUTE_DETAIL_EXP_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.EXP_TYPE);
        String[] values = getArgs(j.expTypeCd, j.expTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
            // 勘定科目、税区分をクリア
            j.accountCd = null;
            j.accountName = null;
            j.taxTypeCd = null;
            j.taxTypeName = null;
            // 税率、税抜金額、税額をクリア
            j.taxRate = null;
            j.companyAmountExcTax = null;
            j.companyAmountTax = null;
        }
    }
}

```



```

        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 勘定科目マスタチェック
if (isNotEmpty(j.accountCd) && isEmpty(j.accountName)) {
    String[] fieldId = getArgs(ROUTE_DETAIL_ACCOUNT_CD, ROUTE_DETAIL_ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(j.accountCd, j.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 税区分マスタチェック
if (isNotEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
    String[] fieldId = getArgs(ROUTE_DETAIL_TAX_TYPE_CD, ROUTE_DETAIL_TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}
}
}
}
return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 旅程明細テーブルの対象データを過去案件退避します。<br>
 * 駅すばあと連携している場合は、旅程明細駅すばあと検索情報テーブルも過去案件退避の対象となります。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    // 旅程明細テーブル
    @SuppressWarnings("unchecked")
    List<RouteDetailManagerDto> listRoute = (List<RouteDetailManagerDto>) dataMap.get(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    int rv = routeDetailManager.matterArchiveRegister(criteria, listRoute);

    List<RouteDetailValManagerDto> listVal = new ArrayList<RouteDetailValManagerDto>();
    for (RouteDetailManagerDto j : listRoute) {
        if (StringUtil.isEmpty(j.routeSearchId)) {
            RouteDetailValManagerDto val = new RouteDetailValManagerDto();
            val.routeSearchId = j.routeSearchId;
            val.routeSearchId = j.routeSearchId;
            val.commutationRoute = j.commutationRoute;
            val.searchCondition = j.searchCondition;
            val.routeInfo = j.routeInfo;
            val.assessDate = j.assessDate;
            val.tranceMeansFlag = j.tranceMeansFlag;
            val.assessFlag = j.assessFlag;
            val.localeId = criteria.localeId;
            listVal.add(val);
        }
    }
    routeDetailValManager.matterArchiveRegister(criteria, listVal);

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 固有処理として、対象となる旅程明細駅すばあと検索情報テーブルのデータを削除します。
 * </p>
 */
@Override
protected void matterArchivePost(KaidenDto dto, CriteriaForSqlDto criteria) {
    routeDetailValManager.matterArchivePostDelete(criteria);
}

```

```

routeDetailManager.getMatterEnv().delete(criteria);
    super.matterArchivePost(dto, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(routeDetailManager);
    return list;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Deprecated
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();
    list.add(SYSTEM_MATTER_ID);
    list.add(USER_DATA_ID);
    list.add(GADGET_CLASS);
    list.add(GADGET_INSTANCE);
    return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccountIF#getClaimExpenccount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

```

```

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG);
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmtnt)) {
                sum = sum.add(new BigDecimal(strAmtnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmtnt)) {
            list.add(new BigDecimal(strAmtnt));
        }
    }

    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmtnt)) {
            sum = sum.add(new BigDecimal(strAmtnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * 往復フラグに基づく数量を返却.
 *
 * @param roundTripFlag 往復フラグ
 * @return 往復フラグに基づく数量
 */
protected BigDecimal getQuantityFromRoundTrip(String roundTripFlag) {
    if (GenericFlag.isOff(roundTripFlag)) {
        return BigDecimal.ONE;
    } else if (GenericFlag.isOn(roundTripFlag)) {

```

```

        return new BigDecimal("2");
    }
    return null;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/routedetail/RouteDetailV01Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUND_TRIP_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

/**
 * 旅程 (標準) ガジェット (Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV01Logic extends AbstractRouteDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public RouteDetailV01Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     *
     * <p>
     * サブクラスで追加定義します。<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     *

```

```

* <tr>
* <td>会社通貨額</td>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード/プロジェクトコード</td>
* <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*/

@Override
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    super.defineValidateForNormal(params, tmpFlg);

    // 取引通貨額
    addValidateOnlyInput(ROUTE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
    // 会社通貨額
    addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

    if (!tmpFlg) {
        // 組織コード
        addValidateOnlyInput(ROUTE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_PROJECT_CD);
        // プロジェクトコード
        addValidateOnlyInput(ROUTE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_PROJECT_CD);
    }
}

/**
 * {@inheritDoc}
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>取引通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>会社通貨換算レート: 固定値: 1を設定</li>
 * <li>数量: 往復フラグに基づく数値を設定</li>
 * <li>立替フラグ: 支払分類コードに紐づく立替フラグを支払分類マスタから取得して設定</li>
 * </ul>
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {

            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(ROUTE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 取引通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(ROUTE_DETAIL_TRANS_CURRENCY_CD, companyCurrencyCd);

            // 会社通貨換算レート
            j.put(ROUTE_DETAIL_COMPANY_RATE, "1");

            // 数量
            BigDecimal roundTripFlag = getQuantityFromRoundTrip((String) j.get(ROUTE_DETAIL_ROUND_TRIP_FLAG));
            j.put(ROUTE_DETAIL_QUANTITY, StringUtil.toString(roundTripFlag));

            // 立替フラグ
            if (StringUtil.isNotEmpty((String) j.get(ROUTE_DETAIL_PAY_TYPE_CD))) {
                PayTypeManagerDto criteria = new PayTypeManagerDto();
                criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
                criteria.payTypeCd = (String) j.get(ROUTE_DETAIL_PAY_TYPE_CD);
                criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                    .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
                List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                    criteria, new ArrayList<OrderDto>());
                if (CollectionUtil.isEmpty(retList)) {
                    j.put(ROUTE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
                }
            }
        }
    }
}

```

```

    }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * <p>
 *
 * @return グローバルデータ補完処理
 */

@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * <p>
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(ROUTE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(ROUTE_DETAIL_TRANS_AMOUNT),
                (String) j.get(ROUTE_DETAIL_COMPANY_RATE)));
            j.put(ROUTE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(ROUTE_DETAIL_QUANTITY)));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * <p>
 * <ul>
 * <li>支払方法コード:ガジェット間連携項目、支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード

```

```

        if (gcPayMethodCd == null) {
            gcPayMethodCd = getGadgetCouplingPayMethod();
        }
        j.put(ROUTE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
    }
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenCount(Map<String, Object> params) {
    return super.putClaimExpenCount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/routedetail/RouteDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 旅程明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class RouteDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected RouteDetailFieldName() {

        /** 旅程日付. */
        public static final String ROUTE_DETAIL_ROUTE_DATE = "routeDate";
        /** 出発地. */
        public static final String ROUTE_DETAIL_STARTING_SPOT = "startingSpot";
        /** 到着地. */
        public static final String ROUTE_DETAIL_ARRIVAL_SPOT = "arrivalSpot";
        /** 補足. */
        public static final String ROUTE_DETAIL_SUPPLEMENT = "supplement";
        /** 経費区分コード. */
        public static final String ROUTE_DETAIL_EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String ROUTE_DETAIL_ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String ROUTE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String ROUTE_DETAIL_TAX_RATE = "taxRate";
        /** 数量. */
        public static final String ROUTE_DETAIL_QUANTITY = "quantity";
        /** 取引通貨コード. */
        public static final String ROUTE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String ROUTE_DETAIL_TRANS_AMOUNT = "transAmount";
        /** 会社通貨換算レート. */
    }
}

```

```

public static final String ROUTE_DETAIL_COMPANY_RATE = "companyRate";
/** 会社通貨コード. */
public static final String ROUTE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
/** 会社通貨額. */
public static final String ROUTE_DETAIL_COMPANY_AMOUNT = "companyAmount";
/** 会社通貨税抜額. */
public static final String ROUTE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
/** 会社通貨税額. */
public static final String ROUTE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
/** 会社通貨額合計. */
public static final String ROUTE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
/** 支払分類コード. */
public static final String ROUTE_DETAIL_PAY_TYPE_CD = "payTypeCd";
/** 支払方法コード. */
public static final String ROUTE_DETAIL_PAY_METHOD_CD = "payMethodCd";
/** 支払先コード. */
public static final String ROUTE_DETAIL_PAYEE_CD = "payeeCd";
/** 支払先名. */
public static final String ROUTE_DETAIL_PAYEE_NAME = "payeeName";
/** 組織セットコード. */
public static final String ROUTE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
/** 組織コード. */
public static final String ROUTE_DETAIL_DEPARTMENT_CD = "departmentCd";
/** プロジェクトコード. */
public static final String ROUTE_DETAIL_PROJECT_CD = "projectCd";
/** 往復フラグ. */
public static final String ROUTE_DETAIL_ROUND_TRIP_FLAG = "roundTripFlag";
/** 経路情報連携済フラグ. */
public static final String ROUTE_DETAIL_ROUTE_LINK_FLAG = "routeLinkFlag";
/** 経路検索id. */
public static final String ROUTE_DETAIL_ROUTE_SEARCH_ID = "routeSearchId";

/** 会社名. */
public static final String ROUTE_DETAIL_COMPANY_NAME = "companyName";
/** 経費区分名称. */
public static final String ROUTE_DETAIL_EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String ROUTE_DETAIL_ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String ROUTE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
/** 取引通貨名称. */
public static final String ROUTE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String ROUTE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 支払分類名称. */
public static final String ROUTE_DETAIL_PAY_TYPE_NAME = "payTypeName";
/** 支払方法名称. */
public static final String ROUTE_DETAIL_PAY_METHOD_NAME = "payMethodName";
/** 組織セット名称. */
public static final String ROUTE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
/** 組織名称. */
public static final String ROUTE_DETAIL_DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String ROUTE_DETAIL_PROJECT_NAME = "projectName";
/** 立替フラグ. */
public static final String ROUTE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailManager.java


```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 旅程明細テーブルマネージャークラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface RouteDetailManager extends PluginGadgetManager<RouteDetailManagerDto> {

    /** {@link #selectView(CriteriaForSqlDto) 表示案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link #selectViewForCopy(CriteriaForSqlDto) コピー案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link #selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /** {@link #selectViewArchive(CriteriaForSqlDto) 過去案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NMF_FOR_ARCHIVE = "selectViewForArchive.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;

/**
 * 旅程明細駅すばあと検索情報テーブルへの各種操作を提供するAPIのinterface.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface RouteDetailValManager {

    /** {@link #matterArchivePostDelete(CriteriaForSqlDto) 過去案件一括削除}}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /**
     * 主キー検索.
     * <p>
     * 検索条件のうち、主キー(経路検索id)のみを利用して問い合わせします.<br>
     * 主キー条件が未設定の場合、RunTimeExceptionがスローされます。
     * </p>
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public RouteDetailValManagerDto selectIgnoreAuthCompany(RouteDetailValManagerDto criteria);

    /**
     * 登録.
     *
     * @param data 登録データ
     * @return 更新件数
     */
    public int entryIgnoreAuthCompany(RouteDetailValManagerDto data);

    /**
     * 削除.
     *
     * @param criteria 削除条件
     * @return 削除件数
     */
    public int removeIgnoreAuthCompany(RouteDetailValManagerDto criteria);

    /**
     * 過去案件登録.
     * <p>
     * 引数の登録データリストを過去案件用テーブルへ移送します.<br>
     * </p>
     *
     * @param dataList 登録データリスト
     * @param criteria 削除データ
     * @return 削除件数
     */
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<RouteDetailValManagerDto> dataList);

    /**
     * 過去案件処理時削除.
     * <p>
     * 引数のDTOのシステム案件IDに該当するデータを全件削除します.<br>
     * </p>
     *
     * @param criteria 削除データ
     * @return 削除件数
     */
    public int matterArchivePostDelete(CriteriaForSqlDto criteria);

    /**
     * 過去案件一括削除.
     * <p>
     * 引数の削除データリストを全件削除します.<br>
     * </p>
     *
     * @param criteria 削除条件
     * @return 削除件数
     */
    public int removeAllArchive(CriteriaForSqlDto criteria);
}

```

Manager(実装)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailManagerImpl.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.impl;

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.LOCALE_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tRouteDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tRouteDetailService;
import org.seasar.util.beans.util.BeanMap;
```

```

/*
 * 旅程明細テーブルへの各種操作を提供するAPI.
 *
 * 本APIは、主にデータベース操作を行いますが、より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link RouteDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link RouteDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>表示案件検索</td>
 * <td>selectView</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>コピー案件検索</td>
 * <td>selectViewForCopy</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件検索</td>
 * <td>selectViewArchive</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括挿入</td>
 * <td>entry</td>
 * <td>登録</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括更新</td>
 * <td>renew</td>
 * <td>更新</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>削除</td>
 * <td>remove</td>
 * <td>削除</td>
 * <td><b>&nbsp;</b></td>
 * </tr>

```

[illegible]

[illegible]

```

public class RouteDetailManagerImpl extends AbstractWifiManager<RouteDetailManagerDto> implements RouteDetailManager {

    /** K30tRouteDetailService. */
    @Resource
    protected K30tRouteDetailService k30tRouteDetailService;

    /** Ka30tRouteDetailService. */
    @Resource
    protected Ka30tRouteDetailService ka30tRouteDetailService;

    /**
     * {@inheritDoc}
     */
    @Override
    public RouteDetailManagerDto selectByPk(RouteDetailManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k30tRouteDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<RouteDetailManagerDto> selectList(RouteDetailManagerDto criteria) {
        return super.selectPkIgnoreAuthority(k30tRouteDetailService, criteria);
    }
}

```

```

    return super.selectRouteDetailData(criteriaForSqlDto criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<RouteDetailManagerDto> selectView(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForView(k30tRouteDetailService, SELECT_SQLFILE_NM, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<RouteDetailManagerDto> selectViewForCopy(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForViewForCopy(k30tRouteDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 * <p>
 * (過去案件) 旅程明細駅すばあと検索情報テーブルも併せて取得するため、{@link RouteDetailManager#SELECT_SQLFILE_NMF_FOR_ARCHIVE SQLファイル}を利用します。
 * </p>
 */
@Override
public List<RouteDetailManagerDto> selectViewArchive(criteriaForSqlDto criteria) {
    // CriteriaForSqlDto入力チェック
    String[] fieldNames = {SYSTEM_MATTER_ID, USER_DATA_ID, GADGET_CLASS, GADGET_INSTANCE, LOCALE_ID};
    this.validateCriteriaForSqlDto(criteria, fieldNames);
    criteria.archiveMonth = null;
    return super.selectBySqlIgnoreAuthority(ka30tRouteDetailService, SELECT_SQLFILE_NMF_FOR_ARCHIVE, criteria,
        ka30tRouteDetailService.makeOrderListFromPk());
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<RouteDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.insertBatch(k30tRouteDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int renew(List<RouteDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.renewBatch(k30tRouteDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(RouteDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tRouteDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(RouteDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tRouteDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(criteriaForSqlDto criteria, List<RouteDetailManagerDto> dataList) {
    return super.entryArchive(ka30tRouteDetailService, criteria, dataList);
}

```

```

    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataId(k30tRouteDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataIdForArchive(ka30tRouteDetailService, criteria);
    }
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailValManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.impl;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetailVal;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetailVal;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tRouteDetailValService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tRouteDetailValService;
import org.seasar.framework.beans.util.Beans;

/**
 * 旅程明細駅すばあと検索情報テーブルへの各種操作を提供するAPI.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailValManagerImpl implements RouteDetailValManager {

    /** K30tRouteDetailValService. */
    @Resource
    protected K30tRouteDetailValService k30tRouteDetailValService;

    /** Ka30tRouteDetailValService. */
    @Resource
    protected Ka30tRouteDetailValService ka30tRouteDetailValService;

    /**
     * {@inheritDoc}
     */
    @Override
    public RouteDetailValManagerDto selectIgnoreAuthCompany(RouteDetailValManagerDto criteria) {
        if (criteria == null || criteria.routeSearchId == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
        Map<String, Object> criteriaMap = BeanUtil.toMap(criteria);
        List<K30tRouteDetailVal> selectList = k30tRouteDetailValService.genericSelectList(criteriaMap,
            new ArrayList<OrderDto>());
        if (CollectionUtil.isEmpty(selectList)) {
            return null;
        }

        return Beans.createAndCopy(RouteDetailValManagerDto.class, selectList.get(0)).execute();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entryIgnoreAuthCompany(RouteDetailValManagerDto data) {
        if (data == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

```

```

        K30tRouteDetailVal insData = Beans.createAndCopy(K30tRouteDetailVal.class, data).execute();
        return k30tRouteDetailValService.insert(insData);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeIgnoreAuthCompany(RouteDetailValManagerDto criteria) {

        if (criteria == null || criteria.routeSearchId == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
        Map<String, Object> criteriaMap = BeanUtil.toMap(criteria);
        List<K30tRouteDetailVal> selectList = k30tRouteDetailValService.genericSelectList4Update(criteriaMap,
            new ArrayList<OrderDto>());
        if (CollectionUtil.isEmpty(selectList)) {
            return 0;
        }
        return k30tRouteDetailValService.delete(selectList.get(0));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<RouteDetailValManagerDto> dataList) {
        List<Ka30tRouteDetailVal> castListContent = new ArrayList<Ka30tRouteDetailVal>();
        for (RouteDetailValManagerDto j : dataList) {
            Ka30tRouteDetailVal data = Beans.createAndCopy(Ka30tRouteDetailVal.class, j).execute();
            data.localeId = criteria.localeId;
            data.archiveMonth = criteria.archiveMonth;
            castListContent.add(data);
        }
        return CollectionUtil.summaryInt(k30tRouteDetailValService.insertBatch(castListContent));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(ImwEntityCommonFieldNames.SYSTEM_MATTER_ID, criteria.systemMatterId);
        map.put(ImwEntityCommonFieldNames.GADGET_CLASS, criteria.gadgetClass);
        map.put(ImwEntityCommonFieldNames.GADGET_INSTANCE, criteria.gadgetInstance);
        List<Ka30tRouteDetailVal> dataList = k30tRouteDetailValService.selectBySql(K30tRouteDetailVal.class,
            SELECT_SQLFILE_NM, map, k30tRouteDetailValService.makeOrderListFromPk());
        return CollectionUtil.summaryInt(k30tRouteDetailValService.deleteBatch(dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(ImwEntityCommonFieldNames.SYSTEM_MATTER_ID, criteria.systemMatterId);
        List<Ka30tRouteDetailVal> dataList = k30tRouteDetailValService.selectBySql(Ka30tRouteDetailVal.class,
            SELECT_SQLFILE_NM, map, k30tRouteDetailValService.makeOrderListFromPk());
        return CollectionUtil.summaryInt(k30tRouteDetailValService.deleteBatch(dataList));
    }
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 旅程明細テーブルDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailManagerDto extends KaidenDto {

```



```

/** システム案件id. */
public String systemMatterId;
/** ユーザーデータid. */
public String userDataId;
/** ガジェットクラス. */
public String gadgetClass;
/** ガジェットインスタンス. */
public Long gadgetInstance;
/** タブルid. */
public Long tupleId;
/** ガジェットid. */
public String gadgetId;
/** ガジェットバージョン. */
public String gadgetVariation;
/** 行番号. */
public Long rowNumber;

/** 旅程日付. */
public Timestamp routeDate;
/** 出発地. */
public String startingSpot;
/** 到着地. */
public String arrivalSpot;
/** 補足. */
public String supplement;
/** 経費区分コード. */
public String expTypeCd;
/** 勘定科目コード. */
public String accountCd;
/** 税区分コード. */
public String taxTypeCd;
/** 税率. */
public BigDecimal taxRate;
/** 数量. */
public BigDecimal quantity;
/** 取引通貨コード. */
public String transCurrencyCd;
/** 取引通貨額. */
public BigDecimal transAmount;
/** 会社通貨換算レート. */
public BigDecimal companyRate;
/** 会社通貨コード. */
public String companyCurrencyCd;
/** 会社通貨額. */
public BigDecimal companyAmount;
/** 会社通貨税抜額. */
public BigDecimal companyAmountExcTax;
/** 会社通貨税額. */
public BigDecimal companyAmountTax;
/** 会社通貨額合計. */
public BigDecimal companyAmountSum;
/** 支払分類コード. */
public String payTypeCd;
/** 支払方法コード. */
public String payMethodCd;
/** 支払先コード. */
public String payeeCd;
/** 支払先名. */
public String payeeName;
/** 組織セットコード. */
public String departmentSetCd;
/** 組織コード. */
public String departmentCd;
/** プロジェクトコード. */
public String projectCd;
/** 往復フラグ. */
public String roundTripFlag;
/** 経路情報連携済フラグ. */
public String routeLinkFlag;
/** 経路検索id. */
public String routeSearchId;

/** 予備項目a. */
public String extensionA;
/** 予備項目b. */
public String extensionB;
/** 予備項目c. */
public String extensionC;
/** 予備項目d. */
public String extensionD;
/** 予備項目e. */
public String extensionE;
/** 予備項目f. */
public String extensionF;
/** 予備項目g. */
public String extensionG;

```

```

/** 予備項目h. */
public String extensionH;
/** 予備項目i. */
public String extensionI;
/** 予備項目j. */
public String extensionJ;
/** 登録日時. */
public Timestamp entryTs;
/** 登録者コード. */
public String entryUserCd;
/** 更新カウンタ. */
public Long renewCnt;
/** 更新日時. */
public Timestamp renewTs;
/** 更新者コード. */
public String renewUserCd;

/** 検索基準会社コード. */
public String searchCriteriaCompany;
/** 取引通貨名称. */
public String transCurrencyName;
/** 会社通貨名称. */
public String companyCurrencyName;
/** 組織名称. */
public String departmentName;
/** プロジェクトマスタプロジェクト名称. */
public String projectName;
/** 経費区分名称. */
public String expTypeName;
/** 勘定科目名称. */
public String accountName;
/** 税区分名称. */
public String taxTypeName;
/** 支払分類名称. */
public String payTypeName;
/** 支払方法名称. */
public String payMethodName;
/** 立替フラグ. */
public String paymentFlag;

/** 定期区間. */
public String commutationRoute;
/** 検索条件. */
public String searchCondition;
/** 経路情報. */
public String routeInfo;
/** 評価判定基準日. */
public Timestamp assessDate;
/** 利用交通機関フラグ. */
public String tranceMeansFlag;
/** 評価フラグ. */
public String assessFlag;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

/**
 * 旅程明細駅すばあと検索情報テーブルマネージャDTO.
 * <p>
 * マネージャクラス(@link RouteDetailValManager)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailValManagerDto extends KaidenMasterManagerDto {

    /** 経路検索id. */
    public String routeSearchId;
    /** 定期区間. */
    public String commutationRoute;
    /** 検索条件. */
    public String searchCondition;
    /** 経路情報. */
    public String routeInfo;
    /** 評価判定基準日. */
    public Timestamp assessDate;
    /** 利用交通機関フラグ. */
    public String tranceMeansFlag;
    /** 評価フラグ. */
    public String assessFlag;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;

    /** ロケールID. */
    public String localeId;
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityExtensionFieldNames;

/**
 * 旅程明細駅すばあと検索情報テーブルDTO フィールド名定義クラス。
 * <p>
 * マネージャクラス([@link RouteDetailValManager])へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class RouteDetailValManagerDtoFieldName implements EntityExtensionFieldNames {

    /**
     * コンストラクタ。
     */
    private RouteDetailValManagerDtoFieldName() {

    }

    /** 経路検索ID。 */
    public static final String ROUTE_SEARCH_ID = "routeSearchId";
    /** 定期区間。 */
    public static final String COMMUTATION_ROUTE = "commutationRoute";
    /** 検索条件。 */
    public static final String SEARCH_CONDITION = "searchCondition";
    /** 経路情報。 */
    public static final String ROUTE_INFO = "routeInfo";
    /** 評価判定基準日。 */
    public static final String ASSESS_DATE = "assessDate";
    /** 利用期間フラグ。 */
    public static final String TRANCE_MEANS_FLAG = "tranceMeansFlag";
    /** 評価フラグ。 */
    public static final String ASSESS_FLAG = "assessFlag";

}

```

Java (S2JDBC)

Entity

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 旅程明細テーブル。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_route_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tRouteDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID。 */
    private static final long serialVersionUID = 1L;

    /** システム案件id。 */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid。 */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス。 */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

```

```

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 旅程日付. */
@Column(name = "route_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp routeDate;

/** 出発地. */
@Column(name = "starting_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String startingSpot;

/** 到着地. */
@Column(name = "arrival_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String arrivalSpot;

/** 補足. */
@Column(name = "supplement", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String supplement;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

```

```

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 往復フラグ. */
@Column(name = "round_trip_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String roundTripFlag;

/** 経路情報連携済フラグ. */
@Column(name = "route_link_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String routeLinkFlag;

/** 経路検索id. */
@Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String routeSearchId;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailVal.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 旅程明細駅すばあと検索情報テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_route_detail_val")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tRouteDetailVal extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 経路検索id. */
    @Id
    @Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = false, unique = true)
    public String routeSearchId;

    /** 定期区間. */
    @Lob
    @Column(name = "commutation_route", columnDefinition = "clob", nullable = true, unique = false)
    public String commutationRoute;

    /** 検索条件. */
    @Lob
    @Column(name = "search_condition", columnDefinition = "clob", nullable = true, unique = false)
    public String searchCondition;

    /** 経路情報. */
    @Lob
    @Column(name = "route_info", columnDefinition = "clob", nullable = true, unique = false)
    public String routeInfo;

    /** 評価判定基準日. */
    @Column(name = "assess_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp assessDate;

    /** 利用交通機関フラグ. */
    @Column(name = "trance_means_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String tranceMeansFlag;

    /** 評価フラグ. */
    @Column(name = "assess_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String assessFlag;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

    /** 予備項目b. */
    @Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionB;

    /** 予備項目c. */
    @Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionC;

    /** 予備項目d. */
    @Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionD;

    /** 予備項目e. */
    @Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionE;

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)旅程明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_route_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tRouteDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** アーカイブ年月. */
    @Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
    public String archiveMonth;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;
}

```



```

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 旅程日付. */
@Column(name = "route_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp routeDate;

/** 出発地. */
@Column(name = "starting_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String startingSpot;

/** 到着地. */
@Column(name = "arrival_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String arrivalSpot;

/** 補足. */
@Column(name = "supplement", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String supplement;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

```

```

public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 往復フラグ. */
@Column(name = "round_trip_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String roundTripFlag;

/** 経路情報連携フラグ. */
@Column(name = "route_link_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String routeLinkFlag;

/** 経路検索id. */
@Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String routeSearchId;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */

```

```

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionG;

    /** 予備項目h. */
    @Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionH;

    /** 予備項目i. */
    @Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionI;

    /** 予備項目j. */
    @Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailVal.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)旅程明細駅すばあと検索情報テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_route_detail_val")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl"})
public final class Ka30tRouteDetailVal extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 経路検索id. */
    @Id
    @Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String routeSearchId;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** アーカイブ年月. */
    @Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
    public String archiveMonth;

    /** 定期区間. */
    @Lob
    @Column(name = "commutation_route", columnDefinition = "clob", nullable = true, unique = false)
    public String commutationRoute;

    /** 検索条件. */
    @Lob
    @Column(name = "search_condition", columnDefinition = "clob", nullable = true, unique = false)
    public String searchCondition;

    /** 経路情報. */
    @Lob
    @Column(name = "route_info", columnDefinition = "clob", nullable = true, unique = false)
    public String routeInfo;

    /** 評価判定基準日. */
    @Column(name = "assess_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp assessDate;

    /** 利用交通機関フラグ. */
    @Column(name = "trance_means_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String tranceMeansFlag;
}

```

```

/** 評価フラグ. */
@Column(name = "assess_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
public String assessFlag;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tRouteDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tRouteDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tRouteDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
}

```

```

*/
public static PropertyName<String> userDataId() {
    return new PropertyName<String>("userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public static PropertyName<String> gadgetClass() {
    return new PropertyName<String>("gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumのプロパティ名を返却.
 *
 * @return rowNumのプロパティ名
 */
public static PropertyName<Long> rowNum() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public static PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>("routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public static PropertyName<String> startingSpot() {
    return new PropertyName<String>("startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public static PropertyName<String> arrivalSpot() {
    return new PropertyName<String>("arrivalSpot");
}

```

```

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public static PropertyName<String> supplement() {
    return new PropertyName<String>("supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */

```

```

public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**

```

```

* departmentCdのプロパティ名を返却.
*
* @return departmentCdのプロパティ名
*/
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
* projectCdのプロパティ名を返却.
*
* @return projectCdのプロパティ名
*/
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
* roundTripFlagのプロパティ名を返却.
*
* @return roundTripFlagのプロパティ名
*/
public static PropertyName<String> roundTripFlag() {
    return new PropertyName<String>("roundTripFlag");
}

/**
* routeLinkFlagのプロパティ名を返却.
*
* @return routeLinkFlagのプロパティ名
*/
public static PropertyName<String> routeLinkFlag() {
    return new PropertyName<String>("routeLinkFlag");
}

/**
* routeSearchIdのプロパティ名を返却.
*
* @return routeSearchIdのプロパティ名
*/
public static PropertyName<String> routeSearchId() {
    return new PropertyName<String>("routeSearchId");
}

/**
* extensionAのプロパティ名を返却.
*
* @return extensionAのプロパティ名
*/
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
* extensionBのプロパティ名を返却.
*
* @return extensionBのプロパティ名
*/
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
* extensionCのプロパティ名を返却.
*
* @return extensionCのプロパティ名
*/
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
* extensionDのプロパティ名を返却.
*
* @return extensionDのプロパティ名
*/
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
* extensionEのプロパティ名を返却.
*
* @return extensionEのプロパティ名
*/
public static PropertyName<String> extensionE() {

```



```

    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *

```

```

    * @return renewUserCdのプロパティ名
    */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30tRouteDetailNames extends PropertyName<K30tRouteDetail> {

        /**
         * インスタンスを構築.
         */
        public _K30tRouteDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30tRouteDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30tRouteDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *
         * @return systemMatterIdのプロパティ名
         */
        public PropertyName<String> systemMatterId() {
            return new PropertyName<String>(this, "systemMatterId");
        }

        /**
         * userDataIdのプロパティ名を返却.
         *
         * @return userDataIdのプロパティ名
         */
        public PropertyName<String> userDataId() {
            return new PropertyName<String>(this, "userDataId");
        }

        /**
         * gadgetClassのプロパティ名を返却.
         *
         * @return gadgetClassのプロパティ名
         */
        public PropertyName<String> gadgetClass() {
            return new PropertyName<String>(this, "gadgetClass");
        }

        /**
         * gadgetInstanceのプロパティ名を返却.
         *
         * @return gadgetInstanceのプロパティ名
         */
        public PropertyName<Long> gadgetInstance() {
            return new PropertyName<Long>(this, "gadgetInstance");
        }

        /**
         * tupleIdのプロパティ名を返却.
         *
         * @return tupleIdのプロパティ名
         */
        public PropertyName<Long> tupleId() {
            return new PropertyName<Long>(this, "tupleId");
        }

        /**
         * gadgetIdのプロパティ名を返却.
         *
         * @return gadgetIdのプロパティ名
         */
    }

```

```

/
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>(this, "routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public PropertyName<String> startingSpot() {
    return new PropertyName<String>(this, "startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public PropertyName<String> arrivalSpot() {
    return new PropertyName<String>(this, "arrivalSpot");
}

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public PropertyName<String> supplement() {
    return new PropertyName<String>(this, "supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**

```

```

    * taxRateのプロパティ名を返却。
    *
    * @return taxRateのプロパティ名
    */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * quantityのプロパティ名を返却。
     *
     * @return quantityのプロパティ名
     */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却。
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却。
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyRateのプロパティ名を返却。
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却。
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyAmountのプロパティ名を返却。
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却。
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却。
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却。
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {

```

```

    /**
     * propertyNameをBigDecimalでcompanyAmountSumを返却。
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

```

```

    /**
     * payTypeCdのプロパティ名を返却。
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }

```

```

    /**
     * payMethodCdのプロパティ名を返却。
     *
     * @return payMethodCdのプロパティ名
     */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>(this, "payMethodCd");
    }

```

```

    /**
     * payeeCdのプロパティ名を返却。
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>(this, "payeeCd");
    }

```

```

    /**
     * payeeNameのプロパティ名を返却。
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>(this, "payeeName");
    }

```

```

    /**
     * departmentSetCdのプロパティ名を返却。
     *
     * @return departmentSetCdのプロパティ名
     */
    public PropertyName<String> departmentSetCd() {
        return new PropertyName<String>(this, "departmentSetCd");
    }

```

```

    /**
     * departmentCdのプロパティ名を返却。
     *
     * @return departmentCdのプロパティ名
     */
    public PropertyName<String> departmentCd() {
        return new PropertyName<String>(this, "departmentCd");
    }

```

```

    /**
     * projectCdのプロパティ名を返却。
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>(this, "projectCd");
    }

```

```

    /**
     * roundTripFlagのプロパティ名を返却。
     *
     * @return roundTripFlagのプロパティ名
     */
    public PropertyName<String> roundTripFlag() {
        return new PropertyName<String>(this, "roundTripFlag");
    }

```

```

    /**
     * routeLinkFlagのプロパティ名を返却。
     *
     * @return routeLinkFlagのプロパティ名
     */
    public PropertyName<String> routeLinkFlag() {
        return new PropertyName<String>(this, "routeLinkFlag");
    }

```

```

    /**
     * routeSearchIdのプロパティ名を返却。

```

```

*
* @return routeSearchIdのプロパティ名
*/
public PropertyName<String> routeSearchId() {
    return new PropertyName<String>(this, "routeSearchId");
}

/**
* extensionAのプロパティ名を返却。
*
* @return extensionAのプロパティ名
*/
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
* extensionBのプロパティ名を返却。
*
* @return extensionBのプロパティ名
*/
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
* extensionCのプロパティ名を返却。
*
* @return extensionCのプロパティ名
*/
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
* extensionDのプロパティ名を返却。
*
* @return extensionDのプロパティ名
*/
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
* extensionEのプロパティ名を返却。
*
* @return extensionEのプロパティ名
*/
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
* extensionFのプロパティ名を返却。
*
* @return extensionFのプロパティ名
*/
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
* extensionGのプロパティ名を返却。
*
* @return extensionGのプロパティ名
*/
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
* extensionHのプロパティ名を返却。
*
* @return extensionHのプロパティ名
*/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
* extensionIのプロパティ名を返却。
*
* @return extensionIのプロパティ名
*/
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

```

```

    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailValNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tRouteDetailVal}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tRouteDetailValNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tRouteDetailValNames() {
        super();
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
}

```

```

    * @return routeSearchIdのプロパティ名
    */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * commutationRouteのプロパティ名を返却.
     *
     * @return commutationRouteのプロパティ名
     */
    public static PropertyName<String> commutationRoute() {
        return new PropertyName<String>("commutationRoute");
    }

    /**
     * searchConditionのプロパティ名を返却.
     *
     * @return searchConditionのプロパティ名
     */
    public static PropertyName<String> searchCondition() {
        return new PropertyName<String>("searchCondition");
    }

    /**
     * routeInfoのプロパティ名を返却.
     *
     * @return routeInfoのプロパティ名
     */
    public static PropertyName<String> routeInfo() {
        return new PropertyName<String>("routeInfo");
    }

    /**
     * assessDateのプロパティ名を返却.
     *
     * @return assessDateのプロパティ名
     */
    public static PropertyName<Timestamp> assessDate() {
        return new PropertyName<Timestamp>("assessDate");
    }

    /**
     * tranceMeansFlagのプロパティ名を返却.
     *
     * @return tranceMeansFlagのプロパティ名
     */
    public static PropertyName<String> tranceMeansFlag() {
        return new PropertyName<String>("tranceMeansFlag");
    }

    /**
     * assessFlagのプロパティ名を返却.
     *
     * @return assessFlagのプロパティ名
     */
    public static PropertyName<String> assessFlag() {
        return new PropertyName<String>("assessFlag");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }
}

```



```

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */

```

```

    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30tRouteDetailValNames extends PropertyName<K30tRouteDetailVal> {

        /**
         * インスタンスを構築.
         */
        public _K30tRouteDetailValNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30tRouteDetailValNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30tRouteDetailValNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * routeSearchIdのプロパティ名を返却.
         *
         * @return routeSearchIdのプロパティ名
         */
        public PropertyName<String> routeSearchId() {
            return new PropertyName<String>((this, "routeSearchId"));
        }

        /**
         * commutationRouteのプロパティ名を返却.
         *
         * @return commutationRouteのプロパティ名
         */
        public PropertyName<String> commutationRoute() {
            return new PropertyName<String>((this, "commutationRoute"));
        }

        /**
         * searchConditionのプロパティ名を返却.
         *
         * @return searchConditionのプロパティ名
         */
        public PropertyName<String> searchCondition() {
            return new PropertyName<String>((this, "searchCondition"));
        }

        /**
         * routeInfoのプロパティ名を返却.
         *
         * @return routeInfoのプロパティ名
         */
        public PropertyName<String> routeInfo() {
            return new PropertyName<String>((this, "routeInfo"));
        }
    }

```

```

    return new PropertyName<String> (this, "extension");
}

/**
 * assessDateのプロパティ名を返却.
 *
 * @return assessDateのプロパティ名
 */
public PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>(this, "assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却.
 *
 * @return tranceMeansFlagのプロパティ名
 */
public PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>(this, "tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却.
 *
 * @return assessFlagのプロパティ名
 */
public PropertyName<String> assessFlag() {
    return new PropertyName<String>(this, "assessFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *

```

```

    * @return extensionGのプロパティ名
    */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailNames.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;
```

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tRouteDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tRouteDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tRouteDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClass のプロパティ名を返却.
     *
     * @return gadgetClass のプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstance のプロパティ名を返却.
     *
     * @return gadgetInstance のプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleId のプロパティ名を返却.
     *
     * @return tupleId のプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * localeId のプロパティ名を返却.
     *
     * @return localeId のプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonth のプロパティ名を返却.
     *
     * @return archiveMonth のプロパティ名
     */
    public static PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }

    /**
     * gadgetId のプロパティ名を返却.
     *
     * @return gadgetId のプロパティ名
     */

```

```

    * @return gadgetIdのプロパティ名
    */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却.
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却.
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * routeDateのプロパティ名を返却.
     *
     * @return routeDateのプロパティ名
     */
    public static PropertyName<Timestamp> routeDate() {
        return new PropertyName<Timestamp>("routeDate");
    }

    /**
     * startingSpotのプロパティ名を返却.
     *
     * @return startingSpotのプロパティ名
     */
    public static PropertyName<String> startingSpot() {
        return new PropertyName<String>("startingSpot");
    }

    /**
     * arrivalSpotのプロパティ名を返却.
     *
     * @return arrivalSpotのプロパティ名
     */
    public static PropertyName<String> arrivalSpot() {
        return new PropertyName<String>("arrivalSpot");
    }

    /**
     * supplementのプロパティ名を返却.
     *
     * @return supplementのプロパティ名
     */
    public static PropertyName<String> supplement() {
        return new PropertyName<String>("supplement");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }
}

```

```

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */

```

```

    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * roundTripFlagのプロパティ名を返却.
     *
     * @return roundTripFlagのプロパティ名
     */
    public static PropertyName<String> roundTripFlag() {
        return new PropertyName<String>("roundTripFlag");
    }

    /**
     * routeLinkFlagのプロパティ名を返却.
     *
     * @return routeLinkFlagのプロパティ名
     */
    public static PropertyName<String> routeLinkFlag() {
        return new PropertyName<String>("routeLinkFlag");
    }

    /**

```



```

    * routeSearchIdのプロパティ名を返却。
    *
    * @return routeSearchIdのプロパティ名
    */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * expTypeNameのプロパティ名を返却。
     *
     * @return expTypeNameのプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却。
     *
     * @return accountNameのプロパティ名
     */
    public static PropertyName<String> accountName() {
        return new PropertyName<String>("accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却。
     *
     * @return taxTypeNameのプロパティ名
     */
    public static PropertyName<String> taxTypeName() {
        return new PropertyName<String>("taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却。
     *
     * @return transCurrencyNameのプロパティ名
     */
    public static PropertyName<String> transCurrencyName() {
        return new PropertyName<String>("transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却。
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public static PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>("companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却。
     *
     * @return payTypeNameのプロパティ名
     */
    public static PropertyName<String> payTypeName() {
        return new PropertyName<String>("payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却。
     *
     * @return paymentFlagのプロパティ名
     */
    public static PropertyName<String> paymentFlag() {
        return new PropertyName<String>("paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却。
     *
     * @return payMethodNameのプロパティ名
     */
    public static PropertyName<String> payMethodName() {
        return new PropertyName<String>("payMethodName");
    }

    /**
     * departmentNameのプロパティ名を返却。
     *
     * @return departmentNameのプロパティ名
     */
    public static PropertyName<String> departmentName() {

```

```

    return new PropertyName<String>("departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public static PropertyName<String> projectName() {
    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.

```

```

*
* @return extensionIのプロパティ名
*/
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
* entryUserCdのプロパティ名を返却.
*
* @return entryUserCdのプロパティ名
*/
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
* renewCntのプロパティ名を返却.
*
* @return renewCntのプロパティ名
*/
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
* renewTsのプロパティ名を返却.
*
* @return renewTsのプロパティ名
*/
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
* renewUserCdのプロパティ名を返却.
*
* @return renewUserCdのプロパティ名
*/
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
* @author S2JDBC-Gen
*/
public static class _Ka30tRouteDetailNames extends PropertyName<Ka30tRouteDetail> {

    /**
    * インスタンスを構築.
    */
    public _Ka30tRouteDetailNames() {
    }

    /**
    * インスタンスを構築.
    *
    * @param name 名前
    */
    public _Ka30tRouteDetailNames(final String name) {
        super(name);
    }

    /**
    * インスタンスを構築.
    *
    * @param parent 親
    * @param name 名前

```

```

*/
public _Ka30tRouteDetailNames(final PropertyName<?> parent, final String name) {
    super(parent, name);
}

/**
 * systemMatterIdのプロパティ名を返却。
 *
 * @return systemMatterIdのプロパティ名
 */
public PropertyName<String> systemMatterId() {
    return new PropertyName<String>(this, "systemMatterId");
}

/**
 * userDataIdのプロパティ名を返却。
 *
 * @return userDataIdのプロパティ名
 */
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
 * gadgetClassのプロパティ名を返却。
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却。
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却。
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * localeIdのプロパティ名を返却。
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * archiveMonthのプロパティ名を返却。
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>(this, "archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却。
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却。
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

```

```

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>(this, "routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public PropertyName<String> startingSpot() {
    return new PropertyName<String>(this, "startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public PropertyName<String> arrivalSpot() {
    return new PropertyName<String>(this, "arrivalSpot");
}

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public PropertyName<String> supplement() {
    return new PropertyName<String>(this, "supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */

```

```

public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 *

```

```

    * payMethodCdのプロパティ名を返却.
    *
    * @return payMethodCdのプロパティ名
    */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>((this, "payMethodCd"));
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>((this, "payeeCd"));
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>((this, "payeeName"));
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public PropertyName<String> departmentSetCd() {
        return new PropertyName<String>((this, "departmentSetCd"));
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public PropertyName<String> departmentCd() {
        return new PropertyName<String>((this, "departmentCd"));
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>((this, "projectCd"));
    }

    /**
     * roundTripFlagのプロパティ名を返却.
     *
     * @return roundTripFlagのプロパティ名
     */
    public PropertyName<String> roundTripFlag() {
        return new PropertyName<String>((this, "roundTripFlag"));
    }

    /**
     * routeLinkFlagのプロパティ名を返却.
     *
     * @return routeLinkFlagのプロパティ名
     */
    public PropertyName<String> routeLinkFlag() {
        return new PropertyName<String>((this, "routeLinkFlag"));
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public PropertyName<String> routeSearchId() {
        return new PropertyName<String>((this, "routeSearchId"));
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public PropertyName<String> expTypeName() {
        return new PropertyName<String>((this, "expTypeName"));
    }

```

```

    return new PropertyName<String>(this, "expItyName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public PropertyName<String> accountName() {
    return new PropertyName<String>(this, "accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public PropertyName<String> taxTypeName() {
    return new PropertyName<String>(this, "taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public PropertyName<String> transCurrencyName() {
    return new PropertyName<String>(this, "transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>(this, "companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *

```



```

    * @return extensionAのプロパティ名
    */
    public PropertyName<String> extensionA() {
        return new PropertyName<String>((this, "extensionA"));
    }

    /**
     * extensionBのプロパティ名を返却。
     *
     * @return extensionBのプロパティ名
     */
    public PropertyName<String> extensionB() {
        return new PropertyName<String>((this, "extensionB"));
    }

    /**
     * extensionCのプロパティ名を返却。
     *
     * @return extensionCのプロパティ名
     */
    public PropertyName<String> extensionC() {
        return new PropertyName<String>((this, "extensionC"));
    }

    /**
     * extensionDのプロパティ名を返却。
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>((this, "extensionD"));
    }

    /**
     * extensionEのプロパティ名を返却。
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>((this, "extensionE"));
    }

    /**
     * extensionFのプロパティ名を返却。
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>((this, "extensionF"));
    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>((this, "extensionG"));
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>((this, "extensionH"));
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>((this, "extensionI"));
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>((this, "extensionJ"));
    }
}

```

```

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailValNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tRouteDetailVal} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tRouteDetailValNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tRouteDetailValNames() {
        super();
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
}

```

```

*/
public static PropertyName<String> localeId() {
    return new PropertyName<String>("localeId");
}

/**
 * archiveMonthのプロパティ名を返却。
 *
 * @return archiveMonthのプロパティ名
 */
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
 * commutationRouteのプロパティ名を返却。
 *
 * @return commutationRouteのプロパティ名
 */
public static PropertyName<String> commutationRoute() {
    return new PropertyName<String>("commutationRoute");
}

/**
 * searchConditionのプロパティ名を返却。
 *
 * @return searchConditionのプロパティ名
 */
public static PropertyName<String> searchCondition() {
    return new PropertyName<String>("searchCondition");
}

/**
 * routeInfoのプロパティ名を返却。
 *
 * @return routeInfoのプロパティ名
 */
public static PropertyName<String> routeInfo() {
    return new PropertyName<String>("routeInfo");
}

/**
 * assessDateのプロパティ名を返却。
 *
 * @return assessDateのプロパティ名
 */
public static PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>("assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却。
 *
 * @return tranceMeansFlagのプロパティ名
 */
public static PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>("tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却。
 *
 * @return assessFlagのプロパティ名
 */
public static PropertyName<String> assessFlag() {
    return new PropertyName<String>("assessFlag");
}

/**
 * extensionAのプロパティ名を返却。
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却。
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

```

```

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */

```

```

public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _Ka30tRouteDetailValNames extends PropertyName<Ka30tRouteDetailVal> {

    /**
     * インスタンスを構築.
     */
    public _Ka30tRouteDetailValNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _Ka30tRouteDetailValNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _Ka30tRouteDetailValNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonthのプロパティ名を返却.
     *
     * @return archiveMonthのプロパティ名
     */
    public PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }
}

```

```

}

/**
 * commutationRouteのプロパティ名を返却.
 *
 * @return commutationRouteのプロパティ名
 */
public PropertyName<String> commutationRoute() {
    return new PropertyName<String>(this, "commutationRoute");
}

/**
 * searchConditionのプロパティ名を返却.
 *
 * @return searchConditionのプロパティ名
 */
public PropertyName<String> searchCondition() {
    return new PropertyName<String>(this, "searchCondition");
}

/**
 * routeInfoのプロパティ名を返却.
 *
 * @return routeInfoのプロパティ名
 */
public PropertyName<String> routeInfo() {
    return new PropertyName<String>(this, "routeInfo");
}

/**
 * assessDateのプロパティ名を返却.
 *
 * @return assessDateのプロパティ名
 */
public PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>(this, "assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却.
 *
 * @return tranceMeansFlagのプロパティ名
 */
public PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>(this, "tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却.
 *
 * @return assessFlagのプロパティ名
 */
public PropertyName<String> assessFlag() {
    return new PropertyName<String>(this, "assessFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名

```

```

    /**
     * extensionDのプロパティ名を返却。
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却。
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却。
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

```

```

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetail;

/**
 * {@link K30tRouteDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tRouteDetailService extends GenerateService<K30tRouteDetail> {
    //
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetailVal;

/**
 * {@link K30tRouteDetailVal}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tRouteDetailValService extends GenerateService<K30tRouteDetailVal> {
    //
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetail;

/**
 * {@link Ka30tRouteDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tRouteDetailService extends GenerateService<Ka30tRouteDetail> {
    //
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailValService.java


```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetailVal;

/**
 * {@link Ka30tRouteDetailVal} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tRouteDetailValService extends GenerateService<Ka30tRouteDetailVal> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30tRouteDetail} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tRouteDetailCondition extends
    AbstractEntityCondition<K30tRouteDetailCondition> {

    /**
     * インスタンスを構築
     */
    public K30tRouteDetailCondition() {
    }

    /**
     * インスタンスを構築
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tRouteDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterId の条件. */
    public NotNulltableStringCondition<K30tRouteDetailCondition> systemMatterId =
        new NotNulltableStringCondition<K30tRouteDetailCondition>("systemMatterId", this);

    /** userDataId の条件. */
    public NotNulltableStringCondition<K30tRouteDetailCondition> userDataId =
        new NotNulltableStringCondition<K30tRouteDetailCondition>("userDataId", this);

    /** gadgetClass の条件. */
    public NotNulltableStringCondition<K30tRouteDetailCondition> gadgetClass =
        new NotNulltableStringCondition<K30tRouteDetailCondition>("gadgetClass", this);

    /** gadgetInstance の条件. */
    public NotNulltableCondition<K30tRouteDetailCondition, Long> gadgetInstance =
        new NotNulltableCondition<K30tRouteDetailCondition, Long>("gadgetInstance", this);

    /** tupleId の条件. */
    public NotNulltableCondition<K30tRouteDetailCondition, Long> tupleId =
        new NotNulltableCondition<K30tRouteDetailCondition, Long>("tupleId", this);

    /** gadgetId の条件. */
    public NotNulltableStringCondition<K30tRouteDetailCondition> gadgetId =
        new NotNulltableStringCondition<K30tRouteDetailCondition>("gadgetId", this);

    /** gadgetVariation の条件. */
    public NotNulltableStringCondition<K30tRouteDetailCondition> gadgetVariation =
        new NotNulltableStringCondition<K30tRouteDetailCondition>("gadgetVariation", this);
}

```

```

new NotNullableStringCondition<K30tRouteDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullableCondition<K30tRouteDetailCondition, Long> rowNumber =
    new NotNullableCondition<K30tRouteDetailCondition, Long>("rowNumber", this);

/** routeDateの条件. */
public NullableCondition<K30tRouteDetailCondition, Timestamp> routeDate =
    new NullableCondition<K30tRouteDetailCondition, Timestamp>("routeDate", this);

/** startingSpotの条件. */
public NullableStringCondition<K30tRouteDetailCondition> startingSpot =
    new NullableStringCondition<K30tRouteDetailCondition>("startingSpot", this);

/** arrivalSpotの条件. */
public NullableStringCondition<K30tRouteDetailCondition> arrivalSpot =
    new NullableStringCondition<K30tRouteDetailCondition>("arrivalSpot", this);

/** supplementの条件. */
public NullableStringCondition<K30tRouteDetailCondition> supplement =
    new NullableStringCondition<K30tRouteDetailCondition>("supplement", this);

/** expTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> expTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> accountCd =
    new NullableStringCondition<K30tRouteDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> taxTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tRouteDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("transAmount", this);

/** companyRateの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyRate", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tRouteDetailCondition>("companyCurrencyCd", this);

/** companyAmountの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payMethodCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payeeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payeeCd", this);

/** ...の条件. */

```

```

    /** payeeNameの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> payeeName =
        new NullableStringCondition<K30tRouteDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> departmentSetCd =
        new NullableStringCondition<K30tRouteDetailCondition>("departmentSetCd", this);

    /** departmentCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> departmentCd =
        new NullableStringCondition<K30tRouteDetailCondition>("departmentCd", this);

    /** projectCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> projectCd =
        new NullableStringCondition<K30tRouteDetailCondition>("projectCd", this);

    /** roundTripFlagの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> roundTripFlag =
        new NotNullableStringCondition<K30tRouteDetailCondition>("roundTripFlag", this);

    /** routeLinkFlagの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> routeLinkFlag =
        new NotNullableStringCondition<K30tRouteDetailCondition>("routeLinkFlag", this);

    /** routeSearchIdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> routeSearchId =
        new NullableStringCondition<K30tRouteDetailCondition>("routeSearchId", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionA =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionB =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionC =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionD =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionE =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionF =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionG =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionH =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionI =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionJ =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tRouteDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tRouteDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Long> renewCnt =
        new NotNullableCondition<K30tRouteDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<K30tRouteDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> renewUserCd =
        new NotNullableStringCondition<K30tRouteDetailCondition>("renewUserCd", this);

```

```
new NotNullableStringCondition<K30tRouteDetailValCondition> { renewUserCd, this },
}
```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValCondition.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30tRouteDetailVal} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tRouteDetailValCondition extends
    AbstractEntityCondition<K30tRouteDetailValCondition> {

    /**
     * インスタンスを構築.
     */
    public K30tRouteDetailValCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tRouteDetailValCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** routeSearchId の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> routeSearchId =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("routeSearchId", this);

    /** commutationRoute の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> commutationRoute =
        new NullableStringCondition<K30tRouteDetailValCondition>("commutationRoute", this);

    /** searchCondition の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> searchCondition =
        new NullableStringCondition<K30tRouteDetailValCondition>("searchCondition", this);

    /** routeInfo の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> routeInfo =
        new NullableStringCondition<K30tRouteDetailValCondition>("routeInfo", this);

    /** assessDate の条件. */
    public NullableCondition<K30tRouteDetailValCondition, Timestamp> assessDate =
        new NullableCondition<K30tRouteDetailValCondition, Timestamp>("assessDate", this);

    /** tranceMeansFlag の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> tranceMeansFlag =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("tranceMeansFlag", this);

    /** assessFlag の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> assessFlag =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("assessFlag", this);

    /** extensionA の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionA =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionA", this);

    /** extensionB の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionB =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionB", this);

    /** extensionC の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionC =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionC", this);

    /** extensionD の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionD =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionD", this);
}
```

```

/** extensionEの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionE =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionF =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionG =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionH =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionI =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionJ =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Timestamp> entryTs =
    new NotNullCondition<K30tRouteDetailValCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullStringCondition<K30tRouteDetailValCondition> entryUserCd =
    new NotNullStringCondition<K30tRouteDetailValCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Long> renewCnt =
    new NotNullCondition<K30tRouteDetailValCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Timestamp> renewTs =
    new NotNullCondition<K30tRouteDetailValCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30tRouteDetailValCondition> renewUserCd =
    new NotNullStringCondition<K30tRouteDetailValCondition>("renewUserCd", this);
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullCondition;
import org.seasar.extension.jdbc.where.condition.NotNullStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tRouteDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl"})
public final class Ka30tRouteDetailCondition extends
    AbstractEntityCondition<Ka30tRouteDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tRouteDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tRouteDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemEntityIdの条件. */

```

```

/ ** systemMatterIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> systemMatterId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("systemMatterId", this);

/ ** userDataIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> userDataId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("userDataId", this);

/ ** gadgetClassの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetClass =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetClass", this);

/ ** gadgetInstanceの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> gadgetInstance =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("gadgetInstance", this);

/ ** tupleIdの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> tupleId =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("tupleId", this);

/ ** localeIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> localeId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("localeId", this);

/ ** archiveMonthの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> archiveMonth =
    new NullableStringCondition<Ka30tRouteDetailCondition>("archiveMonth", this);

/ ** gadgetIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetId", this);

/ ** gadgetVariationの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetVariation =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetVariation", this);

/ ** rowNumberの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> rowNumber =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("rowNumber", this);

/ ** routeDateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Timestamp> routeDate =
    new NullableCondition<Ka30tRouteDetailCondition, Timestamp>("routeDate", this);

/ ** startingSpotの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> startingSpot =
    new NullableStringCondition<Ka30tRouteDetailCondition>("startingSpot", this);

/ ** arrivalSpotの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> arrivalSpot =
    new NullableStringCondition<Ka30tRouteDetailCondition>("arrivalSpot", this);

/ ** supplementの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> supplement =
    new NullableStringCondition<Ka30tRouteDetailCondition>("supplement", this);

/ ** expTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("expTypeCd", this);

/ ** accountCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> accountCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("accountCd", this);

/ ** taxTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("taxTypeCd", this);

/ ** taxRateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("taxRate", this);

/ ** quantityの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("quantity", this);

/ ** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("transCurrencyCd", this);

/ ** transAmountの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("transAmount", this);

/ ** companyRateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyRate", this);

```

```

new NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyRate", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("companyCurrencyCd", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payeeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> projectCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("projectCd", this);

/** roundTripFlagの条件. */
public NotNullableStringCondition<Ka30tRouteDetailCondition> roundTripFlag =
    new NotNullableStringCondition<Ka30tRouteDetailCondition>("roundTripFlag", this);

/** routeLinkFlagの条件. */
public NotNullableStringCondition<Ka30tRouteDetailCondition> routeLinkFlag =
    new NotNullableStringCondition<Ka30tRouteDetailCondition>("routeLinkFlag", this);

/** routeSearchIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> routeSearchId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("routeSearchId", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> accountName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payTypeName", this);

/** pavmentFlaaの条件. */

```



```

    public NullableStringCondition<Ka30tRouteDetailCondition> paymentFlag =
        new NullableStringCondition<Ka30tRouteDetailCondition>("paymentFlag", this);

    /** payMethodNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> payMethodName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("payMethodName", this);

    /** departmentNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> departmentName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("departmentName", this);

    /** projectNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> projectName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("projectName", this);

    /** extensionAの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionA =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionB =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionC =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionD =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionE =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionF =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionG =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionH =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionI =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionJ =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<Ka30tRouteDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<Ka30tRouteDetailCondition> entryUserCd =
        new NotNullableStringCondition<Ka30tRouteDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Long> renewCnt =
        new NotNullableCondition<Ka30tRouteDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<Ka30tRouteDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<Ka30tRouteDetailCondition> renewUserCd =
        new NotNullableStringCondition<Ka30tRouteDetailCondition>("renewUserCd", this);
}

```

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailValCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;

```



```

import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tRouteDetailVal}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tRouteDetailValCondition extends
    AbstractEntityCondition<Ka30tRouteDetailValCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tRouteDetailValCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tRouteDetailValCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** routeSearchIdの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> routeSearchId =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("routeSearchId", this);

    /** localeIdの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> localeId =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("localeId", this);

    /** archiveMonthの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> archiveMonth =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("archiveMonth", this);

    /** commutationRouteの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> commutationRoute =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("commutationRoute", this);

    /** searchConditionの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> searchCondition =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("searchCondition", this);

    /** routeInfoの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> routeInfo =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("routeInfo", this);

    /** assessDateの条件. */
    public NullableCondition<Ka30tRouteDetailValCondition, Timestamp> assessDate =
        new NullableCondition<Ka30tRouteDetailValCondition, Timestamp>("assessDate", this);

    /** tranceMeansFlagの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> tranceMeansFlag =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("tranceMeansFlag", this);

    /** assessFlagの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> assessFlag =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("assessFlag", this);

    /** extensionAの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionA =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionB =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionC =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionD =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionE =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionE", this);

```

```

/** extensionFの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionF =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionG =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionH =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionI =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionJ =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp> entryTs =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tRouteDetailValCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tRouteDetailValCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Long> renewCnt =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp> renewTs =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<Ka30tRouteDetailValCondition> renewUserCd =
    new NotNullableStringCondition<Ka30tRouteDetailValCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.route_date,
    t.starting_spot,
    t.arrival_spot,
    t.supplement,
    t.exp_type_cd,
    t.account_cd,
    t.tax_type_cd,
    t.tax_rate,
    t.quantity,
    t.trans_currency_cd,
    t.trans_amount,
    t.company_rate,
    t.company_currency_cd,
    t.company_amount,
    t.company_amount_exc_tax,
    t.company_amount_tax,
    t.company_amount_sum,
    t.pay_type_cd,
    t.pay_method_cd,
    t.payee_cd,
    t.payee_name,
    t.department_set_cd,
    t.department_cd,
    t.project_cd,
    t.round_trip_flag,
    t.route_link_flag,
    t.route_search_id,
    t.outside_id

```

```

t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
m7.exp_type_name,
m8.account_name,
m9.tax_type_name,
m4.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag
from
k30t_route_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
c1.search_criteria_date >= m2.start_date
and
c1.search_criteria_date < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
c1.search_criteria_date >= m3.start_date
and
c1.search_criteria_date < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id

```

```

/*localeId*/ja' = m4.locale_id
and
  c1.search_criteria_date >= m4.start_date
and
  c1.search_criteria_date < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeId*/ja' = m5.locale_id
and
  c1.search_criteria_date >= m5.start_date
and
  c1.search_criteria_date < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeId*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_exp_type m7
on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeId*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeId*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeId*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date
and
  c1.search_criteria_date < m9.end_date
and
  m9.delete_flag = '0'
left outer join
  k30t_route_detail_val val
on t.route_search_id = val.route_search_id
where
  t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*END*/

```

```

/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

- **Path**

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValService/selectView.sql

```

select
  t.route_search_id,
  t.commutation_route,
  t.search_condition,
  t.route_info,
  t.assess_date,
  t.trance_means_flag, assess_flag,
  t.extension_a,
  t.extension_b,
  t.extension_c,
  t.extension_d,
  t.extension_e,
  t.extension_f,
  t.extension_g,
  t.extension_h,
  t.extension_i,
  t.extension_j,
  t.entry_ts,
  t.entry_user_cd,
  t.renew_cnt,
  t.renew_ts,
  t.renew_user_cd
from
  k30t_route_detail d
  inner join
    k30t_route_detail_val t
  on
    d.route_search_id = t.route_search_id
where
  d.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  d.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  d.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
  d.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  d.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.route_search_id
/*END*/

```

- **Path**

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValService/selectView.sql

```

select
  t.route_search_id,
  t.locale_id,
  t.archive_month,
  t.commutation_route,
  t.search_condition,
  t.route_info,
  t.assess_date,
  t.trance_means_flag, assess_flag,
  t.extension_a,
  t.extension_b,
  t.extension_c,
  t.extension_d,
  t.extension_e,
  t.extension_f,
  t.extension_g,
  t.extension_h,
  t.extension_i,
  t.extension_j,
  t.entry_ts,
  t.entry_user_cd,
  t.renew_cnt,
  t.renew_ts,
  t.renew_user_cd
from
  ka30t_route_detail d
  inner join
    ka30t_route_detail_val t
  on
    d.route_search_id = t.route_search_id
  and
    d.locale_id = t.locale_id
where
  d.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
  and
    d.user_data_id = /*userDataId*/
  /*END*/
  /*IF orderBy != null*/
  order by /*$orderBy*/t.route_search_id
  /*END*/

```

コピー

- **Path**
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService/selectViewForCopy.sql

```

select
  t.system_matter_id,
  t.user_data_id,
  t.gadget_class,
  t.gadget_instance,
  t.tuple_id,
  t.gadget_id,
  t.gadget_variation,
  t.row_number,
  t.route_date,
  t.starting_spot,
  t.arrival_spot,
  t.supplement,
  t.exp_type_cd,
  t.account_cd,
  t.tax_type_cd,
  t.tax_rate,
  t.quantity,
  t.trans_currency_cd,
  t.trans_amount,
  t.company_rate,
  t.company_currency_cd,
  t.company_amount,
  t.company_amount_exc_tax,
  t.company_amount_tax,
  t.company_amount_sum,
  t.pay_type_cd,
  t.pay_method_cd,
  t.payee_cd,
  t.payee_name,
  t.department_set_cd,
  t.department_cd,
  t.project_cd,
  t.round_trip_flag,
  t.route_link_flag,
  t.route_search_id,
  t.extension_a,

```

```

t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
ma.exp_type_name,
case when t.exp_type_cd is not null and t.exp_type_cd <>'' then m8.account_name else mb.account_name end as account_name,
case when t.exp_type_cd is not null and t.exp_type_cd <>'' then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
m4.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag
from
k30t_route_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
/*searchCriteriaDate*/" >= m2.start_date
and
/*searchCriteriaDate*/" < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
/*searchCriteriaDate*/" >= m3.start_date
and
/*searchCriteriaDate*/" < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id
and

```

```

and
/*searchCriteriaDate*/" >= m4.start_date
and
/*searchCriteriaDate*/" < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeId*/ja' = m5.locale_id
and
/*searchCriteriaDate*/" >= m5.start_date
and
/*searchCriteriaDate*/" < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6
on
c1.search_criteria_company = m6.company_cd
and
t.company_currency_cd = m6.currency_cd
and
/*localeId*/ja' = m6.locale_id
and
m6.delete_flag = '0'
left outer join
k30m_account m8
on
c1.search_criteria_company = m8.company_cd
and
t.account_cd = m8.account_cd
and
/*localeId*/ja' = m8.locale_id
and
/*searchCriteriaDate*/" >= m8.start_date
and
/*searchCriteriaDate*/" < m8.end_date
and
m8.delete_flag = '0'
left outer join
k30m_tax_type m9
on
c1.search_criteria_company = m9.company_cd
and
t.tax_type_cd = m9.tax_type_cd
and
/*localeId*/ja' = m9.locale_id
and
/*searchCriteriaDate*/" >= m9.start_date
and
/*searchCriteriaDate*/" < m9.end_date
and
m9.delete_flag = '0'

left outer join
k30m_exp_type ma
on
c1.search_criteria_company = ma.company_cd
and
t.exp_type_cd = ma.exp_type_cd
and
/*localeId*/ja' = ma.locale_id
and
/*searchCriteriaDate*/" >= ma.start_date
and
/*searchCriteriaDate*/" < ma.end_date
and
ma.delete_flag = '0'
left outer join
k30m_account mb
on
ma.company_cd = mb.company_cd
and
ma.account_cd = mb.account_cd
and
/*localeId*/ja' = mb.locale_id
and
/*searchCriteriaDate*/" >= mb.start_date
and
/*searchCriteriaDate*/" < mb.end_date
and
m8.delete_flag = '0'
left outer join

```



```

left outer join
    k30m_tax_type mc
on
    ma.company_cd = mc.company_cd
and
    ma.tax_type_cd = mc.tax_type_cd
and
    /*localeId*/ja' = mc.locale_id
and
    /*searchCriteriaDate*/ >= mc.start_date
and
    /*searchCriteriaDate*/ < mc.end_date
and
    mc.delete_flag = '0'
left outer join
    k30t_route_detail_val val
on t.route_search_id = val.route_search_id
where
    t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
    t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
    t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
    t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
    t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
    c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
    t.system_matter_id as system_matter_id,
    t.user_data_id as user_data_id,
    t.gadget_class as gadget_class,
    t.gadget_instance as gadget_instance,
    t.tuple_id as tuple_id,
    t.gadget_id as gadget_id,
    t.gadget_variation as gadget_variation,
    i1.contents_id as contents_id,
    i1.contents_version_id as contents_version_id,
    null as journal_data_type_cd,
    t.row_number as row_number,
    t.exp_type_cd as exp_type_cd,
    t.account_cd as account_cd,
    t.tax_type_cd as tax_type_cd,
    mt.tax_rate as tax_rate,
    t.quantity as quantity,
    t.trans_currency_cd as trans_currency_cd,
    t.trans_amount as trans_amount,
    t.company_currency_cd as company_currency_cd,
    t.company_rate as company_rate,
    t.company_amount as company_amount,
    t.company_amount_exc_tax as company_amount_exc_tax,
    t.company_amount_tax as company_amount_tax,
    t.company_amount_sum as company_amount_sum,
    t.payee_cd as customer_cd,
    t.payee_name as customer_name,
    t.department_set_cd as department_set_cd,
    t.department_cd as department_cd,
    t.project_cd as project_cd,
    t.pay_method_cd as pay_method_cd,
    t.pay_type_cd as pay_type_cd,
    null as adv_matter_number,
    null as adv_matter_name,
    t.route_date as detail_date1,
    null as detail_date2,
    null as detail_summary1,
    null as detail_summary2

```

```

null as detail_summary2,
null as detail_summary3,
null as detail_summary4,
null as detail_summary5,
t.supplement as detail_note,
t.extension_a as detail_extension_a,
t.extension_b as detail_extension_b,
t.extension_c as detail_extension_c,
t.extension_d as detail_extension_d,
t.extension_e as detail_extension_e,
t.extension_f as detail_extension_f,
t.extension_g as detail_extension_g,
t.extension_h as detail_extension_h,
t.extension_i as detail_extension_i,
t.extension_j as detail_extension_j,
'0' as journal_status,
null as wish_date,
c1.search_criteria_company,
c1.search_criteria_date
from
k30t_route_detail t
inner join
k20c_matter_info i1
on
t.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join
k30m_tax_type mt
on
c1.search_criteria_company = mt.company_cd
and
t.tax_type_cd = mt.tax_type_cd
and
c1.search_criteria_date >= mt.start_date
and
c1.search_criteria_date < mt.end_date
and
i1.locale_id_apply = mt.locale_id
and
mt.delete_flag = '0'

where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.row_number
/*END*/

```

アーカイブ

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailService/selectViewForArchive.sql

```

select
t.system_matter_id,
t.user_data_id,
t.gadget_class,
t.gadget_instance,
t.tuple_id,
t.locale_id,
t.archive_month,

```

```

t.gadget_id,
t.gadget_variation,
t.row_number,
t.route_date,
t.starting_spot,
t.arrival_spot,
t.supplement,
t.exp_type_cd,
t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_rate,
t.company_currency_cd,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.round_trip_flag,
t.route_link_flag,
t.route_search_id,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
t.trans_currency_name,
t.department_name,
t.project_name,
t.pay_type_name,
t.pay_method_name,
t.company_currency_name,
t.exp_type_name,
t.account_name,
t.tax_type_name,
t.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag

from
ka30t_route_detail t
inner join
ka20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
and
t.locale_id = c1.locale_id
left outer join
ka30t_route_detail_val val
on
t.route_search_id = val.route_search_id
and
t.locale_id = val.locale_id
where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/

```

```

t.gadget_id = /gadgetid/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF localeId != null*/
and
t.locale_id = /*localeId*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.system_matter_id
/*END*/

```

JSP

PC

- **Path**

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/routeDetail/routeDetailV01.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wkf" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- routeDetail v01 --%>
<c:set var="gadgetClass">routeDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/routeDetailV01</c:set>
<c:set var="gadgetBlockId">routeDetailTable</c:set>
<c:set var="tmtableId">routeDetailTable</c:set>
<c:set var="tmtableControllerBlockId">tmtableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="applyAuthUserCode">${param.applyAuthUserCode}</c:set>
<c:set var="useRouteSearch">${k:isUseRouteSearch(param.searchCriteriaCompany)}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks[routeDetailTable]}" />

<!-- tmtable --%>
<c:set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</c:set>
<c:set var="surfaceRouteDate">${gadget.surfaces["routeDate"]}</c:set>
<c:set var="surfaceTravelRoute">${gadget.surfaces["travelRoute"]}</c:set>
<c:set var="surfaceStartingSpot">${gadget.surfaces["startingSpot"]}</c:set>
<c:set var="surfaceHyphen">${gadget.surfaces["hyphen"]}</c:set>
<c:set var="surfaceArrivalSpot">${gadget.surfaces["arrivalSpot"]}</c:set>
<c:set var="surfaceRoundTrip">${gadget.surfaces["roundTrip"]}</c:set>
<c:set var="surfaceSupplement">${gadget.surfaces["supplement"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceRouteSearchLink">${gadget.surfaces["routeSearchLink"]}</c:set>
<c:set var="surfaceRouteSearch">${gadget.surfaces["routeSearch"]}</c:set>
<c:set var="surfaceRouteSearchDetail">${gadget.surfaces["routeSearchDetail"]}</c:set>
<c:set var="surfaceRouteSearchRemove">${gadget.surfaces["routeSearchRemove"]}</c:set>
<c:set var="surfaceCompanyAmountSum">${gadget.surfaces["companyAmountSum"]}</c:set>

```

```

<%-- Surface(Property) --%>
<c:set var="surfaceRouteSearch"><%= BaseProp.S.EkispertWeb.ROUTE_SEARCH %></c:set>
<c:set var="surfaceShowDetail"><%= BaseProp.S.EkispertWeb.SHOW_DETAIL %></c:set>
<c:set var="surfaceReleaseLink"><%= BaseProp.S.EkispertWeb.RELEASE_LINK %></c:set>
<c:set var="surfaceAltFast"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_FAST) %></c:set>
<c:set var="surfaceAltEasy"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_EASE) %></c:set>
<c:set var="surfaceAltCheap"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP) %></c:set>
<c:set var="surfaceAltCheapTeiki1"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI1) %></c:set>
<c:set var="surfaceAltCheapTeiki3"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI3) %></c:set>
<c:set var="surfaceAltCheapTeiki6"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI6) %></c:set>
<c:set var="surfaceAltEco"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_ECO) %></c:set>
<c:set var="surfaceAltCommuter"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_TEIKI) %></c:set>
<c:set var="surfaceAltExpress"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_EXPRESS) %></c:set>
<c:set var="surfaceAltBus"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_BUS) %></c:set>
<c:set var="surfaceAltAir"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_AIR) %></c:set>
<c:set var="surfaceAltShip"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_SHIP) %></c:set>

<%-- Mode --%>
<c:set var="modeRouteDate">${detail.fields["routeDate"].inputType}</c:set>
<c:set var="modeStartingSpot">${detail.fields["startingSpot"].inputType}</c:set>
<c:set var="modeArrivalSpot">${detail.fields["arrivalSpot"].inputType}</c:set>
<c:set var="modeSupplement">${detail.fields["supplement"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modeRouteLinkFlag">${detail.fields["routeLinkFlag"].inputType}</c:set>
<c:set var="modeRoundTripFlag">${detail.fields["roundTripFlag"].inputType}</c:set>
<c:set var="modeCompanyAmountSum">${modeTransAmount == modeRoundTripFlag ? modeTransAmount : "0"}</c:set>

<%-- RequiredMark --%>
<c:set var="reqRouteDate" >${detail.fields["routeDate"].requiredFlag}</c:set>
<c:set var="reqStartingSpot" >${detail.fields["startingSpot"].requiredFlag}</c:set>
<c:set var="reqArrivalSpot" >${detail.fields["arrivalSpot"].requiredFlag}</c:set>
<c:set var="reqSupplement" >${detail.fields["supplement"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqRoundTripFlag" >${detail.fields["roundTripFlag"].requiredFlag}</c:set>
<c:set var="reqRouteLinkFlag" >${detail.fields["routeLinkFlag"].requiredFlag}</c:set>
<c:set var="reqCompanyAmountSum" >${detail.fields["companyAmountSum"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validRouteDate_def">{k_date:[""]}</c:set>
<c:set var="validRouteDate_apply">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>
<c:set var="validRouteDate_tempsave"></c:set>
<c:set var="validRouteDate_reapply">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>
<c:set var="validRouteDate_process">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>

<c:set var="validStartingSpot_def">{k_maxLength:["250"]}</c:set>
<c:set var="validStartingSpot_apply">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validStartingSpot_tempsave"></c:set>
<c:set var="validStartingSpot_reapply">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validStartingSpot_process">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>

<c:set var="validArrivalSpot_def">{k_maxLength:["250"]}</c:set>
<c:set var="validArrivalSpot_apply">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validArrivalSpot_tempsave"></c:set>
<c:set var="validArrivalSpot_reapply">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validArrivalSpot_process">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>

<c:set var="validSupplement_def">{k_maxLength:["1000"]}</c:set>
<c:set var="validSupplement_apply">${"true" == reqSupplement ? "k_required:[""] : ""}</c:set>
<c:set var="validSupplement_tempsave"></c:set>
<c:set var="validSupplement_reapply">${"true" == reqSupplement ? "k_required:[""] : ""}</c:set>
<c:set var="validSupplement_process">${"true" == reqSupplement ? "k_required:[""] : ""}</c:set>

<c:set var="validRoundTripFlag_def">{k_flag:[""]}</c:set>
<c:set var="validRoundTripFlag_apply">${"true" == reqRoundTripFlag ? "k_required:[""] : ""}</c:set>
<c:set var="validRoundTripFlag_tempsave"></c:set>
<c:set var="validRoundTripFlag_reapply">${"true" == reqRoundTripFlag ? "k_required:[""] : ""}</c:set>
<c:set var="validRoundTripFlag_process">${"true" == reqRoundTripFlag ? "k_required:[""] : ""}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:["payTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">${"true" == reqPayTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPavTvoveCd_reapply">${"true" == reqPavTvoveCd ? "k_required:[""] : ""}</c:set>

```

```

<c:set var="validPayTypeCd_process">{{"true" == reqPayTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validPayeeName_def">[k_maxLength:['250']]</c:set>
<c:set var="validPayeeName_apply">{{"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">{{"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayeeName_process">{{"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>

<c:set var="validDepartmentCd_def">[k_departmentExists:['searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate', 'false']]</c:set>
<c:set var="validDepartmentCd_apply">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>
<c:set var="validDepartmentCd_process">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>

<c:set var="validProjectCd_def">[k_projectExists:['projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']]</c:set>
<c:set var="validProjectCd_apply">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>
<c:set var="validProjectCd_process">[k_requiredAny:['departmentCd', 'projectCd']]</c:set>

<c:set var="validExpTypeCd_def">[k_expTypeExists:['expTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate', 'false']]</c:set>
<c:set var="validExpTypeCd_apply">{{"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">{{"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validExpTypeCd_process">{{"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validAccountCd_def">[k_accountExists:['accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']]</c:set>
<c:set var="validAccountCd_apply">{{"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">{{"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validAccountCd_process">{{"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validTaxType_def">[k_taxTypeExists:['taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']]</c:set>
<c:set var="validTaxType_apply">{{"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{{"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validTaxType_process">{{"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validCompanyAmountSum_def">[k_currencyAmnt:['', 'searchCriteriaCompany']]</c:set>
<c:set var="validCompanyAmountSum_apply">{{"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>
<c:set var="validCompanyAmountSum_tempsave"></c:set>
<c:set var="validCompanyAmountSum_reapply">{{"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>
<c:set var="validCompanyAmountSum_process">{{"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>

<k:gadgetContainer gadgetClass="{gadgetClass}"
    gadgetVariation="{gadgetVariation}"
    gadgetInstance="{gadgetInstance}"
    gadgetTitle="{surfaceChapterTitle}"
    gadgetId="{gadgetId}"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:ttable id="{ttableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed;" rowNumTarget="rowNumber">
<colgroup>
<col style="width:40px;overflow:hidden">
<col style="width:235px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
</colgroup>
<thead>
<tr>
<td rowspan="4" style="vertical-align:middle;">
<k:ttableOperationPanel
    position="header"
    allowAdd="{allowAddDetail}"
    allowDel="{allowDelDetail}"
/>
</td>
<th>
<k:surface
    name="routeDateSurface"
    surfaceKey="{surfaceRouteDate}"
    requiredMark="{reqRouteDate}"
/>
</th>
<th colspan="2">
<k:surface
    name="travelRouteSurface"
    surfaceKey="{surfaceTravelRoute}"
    requiredMark="{reqStartingSpot}"
/>
</th>
<th>
<k:surface
    name="supplementSurface"

```

```

        surfaceKey="{surfaceSupplement}"
        requiredMark="{reqSupplement}"
    />
</th>
</tr>
<tr>
<th colspan="2">
<k:surface
    name="expTypeSurface"
    surfaceKey="{surfaceExpType}"
    requiredMark="{reqExpTypeCd}"
    style="display:inline;"
/>
<k:surface
    name="accountSurface"
    surfaceKey="{surfaceAccount}"
    style="display:inline;"
/>
<k:surface
    name="taxTypeSurface"
    surfaceKey="{surfaceTaxType}"
    style="display:inline;"
/>
</th>
<th>
<k:surface
    name="departmentSurface"
    surfaceKey="{surfaceDepartment}"
    requiredMark="{reqDepartmentCd}"
/>
</th>
<th>
<k:surface
    name="projectSurface"
    surfaceKey="{surfaceProject}"
    requiredMark="{reqProjectCd}"
/>
</th>
</tr>
<tr>
<th>
<k:surface
    name="payTypeSurface"
    surfaceKey="{surfacePayType}"
    requiredMark="{reqPayTypeCd}"
/>
</th>
<th>
<k:surface
    name="payeeSurface"
    surfaceKey="{surfacePayee}"
    requiredMark="{reqPayeeName}"
/>
</th>
<th colspan="2">
<k:surface
    name="transAmountSurface"
    surfaceKey="{surfaceTransAmount}"
    requiredMark="{reqTransAmount}"
/>
</th>
</tr>
<tr>
<th colspan="3">
<k:surface
    name="routeSearchLinkSurface"
    surfaceKey="{surfaceRouteSearchLink}"
    requiredMark="{reqRouteSearchLink}"
/>
</th>
<th>
<k:surface
    name="companyAmountSurface"
    surfaceKey="{surfaceCompanyAmountSum}"
    requiredMark="{reqCompanyAmountSum}"
/>
</th>
</tr>
</thead>
<tbody>
<tr>
<td rowspan="4">
<k:tmtableOperationPanel
    position="detail"
    allowAdd="{allowAddDetail}"
    allowDel="{allowDelDetail}"

```

```

        name="rowNumber"
    />
</td>
<td>
    <k:date
        name="routeDate"
        mode="{modeRouteDate}"
        surfaceKey="{surfaceRouteDate}"
        validationDefault="{validRouteDate_def}"
        validationApply="{validRouteDate_apply}"
        validationTempsave="{validRouteDate_tempsave}"
        validationReapply="{validRouteDate_reapply}"
        validationProcess="{validRouteDate_process}"
    />
</td>
<td colspan="2">
    <k:text
        name="startingSpot"
        mode="{modeStartingSpot}"
        imeMode="on"
        surfaceKey="{surfaceStartingSpot}"
        style="width:165px;"
        maxlength="250"
        validationDefault="{validStartingSpot_def}"
        validationApply="{validStartingSpot_apply}"
        validationTempsave="{validStartingSpot_tempsave}"
        validationReapply="{validStartingSpot_reapply}"
        validationProcess="{validStartingSpot_process}"
    />
    <k:surface
        name="hyphenSurface"
        surfaceKey="{surfaceHyphen}"
        style="margin:0 2px;"
    />
    <k:text
        name="arrivalSpot"
        mode="{modeArrivalSpot}"
        imeMode="on"
        surfaceKey="{surfaceArrivalSpot}"
        style="width:165px;"
        maxlength="250"
        validationDefault="{validArrivalSpot_def}"
        validationApply="{validArrivalSpot_apply}"
        validationTempsave="{validArrivalSpot_tempsave}"
        validationReapply="{validArrivalSpot_reapply}"
        validationProcess="{validArrivalSpot_process}"
    />
</td>
<td>
    <k:text
        name="supplement"
        mode="{modeSupplement}"
        imeMode="on"
        surfaceKey="{surfaceSupplement}"
        style="width:169px;"
        maxlength="1000"
        validationDefault="{validSupplement_def}"
        validationApply="{validSupplement_apply}"
        validationTempsave="{validSupplement_tempsave}"
        validationReapply="{validSupplement_reapply}"
        validationProcess="{validSupplement_process}"
    />
</td>
</tr>
<tr>
    <td colspan="2">
        <k:hidden
            name="expTypeCd"
            mode="{modeExpTypeCd}"
        />
        <kmsw:expType
            name="expTypeName"
            mode="{modeExpTypeCd}"
            imeMode="on"
            surfaceKey="{surfaceExpType}"
            resultValue="expTypeName"
            resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
            primaryTarget="expTypeCd"
            searchCriteriaDate="searchCriteriaDate"
            searchCriteriaCompany="searchCriteriaCompany"
            searchCriteriaExpTypeCd="expTypeCd"
            style="width:150px;"
            validationDefault="{validExpTypeCd_def}"
            validationApply="{validExpTypeCd_apply}"
            validationTempsave="{validExpTypeCd_tempsave}"
            validationReapply="{validExpTypeCd_reapply}"

```



```

        validationProcess="${validExpTypeCd_process}"
    />
    <k:hidden
        name="accountCd"
        mode="${modeAccountCd}"
        surfaceKey="${surfaceAccount}"
        validationDefault="${validAccountCd_def}"
        validationApply="${validAccountCd_apply}"
        validationTempsave="${validAccountCd_tempsave}"
        validationReapply="${validAccountCd_reapply}"
        validationProcess="${validAccountCd_process}"
    />
    <k:text
        name="accountName"
        mode="1"
        class="kaiden-ui-mst-search-chain-item"
        style="width:115px;"
    />
    <k:hidden
        name="taxTypeCd"
        mode="${modeTaxTypeCd}"
        surfaceKey="${surfaceTaxType}"
        validationDefault="${validTaxType_def}"
        validationApply="${validTaxType_apply}"
        validationTempsave="${validTaxType_tempsave}"
        validationReapply="${validTaxType_reapply}"
        validationProcess="${validTaxType_process}"
    />
    <k:text
        name="taxTypeName"
        mode="1"
        class="kaiden-ui-mst-search-chain-item"
        style="width:115px;"
    />
</td>
<td>
    <k:hidden
        name="departmentSetCd"
        mode="${modeDepartmentCd}"
    />
    <k:hidden
        name="departmentCd"
        mode="${modeDepartmentCd}"
        surfaceKey="${surfaceDepartment}"
    />
    <kmsb:department
        name="departmentName"
        mode="${modeDepartmentCd}"
        imeMode="on"
        surfaceKey="${surfaceDepartment}"
        resultValue="departmentName"
        resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd'}"
        primaryTarget="departmentCd,departmentSetCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaDepartmentCd="departmentCd"
        searchCriteriaDepartmentSetCd="departmentSetCd"
        style="width:150px;"
        validationDefault="${validDepartmentCd_def}"
        validationApply="${validDepartmentCd_apply}"
        validationTempsave="${validDepartmentCd_tempsave}"
        validationReapply="${validDepartmentCd_reapply}"
        validationProcess="${validDepartmentCd_process}"
    />
</td>
<td>
    <k:hidden
        name="projectCd"
        mode="${modeProjectCd}"
        surfaceKey="${surfaceProject}"
    />
    <kmsw:project
        name="projectName"
        mode="${modeProjectCd}"
        imeMode="on"
        surfaceKey="${surfaceProject}"
        resultValue="projectName"
        resultChain="{projectCd:'projectCd'}"
        primaryTarget="projectCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaProjectCd="projectCd"
        style="width:150px;"
        validationDefault="${validProjectCd_def}"
        validationApply="${validProjectCd_apply}"
        validationTempsave="${validProjectCd_tempsave}"

```

```

        validationReapply="${validProjectCd_reapply}"
        validationProcess="${validProjectCd_process}"
    />
</td>
</tr>
<tr>
<td>
<klst_wkf:listDataPayType
    blank="false"
    name="payType"
    companyCd="${param.searchCriteriaCompany}"
    searchCriteriaDate="${param.searchCriteriaDate}"
/>
<k:select
    name="payTypeCd"
    mode="${modePayTypeCd}"
    surfaceKey="${surfacePayType}"
    viewModeName="payTypeName"
    dataName="payType"
    selectChain='{paymentFlag:paymentFlag}'
    style="width:230px;"
    validationDefault="${validPayTypeCd_def}"
    validationApply="${validPayTypeCd_apply}"
    validationTempsave="${validPayTypeCd_tempsave}"
    validationReapply="${validPayTypeCd_reapply}"
    validationProcess="${validPayTypeCd_process}"
/>
<k:hidden
    name="paymentFlag"
    mode="${modePayTypeCd}"
/>
</td>
<td>
<k:text
    name="payeeName"
    mode="${modePayeeName}"
    imeMode="on"
    surfaceKey="${surfacePayee}"
    style="width:160px;"
    maxlength="250"
    validationDefault="${validPayeeName_def}"
    validationApply="${validPayeeName_apply}"
    validationTempsave="${validPayeeName_tempsave}"
    validationReapply="${validPayeeName_reapply}"
    validationProcess="${validPayeeName_process}"
/>
</td>
<td colspan="2">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericAmountInputParts.jsp">
<jsp:param name="gadgetId" value="${gadgetId}" />
<jsp:param name="tmtableId" value="${tmtableId}" />
<jsp:param name="searchCriteriaCompany" value="searchCriteriaCompany" />
<!-- 金額 --%>
<jsp:param name="modeTransAmnt" value="${modeTransAmount}" />
<jsp:param name="surfaceTransAmnt" value="${surfaceTransAmount}" />
<jsp:param name="requiredTransAmntApply" value="${reqTransAmount}" />
<jsp:param name="requiredTransAmntReapply" value="${reqTransAmount}" />
<jsp:param name="requiredTransAmntProcess" value="${reqTransAmount}" />
</jsp:include>
</td>
</tr>
<tr>
<td colspan="3">
<a id="routeDetail" style="display:none">
<span class="kaiden-ui-icon-train24" style="vertical-align:middle;margin-right:0.5em"></span>
<k:surface
    name="showDetailSurface"
    surfaceKey="${surfaceShowDetail}"
    style="vertical-align:middle;"
/>
</a>
<c:if test="${allowAddDetail && 'true' == useRouteSearch}">
<a id="routeSearch">
<span class="kaiden-ui-icon-search16" style="vertical-align:middle;margin-right:0.5em"></span>
<k:surface
    name="routeSearchSurface"
    surfaceKey="${surfaceRouteSearch}"
    style="vertical-align:middle;"
/>
</a>
</c:if>
<table style="table-layout:fixed;display:inline-block;vertical-align:middle;margin-left:1em;margin-right:1em;height:24px;" class="noborder" id="flagIconTbl">
<colgroup>
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">

```

```

<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
</colgroup>
<tbody>
<tr>
<td>
<span class="kaiden-ui-icon-fast24" id="iconFast" title="{surfaceAltFast}" style="display:none" ></span>
<span class="kaiden-ui-icon-fast-off24" id="iconFastOff" title="{surfaceAltFast}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-cheap24" id="iconCheap" title="{surfaceAltCheap}" style="display:none"></span>
<span class="kaiden-ui-icon-cheap-off24" id="iconCheapOff" title="{surfaceAltCheap}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-easy24" id="iconEasy" title="{surfaceAltEasy}" style="display:none"></span>
<span class="kaiden-ui-icon-easy-off24" id="iconEasyOff" title="{surfaceAltEasy}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-eco24" id="iconEco" title="{surfaceAltEco}" style="display:none"></span>
<span class="kaiden-ui-icon-eco-off24" id="iconEcoOff" title="{surfaceAltEco}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-commuter24" id="iconCommuter" title="{surfaceAltCommuter}" style="display:none"></span>
<span class="kaiden-ui-icon-commuter-off24" id="iconCommuterOff" title="{surfaceAltCommuter}" style="display:none"></span>
</td>
<td></td>
<td>
<span class="kaiden-ui-icon-express24" id="iconExpress" title="{surfaceAltExpress}" style="display:none"></span>
<span class="kaiden-ui-icon-express-off24" id="iconExpressOff" title="{surfaceAltExpress}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-bus24" id="iconBus" title="{surfaceAltBus}" style="display:none">
</span><span class="kaiden-ui-icon-bus-off24" id="iconBusOff" title="{surfaceAltBus}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-plane24" id="iconPlane" title="{surfaceAltAir}" style="display:none"></span>
<span class="kaiden-ui-icon-plane-off24" id="iconPlaneOff" title="{surfaceAltAir}" style="display:none"></span>
</td>
<td>
<span class="kaiden-ui-icon-ship24" id="iconShip" title="{surfaceAltShip}" style="display:none"></span>
<span class="kaiden-ui-icon-ship-off24" id="iconShipOff" title="{surfaceAltShip}" style="display:none"></span>
</td>
</tr>
</tbody>
</table>
<c:if test="{allowAddDetail}">
<a id="releaseLink" style="display:none">
<span class="kaiden-ui-icon-x16" style="vertical-align:middle;margin-right:0.5em"></span>
<k:surface
name="releaseLinkSurface"
surfaceKey="{surfaceReleaseLink}"
style="vertical-align:middle;"
/>
</a>
</c:if>
<div id="routeDetailArea" class="kaiden-ui-routedetail"></div>
<k:hidden
name="routeLinkFlag"
mode="0"
value="0"
/>
<k:hidden
name="commutationRoute"
mode="0"
/>
<k:hidden
name="searchCondition"
mode="0"
/>
<k:hidden
name="routeInfo"
mode="0"
/>
<k:hidden
name="assessDate"
mode="0"
/>
<k:hidden
name="tranceMeansFlag"
mode="0"
/>

```

```

        <k:hidden
            name="assessFlag"
            mode="0"
        />
    </td>
    <td style="text-align:right;">
        <k:checkbox
            name="roundTripFlag"
            mode="{modeRoundTripFlag}"
            label="{surfaceRoundTrip}"
            surfaceKey="{surfaceRoundTrip}"
            validationDefault="{validRoundTripFlag_def}"
            validationApply="{validRoundTripFlag_apply}"
            validationTempsave="{validRoundTripFlag_tempsave}"
            validationReapply="{validRoundTripFlag_reapply}"
            validationProcess="{validRoundTripFlag_process}"
        />
        <k:numericText
            name="companyAmountSum"
            mode="{modeCompanyAmountSum}"
            formatted="view"
            surfaceKey="{surfaceCompanyAmountSum}"
            style="width:120px"
            tabindex="-1"
            readonly="readonly"
            validationDefault="{validCompanyAmountSum_def}"
            validationApply="{validCompanyAmountSum_apply}"
            validationTempsave="{validCompanyAmountSum_tempsave}"
            validationReapply="{validCompanyAmountSum_reapply}"
            validationProcess="{validCompanyAmountSum_process}"
        />
    </td>
</tr>
</tbody>
</k:tmttable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){

    var eventMan = KAIDEN.getGadget("{gadgetId}").eventManager
    , blockAccessor = KAIDEN.getGadget("{gadgetId}").blockAccessors["{gadgetBlockHeaderId}"]
    , tmt = KAIDEN.getGadget("{gadgetId}").tmtables["{tmttableId}"]
    , tma = tmt.tmttableAccessor;

    /*****
    * 内部イベント
    *****/

    //明細行:行挿入(データ設定前)
    tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

        //デフォルト部門を要求
        var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "{gadgetId}");
        defaultDept = defaultDept[0].result; //先頭行のみ利用
        tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
        tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
        tma("departmentName", tupleId).setValue(defaultDept.departmentName);

        //デフォルトプロジェクトを要求
        var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "{gadgetId}");
        defaultPrj = defaultPrj[0].result; //先頭行のみ利用
        tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
        tma("projectName", tupleId).setValue(defaultPrj.projectName);

    });

    //明細行:行挿入(データ設定後)
    tmt.addEventPostNewRow(function(tupleId, tma, data) {
        //マスタ検索部品の整合性確認。
        KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
        KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
        KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

        //初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
        if (tma("payTypeCd", tupleId).elem(true).length > 0) {
            KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
        }

        //事前申請書取込では、会社通貨額が設定されているため、取引通貨額に会社通貨額を設定する
        //※外貨版から標準版にコピーする際、会社通貨額(レート換算後)の金額を設定する
        var compAmnt = tma("companyAmount", tupleId).getValue();
        if (compAmnt) {
            tma("transAmount", tupleId).setValue(compAmnt);
            tma("companyAmount", tupleId).setValue("");
        }
    });

```

```

Kaiden.getGadget("${gadgetId}").variable.adjustingRouteSearchResult(tupleId);

});

//金額*係数(往復フラグがONの場合は"2", OFFの場合は"1"を利用)
tmt.variable.multiplyRoundTrip = function(target) {
  var transAmount = Kaiden.gadgetMan.getNeighboringAccessor(target, "transAmount").getValue();
  var roundTripFlag = Kaiden.gadgetMan.getNeighboringAccessor(target, "roundTripFlag").getValue();

  if($.isNumeric(transAmount)) {
    //往復フラグがONの場合
    if (roundTripFlag == "1") {
      var total = new BigDecimal(transAmount).multiply(new BigDecimal("2"));
      Kaiden.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue([total.toString(), Kaiden.Util.addFigure(total.toString())]);
    } else {
      Kaiden.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue([transAmount, Kaiden.Util.addFigure(transAmount)]);
    }
  } else {
    Kaiden.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue("");
  }
};

//往復トグルボタン押下時後処理
tma("roundTripFlag").elem(true).live("click", function(e) {
  tmt.variable.multiplyRoundTrip(this);
});
//取引通貨額入力後処理(readonlyだがtrigger時にも利用できるよう付与)
tma("transAmount").elem(true).live("change", function(e) {
  tmt.variable.multiplyRoundTrip(this);
});

/*****
* 受信イベント (push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
  //明細行のクリア
  tmt.deleteAll();
  return Kaiden.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
  return Kaiden.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
  return Kaiden.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//reloadAllCurrencyRate レート再取得 イベントリスナの登録
eventMan.registPushListener("reloadAllCurrencyRate", function(**Object*/arg) {
  Kaiden.getGadget("${gadgetId}").variable.reloadRateAll();
});

/*****
* 受信イベント (pull)
*****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
  return tmt.getColumnValues(["routeDate", "routeDate"]);
});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues("companyAmountSum");
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
  return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["companyAmountSum", "paymentFlag"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value[0]) && value[1] == "1") {
      result = result.add(new BigDecimal(value[0]));
    }
  });
}

```

```

        result = result.add(new BigDecimal(value[0]));
    }
    });
    return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues(["companyAmountSum", "paymentFlag"]);
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value[0]) && value[1] != "1") {
            result = result.add(new BigDecimal(value[0]));
        }
    });
    return result.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
    return tmt.getColumnValues("companyAmountSum");
});

//claimExpenccCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenccCount", function(**Object*/arg) {
    return tmt.getCount();
});

//経路検索結果返却イベントリスナの登録
eventMan.registPushListener("setRouteSearchResult", function(**Object*/routeResult) {

    var tupleId = KAIDEN.getGadget("${gadgetId}").variable.routeSearchTargetTupleId;

    //連携内容を画面に設定
    var routeDate = routeResult.param.date;
    if (routeDate) {
        routeDate = routeDate.substring(0, 4) + "/" + routeDate.substring(4, 6) + "/" + routeDate.substring(6, 8)
    }
    tma("routeDate", tupleId).setValue(routeDate); //日付
    tma("startingSpot", tupleId).setValue(routeResult.param.from); //出発地
    tma("arrivalSpot", tupleId).setValue(routeResult.param.to); //到着地
    tma("transAmount", tupleId).setValue(routeResult.param.amount); //片道金額

    //利用交通機関フラグ
    var useTrain = routeResult.param.train //電車利用
    , useExpress = "1" == routeResult.param.train && "1" == routeResult.param.charge ? "1" : "0" //特急利用
    , useBus = routeResult.param.bus //バス利用
    , usePlane = routeResult.param.airplane //飛行機利用
    , useShip = routeResult.param.ship //船舶利用
    , useWalk = routeResult.param.walk; //徒歩利用
    tma("tranceMeansFlag", tupleId).setValue(useTrain + useExpress + useBus + usePlane + useShip + useWalk + "0000"); //利用交通機関フラグ

    //評価フラグ
    var fastFlag = routeResult.param.fastFlag //早
    , cheapFlag = routeResult.param.cheapFlag //安
    , easeFlag = routeResult.param.easeFlag //楽
    , ecoFlag = routeResult.param.ecoFlag //eco
    , teikiFlag = routeResult.param.teikiFlag; //定期
    tma("assessFlag", tupleId).setValue(fastFlag + cheapFlag + easeFlag + ecoFlag + teikiFlag + "00000"); //評価フラグ

    //経路連携済フラグ
    tma("routeLinkFlag", tupleId).setValue("1");

    //他駅すばあと情報
    tma("commutationRoute", tupleId).setValue(""); //定期区間
    tma("searchCondition", tupleId).setValue(routeResult.condition); //検索条件
    tma("routeInfo", tupleId).setValue(routeResult.route); //経路情報

    //金額の計算
    tma("transAmount", tupleId).elem().trigger("change");

    //表示状態の調整
    KAIDEN.getGadget("${gadgetId}").variable.adjustingRouteSearchResult(tupleId);
});

//経路検索結果に基づく表示調整
KAIDEN.getGadget("${gadgetId}").variable.adjustingRouteSearchResult = function(**String*/tupleId) {
    if ("1" == tma("routeLinkFlag", tupleId).getValue()) {
        //経路検索連携済
        tma("#routeSearch", tupleId).elem(true).hide();
        tma("#routeDetail", tupleId).elem(true).show();
        tma("#releaseLink", tupleId).elem(true).show();
        // tma("routeDate", tupleId).readonly(); //日付
        tma("startingSpot", tupleId).readonly(); //出発地
        tma("arrivalSpot", tupleId).readonly(); //到着地
        tma("transAmount", tupleId).readonly(); //金額
    }
}

```

```

//アイコンの調整 (利用経路)
var tranceMeansFlag = tma("tranceMeansFlag", tupleId).getValue()
, train = tranceMeansFlag[0] //電車
, express = tranceMeansFlag[1] //特急
, bus = tranceMeansFlag[2] //バス
, plain = tranceMeansFlag[3] //飛行機
, ship = tranceMeansFlag[4] //船舶
, walk = tranceMeansFlag[5]; //徒歩

// 特急 (電車) 利用
if ("1" == express) {
  tma("#iconExpress", tupleId).elem().show();
  tma("#iconExpressOff", tupleId).elem().hide();
} else {
  tma("#iconExpress", tupleId).elem().hide();
  tma("#iconExpressOff", tupleId).elem().show();
}

// バス利用
if ("1" == bus) {
  tma("#iconBus", tupleId).elem().show();
  tma("#iconBusOff", tupleId).elem().hide();
} else {
  tma("#iconBus", tupleId).elem().hide();
  tma("#iconBusOff", tupleId).elem().show();
}

// 飛行機利用
if ("1" == plain) {
  tma("#iconPlane", tupleId).elem().show();
  tma("#iconPlaneOff", tupleId).elem().hide();
} else {
  tma("#iconPlane", tupleId).elem().hide();
  tma("#iconPlaneOff", tupleId).elem().show();
}

// 船舶
if ("1" == ship) {
  tma("#iconShip", tupleId).elem().show();
  tma("#iconShipOff", tupleId).elem().hide();
} else {
  tma("#iconShip", tupleId).elem().hide();
  tma("#iconShipOff", tupleId).elem().show();
}

//アイコンの調整 (評価フラグ)
var assessFlag = tma("assessFlag", tupleId).getValue();
var fastFlag = assessFlag[0] //早
, cheapFlag = assessFlag[1] //安
, easeFlag = assessFlag[2] //楽
, ecoFlag = assessFlag[3] //eco
, teikiFlag = assessFlag[4]; //定期

//早
if ("1" == fastFlag) {
  tma("#iconFast", tupleId).elem().show();
  tma("#iconFastOff", tupleId).elem().hide();
} else {
  tma("#iconFast", tupleId).elem().hide();
  tma("#iconFastOff", tupleId).elem().show();
}

//安
if ("1" == cheapFlag) {
  tma("#iconCheap", tupleId).elem().show();
  tma("#iconCheapOff", tupleId).elem().hide();
} else {
  tma("#iconCheap", tupleId).elem().hide();
  tma("#iconCheapOff", tupleId).elem().show();
}
}

//ソート順によりアイコンのtitle属性を切り替える

var searchCondition = tma("searchCondition", tupleId).getValue();
if (searchCondition) {
  switch(KAIDEN.Util.parseJSON(searchCondition).sort) {
    case "teiki1":
      tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki1}");
      tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki1}");
      break;
    case "teiki3":
      tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki3}");
      tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki3}");
      break;
    case "teiki6":

```

```

        tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki6}");
        tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki6}");
        break;
    default :
        tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheap}");
        tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheap}");
        break;
    }
}
//案
if ("1" == easeFlag) {
    tma("#iconEasy", tupleId).elem().show();
    tma("#iconEasyOff", tupleId).elem().hide();
} else {
    tma("#iconEasy", tupleId).elem().hide();
    tma("#iconEasyOff", tupleId).elem().show();
}
//eco
if ("1" == ecoFlag) {
    tma("#iconEco", tupleId).elem().show();
    tma("#iconEcoOff", tupleId).elem().hide();
} else {
    tma("#iconEco", tupleId).elem().hide();
    tma("#iconEcoOff", tupleId).elem().show();
}

//eco
if ("1" == teikiFlag) {
    tma("#iconCommuter", tupleId).elem().show();
    tma("#iconCommuterOff", tupleId).elem().hide();
} else {
    tma("#iconCommuter", tupleId).elem().hide();
    tma("#iconCommuterOff", tupleId).elem().show();
}

//アイコンエリアを表示
tma("#flagIconTbl", tupleId).elem().show();
} else {
    //経路検索未連携

    tma("#routeSearch", tupleId).elem(true).show();
    tma("#routeDetail", tupleId).elem(true).hide();
    tma("#releaseLink", tupleId).elem(true).hide();

    tma("routeDate", tupleId).removeReadonly(); //日付
    tma("startingSpot", tupleId).removeReadonly(); //出発地
    tma("arrivalSpot", tupleId).removeReadonly(); //到着地
    tma("transAmount", tupleId).removeReadonly(); // 定期代 (1ヶ月)
    //アイコンエリアを非表示
    tma("#flagIconTbl", tupleId).elem(true).hide();

}
};

//経路検索画面を表示
tma("#routeSearch").elem(true).live("click", function(e) {
    var param = {};
    param.companyCd = "${searchCriteriaCompany}";
    param.userCd = "${applyAuthUserCode}";
    param.mode = "1"; //経路検索
    param.date = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "routeDate").getValue();
    param.from = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "startingSpot").getValue();
    param.to = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "arrivalSpot").getValue();
    KAIDEN.gadgetMan.execPush("openRouteSearch", param, "${gadgetId}");

    KAIDEN.getGadget("${gadgetId").variable.routeSearchTargetTupleId = KAIDEN.gadgetMan.whereMe(e.target).tmtTupleId;
});

//詳細画面を表示
tma("#routeDetail").elem(true).live("click", function(e) {

    var routeDetailArea = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "#routeDetailArea").elem();
    routeDetailArea.toggle();

    if (routeDetailArea.attr("data-kaiden-routedetail-loaded")) {
        return;
    }

    var routeDataJson = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "routeInfo").getValue();
    if (routeDataJson) {
        var routeData = KAIDEN.Util.parseJSON(routeDataJson);
        var requestJson = KAIDEN.Util.stringifyJSON(routeData.resultSet.result[0]);

        var param = {};
        param.requestParam = {};

```



```

param.requestParam.result = requestJson;
param.requestParam.mode = "1";
param.companyCd = "${searchCriteriaCompany}";
param.userCd = "${applyAuthUserCode}";
param.callBack = function("**String"/html) {
    KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "#routeDetailArea").elem().attr("data-kaiden-routedetail-loaded", "true").imuiIndicator("destroy").html(html);
};

routeDetailArea.imuiIndicator();
KAIDEN.gadgetMan.execPush("showRouteDetailHtml", param, "${gadgetId}");
routeDetailArea.show();
}
});

//駅すばあと連携解除
tma("#releaseLink").elem(true).live("click", function(e) {

var tupleId = KAIDEN.gadgetMan.whereMe(e.target).tmtTupleId;

tma("routeLinkFlag", tupleId).setValue("0");
tma("commutationRoute", tupleId).setValue(""); //定期区間
tma("searchCondition", tupleId).setValue(""); //検索条件
tma("routeInfo", tupleId).setValue(""); //経路情報
tma("assessDate", tupleId).setValue(""); //評価判定基準日
tma("#routeDetailArea").elem().removeAttr("data-kaiden-routedetail-loaded").hide().children().remove();

KAIDEN.getGadget("${gadgetId}").variable.adjustingRouteSearchResult(tupleId);
});
});
</script>

```

SP

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/routeDetail/routeDetailV01_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
<c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- routeDetail v01 --%>
<c:set var="gadgetClass">routeDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">routeDetailV01Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">routeDetailTable</c:set>

<c:set var="gadgetResponse" value="${kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>
<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRouteDate">${gadget.surfaces["routeDate"]}</c:set>
<c:set var="surfaceTravelRoute">${gadget.surfaces["travelRoute"]}</c:set>
<c:set var="surfaceHyphen">${gadget.surfaces["hyphen"]}</c:set>
<c:set var="surfaceSupplement">${gadget.surfaces["supplement"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceRouteSearchLink">${gadget.surfaces["routeSearchLink"]}</c:set>
<c:set var="surfaceCompanyAmountSum">${gadget.surfaces["companyAmountSum"]}</c:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconop

```

```

<h3>
<span>${surfaceChapterTitle}</span>
<span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="${fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>
<ul data-role="listview" data-divider-theme="b">
<c:forEach items="${gadgetResponse.data[tmtableId]}" var="detailData">
<li>
<span class="ui-li-heading"><c:out value="${detailData.routeDate}"></c:out></span>
<span class="ui-li-aside"><c:out value="${detailData.companyAmountSum[1]}"></c:out></span>
<div class="indent">
<p class="kaiden-ui-overview-contents">
<c:out value="${detailData.startingSpot}"></c:out>
<c:out value="${surfaceHyphen}"></c:out>
<c:out value="${detailData.arrivalSpot}"></c:out>
</p>
<c:if test="${'1' == detailData.routeLinkFlag}">
<p class="kaiden-ui-overview-contents">
<c:if test="${'1' == fn:substring(detailData.assessFlag, 0, 1)}">
<span class="kaiden-ui-icon-fast24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 0, 1)}">
<span class="kaiden-ui-icon-fast-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 2, 3)}">
<span class="kaiden-ui-icon-cheap24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 2, 3)}">
<span class="kaiden-ui-icon-cheap-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 1, 2)}">
<span class="kaiden-ui-icon-easy24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 1, 2)}">
<span class="kaiden-ui-icon-easy-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 3, 4)}">
<span class="kaiden-ui-icon-eco24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 3, 4)}">
<span class="kaiden-ui-icon-eco-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 4, 5)}">
<span class="kaiden-ui-icon-commuter24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 4, 5)}">
<span class="kaiden-ui-icon-commuter-off24"></span>
</c:if>
<span style="display:inline-block;width:24px"></span>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 0, 1)}">
<span class="kaiden-ui-icon-express24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 0, 1)}">
<span class="kaiden-ui-icon-express-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 1, 2)}">
<span class="kaiden-ui-icon-bus24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 1, 2)}">
<span class="kaiden-ui-icon-bus-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 2, 3)}">
<span class="kaiden-ui-icon-plane24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 2, 3)}">
<span class="kaiden-ui-icon-plane-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 3, 4)}">
<span class="kaiden-ui-icon-ship24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 3, 4)}">
<span class="kaiden-ui-icon-ship-off24"></span>
</c:if>
</p>
</c:if>
<p class="kaiden-ui-overview-contents-sub"><c:out value="${detailData.payTypeName}"></c:out></p>
</div>
</li>
<li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="${surfaceBack}"></c:out></a></li>
<li>
<p class="kaiden-ui-detail-item-title"><c:out value="${surfaceRouteDate}"></c:out></p>
<div class="indent">
<label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.routeDate}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="${surfaceTravelRoute}"></c:out></p>
<div class="indent">
<label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.startingSpot}"></c:out><c:out value="${surfaceHyphen}"></c:out><c:out value="${detailData.arrivalSpot}"></c:out></div>

```

```

</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceSupplement}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.supplement}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceExpType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.expTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceAccount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.accountName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTaxType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.taxTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceDepartment}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.departmentName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.projectName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.payTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayee}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="\${detailData.payeeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTransAmount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.transAmount[1]}"></c:out></strong></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceCompanyAmountSum}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.companyAmountSum[1]}"></c:out></strong></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>

```

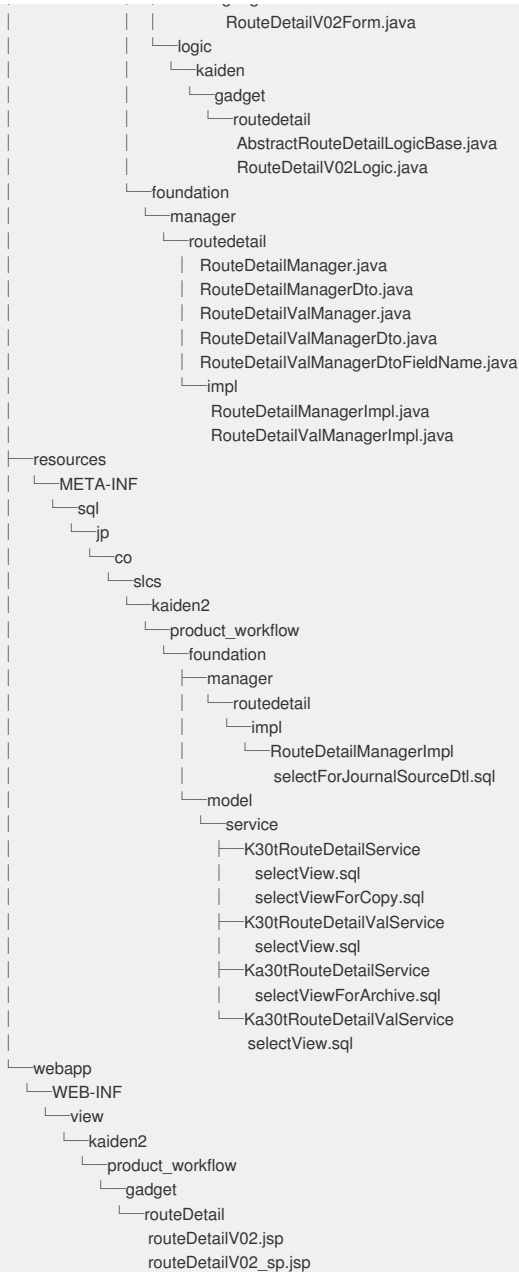
旅程明細 (複数通貨)

旅程明細 (複数通貨) のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action \(基底\)](#)
 - [Action](#)
 - [Form](#)
 - [Logic \(基底\)](#)
 - [Logic](#)
 - [FieldName](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
 - [コピー](#)
 - [仕訳連携](#)
 - [アーカイブ](#)
- [JSP](#)
 - [PC](#)
 - [SP](#)

[一覧](#)





Java

Action(基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractRouteDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

```

/**

```

* 旅程明細がジェットActionの抽象クラス。
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public abstract class AbstractRouteDetailActionBase implements GadgetActionIF {

    /**
     * 対象ロジッククラスをサブクラスより返却。
     *
     * @return ロジッククラス
     */
    @Override
    public abstract PluginLogicIF getLogic();

    /**
     * Fromをサブクラスより返却。
     *
     * @return From
     */
    @Override
    public abstract AbstractImwPageForm getForm();

    /**
     * 申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute validator = false
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute validator = false
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadProcess() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認画面のデータロード。
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute validator = false

```

```

public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConveror());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConveror())));
    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {

```

```

        ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
        return null;
    }

    /**
     * 書式変換設定を返却.
     * <p>
     * 標準では{@link #RouteDetailConverter 旅程明細 (標準) ガジェット}の書式変換を返却します.<br>
     * 書式を変更する場合は、サブクラスでオーバーライドしてください
     * </p>
     *
     * @return 書式変換後データ
     */
    protected ConvertForJsonSource getConverter() {
        return new RouteDetailConverter(getForm());
    }

    /**
     * 旅程明細ガジェットの書式変換.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    private static class RouteDetailConverter extends ConvertForJsonSource {

        /** リクエストパラメータ. */
        private AbstractImwPageForm frm;

        /** NumberHelper. */
        private NumberHelper numberHelper;

        /**
         * コンストラクタ.
         *
         * @param frm リクエストパラメータ
         */
        public RouteDetailConverter(AbstractImwPageForm frm) {
            this.frm = frm;
            numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
        }

        /**
         * {@inheritDoc}
         * <p>
         * 本サブクラスでは、以下の書式変換を行う.<br>
         * <table border="1">
         * <tr>
         * <td>項目名</td>
         * <td>書式</td>
         * </tr>
         * <tr>
         * <td>取引通貨</td>
         * <td>スケールを通貨のスケールに変換</td>
         * </tr>
         * <tr>
         * <td>レート</td>
         * <td>スケールをレートスケールに変換</td>
         * </tr>
         * <tr>
         * <td>換算額</td>
         * <td>スケールを会社通貨のスケールに変換</td>
         * </tr>
         * <tr>
         * <td>会社通貨額合計</td>
         * <td>スケールを会社通貨のスケールに変換</td>
         * </tr>
         * </table>
         * </p>
         */
        @Override
        protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

            Map<String, Object> result = new HashMap<String, Object>(map);

            String currencyCd = (String) result.get(ROUTE_DETAIL_TRANS_CURRENCY_CD);
            String companyCurrencyCd = (String) result.get(ROUTE_DETAIL_COMPANY_CURRENCY_CD);
            // 取引通貨額の書式変換
            if (result.containsKey(ROUTE_DETAIL_TRANS_AMOUNT)) {
                BigDecimal amount = (BigDecimal) result.get(ROUTE_DETAIL_TRANS_AMOUNT);
                result.put(ROUTE_DETAIL_TRANS_AMOUNT,
                    numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, currencyCd));
            }
            // レートの書式変換
            if (result.containsKey(ROUTE_DETAIL_COMPANY_RATE)) {
                BigDecimal rate = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_RATE);
                result.put(ROUTE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
            }
        }
    }

```



```

// 換算額の書式変換
if (result.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT)) {
    BigDecimal companyAmount = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_AMOUNT);
    result.put(ROUTE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
        frm.searchCriteriaCompany, companyCurrencyCd));
}

// 会社通貨額合計の書式変換
if (result.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT_SUM)) {
    BigDecimal companyAmountSum = (BigDecimal) result.get(ROUTE_DETAIL_COMPANY_AMOUNT_SUM);
    result.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4Input(companyAmountSum,
        frm.searchCriteriaCompany, companyCurrencyCd));
}

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>換算額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額合計</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * <p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(ROUTE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(ROUTE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(ROUTE_DETAIL_TRANS_CURRENCY_CD);
        result.put(ROUTE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_RATE);
        result.put(ROUTE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }

    Object companyCurrencyCd = map.get(ROUTE_DETAIL_COMPANY_CURRENCY_CD);

    // 換算額の書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_AMOUNT);
        result.put(ROUTE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    // 会社通貨額合計の書式変換
    if (map.containsKey(ROUTE_DETAIL_COMPANY_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(ROUTE_DETAIL_COMPANY_AMOUNT_SUM);
        result.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4View(companyAmountSum,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }

    return result;
}
}

```

Action

- Path

```
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/RouteDetailV02Action.java
```

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.RouteDetailV02Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail.RouteDetailV02Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 旅程明細(標準・外貨)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV02Action extends AbstractRouteDetailActionBase {

    /**
     * RouteDetailV02Logic.
     */
    @Resource
    public RouteDetailV02Logic routeDetailV02Logic;

    /**
     * RouteDetailV02Form.
     */
    @ActionForm
    @Resource
    public RouteDetailV02Form routeDetailV02Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return routeDetailV02Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return routeDetailV02Form;
    }
}

```

Form

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/RouteDetailV02Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 旅程明細(標準・外貨)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV02Form extends AbstractImwPageForm {

}

```

Logic(基底)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/routedetail/AbstractRouteDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNameNames.GADGET_CLASS;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ARRIVAL_SPOT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUND_TRIP_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUTE_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_STARTING_SPOT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_SUPPLEMENT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.IdentifierHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.FlagValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;

```

```

import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

/**
 * 旅程ガジェット用ロジックの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractRouteDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 旅程明細(ヘッダー)。 */
    protected static final String BLOCK_KEY_ROUTE_INFO_TABLE = "routeInfoTable";
    /** ガジェットブロックキー: 旅程明細(明細)。 */
    protected static final String BLOCK_KEY_ROUTE_DETAIL_TABLE = "routeDetailTable";

    /** 旅程日付項目長。 */
    public static final int ROUTE_DATE_LEN = 100;
    /** 出発地項目長。 */
    public static final int STARTING_SPOT_LEN = 250;
    /** 到着地項目長。 */
    public static final int ARRIVAL_SPOT_LEN = 250;
    /** 補足項目長。 */
    public static final int SUPPLEMENT_LEN = 1000;
    /** 経路検索id項目長。 */
    public static final int ROUTE_SEARCH_ID_LEN = 100;
    /** 税率(整数部)。 */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部)。 */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部)。 */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部)。 */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先項目長。 */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部)。 */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部)。 */
    public static final int QUANTITY_DEC_LEN = 10;

    /** 旅程明細. 予備項目A~Jの最大長。 */
    public static final int ROUTE_EXTENSION_COMON_LEN = 250;

    /** RouteDetailManager。 */
    @Resource
    protected RouteDetailManager routeDetailManager;

    /** RouteDetailValManager。 */
    @Resource
    protected RouteDetailValManager routeDetailValManager;

    /** JournalSourceDtlManager。 */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper。 */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** IdentifierHelper。 */
    @Resource
    protected IdentifierHelper identifierHelper;

    /** コンストラクタ。 */
    public AbstractRouteDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細を登録します。
     * </p>
     */

```

```

    /
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる旅程明細テーブルを削除後に、旅程明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、旅程明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、旅程明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */

```

```

</p>
*/
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 旅程明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、旅程明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}

```

```

    /**
     * <p>
     * 旅程明細を登録します。
     * </p>
     */
    @Override
    public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 旅程明細テーブルの対象データを削除します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTable(true);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる旅程明細テーブルを削除後に、旅程明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override
    public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本機能では何も処理を行わず、常に0を返却します。
     * </p>
     */
    @Override
    public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return 0;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
     * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
     * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
     * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
     * </p>
     */
    @Override
    public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

        PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
        MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
            pluginGadgetDTO.imwPluginParameterDto);

        String status = matterEndProcessDto.lastResultStatus;

        int rv = 0;

        if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

            if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
                // 旅程明細データ取得キー
                CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
                criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
                criteriaDto.userId = pluginGadgetDTO.userId;
                criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
                criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
                // 旅程明細データを取得
                List<BeanMap> beanMapList = routeDetailManager.selectForJournalSourceDtl(criteriaDto);
                List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                    beanMapList);

                // 仕訳連携用明細テーブル登録処理
                rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
            }
        }

        return rv;
    }
}

```

```

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる旅程明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc} *
 * <p>
 * 対象となる(過去案件)旅程明細テーブルを一括削除します。
 * </p>
 */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    CriteriaForSqlDto criteria = super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().lmwPluginParameterDto);
    routeDetailValManager.removeAllArchive(criteria);
    int rv = routeDetailManager.removeAllArchive(criteria);
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる旅程明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 *
 * @see #viewMapFormatted
 */
@Override
protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    return viewMapFormatted(conditionDto, null);
}

/**
 * {@inheritDoc}
 * <p>
 * データの取得は、{@link RouteDetailManager#selectList(RouteDetailManagerDto)}を利用します。
 * </p>
 */
@Override
protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
    throws KaidenSystemException {

    RouteDetailManagerDto dto = Beans.createAndCopy(RouteDetailManagerDto.class, conditionDto).execute();
    List<RouteDetailManagerDto> listSelect = routeDetailManager.selectList(dto);
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
    List<Map<String, Object>> lstDtoMap = new ArrayList<Map<String, Object>>();
    for (RouteDetailManagerDto src : listSelect) {
        if (converter == null) {
            lstDtoMap.add(BeanUtil.toMap(src, true));
        } else {
            lstDtoMap.add(converter.mapConvert(BeanUtil.toMap(src, true)));
        }
    }
    rv.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, lstDtoMap);
    return rv;
}

/**
 * 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ登録処理。
 * <D>

```



```

* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は登録対象外です。
* <p>
*
*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int insertTable() throws KaidenSystemException {
    // 旅程明細ガジェットデータをMap形式で取得
    List<RouteDetailManagerDto> dtos = JsonUtil.getDtoList(RouteDetailManagerDto.class,
        getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE));
    for (RouteDetailManagerDto dto : dtos) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {
            dto.routeSearchId = identifierHelper.get();

            // 旅程明細駅すばあと検索テーブルへの登録
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            valTarget.commutationRoute = dto.commutationRoute;
            valTarget.searchCondition = dto.searchCondition;
            valTarget.routeInfo = dto.routeInfo;
            valTarget.assessDate = dto.assessDate;
            valTarget.tranceMeansFlag = dto.tranceMeansFlag;
            valTarget.assessFlag = dto.assessFlag;
            routeDetailValManager.entryIgnoreAuthCompany(valTarget);
        }
    }
    // 旅程明細テーブル登録
    return routeDetailManager.entry(dtos);
}

/**
* 仕訳連携用明細テーブルデータ登録処理.
*
* @param srcList 旅程明細テーブルデータリスト
* @param matterEndProcessDto 案件終了処理用DTO
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
    MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
    for (JournalSourceDtlManagerDto srcDto : srcList) {
        // 仕訳データ区分をセット
        srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
    }

    // 仕訳連携用明細テーブル登録
    return journalSourceDtlManager.entry(srcList);
}

/**
* 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ削除処理.
* <p>
* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は、削除対象外です。
* <p>
*
* @param tempFlg 一時保存データフラグ
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    RouteDetailManagerDto mngDto = new RouteDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userId = super.getGadgetDto().userId;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    List<RouteDetailManagerDto> detailList = routeDetailManager.selectList(mngDto);
    for (RouteDetailManagerDto dto : detailList) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {

            // 旅程明細駅すばあと検索テーブルの削除
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            routeDetailValManager.removeIgnoreAuthCompany(valTarget);
        }
    }
    return routeDetailManager.remove(mngDto);
}

/**
* 旅程明細テーブル・旅程明細駅すばあと検索テーブルデータ削除処理(履歴を含む).
* <p>
* 旅程明細駅すばあと検索テーブルについては、駅すばあと連携をしていない場合は、削除対象外です。
* <p>

```

```

*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    RouteDetailManagerDto mngDto = new RouteDetailManagerDto();
    mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
    mngDto.userDataId = null;
    mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
    mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    List<RouteDetailManagerDto> detailList = routeDetailManager.selectList(mngDto);
    for (RouteDetailManagerDto dto : detailList) {
        if (GenericFlag.isOn(dto.routeLinkFlag)) {

            // 旅程明細駅すばあと検索テーブルの削除
            RouteDetailValManagerDto valTarget = new RouteDetailValManagerDto();
            valTarget.routeSearchId = dto.routeSearchId;
            routeDetailValManager.removeIgnoreAuthCompany(valTarget);
        }
    }
    return routeDetailManager.removeAll(mngDto);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {

```

```

    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>
 * </tr>
 * <tr>
 * <td>旅程日付</td>
 * <td>{@link DateValidator 日付型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>出発地</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>到着地</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>

```

```

* <td align="center">入力</td>
* </tr>
* <tr>
* <td>補足</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>往復フラグ</td>
* <td>{@link FlagValidator フラグチェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>経費区分コード</td>
* <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>取引通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>取引通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td rowspan="2">会社通貨換算レート</td>
* <td>{@link DigitsValidator 小数型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨コード</td>
* <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>会社通貨額</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数型チェック}</td>
* <td align="center">入力</td>
* </tr>

```

```

* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
KaidenApplicationException {

    if (fieldDefsMap.containsKey(BLOCK_KEY_ROUTE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_ROUTE_DETAIL_TABLE, tmpFlg);
        // 旅程日付
        addValidateOnlyInput(ROUTE_DETAIL_ROUTE_DATE, VALID_DATE);
        // 出発地
        addValidateOnlyInput(ROUTE_DETAIL_STARTING_SPOT, VALID_MAX_LENGTH, getArgs(STARTING_SPOT_LEN));
        // 到着地
        addValidateOnlyInput(ROUTE_DETAIL_ARRIVAL_SPOT, VALID_MAX_LENGTH, getArgs(ARRIVAL_SPOT_LEN));
        // 補足
        addValidateOnlyInput(ROUTE_DETAIL_SUPPLEMENT, VALID_MAX_LENGTH, getArgs(SUPPLEMENT_LEN));
        // 往復フラグ
        addValidateOnlyInput(ROUTE_DETAIL_ROUND_TRIP_FLAG, VALID_FLAG);
        // 経費区分コード
        addValidate(ROUTE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, ROUTE_DETAIL_EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidate(ROUTE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ROUTE_DETAIL_ACCOUNT_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidate(ROUTE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, ROUTE_DETAIL_TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateOnlyInput(ROUTE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 取引通貨コード
        addValidate(ROUTE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, ROUTE_DETAIL_TRANS_CURRENCY_CD,
            getGadgetDto().searchCriteriaCompany);
        // 会社通貨換算レート
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_RATE, VALID_DIGITS,
            getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
        // 会社通貨コード
        addValidate(ROUTE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, ROUTE_DETAIL_COMPANY_CURRENCY_CD,
            getGadgetDto().searchCriteriaCompany);
        // 会社通貨額
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
        // 支払分類コード
        addValidate(ROUTE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, ROUTE_DETAIL_PAY_TYPE_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidate(ROUTE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, ROUTE_DETAIL_PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払先名
        addValidateOnlyInput(ROUTE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
        // 組織コード
        addValidate(ROUTE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCompany,
            ROUTE_DETAIL_DEPARTMENT_SET_CD, ROUTE_DETAIL_DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidate(ROUTE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, ROUTE_DETAIL_PROJECT_CD,
            getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 数量
        addValidateOnlyInput(ROUTE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
        // 会社通貨額合計
        addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
    }
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldKey, validationId, args);
}

```

```

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加。
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 *
 * @param fieldKey フィールドキー
 * @param validationId バリデートキー
 * @param args バリデート実行時アーギュメント
 */
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
 * 設定されている場合は、{@link RouteDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, new ArrayList<RouteDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectView(criteria));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link RouteDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link RouteDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ROUTE_DETAIL_TABLE, routeDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した旅程明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_depaertment)</li>
 * <li>プロジェクトマスタ</li>
 * <li>支払分類マスタ</li>
 * <li>支払方法マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * <p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_ROUTE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<RouteDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), RouteDetailManagerDto.class);
        }
    }
}

```

```

for (RouteDetailManagerDto j : sqlResultList) {
    // 取引通貨チェック
    if (isNotEmpty(j.transCurrencyCd) && isEmpty(j.transCurrencyName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_TRANS_CURRENCY_CD, ROUTE_DETAIL_TRANS_CURRENCY_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.CURRENCY);
        String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 会社通貨チェック
    if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_COMPANY_CURRENCY_CD, ROUTE_DETAIL_COMPANY_CURRENCY_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.CURRENCY);
        String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 組織チェック
    if (isNotEmpty(j.departmentSetCd) && isEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_DEPARTMENT_SET_CD, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_DEPARTMENT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            BaseProp.S.DEPARTMENT);
        String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // プロジェクトチェック
    if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PROJECT_CD, ROUTE_DETAIL_PROJECT_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PROJECT);
        String[] values = getArgs(j.projectCd, j.projectName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 支払分類マスタチェック
    if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PAY_TYPE_CD, ROUTE_DETAIL_PAY_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PAY_TYPE);
        String[] values = getArgs(j.payTypeCd, j.payTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 支払方法マスタチェック
    if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_PAY_METHOD_CD, ROUTE_DETAIL_PAY_METHOD_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.PAY_METHOD);
        String[] values = getArgs(j.payMethodCd, j.payMethodName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
            .toString(j.tupleId), masterName, values, message));
    }

    // 経費区分マスタチェック
    if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
        String[] fieldId = getArgs(ROUTE_DETAIL_EXP_TYPE_CD, ROUTE_DETAIL_EXP_TYPE_NAME);
        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.EXP_TYPE);
        String[] values = getArgs(j.expTypeCd, j.expTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

        if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
            // 勘定科目、税区分をクリア
            j.accountCd = null;
            j.accountName = null;
            j.taxTypeCd = null;
            j.taxTypeName = null;
            // 税率、税抜金額、税額をクリア
            j.taxRate = null;
            j.companyAmountExcTax = null;
            j.companyAmountTax = null;
        }
    }
}

```

```

        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 勘定科目マスタチェック
if (isNotEmpty(j.accountCd) && isEmpty(j.accountName)) {
    String[] fieldId = getArgs(ROUTE_DETAIL_ACCOUNT_CD, ROUTE_DETAIL_ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(j.accountCd, j.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 税区分マスタチェック
if (isNotEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
    String[] fieldId = getArgs(ROUTE_DETAIL_TAX_TYPE_CD, ROUTE_DETAIL_TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ROUTE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}
}
}
}
return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 旅程明細テーブルの対象データを過去案件退避します。<br>
 * 駅すばあと連携している場合は、旅程明細駅すばあと検索情報テーブルも過去案件退避の対象となります。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    // 旅程明細テーブル
    @SuppressWarnings("unchecked")
    List<RouteDetailManagerDto> listRoute = (List<RouteDetailManagerDto>) dataMap.get(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    int rv = routeDetailManager.matterArchiveRegister(criteria, listRoute);

    List<RouteDetailValManagerDto> listVal = new ArrayList<RouteDetailValManagerDto>();
    for (RouteDetailManagerDto j : listRoute) {
        if (StringUtil.isEmpty(j.routeSearchId)) {
            RouteDetailValManagerDto val = new RouteDetailValManagerDto();
            val.routeSearchId = j.routeSearchId;
            val.routeSearchId = j.routeSearchId;
            val.commutationRoute = j.commutationRoute;
            val.searchCondition = j.searchCondition;
            val.routeInfo = j.routeInfo;
            val.assessDate = j.assessDate;
            val.tranceMeansFlag = j.tranceMeansFlag;
            val.assessFlag = j.assessFlag;
            val.localeId = criteria.localeId;
            listVal.add(val);
        }
    }
    routeDetailValManager.matterArchiveRegister(criteria, listVal);

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 固有処理として、対象となる旅程明細駅すばあと検索情報テーブルのデータを削除します。
 * </p>
 */
@Override
protected void matterArchivePost(KaidenDto dto, CriteriaForSqlDto criteria) {
    routeDetailValManager.matterArchivePostDelete(criteria);
}

```



```

routeDetailManager.getMatterEnv().delete(criteria);
    super.matterArchivePost(dto, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(routeDetailManager);
    return list;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();
    list.add(SYSTEM_MATTER_ID);
    list.add(USER_DATA_ID);
    list.add(GADGET_CLASS);
    list.add(GADGET_INSTANCE);
    return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccountIF#getClaimExpenccount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG);
        String strAmnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

```

```

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = (String) j.get(RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG);
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isEmpty(strAmtnt)) {
                sum = sum.add(new BigDecimal(strAmtnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmtnt)) {
            list.add(new BigDecimal(strAmtnt));
        }
    }

    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmtnt = StringUtil.toString(j.get(RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmtnt)) {
            sum = sum.add(new BigDecimal(strAmtnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * 往復フラグに基づく数量を返却.
 *
 * @param roundTripFlag 往復フラグ
 * @return 往復フラグに基づく数量
 */
protected BigDecimal getQuantityFromRoundTrip(String roundTripFlag) {
    if (GenericFlag.isOff(roundTripFlag)) {
        return BigDecimal.ONE;
    } else if (GenericFlag.isOn(roundTripFlag)) {

```

```

        return new BigDecimal("2");
    }
    return null;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/routedetail/RouteDetailV02Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.routedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_ROUND_TRIP_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail.RouteDetailFieldName.ROUTE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

/**
 * 旅程情報(標準・外貨)ガジェット(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailV02Logic extends AbstractRouteDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public RouteDetailV02Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     *
     * <p>
     * サブクラスで追加定義します。<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     *

```

```

* <tr>
* <td>会社通貨額</td>
* <td>{@link NumericMinValidator 最小数値チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード/プロジェクトコード</td>
* <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*/

@Override
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    super.defineValidateForNormal(params, tmpFlg);

    // 取引通貨額
    addValidateOnlyInput(ROUTE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
    // 会社通貨額
    addValidateOnlyInput(ROUTE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

    if (!tmpFlg) {
        // 組織コード
        addValidateOnlyInput(ROUTE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_PROJECT_CD);
        // プロジェクトコード
        addValidateOnlyInput(ROUTE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, ROUTE_DETAIL_DEPARTMENT_CD,
            ROUTE_DETAIL_PROJECT_CD);
    }
}

/**
 * {@inheritDoc}
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>数量: 往復フラグに基づく数値を設定</li>
 * <li>立替フラグ: 支払分類コードに紐づく立替フラグを支払分類マスタから取得して設定</li>
 * </ul>
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(ROUTE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 数量
            BigDecimal roundTripFlag = getQuantityFromRoundTrip((String) j.get(ROUTE_DETAIL_ROUND_TRIP_FLAG));
            j.put(ROUTE_DETAIL_QUANTITY, StringUtil.toString(roundTripFlag));

            // 立替フラグ
            if (StringUtil.isNotEmpty((String) j.get(ROUTE_DETAIL_PAY_TYPE_CD))) {
                PayTypeManagerDto criteria = new PayTypeManagerDto();
                criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
                criteria.payTypeCd = (String) j.get(ROUTE_DETAIL_PAY_TYPE_CD);
                criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                    .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
                List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                    criteria, new ArrayList<OrderDto>());
                if (CollectionUtil.isNotEmpty(retList)) {
                    j.put(ROUTE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
                }
            }
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの)。
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>

```

```

* <th>処理順</th>
* <th>補完処理</th>
* </tr>
* <tr>
* <td>100</td>
* <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
* </tr>
* <tr>
* <td>200</td>
* <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
* </tr>
* </table>
* </p>
*
* @return グローバルデータ補完処理
*/

@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * </p>
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(ROUTE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(ROUTE_DETAIL_TRANS_AMOUNT),
                (String) j.get(ROUTE_DETAIL_COMPANY_RATE)));
            j.put(ROUTE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(ROUTE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(ROUTE_DETAIL_QUANTITY)));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>支払方法コード:ガジェット間連携項目. 支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ROUTE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード
            if (gcPayMethodCd == null) {
                gcPayMethodCd = getGadgetCouplingPayMethod();
            }
            j.put(ROUTE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
        }
    }
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenCount(Map<String, Object> params) {
    return super.putClaimExpenCount(params);
}

```

```

}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/routedetail/RouteDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.routedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 旅程明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class RouteDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected RouteDetailFieldName() {

        /** 旅程日付. */
        public static final String ROUTE_DETAIL_ROUTE_DATE = "routeDate";
        /** 出発地. */
        public static final String ROUTE_DETAIL_STARTING_SPOT = "startingSpot";
        /** 到着地. */
        public static final String ROUTE_DETAIL_ARRIVAL_SPOT = "arrivalSpot";
        /** 補足. */
        public static final String ROUTE_DETAIL_SUPPLEMENT = "supplement";
        /** 経費区分コード. */
        public static final String ROUTE_DETAIL_EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String ROUTE_DETAIL_ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String ROUTE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String ROUTE_DETAIL_TAX_RATE = "taxRate";
        /** 数量. */
        public static final String ROUTE_DETAIL_QUANTITY = "quantity";
        /** 取引通貨コード. */
        public static final String ROUTE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String ROUTE_DETAIL_TRANS_AMOUNT = "transAmount";
        /** 会社通貨換算レート. */
        public static final String ROUTE_DETAIL_COMPANY_RATE = "companyRate";
        /** 会社通貨コード. */
        public static final String ROUTE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
        /** 会社通貨額. */
        public static final String ROUTE_DETAIL_COMPANY_AMOUNT = "companyAmount";
        /** 会社通貨税抜額. */
        public static final String ROUTE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
        /** 会社通貨税額. */
        public static final String ROUTE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
        /** 会社通貨額合計. */
        public static final String ROUTE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
        /** 支払分類コード. */

```

```

public static final String ROUTE_DETAIL_PAY_TYPE_CD = "payTypeCd";
/** 支払方法コード. */
public static final String ROUTE_DETAIL_PAY_METHOD_CD = "payMethodCd";
/** 支払先コード. */
public static final String ROUTE_DETAIL_PAYEE_CD = "payeeCd";
/** 支払先名. */
public static final String ROUTE_DETAIL_PAYEE_NAME = "payeeName";
/** 組織セットコード. */
public static final String ROUTE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
/** 組織コード. */
public static final String ROUTE_DETAIL_DEPARTMENT_CD = "departmentCd";
/** プロジェクトコード. */
public static final String ROUTE_DETAIL_PROJECT_CD = "projectCd";
/** 往復フラグ. */
public static final String ROUTE_DETAIL_ROUND_TRIP_FLAG = "roundTripFlag";
/** 経路情報連携フラグ. */
public static final String ROUTE_DETAIL_ROUTE_LINK_FLAG = "routeLinkFlag";
/** 経路検索id. */
public static final String ROUTE_DETAIL_ROUTE_SEARCH_ID = "routeSearchId";

/** 会社名. */
public static final String ROUTE_DETAIL_COMPANY_NAME = "companyName";
/** 経費区分名称. */
public static final String ROUTE_DETAIL_EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String ROUTE_DETAIL_ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String ROUTE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
/** 取引通貨名称. */
public static final String ROUTE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String ROUTE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 支払分類名称. */
public static final String ROUTE_DETAIL_PAY_TYPE_NAME = "payTypeName";
/** 支払方法名称. */
public static final String ROUTE_DETAIL_PAY_METHOD_NAME = "payMethodName";
/** 組織セット名称. */
public static final String ROUTE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
/** 組織名称. */
public static final String ROUTE_DETAIL_DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String ROUTE_DETAIL_PROJECT_NAME = "projectName";
/** 立替フラグ. */
public static final String ROUTE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 旅程明細テーブルマネージャークラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface RouteDetailManager extends PluginGadgetManager<RouteDetailManagerDto> {

    /** {@link #selectView(CriteriaForSqlDto) 表示案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link #selectViewForCopy(CriteriaForSqlDto) コピー案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link #selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /** {@link #selectViewArchive(CriteriaForSqlDto) 過去案件検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NMF_FOR_ARCHIVE = "selectViewForArchive.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManager.java


```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;

/**
 * 旅程明細駅すばあと検索情報テーブルへの各種操作を提供するAPIのinterface.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface RouteDetailValManager {

    /** {@link #matterArchivePostDelete(CriteriaForSqlDto) 過去案件一括削除}}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /**
     * 主キー検索.
     * <p>
     * 検索条件のうち、主キー(経路検索id)のみを利用して問い合わせします.<br>
     * 主キー条件が未設定の場合、RunTimeExceptionがスローされます。
     * </p>
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public RouteDetailValManagerDto selectIgnoreAuthCompany(RouteDetailValManagerDto criteria);

    /**
     * 登録.
     *
     * @param data 登録データ
     * @return 更新件数
     */
    public int entryIgnoreAuthCompany(RouteDetailValManagerDto data);

    /**
     * 削除.
     *
     * @param criteria 削除条件
     * @return 削除件数
     */
    public int removeIgnoreAuthCompany(RouteDetailValManagerDto criteria);

    /**
     * 過去案件登録.
     * <p>
     * 引数の登録データリストを過去案件用テーブルへ移送します.<br>
     * </p>
     *
     * @param dataList 登録データリスト
     * @param criteria 削除データ
     * @return 削除件数
     */
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<RouteDetailValManagerDto> dataList);

    /**
     * 過去案件処理時削除.
     * <p>
     * 引数のDTOのシステム案件IDに該当するデータを全件削除します.<br>
     * </p>
     *
     * @param criteria 削除データ
     * @return 削除件数
     */
    public int matterArchivePostDelete(CriteriaForSqlDto criteria);

    /**
     * 過去案件一括削除.
     * <p>
     * 引数の削除データリストを全件削除します.<br>
     * </p>
     *
     * @param criteria 削除条件
     * @return 削除件数
     */
    public int removeAllArchive(CriteriaForSqlDto criteria);
}

```

Manager(実装)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailManagerImpl.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.impl;

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.LOCALE_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tRouteDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tRouteDetailService;
import org.seasar.util.beans.util.BeanMap;
```

```

/* 旅程明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link RouteDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link RouteDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>表示案件検索</td>
 * <td>selectView</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>コピー案件検索</td>
 * <td>selectViewForCopy</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件検索</td>
 * <td>selectViewArchive</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括挿入</td>
 * <td>entry</td>
 * <td>登録</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括更新</td>
 * <td>renew</td>
 * <td>更新</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>削除</td>
 * <td>remove</td>
 * <td>削除</td>
 * <td><b>&nbsp;&nbsp;&nbsp;</b></td>
 * </tr>

```

[illegible]

[illegible]

```
public class RouteDetailManagerImpl extends AbstractWfManager<RouteDetailManagerDto> implements RouteDetailManager {

    /** K30tRouteDetailService. */
    @Resource
    protected K30tRouteDetailService k30tRouteDetailService;

    /** Ka30tRouteDetailService. */
    @Resource
    protected Ka30tRouteDetailService ka30tRouteDetailService;

    /**
     * {@inheritDoc}
     */
    @Override
    public RouteDetailManagerDto selectByPk(RouteDetailManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k30tRouteDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<RouteDetailManagerDto> selectList(RouteDetailManagerDto criteria) {
        return super.selectPkIgnoreAuthority(k30tRouteDetailService, criteria);
    }
}
```

```

    return super.selectRouteDetailData(criteriaForSqlDto criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<RouteDetailManagerDto> selectView(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForView(k30tRouteDetailService, SELECT_SQLFILE_NM, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<RouteDetailManagerDto> selectViewForCopy(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForViewForCopy(k30tRouteDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(criteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 * <p>
 * (過去案件)旅程明細駅すばあと検索情報テーブルも併せて取得するため、{@link RouteDetailManager#SELECT_SQLFILE_NMF_FOR_ARCHIVE SQLファイル}を利用します。
 * </p>
 */
@Override
public List<RouteDetailManagerDto> selectViewArchive(criteriaForSqlDto criteria) {
    // CriteriaForSqlDto入力チェック
    String[] fieldNames = {SYSTEM_MATTER_ID, USER_DATA_ID, GADGET_CLASS, GADGET_INSTANCE, LOCALE_ID};
    this.validateCriteriaForSqlDto(criteria, fieldNames);
    criteria.archiveMonth = null;
    return super.selectBySqlIgnoreAuthority(ka30tRouteDetailService, SELECT_SQLFILE_NMF_FOR_ARCHIVE, criteria,
        ka30tRouteDetailService.makeOrderListFromPk());
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<RouteDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.insertBatch(k30tRouteDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int renew(List<RouteDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.renewBatch(k30tRouteDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(RouteDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tRouteDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(RouteDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tRouteDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(criteriaForSqlDto criteria, List<RouteDetailManagerDto> dataList) {
    return super.entryArchive(ka30tRouteDetailService, criteria, dataList);
}

```

```

    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataId(k30tRouteDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataIdForArchive(ka30tRouteDetailService, criteria);
    }
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailValManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.impl;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail.RouteDetailValManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetailVal;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetailVal;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tRouteDetailValService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tRouteDetailValService;
import org.seasar.framework.beans.util.Beans;

/**
 * 旅程明細駅すばあと検索情報テーブルへの各種操作を提供するAPI.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailValManagerImpl implements RouteDetailValManager {

    /** K30tRouteDetailValService. */
    @Resource
    protected K30tRouteDetailValService k30tRouteDetailValService;

    /** Ka30tRouteDetailValService. */
    @Resource
    protected Ka30tRouteDetailValService ka30tRouteDetailValService;

    /**
     * {@inheritDoc}
     */
    @Override
    public RouteDetailValManagerDto selectIgnoreAuthCompany(RouteDetailValManagerDto criteria) {
        if (criteria == null || criteria.routeSearchId == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
        Map<String, Object> criteriaMap = BeanUtil.toMap(criteria);
        List<K30tRouteDetailVal> selectList = k30tRouteDetailValService.genericSelectList(criteriaMap,
            new ArrayList<OrderDto>());
        if (CollectionUtil.isEmpty(selectList)) {
            return null;
        }

        return Beans.createAndCopy(RouteDetailValManagerDto.class, selectList.get(0)).execute();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entryIgnoreAuthCompany(RouteDetailValManagerDto data) {
        if (data == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

```

```

        K30tRouteDetailVal insData = Beans.createAndCopy(K30tRouteDetailVal.class, data).execute();
        return k30tRouteDetailValService.insert(insData);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeIgnoreAuthCompany(RouteDetailValManagerDto criteria) {

        if (criteria == null || criteria.routeSearchId == null) {
            throw KaidenIllegalParametersException.makeInstance();
        }
        Map<String, Object> criteriaMap = BeanUtil.toMap(criteria);
        List<K30tRouteDetailVal> selectList = k30tRouteDetailValService.genericSelectList4Update(criteriaMap,
            new ArrayList<OrderDto>());
        if (CollectionUtil.isEmpty(selectList)) {
            return 0;
        }
        return k30tRouteDetailValService.delete(selectList.get(0));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<RouteDetailValManagerDto> dataList) {
        List<Ka30tRouteDetailVal> castListContent = new ArrayList<Ka30tRouteDetailVal>();
        for (RouteDetailValManagerDto j : dataList) {
            Ka30tRouteDetailVal data = Beans.createAndCopy(Ka30tRouteDetailVal.class, j).execute();
            data.localeId = criteria.localeId;
            data.archiveMonth = criteria.archiveMonth;
            castListContent.add(data);
        }
        return CollectionUtil.summaryInt(k30tRouteDetailValService.insertBatch(castListContent));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(lmwEntityCommonFieldNames.SYSTEM_MATTER_ID, criteria.systemMatterId);
        map.put(lmwEntityCommonFieldNames.GADGET_CLASS, criteria.gadgetClass);
        map.put(lmwEntityCommonFieldNames.GADGET_INSTANCE, criteria.gadgetInstance);
        List<Ka30tRouteDetailVal> dataList = k30tRouteDetailValService.selectBySql(K30tRouteDetailVal.class,
            SELECT_SQLFILE_NM, map, k30tRouteDetailValService.makeOrderListFromPk());
        return CollectionUtil.summaryInt(k30tRouteDetailValService.deleteBatch(dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put(lmwEntityCommonFieldNames.SYSTEM_MATTER_ID, criteria.systemMatterId);
        List<Ka30tRouteDetailVal> dataList = k30tRouteDetailValService.selectBySql(Ka30tRouteDetailVal.class,
            SELECT_SQLFILE_NM, map, k30tRouteDetailValService.makeOrderListFromPk());
        return CollectionUtil.summaryInt(k30tRouteDetailValService.deleteBatch(dataList));
    }
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 旅程明細テーブルDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailManagerDto extends KaidenDto {

```

```

/** システム案件id. */
public String systemMatterId;
/** ユーザーデータid. */
public String userDataId;
/** ガジェットクラス. */
public String gadgetClass;
/** ガジェットインスタンス. */
public Long gadgetInstance;
/** タプルid. */
public Long tupleId;
/** ガジェットid. */
public String gadgetId;
/** ガジェットバリエーション. */
public String gadgetVariation;
/** 行番号. */
public Long rowNumber;

/** 旅程日付. */
public Timestamp routeDate;
/** 出発地. */
public String startingSpot;
/** 到着地. */
public String arrivalSpot;
/** 補足. */
public String supplement;
/** 経費区分コード. */
public String expTypeCd;
/** 勘定科目コード. */
public String accountCd;
/** 税区分コード. */
public String taxTypeCd;
/** 税率. */
public BigDecimal taxRate;
/** 数量. */
public BigDecimal quantity;
/** 取引通貨コード. */
public String transCurrencyCd;
/** 取引通貨額. */
public BigDecimal transAmount;
/** 会社通貨換算レート. */
public BigDecimal companyRate;
/** 会社通貨コード. */
public String companyCurrencyCd;
/** 会社通貨額. */
public BigDecimal companyAmount;
/** 会社通貨税抜額. */
public BigDecimal companyAmountExcTax;
/** 会社通貨税額. */
public BigDecimal companyAmountTax;
/** 会社通貨額合計. */
public BigDecimal companyAmountSum;
/** 支払分類コード. */
public String payTypeCd;
/** 支払方法コード. */
public String payMethodCd;
/** 支払先コード. */
public String payeeCd;
/** 支払先名. */
public String payeeName;
/** 組織セットコード. */
public String departmentSetCd;
/** 組織コード. */
public String departmentCd;
/** プロジェクトコード. */
public String projectCd;
/** 往復フラグ. */
public String roundTripFlag;
/** 経路情報連携済フラグ. */
public String routeLinkFlag;
/** 経路検索id. */
public String routeSearchId;

/** 予備項目a. */
public String extensionA;
/** 予備項目b. */
public String extensionB;
/** 予備項目c. */
public String extensionC;
/** 予備項目d. */
public String extensionD;
/** 予備項目e. */
public String extensionE;
/** 予備項目f. */
public String extensionF;
/** 予備項目g. */
public String extensionG;

```



```

/** 予備項目h. */
public String extensionH;
/** 予備項目i. */
public String extensionI;
/** 予備項目j. */
public String extensionJ;
/** 登録日時. */
public Timestamp entryTs;
/** 登録者コード. */
public String entryUserCd;
/** 更新カウンタ. */
public Long renewCnt;
/** 更新日時. */
public Timestamp renewTs;
/** 更新者コード. */
public String renewUserCd;

/** 検索基準会社コード. */
public String searchCriteriaCompany;
/** 取引通貨名称. */
public String transCurrencyName;
/** 会社通貨名称. */
public String companyCurrencyName;
/** 組織名称. */
public String departmentName;
/** プロジェクトマスタプロジェクト名称. */
public String projectName;
/** 経費区分名称. */
public String expTypeName;
/** 勘定科目名称. */
public String accountName;
/** 税区分名称. */
public String taxTypeName;
/** 支払分類名称. */
public String payTypeName;
/** 支払方法名称. */
public String payMethodName;
/** 立替フラグ. */
public String paymentFlag;

/** 定期区間. */
public String commutationRoute;
/** 検索条件. */
public String searchCondition;
/** 経路情報. */
public String routeInfo;
/** 評価判定基準日. */
public Timestamp assessDate;
/** 利用交通機関フラグ. */
public String tranceMeansFlag;
/** 評価フラグ. */
public String assessFlag;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

/**
 * 旅程明細駅すばあと検索情報テーブルマネージャDTO.
 * <p>
 * マネージャクラス({@link RouteDetailValManager})へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class RouteDetailValManagerDto extends KaidenMasterManagerDto {

    /** 経路検索id. */
    public String routeSearchId;
    /** 定期区間. */
    public String commutationRoute;
    /** 検索条件. */
    public String searchCondition;
    /** 経路情報. */
    public String routeInfo;
    /** 評価判定基準日. */
    public Timestamp assessDate;
    /** 利用交通機関フラグ. */
    public String tranceMeansFlag;
    /** 評価フラグ. */
    public String assessFlag;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;

    /** ロケールID. */
    public String localeId;
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/RouteDetailValManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.routedetail;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityExtensionFieldNames;

/**
 * 旅程明細駅すばあと検索情報テーブルDTO フィールド名定義クラス。
 * <p>
 * マネージャクラス([@link RouteDetailValManager])へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class RouteDetailValManagerDtoFieldName implements EntityExtensionFieldNames {

    /**
     * コンストラクタ。
     */
    private RouteDetailValManagerDtoFieldName() {

    }

    /** 経路検索ID。 */
    public static final String ROUTE_SEARCH_ID = "routeSearchId";
    /** 定期区間。 */
    public static final String COMMUTATION_ROUTE = "commutationRoute";
    /** 検索条件。 */
    public static final String SEARCH_CONDITION = "searchCondition";
    /** 経路情報。 */
    public static final String ROUTE_INFO = "routeInfo";
    /** 評価判定基準日。 */
    public static final String ASSESS_DATE = "assessDate";
    /** 利用期間フラグ。 */
    public static final String TRANCE_MEANS_FLAG = "tranceMeansFlag";
    /** 評価フラグ。 */
    public static final String ASSESS_FLAG = "assessFlag";

}

```

Java (S2JDBC)

Entity

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 旅程明細テーブル。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_route_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tRouteDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID。 */
    private static final long serialVersionUID = 1L;

    /** システム案件id。 */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid。 */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス。 */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

```

```

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 旅程日付. */
@Column(name = "route_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp routeDate;

/** 出発地. */
@Column(name = "starting_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String startingSpot;

/** 到着地. */
@Column(name = "arrival_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String arrivalSpot;

/** 補足. */
@Column(name = "supplement", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String supplement;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

```

```

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 往復フラグ. */
@Column(name = "round_trip_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String roundTripFlag;

/** 経路情報連携済フラグ. */
@Column(name = "route_link_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String routeLinkFlag;

/** 経路検索id. */
@Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String routeSearchId;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailVal.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 旅程明細駅すばあと検索情報テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_route_detail_val")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tRouteDetailVal extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 経路検索id. */
    @Id
    @Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = false, unique = true)
    public String routeSearchId;

    /** 定期区間. */
    @Lob
    @Column(name = "commutation_route", columnDefinition = "clob", nullable = true, unique = false)
    public String commutationRoute;

    /** 検索条件. */
    @Lob
    @Column(name = "search_condition", columnDefinition = "clob", nullable = true, unique = false)
    public String searchCondition;

    /** 経路情報. */
    @Lob
    @Column(name = "route_info", columnDefinition = "clob", nullable = true, unique = false)
    public String routeInfo;

    /** 評価判定基準日. */
    @Column(name = "assess_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp assessDate;

    /** 利用交通機関フラグ. */
    @Column(name = "trance_means_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String tranceMeansFlag;

    /** 評価フラグ. */
    @Column(name = "assess_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String assessFlag;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

    /** 予備項目b. */
    @Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionB;

    /** 予備項目c. */
    @Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionC;

    /** 予備項目d. */
    @Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionD;

    /** 予備項目e. */
    @Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionE;

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)旅程明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_route_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tRouteDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** アーカイブ年月. */
    @Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
    public String archiveMonth;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;
}

```

```

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 旅程日付. */
@Column(name = "route_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp routeDate;

/** 出発地. */
@Column(name = "starting_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String startingSpot;

/** 到着地. */
@Column(name = "arrival_spot", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String arrivalSpot;

/** 補足. */
@Column(name = "supplement", columnDefinition = "varchar2(4000)", nullable = true, unique = false)
public String supplement;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

```



```

public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 往復フラグ. */
@Column(name = "round_trip_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String roundTripFlag;

/** 経路情報連携フラグ. */
@Column(name = "route_link_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String routeLinkFlag;

/** 経路検索id. */
@Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String routeSearchId;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */

```

```

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionG;

    /** 予備項目h. */
    @Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionH;

    /** 予備項目i. */
    @Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionI;

    /** 予備項目j. */
    @Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailVal.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)旅程明細駅すばあと検索情報テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_route_detail_val")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl"})
public final class Ka30tRouteDetailVal extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 経路検索id. */
    @Id
    @Column(name = "route_search_id", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String routeSearchId;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** アーカイブ年月. */
    @Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
    public String archiveMonth;

    /** 定期区間. */
    @Lob
    @Column(name = "commutation_route", columnDefinition = "clob", nullable = true, unique = false)
    public String commutationRoute;

    /** 検索条件. */
    @Lob
    @Column(name = "search_condition", columnDefinition = "clob", nullable = true, unique = false)
    public String searchCondition;

    /** 経路情報. */
    @Lob
    @Column(name = "route_info", columnDefinition = "clob", nullable = true, unique = false)
    public String routeInfo;

    /** 評価判定基準日. */
    @Column(name = "assess_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp assessDate;

    /** 利用交通機関フラグ. */
    @Column(name = "trance_means_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
    public String tranceMeansFlag;
}

```

```

/** 評価フラグ. */
@Column(name = "assess_flag", columnDefinition = "varchar2(10)", nullable = false, unique = false)
public String assessFlag;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tRouteDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl"})
public final class K30tRouteDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tRouteDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
}

```

```

*/
public static PropertyName<String> userDataId() {
    return new PropertyName<String>("userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public static PropertyName<String> gadgetClass() {
    return new PropertyName<String>("gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumのプロパティ名を返却.
 *
 * @return rowNumのプロパティ名
 */
public static PropertyName<Long> rowNum() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public static PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>("routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public static PropertyName<String> startingSpot() {
    return new PropertyName<String>("startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public static PropertyName<String> arrivalSpot() {
    return new PropertyName<String>("arrivalSpot");
}

```

```

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public static PropertyName<String> supplement() {
    return new PropertyName<String>("supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */

```

```

public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**

```

```

* departmentCdのプロパティ名を返却.
*
* @return departmentCdのプロパティ名
*/
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
* projectCdのプロパティ名を返却.
*
* @return projectCdのプロパティ名
*/
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
* roundTripFlagのプロパティ名を返却.
*
* @return roundTripFlagのプロパティ名
*/
public static PropertyName<String> roundTripFlag() {
    return new PropertyName<String>("roundTripFlag");
}

/**
* routeLinkFlagのプロパティ名を返却.
*
* @return routeLinkFlagのプロパティ名
*/
public static PropertyName<String> routeLinkFlag() {
    return new PropertyName<String>("routeLinkFlag");
}

/**
* routeSearchIdのプロパティ名を返却.
*
* @return routeSearchIdのプロパティ名
*/
public static PropertyName<String> routeSearchId() {
    return new PropertyName<String>("routeSearchId");
}

/**
* extensionAのプロパティ名を返却.
*
* @return extensionAのプロパティ名
*/
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
* extensionBのプロパティ名を返却.
*
* @return extensionBのプロパティ名
*/
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
* extensionCのプロパティ名を返却.
*
* @return extensionCのプロパティ名
*/
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
* extensionDのプロパティ名を返却.
*
* @return extensionDのプロパティ名
*/
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
* extensionEのプロパティ名を返却.
*
* @return extensionEのプロパティ名
*/
public static PropertyName<String> extensionE() {

```

```

    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *

```



```

    * @return renewUserCdのプロパティ名
    */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30tRouteDetailNames extends PropertyName<K30tRouteDetail> {

        /**
         * インスタンスを構築.
         */
        public _K30tRouteDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30tRouteDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30tRouteDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *
         * @return systemMatterIdのプロパティ名
         */
        public PropertyName<String> systemMatterId() {
            return new PropertyName<String>(this, "systemMatterId");
        }

        /**
         * userDataIdのプロパティ名を返却.
         *
         * @return userDataIdのプロパティ名
         */
        public PropertyName<String> userDataId() {
            return new PropertyName<String>(this, "userDataId");
        }

        /**
         * gadgetClassのプロパティ名を返却.
         *
         * @return gadgetClassのプロパティ名
         */
        public PropertyName<String> gadgetClass() {
            return new PropertyName<String>(this, "gadgetClass");
        }

        /**
         * gadgetInstanceのプロパティ名を返却.
         *
         * @return gadgetInstanceのプロパティ名
         */
        public PropertyName<Long> gadgetInstance() {
            return new PropertyName<Long>(this, "gadgetInstance");
        }

        /**
         * tupleIdのプロパティ名を返却.
         *
         * @return tupleIdのプロパティ名
         */
        public PropertyName<Long> tupleId() {
            return new PropertyName<Long>(this, "tupleId");
        }

        /**
         * gadgetIdのプロパティ名を返却.
         *
         * @return gadgetIdのプロパティ名
         */
    }

```

```

/
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>(this, "routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public PropertyName<String> startingSpot() {
    return new PropertyName<String>(this, "startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public PropertyName<String> arrivalSpot() {
    return new PropertyName<String>(this, "arrivalSpot");
}

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public PropertyName<String> supplement() {
    return new PropertyName<String>(this, "supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**

```

```

    * taxRateのプロパティ名を返却.
    *
    * @return taxRateのプロパティ名
    */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {

```

```

    private PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

```

```

/**
 * payTypeCdのプロパティ名を返却。
 *
 * @return payTypeCdのプロパティ名
 */

```

```

public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

```

```

/**
 * payMethodCdのプロパティ名を返却。
 *
 * @return payMethodCdのプロパティ名
 */

```

```

public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

```

```

/**
 * payeeCdのプロパティ名を返却。
 *
 * @return payeeCdのプロパティ名
 */

```

```

public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

```

```

/**
 * payeeNameのプロパティ名を返却。
 *
 * @return payeeNameのプロパティ名
 */

```

```

public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

```

```

/**
 * departmentSetCdのプロパティ名を返却。
 *
 * @return departmentSetCdのプロパティ名
 */

```

```

public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

```

```

/**
 * departmentCdのプロパティ名を返却。
 *
 * @return departmentCdのプロパティ名
 */

```

```

public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

```

```

/**
 * projectCdのプロパティ名を返却。
 *
 * @return projectCdのプロパティ名
 */

```

```

public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

```

```

/**
 * roundTripFlagのプロパティ名を返却。
 *
 * @return roundTripFlagのプロパティ名
 */

```

```

public PropertyName<String> roundTripFlag() {
    return new PropertyName<String>(this, "roundTripFlag");
}

```

```

/**
 * routeLinkFlagのプロパティ名を返却。
 *
 * @return routeLinkFlagのプロパティ名
 */

```

```

public PropertyName<String> routeLinkFlag() {
    return new PropertyName<String>(this, "routeLinkFlag");
}

```

```

/**
 * routeSearchIdのプロパティ名を返却。

```

```

*
* @return routeSearchIdのプロパティ名
*/
public PropertyName<String> routeSearchId() {
    return new PropertyName<String>(this, "routeSearchId");
}

/**
* extensionAのプロパティ名を返却。
*
* @return extensionAのプロパティ名
*/
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
* extensionBのプロパティ名を返却。
*
* @return extensionBのプロパティ名
*/
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
* extensionCのプロパティ名を返却。
*
* @return extensionCのプロパティ名
*/
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
* extensionDのプロパティ名を返却。
*
* @return extensionDのプロパティ名
*/
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
* extensionEのプロパティ名を返却。
*
* @return extensionEのプロパティ名
*/
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
* extensionFのプロパティ名を返却。
*
* @return extensionFのプロパティ名
*/
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
* extensionGのプロパティ名を返却。
*
* @return extensionGのプロパティ名
*/
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
* extensionHのプロパティ名を返却。
*
* @return extensionHのプロパティ名
*/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
* extensionIのプロパティ名を返却。
*
* @return extensionIのプロパティ名
*/
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

```

```

    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tRouteDetailValNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tRouteDetailVal}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tRouteDetailValNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tRouteDetailValNames() {
        super();
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
}

```

```

    * @return routeSearchIdのプロパティ名
    */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * commutationRouteのプロパティ名を返却.
     *
     * @return commutationRouteのプロパティ名
     */
    public static PropertyName<String> commutationRoute() {
        return new PropertyName<String>("commutationRoute");
    }

    /**
     * searchConditionのプロパティ名を返却.
     *
     * @return searchConditionのプロパティ名
     */
    public static PropertyName<String> searchCondition() {
        return new PropertyName<String>("searchCondition");
    }

    /**
     * routeInfoのプロパティ名を返却.
     *
     * @return routeInfoのプロパティ名
     */
    public static PropertyName<String> routeInfo() {
        return new PropertyName<String>("routeInfo");
    }

    /**
     * assessDateのプロパティ名を返却.
     *
     * @return assessDateのプロパティ名
     */
    public static PropertyName<Timestamp> assessDate() {
        return new PropertyName<Timestamp>("assessDate");
    }

    /**
     * tranceMeansFlagのプロパティ名を返却.
     *
     * @return tranceMeansFlagのプロパティ名
     */
    public static PropertyName<String> tranceMeansFlag() {
        return new PropertyName<String>("tranceMeansFlag");
    }

    /**
     * assessFlagのプロパティ名を返却.
     *
     * @return assessFlagのプロパティ名
     */
    public static PropertyName<String> assessFlag() {
        return new PropertyName<String>("assessFlag");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }
}

```

```

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */

```



```

    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30tRouteDetailValNames extends PropertyName<K30tRouteDetailVal> {

        /**
         * インスタンスを構築.
         */
        public _K30tRouteDetailValNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30tRouteDetailValNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30tRouteDetailValNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * routeSearchIdのプロパティ名を返却.
         *
         * @return routeSearchIdのプロパティ名
         */
        public PropertyName<String> routeSearchId() {
            return new PropertyName<String>((this, "routeSearchId"));
        }

        /**
         * commutationRouteのプロパティ名を返却.
         *
         * @return commutationRouteのプロパティ名
         */
        public PropertyName<String> commutationRoute() {
            return new PropertyName<String>((this, "commutationRoute"));
        }

        /**
         * searchConditionのプロパティ名を返却.
         *
         * @return searchConditionのプロパティ名
         */
        public PropertyName<String> searchCondition() {
            return new PropertyName<String>((this, "searchCondition"));
        }

        /**
         * routeInfoのプロパティ名を返却.
         *
         * @return routeInfoのプロパティ名
         */
        public PropertyName<String> routeInfo() {
            return new PropertyName<String>((this, "routeInfo"));
        }
    }

```

```

    return new PropertyName<String> (this, "extensionF");
}

/**
 * assessDateのプロパティ名を返却.
 *
 * @return assessDateのプロパティ名
 */
public PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>(this, "assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却.
 *
 * @return tranceMeansFlagのプロパティ名
 */
public PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>(this, "tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却.
 *
 * @return assessFlagのプロパティ名
 */
public PropertyName<String> assessFlag() {
    return new PropertyName<String>(this, "assessFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *

```

```

    * @return extensionGのプロパティ名
    */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailNames.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;
```

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tRouteDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tRouteDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tRouteDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClass のプロパティ名を返却.
     *
     * @return gadgetClass のプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstance のプロパティ名を返却.
     *
     * @return gadgetInstance のプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleId のプロパティ名を返却.
     *
     * @return tupleId のプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * localeId のプロパティ名を返却.
     *
     * @return localeId のプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonth のプロパティ名を返却.
     *
     * @return archiveMonth のプロパティ名
     */
    public static PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }

    /**
     * gadgetId のプロパティ名を返却.
     *
     * @return gadgetId のプロパティ名
     */

```

```

    * @return gadgetIdのプロパティ名
    */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却.
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却.
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * routeDateのプロパティ名を返却.
     *
     * @return routeDateのプロパティ名
     */
    public static PropertyName<Timestamp> routeDate() {
        return new PropertyName<Timestamp>("routeDate");
    }

    /**
     * startingSpotのプロパティ名を返却.
     *
     * @return startingSpotのプロパティ名
     */
    public static PropertyName<String> startingSpot() {
        return new PropertyName<String>("startingSpot");
    }

    /**
     * arrivalSpotのプロパティ名を返却.
     *
     * @return arrivalSpotのプロパティ名
     */
    public static PropertyName<String> arrivalSpot() {
        return new PropertyName<String>("arrivalSpot");
    }

    /**
     * supplementのプロパティ名を返却.
     *
     * @return supplementのプロパティ名
     */
    public static PropertyName<String> supplement() {
        return new PropertyName<String>("supplement");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }
}

```

```

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */

```

```

    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * roundTripFlagのプロパティ名を返却.
     *
     * @return roundTripFlagのプロパティ名
     */
    public static PropertyName<String> roundTripFlag() {
        return new PropertyName<String>("roundTripFlag");
    }

    /**
     * routeLinkFlagのプロパティ名を返却.
     *
     * @return routeLinkFlagのプロパティ名
     */
    public static PropertyName<String> routeLinkFlag() {
        return new PropertyName<String>("routeLinkFlag");
    }

    /**

```

```

    * routeSearchIdのプロパティ名を返却。
    *
    * @return routeSearchIdのプロパティ名
    */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * expTypeNameのプロパティ名を返却。
     *
     * @return expTypeNameのプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却。
     *
     * @return accountNameのプロパティ名
     */
    public static PropertyName<String> accountName() {
        return new PropertyName<String>("accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却。
     *
     * @return taxTypeNameのプロパティ名
     */
    public static PropertyName<String> taxTypeName() {
        return new PropertyName<String>("taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却。
     *
     * @return transCurrencyNameのプロパティ名
     */
    public static PropertyName<String> transCurrencyName() {
        return new PropertyName<String>("transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却。
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public static PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>("companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却。
     *
     * @return payTypeNameのプロパティ名
     */
    public static PropertyName<String> payTypeName() {
        return new PropertyName<String>("payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却。
     *
     * @return paymentFlagのプロパティ名
     */
    public static PropertyName<String> paymentFlag() {
        return new PropertyName<String>("paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却。
     *
     * @return payMethodNameのプロパティ名
     */
    public static PropertyName<String> payMethodName() {
        return new PropertyName<String>("payMethodName");
    }

    /**
     * departmentNameのプロパティ名を返却。
     *
     * @return departmentNameのプロパティ名
     */
    public static PropertyName<String> departmentName() {

```



```

    return new PropertyName<String>("departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public static PropertyName<String> projectName() {
    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.

```

```

*
* @return extensionIのプロパティ名
*/
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
* entryUserCdのプロパティ名を返却.
*
* @return entryUserCdのプロパティ名
*/
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
* renewCntのプロパティ名を返却.
*
* @return renewCntのプロパティ名
*/
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
* renewTsのプロパティ名を返却.
*
* @return renewTsのプロパティ名
*/
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
* renewUserCdのプロパティ名を返却.
*
* @return renewUserCdのプロパティ名
*/
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
* @author S2JDBC-Gen
*/
public static class _Ka30tRouteDetailNames extends PropertyName<Ka30tRouteDetail> {

    /**
    * インスタンスを構築.
    */
    public _Ka30tRouteDetailNames() {
    }

    /**
    * インスタンスを構築.
    *
    * @param name 名前
    */
    public _Ka30tRouteDetailNames(final String name) {
        super(name);
    }

    /**
    * インスタンスを構築.
    *
    * @param parent 親
    * @param name 名前
    */

```

```

*/
public _Ka30tRouteDetailNames(final PropertyName<?> parent, final String name) {
    super(parent, name);
}

/**
 * systemMatterIdのプロパティ名を返却。
 *
 * @return systemMatterIdのプロパティ名
 */
public PropertyName<String> systemMatterId() {
    return new PropertyName<String>(this, "systemMatterId");
}

/**
 * userDataIdのプロパティ名を返却。
 *
 * @return userDataIdのプロパティ名
 */
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
 * gadgetClassのプロパティ名を返却。
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却。
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却。
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * localeIdのプロパティ名を返却。
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * archiveMonthのプロパティ名を返却。
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>(this, "archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却。
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却。
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

```

```

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * routeDateのプロパティ名を返却.
 *
 * @return routeDateのプロパティ名
 */
public PropertyName<Timestamp> routeDate() {
    return new PropertyName<Timestamp>(this, "routeDate");
}

/**
 * startingSpotのプロパティ名を返却.
 *
 * @return startingSpotのプロパティ名
 */
public PropertyName<String> startingSpot() {
    return new PropertyName<String>(this, "startingSpot");
}

/**
 * arrivalSpotのプロパティ名を返却.
 *
 * @return arrivalSpotのプロパティ名
 */
public PropertyName<String> arrivalSpot() {
    return new PropertyName<String>(this, "arrivalSpot");
}

/**
 * supplementのプロパティ名を返却.
 *
 * @return supplementのプロパティ名
 */
public PropertyName<String> supplement() {
    return new PropertyName<String>(this, "supplement");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */

```

```

public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 *

```

```

    * payMethodCdのプロパティ名を返却.
    *
    * @return payMethodCdのプロパティ名
    */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>((this, "payMethodCd"));
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>((this, "payeeCd"));
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>((this, "payeeName"));
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public PropertyName<String> departmentSetCd() {
        return new PropertyName<String>((this, "departmentSetCd"));
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public PropertyName<String> departmentCd() {
        return new PropertyName<String>((this, "departmentCd"));
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>((this, "projectCd"));
    }

    /**
     * roundTripFlagのプロパティ名を返却.
     *
     * @return roundTripFlagのプロパティ名
     */
    public PropertyName<String> roundTripFlag() {
        return new PropertyName<String>((this, "roundTripFlag"));
    }

    /**
     * routeLinkFlagのプロパティ名を返却.
     *
     * @return routeLinkFlagのプロパティ名
     */
    public PropertyName<String> routeLinkFlag() {
        return new PropertyName<String>((this, "routeLinkFlag"));
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public PropertyName<String> routeSearchId() {
        return new PropertyName<String>((this, "routeSearchId"));
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public PropertyName<String> expTypeName() {
        return new PropertyName<String>((this, "expTypeName"));
    }

```

```

    return new PropertyName<String>(this, "expItyName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public PropertyName<String> accountName() {
    return new PropertyName<String>(this, "accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public PropertyName<String> taxTypeName() {
    return new PropertyName<String>(this, "taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public PropertyName<String> transCurrencyName() {
    return new PropertyName<String>(this, "transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>(this, "companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *

```

```

    * @return extensionAのプロパティ名
    */
    public PropertyName<String> extensionA() {
        return new PropertyName<String>((this, "extensionA"));
    }

    /**
     * extensionBのプロパティ名を返却。
     *
     * @return extensionBのプロパティ名
     */
    public PropertyName<String> extensionB() {
        return new PropertyName<String>((this, "extensionB"));
    }

    /**
     * extensionCのプロパティ名を返却。
     *
     * @return extensionCのプロパティ名
     */
    public PropertyName<String> extensionC() {
        return new PropertyName<String>((this, "extensionC"));
    }

    /**
     * extensionDのプロパティ名を返却。
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>((this, "extensionD"));
    }

    /**
     * extensionEのプロパティ名を返却。
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>((this, "extensionE"));
    }

    /**
     * extensionFのプロパティ名を返却。
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>((this, "extensionF"));
    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>((this, "extensionG"));
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>((this, "extensionH"));
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>((this, "extensionI"));
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>((this, "extensionJ"));
    }
}

```



```

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tRouteDetailValNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tRouteDetailVal} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tRouteDetailValNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tRouteDetailValNames() {
        super();
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public static PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
}

```

```

*/
public static PropertyName<String> localeId() {
    return new PropertyName<String>("localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
 * commutationRouteのプロパティ名を返却.
 *
 * @return commutationRouteのプロパティ名
 */
public static PropertyName<String> commutationRoute() {
    return new PropertyName<String>("commutationRoute");
}

/**
 * searchConditionのプロパティ名を返却.
 *
 * @return searchConditionのプロパティ名
 */
public static PropertyName<String> searchCondition() {
    return new PropertyName<String>("searchCondition");
}

/**
 * routeInfoのプロパティ名を返却.
 *
 * @return routeInfoのプロパティ名
 */
public static PropertyName<String> routeInfo() {
    return new PropertyName<String>("routeInfo");
}

/**
 * assessDateのプロパティ名を返却.
 *
 * @return assessDateのプロパティ名
 */
public static PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>("assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却.
 *
 * @return tranceMeansFlagのプロパティ名
 */
public static PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>("tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却.
 *
 * @return assessFlagのプロパティ名
 */
public static PropertyName<String> assessFlag() {
    return new PropertyName<String>("assessFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

```

```

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */

```

```

public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _Ka30tRouteDetailValNames extends PropertyName<Ka30tRouteDetailVal> {

    /**
     * インスタンスを構築.
     */
    public _Ka30tRouteDetailValNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _Ka30tRouteDetailValNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _Ka30tRouteDetailValNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * routeSearchIdのプロパティ名を返却.
     *
     * @return routeSearchIdのプロパティ名
     */
    public PropertyName<String> routeSearchId() {
        return new PropertyName<String>("routeSearchId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonthのプロパティ名を返却.
     *
     * @return archiveMonthのプロパティ名
     */
    public PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }
}

```

```

}

/**
 * commutationRouteのプロパティ名を返却.
 *
 * @return commutationRouteのプロパティ名
 */
public PropertyName<String> commutationRoute() {
    return new PropertyName<String>(this, "commutationRoute");
}

/**
 * searchConditionのプロパティ名を返却.
 *
 * @return searchConditionのプロパティ名
 */
public PropertyName<String> searchCondition() {
    return new PropertyName<String>(this, "searchCondition");
}

/**
 * routeInfoのプロパティ名を返却.
 *
 * @return routeInfoのプロパティ名
 */
public PropertyName<String> routeInfo() {
    return new PropertyName<String>(this, "routeInfo");
}

/**
 * assessDateのプロパティ名を返却.
 *
 * @return assessDateのプロパティ名
 */
public PropertyName<Timestamp> assessDate() {
    return new PropertyName<Timestamp>(this, "assessDate");
}

/**
 * tranceMeansFlagのプロパティ名を返却.
 *
 * @return tranceMeansFlagのプロパティ名
 */
public PropertyName<String> tranceMeansFlag() {
    return new PropertyName<String>(this, "tranceMeansFlag");
}

/**
 * assessFlagのプロパティ名を返却.
 *
 * @return assessFlagのプロパティ名
 */
public PropertyName<String> assessFlag() {
    return new PropertyName<String>(this, "assessFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名

```

```

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

```

```

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetail;

/**
 * {@link K30tRouteDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tRouteDetailService extends GenerateService<K30tRouteDetail> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tRouteDetailVal;

/**
 * {@link K30tRouteDetailVal}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tRouteDetailValService extends GenerateService<K30tRouteDetailVal> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetail;

/**
 * {@link Ka30tRouteDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tRouteDetailService extends GenerateService<Ka30tRouteDetail> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailValService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tRouteDetailVal;

/**
 * {@link Ka30tRouteDetailVal} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tRouteDetailValService extends GenerateService<Ka30tRouteDetailVal> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30tRouteDetail} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tRouteDetailCondition extends AbstractEntityCondition<K30tRouteDetailCondition> {

    /**
     * インスタンスを構築
     */
    public K30tRouteDetailCondition() {
    }

    /**
     * インスタンスを構築
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tRouteDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterId の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition> systemMatterId =
        new NotNullableViewCondition<K30tRouteDetailCondition>("systemMatterId", this);

    /** userDataId の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition> userDataId =
        new NotNullableViewCondition<K30tRouteDetailCondition>("userDataId", this);

    /** gadgetClass の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition> gadgetClass =
        new NotNullableViewCondition<K30tRouteDetailCondition>("gadgetClass", this);

    /** gadgetInstanceId の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition, Long> gadgetInstanceId =
        new NotNullableViewCondition<K30tRouteDetailCondition, Long>("gadgetInstanceId", this);

    /** tupleId の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition, Long> tupleId =
        new NotNullableViewCondition<K30tRouteDetailCondition, Long>("tupleId", this);

    /** gadgetId の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition> gadgetId =
        new NotNullableViewCondition<K30tRouteDetailCondition>("gadgetId", this);

    /** gadgetVariation の条件. */
    public NotNullableViewCondition<K30tRouteDetailCondition> gadgetVariation =
        new NotNullableViewCondition<K30tRouteDetailCondition>("gadgetVariation", this);
}

```



```

new NotNullableStringCondition<K30tRouteDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullableCondition<K30tRouteDetailCondition, Long> rowNumber =
    new NotNullableCondition<K30tRouteDetailCondition, Long>("rowNumber", this);

/** routeDateの条件. */
public NullableCondition<K30tRouteDetailCondition, Timestamp> routeDate =
    new NullableCondition<K30tRouteDetailCondition, Timestamp>("routeDate", this);

/** startingSpotの条件. */
public NullableStringCondition<K30tRouteDetailCondition> startingSpot =
    new NullableStringCondition<K30tRouteDetailCondition>("startingSpot", this);

/** arrivalSpotの条件. */
public NullableStringCondition<K30tRouteDetailCondition> arrivalSpot =
    new NullableStringCondition<K30tRouteDetailCondition>("arrivalSpot", this);

/** supplementの条件. */
public NullableStringCondition<K30tRouteDetailCondition> supplement =
    new NullableStringCondition<K30tRouteDetailCondition>("supplement", this);

/** expTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> expTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> accountCd =
    new NullableStringCondition<K30tRouteDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> taxTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tRouteDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("transAmount", this);

/** companyRateの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyRate", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tRouteDetailCondition>("companyCurrencyCd", this);

/** companyAmountの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<K30tRouteDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<K30tRouteDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payTypeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payMethodCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<K30tRouteDetailCondition> payeeCd =
    new NullableStringCondition<K30tRouteDetailCondition>("payeeCd", this);

/** ...の条件. */

```

```

    /** payeeNameの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> payeeName =
        new NullableStringCondition<K30tRouteDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> departmentSetCd =
        new NullableStringCondition<K30tRouteDetailCondition>("departmentSetCd", this);

    /** departmentCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> departmentCd =
        new NullableStringCondition<K30tRouteDetailCondition>("departmentCd", this);

    /** projectCdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> projectCd =
        new NullableStringCondition<K30tRouteDetailCondition>("projectCd", this);

    /** roundTripFlagの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> roundTripFlag =
        new NotNullableStringCondition<K30tRouteDetailCondition>("roundTripFlag", this);

    /** routeLinkFlagの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> routeLinkFlag =
        new NotNullableStringCondition<K30tRouteDetailCondition>("routeLinkFlag", this);

    /** routeSearchIdの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> routeSearchId =
        new NullableStringCondition<K30tRouteDetailCondition>("routeSearchId", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionA =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionB =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionC =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionD =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionE =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionF =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionG =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionH =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionI =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tRouteDetailCondition> extensionJ =
        new NullableStringCondition<K30tRouteDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tRouteDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tRouteDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Long> renewCnt =
        new NotNullableCondition<K30tRouteDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<K30tRouteDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<K30tRouteDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<K30tRouteDetailCondition> renewUserCd =
        new NotNullableStringCondition<K30tRouteDetailCondition>("renewUserCd", this);

```

```
new NotNullableStringCondition<K30tRouteDetailValCondition>("routeSearchId", this);
}
```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValCondition.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30tRouteDetailVal} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tRouteDetailValCondition extends
    AbstractEntityCondition<K30tRouteDetailValCondition> {

    /**
     * インスタンスを構築.
     */
    public K30tRouteDetailValCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tRouteDetailValCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** routeSearchId の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> routeSearchId =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("routeSearchId", this);

    /** commutationRoute の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> commutationRoute =
        new NullableStringCondition<K30tRouteDetailValCondition>("commutationRoute", this);

    /** searchCondition の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> searchCondition =
        new NullableStringCondition<K30tRouteDetailValCondition>("searchCondition", this);

    /** routeInfo の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> routeInfo =
        new NullableStringCondition<K30tRouteDetailValCondition>("routeInfo", this);

    /** assessDate の条件. */
    public NullableCondition<K30tRouteDetailValCondition, Timestamp> assessDate =
        new NullableCondition<K30tRouteDetailValCondition, Timestamp>("assessDate", this);

    /** tranceMeansFlag の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> tranceMeansFlag =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("tranceMeansFlag", this);

    /** assessFlag の条件. */
    public NotNullableStringCondition<K30tRouteDetailValCondition> assessFlag =
        new NotNullableStringCondition<K30tRouteDetailValCondition>("assessFlag", this);

    /** extensionA の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionA =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionA", this);

    /** extensionB の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionB =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionB", this);

    /** extensionC の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionC =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionC", this);

    /** extensionD の条件. */
    public NullableStringCondition<K30tRouteDetailValCondition> extensionD =
        new NullableStringCondition<K30tRouteDetailValCondition>("extensionD", this);
}
```

```

/** extensionEの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionE =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionF =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionG =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionH =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionI =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30tRouteDetailValCondition> extensionJ =
    new NullableStringCondition<K30tRouteDetailValCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Timestamp> entryTs =
    new NotNullCondition<K30tRouteDetailValCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullStringCondition<K30tRouteDetailValCondition> entryUserCd =
    new NotNullStringCondition<K30tRouteDetailValCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Long> renewCnt =
    new NotNullCondition<K30tRouteDetailValCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30tRouteDetailValCondition, Timestamp> renewTs =
    new NotNullCondition<K30tRouteDetailValCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30tRouteDetailValCondition> renewUserCd =
    new NotNullStringCondition<K30tRouteDetailValCondition>("renewUserCd", this);
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullCondition;
import org.seasar.extension.jdbc.where.condition.NotNullStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tRouteDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl"})
public final class Ka30tRouteDetailCondition extends
    AbstractEntityCondition<Ka30tRouteDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tRouteDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tRouteDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemModelの条件. */

```

```

/ ** systemMatterIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> systemMatterId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("systemMatterId", this);

/ ** userDataIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> userDataId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("userDataId", this);

/ ** gadgetClassの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetClass =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetClass", this);

/ ** gadgetInstanceの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> gadgetInstance =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("gadgetInstance", this);

/ ** tupleIdの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> tupleId =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("tupleId", this);

/ ** localeIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> localeId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("localeId", this);

/ ** archiveMonthの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> archiveMonth =
    new NullableStringCondition<Ka30tRouteDetailCondition>("archiveMonth", this);

/ ** gadgetIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetId", this);

/ ** gadgetVariationの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> gadgetVariation =
    new NullableStringCondition<Ka30tRouteDetailCondition>("gadgetVariation", this);

/ ** rowNumberの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Long> rowNumber =
    new NullableCondition<Ka30tRouteDetailCondition, Long>("rowNumber", this);

/ ** routeDateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, Timestamp> routeDate =
    new NullableCondition<Ka30tRouteDetailCondition, Timestamp>("routeDate", this);

/ ** startingSpotの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> startingSpot =
    new NullableStringCondition<Ka30tRouteDetailCondition>("startingSpot", this);

/ ** arrivalSpotの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> arrivalSpot =
    new NullableStringCondition<Ka30tRouteDetailCondition>("arrivalSpot", this);

/ ** supplementの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> supplement =
    new NullableStringCondition<Ka30tRouteDetailCondition>("supplement", this);

/ ** expTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("expTypeCd", this);

/ ** accountCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> accountCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("accountCd", this);

/ ** taxTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("taxTypeCd", this);

/ ** taxRateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("taxRate", this);

/ ** quantityの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("quantity", this);

/ ** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("transCurrencyCd", this);

/ ** transAmountの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("transAmount", this);

/ ** companyRateの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyRate", this);

```

```

new NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyRate", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("companyCurrencyCd", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tRouteDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tRouteDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payeeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> projectCd =
    new NullableStringCondition<Ka30tRouteDetailCondition>("projectCd", this);

/** roundTripFlagの条件. */
public NotNullableStringCondition<Ka30tRouteDetailCondition> roundTripFlag =
    new NotNullableStringCondition<Ka30tRouteDetailCondition>("roundTripFlag", this);

/** routeLinkFlagの条件. */
public NotNullableStringCondition<Ka30tRouteDetailCondition> routeLinkFlag =
    new NotNullableStringCondition<Ka30tRouteDetailCondition>("routeLinkFlag", this);

/** routeSearchIdの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> routeSearchId =
    new NullableStringCondition<Ka30tRouteDetailCondition>("routeSearchId", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> accountName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tRouteDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tRouteDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */

```

```

    public NullableStringCondition<Ka30tRouteDetailCondition> paymentFlag =
        new NullableStringCondition<Ka30tRouteDetailCondition>("paymentFlag", this);

    /** payMethodNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> payMethodName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("payMethodName", this);

    /** departmentNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> departmentName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("departmentName", this);

    /** projectNameの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> projectName =
        new NullableStringCondition<Ka30tRouteDetailCondition>("projectName", this);

    /** extensionAの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionA =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionB =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionC =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionD =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionE =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionF =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionG =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionH =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionI =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<Ka30tRouteDetailCondition> extensionJ =
        new NullableStringCondition<Ka30tRouteDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<Ka30tRouteDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<Ka30tRouteDetailCondition> entryUserCd =
        new NotNullableStringCondition<Ka30tRouteDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Long> renewCnt =
        new NotNullableCondition<Ka30tRouteDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<Ka30tRouteDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<Ka30tRouteDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<Ka30tRouteDetailCondition> renewUserCd =
        new NotNullableStringCondition<Ka30tRouteDetailCondition>("renewUserCd", this);
}

```

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailValCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;

```



```

import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tRouteDetailVal}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tRouteDetailValCondition extends
    AbstractEntityCondition<Ka30tRouteDetailValCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tRouteDetailValCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tRouteDetailValCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** routeSearchIdの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> routeSearchId =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("routeSearchId", this);

    /** localeIdの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> localeId =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("localeId", this);

    /** archiveMonthの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> archiveMonth =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("archiveMonth", this);

    /** commutationRouteの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> commutationRoute =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("commutationRoute", this);

    /** searchConditionの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> searchCondition =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("searchCondition", this);

    /** routeInfoの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> routeInfo =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("routeInfo", this);

    /** assessDateの条件. */
    public NullableCondition<Ka30tRouteDetailValCondition, Timestamp> assessDate =
        new NullableCondition<Ka30tRouteDetailValCondition, Timestamp>("assessDate", this);

    /** tranceMeansFlagの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> tranceMeansFlag =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("tranceMeansFlag", this);

    /** assessFlagの条件. */
    public NotNullStringCondition<Ka30tRouteDetailValCondition> assessFlag =
        new NotNullStringCondition<Ka30tRouteDetailValCondition>("assessFlag", this);

    /** extensionAの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionA =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionB =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionC =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionD =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<Ka30tRouteDetailValCondition> extensionE =
        new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionE", this);

```



```

/** extensionFの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionF =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionG =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionH =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionI =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tRouteDetailValCondition> extensionJ =
    new NullableStringCondition<Ka30tRouteDetailValCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp> entryTs =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tRouteDetailValCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tRouteDetailValCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Long> renewCnt =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp> renewTs =
    new NotNullableCondition<Ka30tRouteDetailValCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<Ka30tRouteDetailValCondition> renewUserCd =
    new NotNullableStringCondition<Ka30tRouteDetailValCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.route_date,
    t.starting_spot,
    t.arrival_spot,
    t.supplement,
    t.exp_type_cd,
    t.account_cd,
    t.tax_type_cd,
    t.tax_rate,
    t.quantity,
    t.trans_currency_cd,
    t.trans_amount,
    t.company_rate,
    t.company_currency_cd,
    t.company_amount,
    t.company_amount_exc_tax,
    t.company_amount_tax,
    t.company_amount_sum,
    t.pay_type_cd,
    t.pay_method_cd,
    t.payee_cd,
    t.payee_name,
    t.department_set_cd,
    t.department_cd,
    t.project_cd,
    t.round_trip_flag,
    t.route_link_flag,
    t.route_search_id,
    t.outside_id

```

```

t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
m7.exp_type_name,
m8.account_name,
m9.tax_type_name,
m4.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag
from
k30t_route_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
c1.search_criteria_date >= m2.start_date
and
c1.search_criteria_date < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
c1.search_criteria_date >= m3.start_date
and
c1.search_criteria_date < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id

```

```

/*localeId*/ja' = m4.locale_id
and
  c1.search_criteria_date >= m4.start_date
and
  c1.search_criteria_date < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeId*/ja' = m5.locale_id
and
  c1.search_criteria_date >= m5.start_date
and
  c1.search_criteria_date < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeId*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_exp_type m7
on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeId*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeId*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeId*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date
and
  c1.search_criteria_date < m9.end_date
and
  m9.delete_flag = '0'
left outer join
  k30t_route_detail_val val
on t.route_search_id = val.route_search_id
where
  t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*FND*/

```

```

/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

- **Path**

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailValService/selectView.sql

```

select
  t.route_search_id,
  t.commutation_route,
  t.search_condition,
  t.route_info,
  t.assess_date,
  t.trance_means_flag, assess_flag,
  t.extension_a,
  t.extension_b,
  t.extension_c,
  t.extension_d,
  t.extension_e,
  t.extension_f,
  t.extension_g,
  t.extension_h,
  t.extension_i,
  t.extension_j,
  t.entry_ts,
  t.entry_user_cd,
  t.renew_cnt,
  t.renew_ts,
  t.renew_user_cd
from
  k30t_route_detail d
  inner join
    k30t_route_detail_val t
  on
    d.route_search_id = t.route_search_id
where
  d.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  d.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
  d.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
  d.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  d.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.route_search_id
/*END*/

```

- **Path**

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailValService/selectView.sql

```

select
  t.route_search_id,
  t.locale_id,
  t.archive_month,
  t.commutation_route,
  t.search_condition,
  t.route_info,
  t.assess_date,
  t.trance_means_flag, assess_flag,
  t.extension_a,
  t.extension_b,
  t.extension_c,
  t.extension_d,
  t.extension_e,
  t.extension_f,
  t.extension_g,
  t.extension_h,
  t.extension_i,
  t.extension_j,
  t.entry_ts,
  t.entry_user_cd,
  t.renew_cnt,
  t.renew_ts,
  t.renew_user_cd
from
  ka30t_route_detail d
  inner join
    ka30t_route_detail_val t
  on
    d.route_search_id = t.route_search_id
  and
    d.locale_id = t.locale_id
where
  d.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
  and
    d.user_data_id = /*userDataId*/
  /*END*/
  /*IF orderBy != null*/
  order by /*$orderBy*/t.route_search_id
  /*END*/

```

コピー

- **Path**
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tRouteDetailService/selectViewForCopy.sql

```

select
  t.system_matter_id,
  t.user_data_id,
  t.gadget_class,
  t.gadget_instance,
  t.tuple_id,
  t.gadget_id,
  t.gadget_variation,
  t.row_number,
  t.route_date,
  t.starting_spot,
  t.arrival_spot,
  t.supplement,
  t.exp_type_cd,
  t.account_cd,
  t.tax_type_cd,
  t.tax_rate,
  t.quantity,
  t.trans_currency_cd,
  t.trans_amount,
  t.company_rate,
  t.company_currency_cd,
  t.company_amount,
  t.company_amount_exc_tax,
  t.company_amount_tax,
  t.company_amount_sum,
  t.pay_type_cd,
  t.pay_method_cd,
  t.payee_cd,
  t.payee_name,
  t.department_set_cd,
  t.department_cd,
  t.project_cd,
  t.round_trip_flag,
  t.route_link_flag,
  t.route_search_id,
  t.extension_a,

```

```

t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
ma.exp_type_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> '' then m8.account_name else mb.account_name end as account_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> '' then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
m4.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag
from
k30t_route_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
/*searchCriteriaDate*/" >= m2.start_date
and
/*searchCriteriaDate*/" < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
/*searchCriteriaDate*/" >= m3.start_date
and
/*searchCriteriaDate*/" < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id
and

```

```

and
/*searchCriteriaDate*/" >= m4.start_date
and
/*searchCriteriaDate*/" < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeId*/ja' = m5.locale_id
and
/*searchCriteriaDate*/" >= m5.start_date
and
/*searchCriteriaDate*/" < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6
on
c1.search_criteria_company = m6.company_cd
and
t.company_currency_cd = m6.currency_cd
and
/*localeId*/ja' = m6.locale_id
and
m6.delete_flag = '0'
left outer join
k30m_account m8
on
c1.search_criteria_company = m8.company_cd
and
t.account_cd = m8.account_cd
and
/*localeId*/ja' = m8.locale_id
and
/*searchCriteriaDate*/" >= m8.start_date
and
/*searchCriteriaDate*/" < m8.end_date
and
m8.delete_flag = '0'
left outer join
k30m_tax_type m9
on
c1.search_criteria_company = m9.company_cd
and
t.tax_type_cd = m9.tax_type_cd
and
/*localeId*/ja' = m9.locale_id
and
/*searchCriteriaDate*/" >= m9.start_date
and
/*searchCriteriaDate*/" < m9.end_date
and
m9.delete_flag = '0'

left outer join
k30m_exp_type ma
on
c1.search_criteria_company = ma.company_cd
and
t.exp_type_cd = ma.exp_type_cd
and
/*localeId*/ja' = ma.locale_id
and
/*searchCriteriaDate*/" >= ma.start_date
and
/*searchCriteriaDate*/" < ma.end_date
and
ma.delete_flag = '0'
left outer join
k30m_account mb
on
ma.company_cd = mb.company_cd
and
ma.account_cd = mb.account_cd
and
/*localeId*/ja' = mb.locale_id
and
/*searchCriteriaDate*/" >= mb.start_date
and
/*searchCriteriaDate*/" < mb.end_date
and
m8.delete_flag = '0'
left outer join

```

```

left outer join
    k30m_tax_type mc
on
    ma.company_cd = mc.company_cd
and
    ma.tax_type_cd = mc.tax_type_cd
and
    /*localeId*/ja' = mc.locale_id
and
    /*searchCriteriaDate*/" >= mc.start_date
and
    /*searchCriteriaDate*/" < mc.end_date
and
    mc.delete_flag = '0'
left outer join
    k30t_route_detail_val val
on t.route_search_id = val.route_search_id
where
    t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
    t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
    t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
    t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
    t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
    c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/routedetail/impl/RouteDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
    t.system_matter_id as system_matter_id,
    t.user_data_id as user_data_id,
    t.gadget_class as gadget_class,
    t.gadget_instance as gadget_instance,
    t.tuple_id as tuple_id,
    t.gadget_id as gadget_id,
    t.gadget_variation as gadget_variation,
    i1.contents_id as contents_id,
    i1.contents_version_id as contents_version_id,
    null as journal_data_type_cd,
    t.row_number as row_number,
    t.exp_type_cd as exp_type_cd,
    t.account_cd as account_cd,
    t.tax_type_cd as tax_type_cd,
    mt.tax_rate as tax_rate,
    t.quantity as quantity,
    t.trans_currency_cd as trans_currency_cd,
    t.trans_amount as trans_amount,
    t.company_currency_cd as company_currency_cd,
    t.company_rate as company_rate,
    t.company_amount as company_amount,
    t.company_amount_exc_tax as company_amount_exc_tax,
    t.company_amount_tax as company_amount_tax,
    t.company_amount_sum as company_amount_sum,
    t.payee_cd as customer_cd,
    t.payee_name as customer_name,
    t.department_set_cd as department_set_cd,
    t.department_cd as department_cd,
    t.project_cd as project_cd,
    t.pay_method_cd as pay_method_cd,
    t.pay_type_cd as pay_type_cd,
    null as adv_matter_number,
    null as adv_matter_name,
    t.route_date as detail_date1,
    null as detail_date2,
    null as detail_summary1,
    null as detail_summary2

```



```

null as detail_summary2,
null as detail_summary3,
null as detail_summary4,
null as detail_summary5,
t.supplement as detail_note,
t.extension_a as detail_extension_a,
t.extension_b as detail_extension_b,
t.extension_c as detail_extension_c,
t.extension_d as detail_extension_d,
t.extension_e as detail_extension_e,
t.extension_f as detail_extension_f,
t.extension_g as detail_extension_g,
t.extension_h as detail_extension_h,
t.extension_i as detail_extension_i,
t.extension_j as detail_extension_j,
'0' as journal_status,
null as wish_date,
c1.search_criteria_company,
c1.search_criteria_date
from
k30t_route_detail t
inner join
k20c_matter_info i1
on
t.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join
k30m_tax_type mt
on
c1.search_criteria_company = mt.company_cd
and
t.tax_type_cd = mt.tax_type_cd
and
c1.search_criteria_date >= mt.start_date
and
c1.search_criteria_date < mt.end_date
and
i1.locale_id_apply = mt.locale_id
and
mt.delete_flag = '0'

where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.row_number
/*END*/

```

アーカイブ

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tRouteDetailService/selectViewForArchive.sql

```

select
t.system_matter_id,
t.user_data_id,
t.gadget_class,
t.gadget_instance,
t.tuple_id,
t.locale_id,
t.archive_month,

```

```

t.gadget_id,
t.gadget_variation,
t.row_number,
t.route_date,
t.starting_spot,
t.arrival_spot,
t.supplement,
t.exp_type_cd,
t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_rate,
t.company_currency_cd,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.round_trip_flag,
t.route_link_flag,
t.route_search_id,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
t.trans_currency_name,
t.department_name,
t.project_name,
t.pay_type_name,
t.pay_method_name,
t.company_currency_name,
t.exp_type_name,
t.account_name,
t.tax_type_name,
t.payment_flag,
val.commutation_route,
val.search_condition,
val.route_info,
val.assess_date,
val.trance_means_flag,
val.assess_flag

from
ka30t_route_detail t
inner join
ka20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
and
t.locale_id = c1.locale_id
left outer join
ka30t_route_detail_val val
on
t.route_search_id = val.route_search_id
and
t.locale_id = val.locale_id
where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/

```

```

t.gadget_id = /gadgetid/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF localeId != null*/
and
t.locale_id = /*localeId*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/t.system_matter_id
/*END*/

```

JSP

PC

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/routeDetail/routeDetailV02.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wk" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- routeDetail v02 --%>
<c:set var="gadgetClass">routeDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/routeDetailV02</c:set>
<c:set var="gadgetBlockId">routeDetailTable</c:set>
<c:set var="gadgetBlockHeaderId">routeDetailHeader</c:set>
<c:set var="tmtableId">routeDetailTable</c:set>
<c:set var="tmtableControllerBlockId">tmtableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="applyAuthUserCode">${param.applyAuthUserCode}</c:set>
<c:set var="useRouteSearch">${k:isUseRouteSearch(param.searchCriteriaCompany)}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks[routeDetailTable]}" />

<!-- tmtable --%>
<c:set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<!-- Message --%>
<c:set var="msgRateReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0008) %></c:set>

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</c:set>
<c:set var="surfaceRouteDate">${gadget.surfaces["routeDate"]}</c:set>
<c:set var="surfaceTravelRoute">${gadget.surfaces["travelRoute"]}</c:set>
<c:set var="surfaceStartingSpot">${gadget.surfaces["startingSpot"]}</c:set>
<c:set var="surfaceHyphen">${gadget.surfaces["hyphen"]}</c:set>
<c:set var="surfaceArrivalSpot">${gadget.surfaces["arrivalSpot"]}</c:set>
<c:set var="surfaceRoundTrip">${gadget.surfaces["roundTrip"]}</c:set>
<c:set var="surfaceSupplement">${gadget.surfaces["supplement"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>

```

```

<:set var="surfaceTransAmount">${gadget:surfaces["amount_transAmount"]}</c:set>
<c:set var="surfaceTransCurrency">${gadget:surfaces["amount_transCurrency"]}</c:set>
<c:set var="surfaceCompanyRate">${gadget:surfaces["amount_transRate"]}</c:set>
<c:set var="surfaceCompanyAmount">${gadget:surfaces["amount_companyAmount"]}</c:set>
<c:set var="surfaceRouteSearchLink">${gadget:surfaces["routeSearchLink"]}</c:set>
<c:set var="surfaceRouteSearch">${gadget:surfaces["routeSearch"]}</c:set>
<c:set var="surfaceRouteSearchDetail">${gadget:surfaces["routeSearchDetail"]}</c:set>
<c:set var="surfaceRouteSearchRemove">${gadget:surfaces["routeSearchRemove"]}</c:set>
<c:set var="surfaceCompanyAmountSum">${gadget:surfaces["companyAmountSum"]}</c:set>

<%-- Surface(Property) --%>
<c:set var="surfaceRouteSearch"><%= BaseProp.S.EkispertWeb.ROUTE_SEARCH %></c:set>
<c:set var="surfaceShowDetail"><%= BaseProp.S.EkispertWeb.SHOW_DETAIL %></c:set>
<c:set var="surfaceReleaseLink"><%= BaseProp.S.EkispertWeb.RELEASE_LINK %></c:set>
<c:set var="surfaceAltFast"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_FAST) %></c:set>
<c:set var="surfaceAltEasy"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_EASE) %></c:set>
<c:set var="surfaceAltCheap"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP) %></c:set>
<c:set var="surfaceAltCheapTeiki1"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI1) %></c:set>
<c:set var="surfaceAltCheapTeiki3"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI3) %></c:set>
<c:set var="surfaceAltCheapTeiki6"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_CHEAP_TEIKI6) %></c:set>
<c:set var="surfaceAltEco"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_ECO) %></c:set>
<c:set var="surfaceAltCommuter"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_TEIKI) %></c:set>
<c:set var="surfaceAltExpress"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_EXPRESS) %></c:set>
<c:set var="surfaceAltBus"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_BUS) %></c:set>
<c:set var="surfaceAltAir"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_AIR) %></c:set>
<c:set var="surfaceAltShip"><%= PropertyJspUtil.getSurface(BaseProp.S.EkispertWeb.ALT_SHIP) %></c:set>

<%-- Mode --%>
<c:set var="modeRouteDate">${detail.fields["routeDate"].inputType}</c:set>
<c:set var="modeStartingSpot">${detail.fields["startingSpot"].inputType}</c:set>
<c:set var="modeArrivalSpot">${detail.fields["arrivalSpot"].inputType}</c:set>
<c:set var="modeSupplement">${detail.fields["supplement"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modeTransCurrency">${detail.fields["transCurrencyCd"].inputType}</c:set>
<c:set var="modeCompanyRate">${detail.fields["companyRate"].inputType}</c:set>
<c:set var="modeCompanyAmount">${detail.fields["companyAmount"].inputType}</c:set>
<c:set var="modeRouteLinkFlag">${detail.fields["routeLinkFlag"].inputType}</c:set>
<c:set var="modeRoundTripFlag">${detail.fields["roundTripFlag"].inputType}</c:set>
<c:set var="modeCompanyAmountSum">${modeTransAmount == modeRoundTripFlag ? modeTransAmount : "0"}</c:set>

<%-- RequiredMark --%>
<c:set var="reqRouteDate" >${detail.fields["routeDate"].requiredFlag}</c:set>
<c:set var="reqStartingSpot" >${detail.fields["startingSpot"].requiredFlag}</c:set>
<c:set var="reqArrivalSpot" >${detail.fields["arrivalSpot"].requiredFlag}</c:set>
<c:set var="reqSupplement" >${detail.fields["supplement"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqTransCurrency" >${detail.fields["transCurrencyCd"].requiredFlag}</c:set>
<c:set var="reqCompanyRate" >${detail.fields["companyRate"].requiredFlag}</c:set>
<c:set var="reqCompanyAmount" >${detail.fields["companyAmount"].requiredFlag}</c:set>
<c:set var="reqRoundTripFlag" >${detail.fields["roundTripFlag"].requiredFlag}</c:set>
<c:set var="reqRouteLinkFlag" >${detail.fields["routeLinkFlag"].requiredFlag}</c:set>
<c:set var="reqCompanyAmountSum" >${detail.fields["companyAmountSum"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validRouteDate_def">{k_date:[""]}</c:set>
<c:set var="validRouteDate_apply">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>
<c:set var="validRouteDate_tempsave"></c:set>
<c:set var="validRouteDate_reapply">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>
<c:set var="validRouteDate_process">${"true" == reqRouteDate ? "k_required:[""] : ""}</c:set>

<c:set var="validStartingSpot_def">{k_maxLength:["250"]}</c:set>
<c:set var="validStartingSpot_apply">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validStartingSpot_tempsave"></c:set>
<c:set var="validStartingSpot_reapply">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validStartingSpot_process">${"true" == reqStartingSpot ? "k_required:[""] : ""}</c:set>

<c:set var="validArrivalSpot_def">{k_maxLength:["250"]}</c:set>
<c:set var="validArrivalSpot_apply">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validArrivalSpot_tempsave"></c:set>
<c:set var="validArrivalSpot_reapply">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>
<c:set var="validArrivalSpot_process">${"true" == reqArrivalSpot ? "k_required:[""] : ""}</c:set>

<c:set var="validSupplement_def">{k_maxLength:["1000"]}</c:set>

```

```

<c:set var="validSupplement_apply">{${"true" == reqSupplement ? "k_required:[""] : ""}}</c:set>
<c:set var="validSupplement_tempsave"></c:set>
<c:set var="validSupplement_reapply">{${"true" == reqSupplement ? "k_required:[""] : ""}}</c:set>
<c:set var="validSupplement_process">{${"true" == reqSupplement ? "k_required:[""] : ""}}</c:set>

<c:set var="validRoundTripFlag_def">{k_flag:[""]}</c:set>
<c:set var="validRoundTripFlag_apply">{${"true" == reqRoundTripFlag ? "k_required:[""] : ""}}</c:set>
<c:set var="validRoundTripFlag_tempsave"></c:set>
<c:set var="validRoundTripFlag_reapply">{${"true" == reqRoundTripFlag ? "k_required:[""] : ""}}</c:set>
<c:set var="validRoundTripFlag_process">{${"true" == reqRoundTripFlag ? "k_required:[""] : ""}}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:[payTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">{${"true" == reqPayTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPayTypeCd_reapply">{${"true" == reqPayTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayTypeCd_process">{${"true" == reqPayTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validPayeeName_def">{k_maxLength:[250]}</c:set>
<c:set var="validPayeeName_apply">{${"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">{${"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>
<c:set var="validPayeeName_process">{${"true" == reqPayeeName ? "k_required:[""] : ""}}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:[searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:[projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:[departmentCd', 'projectCd']}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:[expTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validExpTypeCd_apply">{${"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">{${"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validExpTypeCd_process">{${"true" == reqExpTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validAccountCd_def">{k_accountExists:[accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">{${"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">{${"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validAccountCd_process">{${"true" == reqAccountCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:[taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">{${"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{${"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>
<c:set var="validTaxType_process">{${"true" == reqTaxTypeCd ? "k_required:[""] : ""}}</c:set>

<c:set var="validCompanyAmountSum_def">{k_currencyAmnt:["', 'searchCriteriaCompany']}</c:set>
<c:set var="validCompanyAmountSum_apply">{${"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>
<c:set var="validCompanyAmountSum_tempsave"></c:set>
<c:set var="validCompanyAmountSum_reapply">{${"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>
<c:set var="validCompanyAmountSum_process">{${"true" == reqCompanyAmountSum ? "k_required:[""] : ""}}</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${surfaceChapterTitle}"
    gadgetId="${gadgetId}"
    hideGadgetTitle="true"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="${gadgetBlockHeaderId}">
<div class="imui-chapter-title">
    <h2 style="display:inline-block;width:50%"><k:surface surfaceKey="${surfaceChapterTitle}" /></h2>
    <a class="kaiden-ui-icon-refresh-gray16_nosize"
        name="rateReget"
        mode="${modeTransCurrency}"
        text="${surfaceRateReget}"
        style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem"
    />
</div>
</k:blockContainer>
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:tmttable id="${tmttableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumberTarget="rowNumber">
    <colgroup>
        <col style="width:40px;overflow:hidden">
        <col style="width:235px;overflow:hidden">
        <col style="width:185px;overflow:hidden">
        <col style="width:185px;overflow:hidden">
        <col style="width:185px;overflow:hidden">

```

```

</colgroup>
<thead>
<tr>
<td rowspan="4" style="vertical-align:middle;">
<k:tmtableOperationPanel
position="header"
allowAdd="{allowAddDetail}"
allowDel="{allowDelDetail}"
/>
</td>
<th>
<k:surface
name="routeDateSurface"
surfaceKey="{surfaceRouteDate}"
requiredMark="{reqRouteDate}"
/>
</th>
<th colspan="2">
<k:surface
name="travelRouteSurface"
surfaceKey="{surfaceTravelRoute}"
requiredMark="{reqStartingSpot}"
/>
</th>
<th>
<k:surface
name="supplementSurface"
surfaceKey="{surfaceSupplement}"
requiredMark="{reqSupplement}"
/>
</th>
</tr>
<tr>
<th colspan="2">
<k:surface
name="expTypeSurface"
surfaceKey="{surfaceExpType}"
requiredMark="{reqExpTypeCd}"
style="display:inline;"
/>
<k:surface
name="accountSurface"
surfaceKey="{surfaceAccount}"
style="display:inline;"
/>
<k:surface
name="taxTypeSurface"
surfaceKey="{surfaceTaxType}"
style="display:inline;"
/>
</th>
<th>
<k:surface
name="departmentSurface"
surfaceKey="{surfaceDepartment}"
requiredMark="{reqDepartmentCd}"
/>
</th>
<th>
<k:surface
name="projectSurface"
surfaceKey="{surfaceProject}"
requiredMark="{reqProjectCd}"
/>
</th>
</tr>
<tr>
<th>
<k:surface
name="payTypeSurface"
surfaceKey="{surfacePayType}"
requiredMark="{reqPayTypeCd}"
/>
</th>
<th>
<k:surface
name="payeeSurface"
surfaceKey="{surfacePayee}"
requiredMark="{reqPayeeName}"
/>
</th>
<th colspan="2">
<k:surface
name="transAmountSurface"
surfaceKey="{surfaceTransAmount}"
requiredMark="{reqTransAmount}"

```

```

    />
  </th>
</tr>
<tr>
  <th colspan="3">
    <k:surface
      name="routeSearchLinkSurface"
      surfaceKey="{surfaceRouteSearchLink}"
      requiredMark="{reqRouteSearchLink}"
    />
  </th>
  <th>
    <k:surface
      name="companyAmountSurface"
      surfaceKey="{surfaceCompanyAmountSum}"
      requiredMark="{reqCompanyAmountSum}"
    />
  </th>
</tr>
</thead>
<tbody>
  <tr>
    <td rowspan="4">
      <k:tmtableOperationPanel
        position="detail"
        allowAdd="{allowAddDetail}"
        allowDel="{allowDelDetail}"
        name="rowNumber"
      />
    </td>
    <td>
      <k:date
        name="routeDate"
        mode="{modeRouteDate}"
        surfaceKey="{surfaceRouteDate}"
        validationDefault="{validRouteDate_def}"
        validationApply="{validRouteDate_apply}"
        validationTempsave="{validRouteDate_tempsave}"
        validationReapply="{validRouteDate_reapply}"
        validationProcess="{validRouteDate_process}"
      />
    </td>
    <td colspan="2">
      <k:text
        name="startingSpot"
        mode="{modeStartingSpot}"
        imeMode="on"
        surfaceKey="{surfaceStartingSpot}"
        style="width:165px;"
        maxLength="250"
        validationDefault="{validStartingSpot_def}"
        validationApply="{validStartingSpot_apply}"
        validationTempsave="{validStartingSpot_tempsave}"
        validationReapply="{validStartingSpot_reapply}"
        validationProcess="{validStartingSpot_process}"
      />
      <k:surface
        name="hyphenSurface"
        surfaceKey="{surfaceHyphen}"
        style="margin:0 2px;"
      />
      <k:text
        name="arrivalSpot"
        mode="{modeArrivalSpot}"
        imeMode="on"
        surfaceKey="{surfaceArrivalSpot}"
        style="width:165px;"
        maxLength="250"
        validationDefault="{validArrivalSpot_def}"
        validationApply="{validArrivalSpot_apply}"
        validationTempsave="{validArrivalSpot_tempsave}"
        validationReapply="{validArrivalSpot_reapply}"
        validationProcess="{validArrivalSpot_process}"
      />
    </td>
    <td>
      <k:text
        name="supplement"
        mode="{modeSupplement}"
        imeMode="on"
        surfaceKey="{surfaceSupplement}"
        style="width:169px;"
        maxLength="1000"
        validationDefault="{validSupplement_def}"
        validationApply="{validSupplement_apply}"
        validationTempsave="{validSupplement_tempsave}"

```

```

        validationReapply="${validSupplement_reapply}"
        validationProcess="${validSupplement_process}"
    />
</td>
</tr>
<tr>
<td colspan="2">
<k:hidden
    name="expTypeCd"
    mode="${modeExpTypeCd}"
/>
<kmsw:expType
    name="expTypeName"
    mode="${modeExpTypeCd}"
    imeMode="on"
    surfaceKey="${surfaceExpType}"
    resultValue="expTypeName"
    resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
    primaryTarget="expTypeCd"
    searchCriteriaDate="searchCriteriaDate"
    searchCriteriaCompany="searchCriteriaCompany"
    searchCriteriaExpTypeCd="expTypeCd"
    style="width:150px;"
    validationDefault="${validExpTypeCd_def}"
    validationApply="${validExpTypeCd_apply}"
    validationTempsave="${validExpTypeCd_tempsave}"
    validationReapply="${validExpTypeCd_reapply}"
    validationProcess="${validExpTypeCd_process}"
/>
<k:hidden
    name="accountCd"
    mode="${modeAccountCd}"
    surfaceKey="${surfaceAccount}"
    validationDefault="${validAccountCd_def}"
    validationApply="${validAccountCd_apply}"
    validationTempsave="${validAccountCd_tempsave}"
    validationReapply="${validAccountCd_reapply}"
    validationProcess="${validAccountCd_process}"
/>
<k:text
    name="accountName"
    mode="1"
    class="kaiden-ui-mst-search-chain-item"
    style="width:115px;"
/>
<k:hidden
    name="taxTypeCd"
    mode="${modeTaxTypeCd}"
    surfaceKey="${surfaceTaxType}"
    validationDefault="${validTaxType_def}"
    validationApply="${validTaxType_apply}"
    validationTempsave="${validTaxType_tempsave}"
    validationReapply="${validTaxType_reapply}"
    validationProcess="${validTaxType_process}"
/>
<k:text
    name="taxTypeName"
    mode="1"
    class="kaiden-ui-mst-search-chain-item"
    style="width:115px;"
/>
</td>
<td>
<k:hidden
    name="departmentSetCd"
    mode="${modeDepartmentCd}"
/>
<k:hidden
    name="departmentCd"
    mode="${modeDepartmentCd}"
    surfaceKey="${surfaceDepartment}"
/>
<kmsb:department
    name="departmentName"
    mode="${modeDepartmentCd}"
    imeMode="on"
    surfaceKey="${surfaceDepartment}"
    resultValue="departmentName"
    resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd'}"
    primaryTarget="departmentCd,departmentSetCd"
    searchCriteriaDate="searchCriteriaDate"
    searchCriteriaCompany="searchCriteriaCompany"
    searchCriteriaDepartmentCd="departmentCd"
    searchCriteriaDepartmentSetCd="departmentSetCd"
    style="width:150px;"
    validationDefault="${validDepartmentCd_def}"

```



```

validationApply="${validDepartmentCd_apply}"
validationTempsave="${validDepartmentCd_tempsave}"
validationReapply="${validDepartmentCd_reapply}"
validationProcess="${validDepartmentCd_process}"
/>
</td>
<td>
<k:hidden
name="projectCd"
mode="${modeProjectCd}"
surfaceKey="${surfaceProject}"
/>
<kmsw:project
name="projectName"
mode="${modeProjectCd}"
imeMode="on"
surfaceKey="${surfaceProject}"
resultValue="projectName"
resultChain="{projectCd:'projectCd'}"
primaryTarget="projectCd"
searchCriteriaDate="searchCriteriaDate"
searchCriteriaCompany="searchCriteriaCompany"
searchCriteriaProjectCd="projectCd"
style="width:150px;"
validationDefault="${validProjectCd_def}"
validationApply="${validProjectCd_apply}"
validationTempsave="${validProjectCd_tempsave}"
validationReapply="${validProjectCd_reapply}"
validationProcess="${validProjectCd_process}"
/>
</td>
</tr>
<tr>
<td>
<klist_wkf:listDataPayType
blank="false"
name="payType"
companyCd="${param.searchCriteriaCompany}"
searchCriteriaDate="${param.searchCriteriaDate}"
/>
<k:select
name="payTypeCd"
mode="${modePayTypeCd}"
surfaceKey="${surfacePayType}"
viewModeName="payTypeName"
dataName="payType"
selectChain="{paymentFlag:'paymentFlag'}"
style="width:230px;"
validationDefault="${validPayTypeCd_def}"
validationApply="${validPayTypeCd_apply}"
validationTempsave="${validPayTypeCd_tempsave}"
validationReapply="${validPayTypeCd_reapply}"
validationProcess="${validPayTypeCd_process}"
/>
<k:hidden
name="paymentFlag"
mode="${modePayTypeCd}"
/>
</td>
<td>
<k:text
name="payeeName"
mode="${modePayeeName}"
imeMode="on"
surfaceKey="${surfacePayee}"
style="width:160px;"
maxlength="250"
validationDefault="${validPayeeName_def}"
validationApply="${validPayeeName_apply}"
validationTempsave="${validPayeeName_tempsave}"
validationReapply="${validPayeeName_reapply}"
validationProcess="${validPayeeName_process}"
/>
</td>
<td colspan="2">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericCurrencyInputParts.jsp">
<jsp:param name="gadgetId" value="${gadgetId}" />
<jsp:param name="tmtbleId" value="${tmtbleId}" />
<jsp:param name="searchCriteriaCompany" value="${searchCriteriaCompany}" />
<jsp:param name="companyCurrencyCd" value="${companyCurrencyCd}" />
<%-- 金額 --%>
<jsp:param name="modeTransAmnt" value="${modeTransAmount}" />
<jsp:param name="surfaceTransAmnt" value="${surfaceTransAmount}" />
<jsp:param name="requiredTransAmntApply" value="${reqTransAmount}" />
<jsp:param name="requiredTransAmntReapply" value="${reqTransAmount}" />
<jsp:param name="requiredTransAmntProcess" value="${reqTransAmount}" />

```

```

<%-- 通貨 --%>
<jsp:param name="modeTransCryCd" value="${modeTransCurrency}" />
<jsp:param name="surfaceTransCryCd" value="${surfaceTransCurrency}" />
<jsp:param name="requiredTransCryCdApply" value="${reqTransCurrency}" />
<jsp:param name="requiredTransCryCdReapply" value="${reqTransCurrency}" />
<jsp:param name="requiredTransCryCdProcess" value="${reqTransCurrency}" />
<%-- レート --%>
<jsp:param name="modeCmpRate" value="${modeCompanyRate}" />
<jsp:param name="surfaceCmpRate" value="${surfaceCompanyRate}" />
<jsp:param name="requiredCmpRateApply" value="${reqTransCurrency}" />
<jsp:param name="requiredCmpRateReapply" value="${reqTransCurrency}" />
<jsp:param name="requiredCmpRateProcess" value="${reqTransCurrency}" />
<%-- 換算額 --%>
<jsp:param name="modeCmpAmnt" value="${modeCompanyAmount}" />
<jsp:param name="surfaceCmpAmnt" value="${surfaceCompanyAmount}" />
<jsp:param name="requiredCmpAmntApply" value="${reqCompanyAmount}" />
<jsp:param name="requiredCmpAmntReapply" value="${reqCompanyAmount}" />
<jsp:param name="requiredCmpAmntProcess" value="${reqCompanyAmount}" />
</jsp:include>
</td>
</tr>
<tr>
<td colspan="3">
<a id="routeDetail" style="display:none">
<span class="kaiden-ui-icon-train24" style="vertical-align:middle;margin-right:0.5em"></span>
<k:surface
name="showDetailSurface"
surfaceKey="${surfaceShowDetail}"
style="vertical-align:middle;"
/>
</a>
<c:if test="${allowAddDetail && 'true' == useRouteSearch}">
<a id="routeSearch">
<span class="kaiden-ui-icon-search16" style="vertical-align:middle;margin-right:0.5em"></span>
<k:surface
name="routeSearchSurface"
surfaceKey="${surfaceRouteSearch}"
style="vertical-align:middle;"
/>
</a>
</c:if>
<table style="table-layout:fixed;display:inline-block;vertical-align:middle;margin-left:1em;;margin-right:1em;height:24px;" class="noborder" id="flagIconTbl" >
<colgroup>
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
<col style="width:30px;">
</colgroup>
<tbody>
<tr>
<td>
<span class="kaiden-ui-icon-fast24" id="iconFast" title="${surfaceAltFast}" style="display:none" ></span>
<span class="kaiden-ui-icon-fast-off24" id="iconFastOff" title="${surfaceAltFast}" style="display:none" ></span>
</td>
<td>
<span class="kaiden-ui-icon-cheap24" id="iconCheap" title="${surfaceAltCheap}" style="display:none" ></span>
<span class="kaiden-ui-icon-cheap-off24" id="iconCheapOff" title="${surfaceAltCheap}" style="display:none" ></span>
</td>
<td>
<span class="kaiden-ui-icon-easy24" id="iconEasy" title="${surfaceAltEasy}" style="display:none" ></span>
<span class="kaiden-ui-icon-easy-off24" id="iconEasyOff" title="${surfaceAltEasy}" style="display:none" ></span>
</td>
<td>
<span class="kaiden-ui-icon-eco24" id="iconEco" title="${surfaceAltEco}" style="display:none" ></span>
<span class="kaiden-ui-icon-eco-off24" id="iconEcoOff" title="${surfaceAltEco}" style="display:none" ></span>
</td>
<td>
<span class="kaiden-ui-icon-commuter24" id="iconCommuter" title="${surfaceAltCommuter}" style="display:none" ></span>
<span class="kaiden-ui-icon-commuter-off24" id="iconCommuterOff" title="${surfaceAltCommuter}" style="display:none" ></span>
</td>
<td></td>
<td>
<span class="kaiden-ui-icon-express24" id="iconExpress" title="${surfaceAltExpress}" style="display:none" ></span>
<span class="kaiden-ui-icon-express-off24" id="iconExpressOff" title="${surfaceAltExpress}" style="display:none" ></span>
</td>
<td>
<span class="kaiden-ui-icon-bus24" id="iconBus" title="${surfaceAltBus}" style="display:none" >
</span><span class="kaiden-ui-icon-bus-off24" id="iconBusOff" title="${surfaceAltBus}" style="display:none" ></span>
</td>
<td>

```

```

        <span class="kaiden-ui-icon-plane24" id="iconPlane" title="{surfaceAltAir}" style="display:none"></span>
        <span class="kaiden-ui-icon-plane-off24" id="iconPlaneOff" title="{surfaceAltAir}" style="display:none"></span>
    </td>
    <td>
        <span class="kaiden-ui-icon-ship24" id="iconShip" title="{surfaceAltShip}" style="display:none"></span>
        <span class="kaiden-ui-icon-ship-off24" id="iconShipOff" title="{surfaceAltShip}" style="display:none"></span>
    </td>
</tr>
</tbody>
</table>
<c:if test="{allowAddDetail}">
    <a id="releaseLink" style="display:none">
        <span class="kaiden-ui-icon-x16" style="vertical-align:middle;margin-right:0.5em"></span>
        <k:surface
            name="releaseLinkSurface"
            surfaceKey="{surfaceReleaseLink}"
            style="vertical-align:middle;"
        />
    </a>
</c:if>
<div id="routeDetailArea" class="kaiden-ui-routedetail"></div>
<k:hidden
    name="routeLinkFlag"
    mode="0"
    value="0"
/>
<k:hidden
    name="commutationRoute"
    mode="0"
/>
<k:hidden
    name="searchCondition"
    mode="0"
/>
<k:hidden
    name="routeInfo"
    mode="0"
/>
<k:hidden
    name="assessDate"
    mode="0"
/>
<k:hidden
    name="tranceMeansFlag"
    mode="0"
/>
<k:hidden
    name="assessFlag"
    mode="0"
/>
</td>
<td style="text-align:right;">
    <k:checkbox
        name="roundTripFlag"
        mode="{modeRoundTripFlag}"
        label="{surfaceRoundTrip}"
        surfaceKey="{surfaceRoundTrip}"
        validationDefault="{validRoundTripFlag_def}"
        validationApply="{validRoundTripFlag_apply}"
        validationTempsave="{validRoundTripFlag_tempsave}"
        validationReapply="{validRoundTripFlag_reapply}"
        validationProcess="{validRoundTripFlag_process}"
    />
    <k:numericText
        name="companyAmountSum"
        mode="{modeCompanyAmountSum}"
        formatted="view"
        surfaceKey="{surfaceCompanyAmountSum}"
        style="width:120px"
        tabindex="-1"
        readonly="readonly"
        validationDefault="{validCompanyAmountSum_def}"
        validationApply="{validCompanyAmountSum_apply}"
        validationTempsave="{validCompanyAmountSum_tempsave}"
        validationReapply="{validCompanyAmountSum_reapply}"
        validationProcess="{validCompanyAmountSum_process}"
    />
</td>
</tr>
</tbody>
</k:htmltable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){

```

```

var eventman = KAIDEN.getGadget( $("#gadgetId").eventmanager
, blockAccessor = KAIDEN.getGadget("${gadgetId}").blockAccessors["${gadgetBlockHeaderId}"]
, tmt = KAIDEN.getGadget("${gadgetId}").tmtables["${tmttableId}"]
, tma = tmt.tmttableAccessor;

/*****
 * 内部イベント
 *****/

//明細行:行挿入(データ設定前)
tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

    //デフォルト部門を要求
    var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "${gadgetId}");
    defaultDept = defaultDept[0].result; //先頭行のみ利用
    tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
    tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
    tma("departmentName", tupleId).setValue(defaultDept.departmentName);

    //デフォルトプロジェクトを要求
    var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "${gadgetId}");
    defaultPrj = defaultPrj[0].result; //先頭行のみ利用
    tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
    tma("projectName", tupleId).setValue(defaultPrj.projectName);

});

//明細行:行挿入(データ設定後)
tmt.addEventPostNewRow(function(tupleId, tma, data) {
    //マスタ検索部品の整合性確認。
    KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

    //レートボックスのreadonly化
    if (tma("transCurrencyCd", tupleId).elem(true).length > 0) {
        var currencyCd = tma("transCurrencyCd", tupleId).getValue();
        if (currencyCd == "${companyCurrencyCd}") {
            tma("companyRate", tupleId).readonly();
        } else {
            tma("companyRate", tupleId).removeReadOnly();
        }
    }

    //初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
    if (tma("payTypeCd", tupleId).elem(true).length > 0) {
        KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
    }

    //表示状態の調整
    KAIDEN.getGadget("${gadgetId}").variable.adjustingRouteSearchResult(tupleId, true);
});
//レート再取得ロジック
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll = function() {
    tma("transCurrencyCd").elem(true).each(function() {
        tmt.variable.loadRate(this);
        tmt.variable.multiplyRoundTrip(this);
    });
}
//レート再取得:ボタン押下
blockAccessor("rateReget").elem(true).on("click", function(e){
    imuiConfirm("${msgRateReget}", "", function() {
        KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
    }, undefined, false);
});

//金額*係数(往復フラグがONの場合は"2"、OFFの場合は"1"を利用)
tmt.variable.multiplyRoundTrip = function(target) {
    var companyAmount = KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmount").getValue();
    var roundTripFlag = KAIDEN.gadgetMan.getNeighboringAccessor(target, "roundTripFlag").getValue();

    if ($.isNumeric(companyAmount)) {
        //往復フラグがONの場合
        if (roundTripFlag == "1") {
            var total = new BigDecimal(companyAmount).multiply(new BigDecimal("2"));
            KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue([total.toString(), KAIDEN.Util.addFigure(total.toString())]);
        } else {
            KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue([companyAmount, KAIDEN.Util.addFigure(companyAmount)]);
        }
    } else {
        KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue("");
    }
};

//往復トグルボタン押下時後処理
tma("roundTripFlag").elem(true).live("click", function(e) {

```

```

tmt.variable.multiplyRoundTrip(this);
});
//取引通貨額入力後処理(readonlyだがtrigger時にも利用できるよう付与)
tma("transAmount").elem(true).live("change", function(e) {
    tmt.variable.multiplyRoundTrip(this);
});
//通貨コード入力後処理
tma("transCurrencyCd").elem(true).live("change", function(e) {
    tmt.variable.multiplyRoundTrip(this);
});
//レート入力後処理
tma("companyRate").elem(true).live("change", function(e) {
    tmt.variable.multiplyRoundTrip(this);
});

/*****
 * 受信イベント (push)
 *****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
    //明細行のクリア
    tmt.deleteAll();
    return KAIKEN.lmwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
    return KAIKEN.lmwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
    return KAIKEN.lmwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//reloadAllCurrencyRate レート再取得 イベントリスナの登録
eventMan.registPushListener("reloadAllCurrencyRate", function(**Object*/arg) {
    KAIKEN.getGadget("${gadgetId").variable.reloadRateAll();
});

/*****
 * 受信イベント (pull)
 *****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
    return tmt.getColumnValues(["routeDate", "routeDate"]);
});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues("companyAmountSum");
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value)) {
            result = result.add(new BigDecimal(value));
        }
    });
    return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues(["companyAmountSum", "paymentFlag"]);
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value[0]) && value[1] == "1") {
            result = result.add(new BigDecimal(value[0]));
        }
    });
    return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues(["companyAmountSum", "paymentFlag"]);
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value[0]) && value[1] != "1") {
            result = result.add(new BigDecimal(value[0]));
        }
    });
    return result.toString();
});

```

```

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
    return tmt.getColumnValues("companyAmountSum");
});

//claimExpenccount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenccount", function(**Object*/arg) {
    return tmt.getCount();
});

//経路検索結果返却イベントリスナの登録
eventMan.registPushListener("setRouteSearchResult", function(**Object*/routeResult) {

    var tupleId = KAIDEN.getGadget("${gadgetId").variable.routeSearchTargetTupleId;

    //連携内容を画面に設定
    var routeDate = routeResult.param.date;
    if (routeDate) {
        routeDate = routeDate.substring(0, 4) + "/" + routeDate.substring(4, 6) + "/" + routeDate.substring(6, 8)
    }
    tma("routeDate", tupleId).setValue(routeDate); //日付
    tma("startingSpot", tupleId).setValue(routeResult.param.from); //出発地
    tma("arrivalSpot", tupleId).setValue(routeResult.param.to); //到着地
    tma("transAmount", tupleId).setValue(routeResult.param.amount); // 片道金額

    //利用交通機関フラグ
    var useTrain = routeResult.param.train //電車利用
    , useExpress = "1" == routeResult.param.train && "1" == routeResult.param.charge ? "1" : "0" //特急利用
    , useBus = routeResult.param.bus //バス利用
    , usePlane = routeResult.param.airplane //飛行機利用
    , useShip = routeResult.param.ship //船舶利用
    , useWalk = routeResult.param.walk; //徒歩利用
    tma("tranceMeansFlag", tupleId).setValue(useTrain + useExpress + useBus + usePlane + useShip + useWalk + "0000"); //利用交通機関フラグ

    //評価フラグ
    var fastFlag = routeResult.param.fastFlag //早
    , cheapFlag = routeResult.param.cheapFlag //安
    , easeFlag = routeResult.param.easeFlag //楽
    , ecoFlag = routeResult.param.ecoFlag //eco
    , teikiFlag = routeResult.param.teikiFlag; //定期
    tma("assessFlag", tupleId).setValue(fastFlag + cheapFlag + easeFlag + ecoFlag + teikiFlag + "00000"); //評価フラグ

    //経路連携フラグ
    tma("routeLinkFlag", tupleId).setValue("1");

    //他駅すばあと情報
    tma("commutationRoute", tupleId).setValue(""); //定期区間
    tma("searchCondition", tupleId).setValue(routeResult.condition); //検索条件
    tma("routeInfo", tupleId).setValue(routeResult.route); //経路情報

    //金額の計算
    tma("transAmount", tupleId).elem().trigger("change");

    //表示状態の調整
    KAIDEN.getGadget("${gadgetId").variable.adjustingRouteSearchResult(tupleId);
});

//経路検索結果に基づく表示調整
KAIDEN.getGadget("${gadgetId").variable.adjustingRouteSearchResult = function(**String*/tupleId, /**boolean*/isDataLoadProcess) {
    if ("1" == tma("routeLinkFlag", tupleId).getValue()) {
        //経路検索連携済
        tma("#routeSearch", tupleId).elem(true).hide();
        tma("#routeDetail", tupleId).elem(true).show();
        tma("#releaseLink", tupleId).elem(true).show();
        // tma("routeDate", tupleId).readonly(); //日付
        tma("startingSpot", tupleId).readonly(); //出発地
        tma("arrivalSpot", tupleId).readonly(); //到着地
        tma("transAmount", tupleId).readonly(); // 金額

        //アイコンの調整（利用経路）
        var tranceMeansFlag = tma("tranceMeansFlag", tupleId).getValue()
        , train = tranceMeansFlag[0] //電車
        , express = tranceMeansFlag[1] //特急
        , bus = tranceMeansFlag[2] //バス
        , plain = tranceMeansFlag[3] //飛行機
        , ship = tranceMeansFlag[4] //船舶
        , walk = tranceMeansFlag[5]; //徒歩

        // 特急（電車）利用
        if ("1" == express) {
            tma("#iconExpress", tupleId).elem().show();
            tma("#iconExpressOff", tupleId).elem().hide();
        } else {
            tma("#iconExpress", tupleId).elem().hide();
            tma("#iconExpressOff", tupleId).elem().show();
        }
    }
}

```

```

    }

    // バス利用
    if ("1" == bus) {
        tma("#iconBus", tupleId).elem().show();
        tma("#iconBusOff", tupleId).elem().hide();
    } else {
        tma("#iconBus", tupleId).elem().hide();
        tma("#iconBusOff", tupleId).elem().show();
    }
}

// 飛行機利用
if ("1" == plain) {
    tma("#iconPlane", tupleId).elem().show();
    tma("#iconPlaneOff", tupleId).elem().hide();
} else {
    tma("#iconPlane", tupleId).elem().hide();
    tma("#iconPlaneOff", tupleId).elem().show();
}

// 船舶
if ("1" == ship) {
    tma("#iconShip", tupleId).elem().show();
    tma("#iconShipOff", tupleId).elem().hide();
} else {
    tma("#iconShip", tupleId).elem().hide();
    tma("#iconShipOff", tupleId).elem().show();
}

// アイコンの調整 (評価フラグ)
var assessFlag = tma("assessFlag", tupleId).getValue();
var fastFlag = assessFlag[0] //早
    , cheapFlag = assessFlag[1] //安
    , easeFlag = assessFlag[2] //楽
    , ecoFlag = assessFlag[3] //eco
    , teikiFlag = assessFlag[4]; //定期

// 早
if ("1" == fastFlag) {
    tma("#iconFast", tupleId).elem().show();
    tma("#iconFastOff", tupleId).elem().hide();
} else {
    tma("#iconFast", tupleId).elem().hide();
    tma("#iconFastOff", tupleId).elem().show();
}

// 安
if ("1" == cheapFlag) {
    tma("#iconCheap", tupleId).elem().show();
    tma("#iconCheapOff", tupleId).elem().hide();
} else {
    tma("#iconCheap", tupleId).elem().hide();
    tma("#iconCheapOff", tupleId).elem().show();
}

// ソート順によりアイコンのtitle属性を切り替える

var searchCondition = tma("searchCondition", tupleId).getValue();
if (searchCondition) {
    switch(KAIDEN.Util.parseJSON(searchCondition).sort) {
        case "teiki1":
            tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki1}");
            tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki1}");
            break;
        case "teiki3":
            tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki3}");
            tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki3}");
            break;
        case "teiki6":
            tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheapTeiki6}");
            tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheapTeiki6}");
            break;
        default :
            tma("#iconCheap", tupleId).elem().attr("title", "${surfaceAltCheap}");
            tma("#iconCheapOff", tupleId).elem().attr("title", "${surfaceAltCheap}");
            break;
    }
}

// 楽
if ("1" == easeFlag) {
    tma("#iconEasy", tupleId).elem().show();
    tma("#iconEasyOff", tupleId).elem().hide();
} else {
    tma("#iconEasy", tupleId).elem().hide();
    tma("#iconEasyOff", tupleId).elem().show();
}

// eco

```

```

    if ("1" == ecoFlag) {
        tma("#iconEco", tupleId).elem().show();
        tma("#iconEcoOff", tupleId).elem().hide();
    } else {
        tma("#iconEco", tupleId).elem().hide();
        tma("#iconEcoOff", tupleId).elem().show();
    }

    //eco
    if ("1" == teikiFlag) {
        tma("#iconCommuter", tupleId).elem().show();
        tma("#iconCommuterOff", tupleId).elem().hide();
    } else {
        tma("#iconCommuter", tupleId).elem().hide();
        tma("#iconCommuterOff", tupleId).elem().show();
    }

    //アイコンエリアを表示
    tma("#flagIconTbl", tupleId).elem().show();

    if (!isDataLoadProcess) {
        //通貨を会社通貨に固定
        tma("transCurrencyCd", tupleId).setValue("${companyCurrencyCd}").elem().trigger("change");
        tma("transCurrencyCd", tupleId).disabled();
    } else {
        tma("transCurrencyCd", tupleId).disabled();
    }
} else {
    //経路検索未連携

    tma("#routeSearch", tupleId).elem(true).show();
    tma("#routeDetail", tupleId).elem(true).hide();
    tma("#releaseLink", tupleId).elem(true).hide();

    tma("routeDate", tupleId).removeReadonly(); //日付
    tma("startingSpot", tupleId).removeReadonly(); //出発地
    tma("arrivalSpot", tupleId).removeReadonly(); //到着地
    tma("transAmount", tupleId).removeReadonly(); //定期代(1ヶ月)
    //アイコンエリアを非表示
    tma("#flagIconTbl", tupleId).elem(true).hide();

    if (!isDataLoadProcess) {
        tma("transCurrencyCd", tupleId).removeDisabled().elem().trigger("change");
    }
}
};

//経路検索画面を表示
tma("#routeSearch").elem(true).live("click", function(e) {
    var param = {};
    param.companyCd = "${searchCriteriaCompany}";
    param.userCd = "${applyAuthUserCode}";
    param.mode = "1"; //経路検索
    param.date = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "routeDate").getValue();
    param.from = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "startingSpot").getValue();
    param.to = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "arrivalSpot").getValue();
    KAIDEN.gadgetMan.execPush("openRouteSearch", param, "${gadgetId}");

    KAIDEN.getGadget("${gadgetId}").variable.routeSearchTargetTupleId = KAIDEN.gadgetMan.whereMe(e.target).tmtTupleId;
});

//詳細画面を表示
tma("#routeDetail").elem(true).live("click", function(e) {

    var routeDetailArea = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "#routeDetailArea").elem();
    routeDetailArea.toggle();

    if (routeDetailArea.attr("data-kaiden-routedetail-loaded")) {
        return;
    }

    var routeDataJson = KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "routeInfo").getValue();
    if (routeDataJson) {
        var routeData = KAIDEN.Util.parseJSON(routeDataJson);
        var requestJson = KAIDEN.Util.stringifyJSON(routeData.resultSet.result[0]);

        var param = {};
        param.requestParam = {};
        param.requestParam.result = requestJson;
        param.requestParam.mode = "1";
        param.companyCd = "${searchCriteriaCompany}";
        param.userCd = "${applyAuthUserCode}";
        param.callBack = function(/*String*/html) {
            KAIDEN.gadgetMan.getNeighboringAccessor(e.target, "#routeDetailArea").elem().attr("data-kaiden-routedetail-loaded", "true").imuiIndicator("destroy").html(html);
        }
    }
});

```



```

});

routeDetailArea.imuiIndicator();
KAIDEN.gadgetMan.execPush("showRouteDetailHtml", param, "${gadgetId}");
routeDetailArea.show();
}
});

//駅すばあと連携解除
tma("#releaseLink").elem(true).live("click", function(e) {

    var tupleId = KAIDEN.gadgetMan.whereMe(e.target).tmtTupleId;

    tma("routeLinkFlag", tupleId).setValue("0");
    tma("commutationRoute", tupleId).setValue(""); //定期区間
    tma("searchCondition", tupleId).setValue(""); //検索条件
    tma("routeInfo", tupleId).setValue(""); //経路情報
    tma("assessDate", tupleId).setValue(""); //評価判定基準日
    tma("#routeDetailArea").elem().removeAttr("data-kaiden-routedetail-loaded").hide().children().remove();

    KAIDEN.getGadget("${gadgetId}").variable.adjustingRouteSearchResult(tupleId);
});

});
</script>

```

SP

- Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/routeDetail/routeDetailV02_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
<c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- routeDetail v02 --%>
<c:set var="gadgetClass">routeDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">routeDetailV02Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">routeDetailTable</c:set>

<c:set var="gadgetResponse" value="${kimw.load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>
<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRouteDate">${gadget.surfaces["routeDate"]}</c:set>
<c:set var="surfaceTravelRoute">${gadget.surfaces["travelRoute"]}</c:set>
<c:set var="surfaceHyphen">${gadget.surfaces["hyphen"]}</c:set>
<c:set var="surfaceSupplement">${gadget.surfaces["supplement"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceRouteSearchLink">${gadget.surfaces["routeSearchLink"]}</c:set>
<c:set var="surfaceCompanyAmountSum">${gadget.surfaces["companyAmountSum"]}</c:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconp>
<h3>
<span>${surfaceChapterTitle}</span>
<span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="${fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>
<ul data-role="listview" data-divider-theme="b">

```

```

<c:forEach items="${gadgetResponse.data[tmtbleId]}" var="detailData">
<li>
<span class="ui-li-heading"><c:out value="${detailData.routeDate}"></c:out></span>
<span class="ui-li-aside"><c:out value="${detailData.companyAmountSum[1]}"></c:out></span>
<div class="indent">
<p class="kaiden-ui-overview-contents">
<c:out value="${detailData.startingSpot}"></c:out>
<c:out value="${surfaceHyphen}"></c:out>
<c:out value="${detailData.arrivalSpot}"></c:out>
</p>
<c:if test="${'1' == detailData.routeLinkFlag}">
<p class="kaiden-ui-overview-contents">
<c:if test="${'1' == fn:substring(detailData.assessFlag, 0, 1)}">
<span class="kaiden-ui-icon-fast24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 0, 1)}">
<span class="kaiden-ui-icon-fast-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 2, 3)}">
<span class="kaiden-ui-icon-cheap24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 2, 3)}">
<span class="kaiden-ui-icon-cheap-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 1, 2)}">
<span class="kaiden-ui-icon-easy24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 1, 2)}">
<span class="kaiden-ui-icon-easy-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 3, 4)}">
<span class="kaiden-ui-icon-eco24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 3, 4)}">
<span class="kaiden-ui-icon-eco-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.assessFlag, 4, 5)}">
<span class="kaiden-ui-icon-commuter24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.assessFlag, 4, 5)}">
<span class="kaiden-ui-icon-commuter-off24"></span>
</c:if>
<span style="display:inline-block;width:24px"></span>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 0, 1)}">
<span class="kaiden-ui-icon-express24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 0, 1)}">
<span class="kaiden-ui-icon-express-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 1, 2)}">
<span class="kaiden-ui-icon-bus24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 1, 2)}">
<span class="kaiden-ui-icon-bus-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 2, 3)}">
<span class="kaiden-ui-icon-plane24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 2, 3)}">
<span class="kaiden-ui-icon-plane-off24"></span>
</c:if>
<c:if test="${'1' == fn:substring(detailData.tranceMeansFlag, 3, 4)}">
<span class="kaiden-ui-icon-ship24"></span>
</c:if>
<c:if test="${'0' == fn:substring(detailData.tranceMeansFlag, 3, 4)}">
<span class="kaiden-ui-icon-ship-off24"></span>
</c:if>
</p>
</c:if>
<p class="kaiden-ui-overview-contents-sub"><c:out value="${detailData.payTypeName}"></c:out></p>
</div>
<ul>
<li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="${surfaceBack}"></c:out></a></li>
<li>
<p class="kaiden-ui-detail-item-title"><c:out value="${surfaceRouteDate}"></c:out></p>
<div class="indent">
<label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.routeDate}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="${surfaceTravelRoute}"></c:out></p>
<div class="indent">
<label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.startingSpot}"></c:out><c:out value="${surfaceHyphen}"></c:out><c:out value="${detailData.arriv
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="${surfaceSupplement}"></c:out></p>
<div class="indent">
<label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.supplement}"></c:out></label>
</div>

```

```

<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceExpType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.expTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceAccount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.accountName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTaxType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.taxTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceDepartment}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.departmentName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.projectName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.payTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayee}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="\${detailData.payeeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTransAmount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents-sub"><c:out value="\${detailData.transAmount[1]}"></c:out></label>
  <label class="kaiden-ui-detail-item-contents-sub" style="margin-left:1em"><c:out value="\${detailData.transCurrencyName}"></c:out></label>
</div>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents-sub kaiden-ui-cmpRate"><c:out value="\${detailData.companyRate[1]}"></c:out></label>
</div>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.companyAmount[1]}"></c:out></strong></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceCompanyAmountSum}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.companyAmountSum[1]}"></c:out></strong></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>

```

宿泊明細

宿泊明細のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   └── slcs
│   │   │       └── kaiden2
│   │           ├── product_workflow
│   │           │   ├── foundation
│   │           │   │   └── model
│   │           │       ├── entity
│   │           │       │   ├── K30tLodgeDetail.java
│   │           │       │   ├── K30tLodgeDetailNames.java
│   │           │       │   ├── Ka30tLodgeDetail.java
│   │           │       │   └── Ka30tLodgeDetailNames.java
│   │           │       └── service
│   │           │           ├── K30tLodgeDetailCondition.java
│   │           │           ├── K30tLodgeDetailsService.java
│   │           │           ├── Ka30tLodgeDetailCondition.java
│   │           │           └── Ka30tLodgeDetailsService.java
│   │   └── java
│   │       ├── jp
│   │       │   ├── co
│   │       │   │   └── slcs
│   │       │       └── kaiden2
│   │           ├── product_workflow
│   │           │   ├── feature
│   │           │   │   ├── action
│   │           │   │   │   ├── kaizen
│   │           │   │   │   │   └── gadget
│   │           │   │   │       ├── AbstractLodgeDetailActionBase.java
│   │           │   │   │       └── LodgeDetailV01Action.java
│   │           │   │   ├── conf
│   │           │   │   │   ├── kaizen
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── lodgedetail
│   │           │   │   │           └── LodgeDetailFieldName.java
│   │           │   │   ├── form
│   │           │   │   │   ├── kaizen
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── LodgeDetailV01Form.java
│   │           │   │   ├── logic
│   │           │   │   │   ├── kaizen
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── lodgedetail
│   │           │   │   │           ├── AbstractLodgeDetailLogicBase.java
│   │           │   │   │           └── LodgeDetailV01Logic.java
│   │           │   └── foundation
│   │           │       ├── manager
│   │           │       │   └── lodgedetail
│   │           │       │       ├── LodgeDetailManager.java
│   │           │       │       ├── LodgeDetailManagerDto.java
│   │           │       │       └── impl
│   │           │       │           └── LodgeDetailManagerImpl.java
│   │   └── resources
│   │       ├── META-INF
│   │       │   └── sql
│   │       │       ├── jp
│   │       │       │   ├── co
│   │       │       │   │   └── slcs
│   │       │       │       └── kaizen2
│   │       │       │   ├── product_workflow
│   │       │       │   │   ├── foundation
│   │       │       │   │   │   ├── manager
│   │       │       │   │   │   │   ├── lodgedetail
│   │       │       │   │   │   │   │   └── impl
│   │       │       │   │   │       └── LodgeDetailManagerImpl
│   │       │       │   │   │           selectForJournalSourceDtl.sql
│   │       │       │   │   ├── model
│   │       │       │   │   │   └── service
│   │       │       │   │       └── K30tLodgeDetailsService
│   │       │       │   │           ├── selectView.sql
│   │       │       │   │           └── selectViewForCopy.sql
│   │   └── webapp
│   │       ├── WEB-INF
│   │       │   └── view
│   │       │       └── kaizen2
│   │           ├── product_workflow
│   │           │   └── gadget
│   │           │       └── lodgeDetail
│   │           │           ├── lodgeDetailV01.jsp
│   │           │           └── lodgeDetailV01_sp.isp

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractLodgeDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 宿泊明細がジェットActionの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractLodgeDetailActionBase implements GadgetActionIF {

    /**
     * 対象ロジッククラスをサブクラスより返却。
     *
     * @return ロジッククラス
     */
    @Override
    public abstract PluginLogicIF getLogic();

    /**
     * Fromをサブクラスより返却。
     *
     * @return From
     */
    @Override
    public abstract AbstractImwPageForm getForm();

    /**
     * 申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード。
     *
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
    }

```

```

    return null;
}

/**
 * 処理画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcess() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConvertor());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

```

```

}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvector())));
    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConvector())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #LodgeDetailConvector 宿泊明細(標準)ガジェットの書式変換}を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected ConvertForJsonSource getConvector() {
    return new LodgeDetailConvector(getForm());
}

/**
 * 宿泊明細ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class LodgeDetailConvector extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;
    /**
     * NumberHelper.
     */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public LodgeDetailConvector(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     */

```



```

* <tr>
* <td>換算額</td>
* <td>スケールを会社通貨のスケールに変換</td>
* </tr>
* </table>
* </p>
*/

@Override
protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

    Map<String, Object> result = new HashMap<String, Object>(map);

    // 取引通貨額の書式変換
    if (result.containsKey(LODGE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) result.get(LODGE_DETAIL_TRANS_AMOUNT);
        String currencyCd = (String) result.get(LODGE_DETAIL_TRANS_CURRENCY_CD);
        result.put(LODGE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, currencyCd));
    }
    // レートの書式変換
    if (result.containsKey(LODGE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(LODGE_DETAIL_COMPANY_RATE);
        result.put(LODGE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }
    // 換算額の書式変換
    if (result.containsKey(LODGE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(LODGE_DETAIL_COMPANY_AMOUNT);
        String companyCurrencyCd = (String) result.get(LODGE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(LODGE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }
    return result;
}

/**
* {@inheritDoc}
* <p>
* 本サブクラスでは、以下の書式変換を行う。<br>
* <table border="1">
* <tr>
* <td>項目名</td>
* <td>書式</td>
* </tr>
* <tr>
* <td>取引通貨</td>
* <td>スケールを通貨のスケールに変換</td>
* </tr>
* <tr>
* <td>レート</td>
* <td>スケールをレートスケールに変換</td>
* </tr>
* <tr>
* <td>換算額</td>
* <td>スケールを会社通貨のスケールに変換</td>
* </tr>
* </table>
* </p>
*/

@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(LODGE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(LODGE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(LODGE_DETAIL_TRANS_CURRENCY_CD);
        result.put(LODGE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }
    // レートの書式変換
    if (map.containsKey(LODGE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(LODGE_DETAIL_COMPANY_RATE);
        result.put(LODGE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }
    // 換算額の書式変換
    if (map.containsKey(LODGE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(LODGE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = result.get(LODGE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(LODGE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }
    return result;
}
}
}

```

Action

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/LodgeDetailV01Action.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.LodgeDetailV01Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail.LodgeDetailV01Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 宿泊明細(標準)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV01Action extends AbstractLodgeDetailActionBase {

    /**
     * LodgeDetailV01Logic.
     */
    @Resource
    public LodgeDetailV01Logic lodgeDetailV01Logic;

    /**
     * LodgeDetailV01Form.
     */
    @ActionForm
    @Resource
    public LodgeDetailV01Form lodgeDetailV01Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return lodgeDetailV01Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return lodgeDetailV01Form;
    }
}
```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/LodgeDetailV01Form.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 宿泊明細(標準)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV01Form extends AbstractImwPageForm {

}
```

Logic(基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/lodgedetail/AbstractLodgeDetailLogicBase.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
```

```

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_ADDRESS;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_TELEPHONE_NUMBER;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseAmountIF;

```

```

import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beat;
import org.seasar.util.beans.util.BeanMap;

/**
 * 宿泊明細ガジェット用Logicの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractLodgeDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 宿泊明細ガジェット。 */
    protected static final String BLOCK_KEY_LODGE_DETAIL_TABLE = "lodgeDetailTable";

    /** 宿泊先名称。 */
    public static final int LODGE_NAME_LEN = 250;
    /** 宿泊先tel。 */
    public static final int LODGE_TELEPHONE_NUMBER_LEN = 250;
    /** 宿泊先住所。 */
    public static final int LODGE_ADDRESS_LEN = 250;
    /** 税率(整数部)。 */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部)。 */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部)。 */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部)。 */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先。 */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部)。 */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部)。 */
    public static final int QUANTITY_DEC_LEN = 10;

    /** LodgeDetailManager。 */
    @Resource
    protected LodgeDetailManager lodgeDetailManager;

    /** JournalSourceDtlManager。 */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper。 */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** コンストラクタ。 */
    public AbstractLodgeDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 宿泊明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる宿泊明細テーブルを削除後に、宿泊明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override

```

```

public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override

```

```

public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細テーブルの対象データを削除します。<br>

```

```

* 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
* <p>
*/
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる宿泊明細テーブルを削除後に、宿泊明細テーブルを登録します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * <p>
 */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * <p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * <p>
 */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.imwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 宿泊明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userId = pluginGadgetDTO.userId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 宿泊明細データを取得
            List<BeanMap> beanMapList = lodgeDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * <p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}

```

```

    * {@inheritDoc}
    * <p>
    * 対象となる宿泊明細の履歴を含めた全てのデータを削除します。
    * </p>
    */
    @Override
    public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる(過去案件)宿泊明細テーブルを一括削除します。
    * </p>
    */
    @Override
    public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = lodgeDetailManager
            .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().imwPluginParameterDto));
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる宿泊明細の履歴を含めた全てのデータを削除します。
    * </p>
    */
    @Override
    public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
    * {@inheritDoc}
    *
    * @see #viewMapFormatted
    */
    @Override
    protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
        return viewMapFormatted(conditionDto, null);
    }

    /**
    * {@inheritDoc}
    * <p>
    * データの取得は、{@link LodgeDetailManager#selectList(LodgeDetailManagerDto)}を利用します。
    * </p>
    */
    @Override
    protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
        throws KaidenSystemException {

        LodgeDetailManagerDto dto = Beans.createAndCopy(LodgeDetailManagerDto.class, conditionDto).execute();
        List<LodgeDetailManagerDto> listSelect = lodgeDetailManager.selectList(dto);
        LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
        List<Map<String, Object>> listDtoMap = new ArrayList<Map<String, Object>>();
        for (LodgeDetailManagerDto src : listSelect) {
            if (converter == null) {
                listDtoMap.add(BeansUtil.toMap(src, true));
            } else {
                listDtoMap.add(converter.mapConvert(BeansUtil.toMap(src, true)));
            }
        }
        rv.put(BLOCK_KEY_LODGE_DETAIL_TABLE, listDtoMap);
        return rv;
    }

    /**
    * 宿泊明細テーブルデータ登録処理。
    *
    * @return 更新カウント
    * @throws KaidenSystemException システムエラー
    */
    protected int insertTable() throws KaidenSystemException {
        // 宿泊明細ガジェットデータをMap形式で取得
        List<LodgeDetailManagerDto> dtos = JsonUtil.getDtosList(LodgeDetailManagerDto.class,
            getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE));

        // 宿泊明細テーブル登録
        return lodgeDetailManager.entry(dtos);
    }

    /**
    * 仕訳連携用明細テーブルデータ登録処理。
    *

```



```

    * @param srcList 宿泊明細テーブルデータリスト
    * @param matterEndProcessDto 案件終了処理用DTO
    * @return 更新カウント
    * @throws KaidenSystemException システムエラー
    */
    protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
        MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
        for (JournalSourceDtlManagerDto srcDto : srcList) {
            // 仕訳データ区分をセット
            srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
        }

        // 仕訳連携用明細テーブル登録
        return journalSourceDtlManager.entry(srcList);
    }

    /**
     * 宿泊明細テーブルデータ削除処理.
     *
     * @param tempFlg 一時保存データフラグ
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
        LodgeDetailManagerDto mngDto = new LodgeDetailManagerDto();
        mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
        mngDto.userDataId = super.getGadgetDto().userDataId;
        mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
        mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
        if (tempFlg) {
            mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
        }
        return lodgeDetailManager.remove(mngDto);
    }

    /**
     * 宿泊明細テーブルデータ削除処理(履歴を含む).
     *
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int deleteTableIncludeBackground() throws KaidenSystemException {
        LodgeDetailManagerDto mngDto = new LodgeDetailManagerDto();
        mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
        mngDto.userDataId = null;
        mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
        mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
        return lodgeDetailManager.removeAll(mngDto);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *

```

```

    * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
    */
    @Override
    protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, true);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override

```

```

protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>{@link DateValidator 日付型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">終了日</td>
 * <td>{@link DateValidator 日付型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link DateMinValidator 最小日付チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先名称</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先tel</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先住所</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>取引通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td rowspan="2">会社通貨換算レート</td>
 * <td>{@link DigitsValidator 小数型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericMinValidator 最小数値チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
 * <td align="center">入力</td>

```

```

* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数型チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
KaidenApplicationException {

if (fieldDefsMap.containsKey(BLOCK_KEY_LODGE_DETAIL_TABLE)) {
addValidateDefsRequired(BLOCK_KEY_LODGE_DETAIL_TABLE, tmpFlg);
// 開始日
addValidateOnlyInput(LODGE_DETAIL_START_DATE, VALID_DATE);
// 終了日
addValidateOnlyInput(LODGE_DETAIL_END_DATE, VALID_DATE);
addValidateOnlyInput(LODGE_DETAIL_END_DATE, VALID_DATE_MIN, LODGE_DETAIL_START_DATE);
// 宿泊先名称
addValidateOnlyInput(LODGE_DETAIL_LODGE_NAME, VALID_MAX_LENGTH, getArgs(LODGE_NAME_LEN));
// 宿泊先tel
addValidateOnlyInput(LODGE_DETAIL_LODGE_TELEPHONE_NUMBER, VALID_MAX_LENGTH,
getArgs(LODGE_TELEPHONE_NUMBER_LEN));
// 宿泊先住所
addValidateOnlyInput(LODGE_DETAIL_LODGE_ADDRESS, VALID_MAX_LENGTH, getArgs(LODGE_ADDRESS_LEN));
// 経費区分コード
addValidate(LODGE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, LODGE_DETAIL_EXP_TYPE_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 勘定科目コード
addValidate(LODGE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, LODGE_DETAIL_ACCOUNT_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 税区分コード
addValidate(LODGE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, LODGE_DETAIL_TAX_TYPE_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 取引通貨額
addValidateOnlyInput(LODGE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
getArgs(LODGE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
// 取引通貨コード
addValidate(LODGE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, LODGE_DETAIL_TRANS_CURRENCY_CD,
getGadgetDto().searchCriteriaCompany);
// 会社通貨換算レート
addValidateOnlyInput(LODGE_DETAIL_COMPANY_RATE, VALID_DIGITS,
getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
addValidateOnlyInput(LODGE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
// 会社通貨コード
addValidate(LODGE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, LODGE_DETAIL_COMPANY_CURRENCY_CD,
getGadgetDto().searchCriteriaCompany);
// 会社通貨額
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));

```

```

// 支払分類コード
addValidate(LODGE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, LODGE_DETAIL_PAY_TYPE_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 支払方法コード
addValidate(LODGE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, LODGE_DETAIL_PAY_METHOD_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 支払先名
addValidateOnlyInput(LODGE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
// 組織コード
addValidate(LODGE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCompany,
    LODGE_DETAIL_DEPARTMENT_SET_CD, LODGE_DETAIL_DEPARTMENT_CD,
    dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// プロジェクトコード
addValidate(LODGE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, LODGE_DETAIL_PROJECT_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 数量
addValidateOnlyInput(LODGE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
// 会社通貨額合計
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
    getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
}
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
 * 設定されている場合は、{@link LodgeDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, new ArrayList<LodgeDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectView(criteria));
    }
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link LodgeDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link LodgeDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>

```

```

*/
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した宿泊明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_depaertment)</li>
 * <li>プロジェクトマスタ</li>
 * <li>支払分類マスタ</li>
 * <li>支払方法マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * <p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_LODGE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<LodgeDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), LodgeDetailManagerDto.class);
            for (LodgeDetailManagerDto j : sqlResultList) {
                // 取引通貨チェック
                if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_TRANS_CURRENCY_CD, LODGE_DETAIL_TRANS_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 会社通貨チェック
                if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, LODGE_DETAIL_COMPANY_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 組織チェック
                if (isNotEmpty(j.departmentSetCd) && isEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_DEPARTMENT_SET_CD, LODGE_DETAIL_DEPARTMENT_CD,
                        LODGE_DETAIL_DEPARTMENT_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.DEPARTMENT);
                    String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // プロジェクトチェック
                if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_PROJECT_CD, LODGE_DETAIL_PROJECT_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        ProductWorkflowProp.S.PROJECT);
                    String[] values = getArgs(j.projectCd, j.projectName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 支払分類マスタチェック
                if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_PAY_TYPE_CD, LODGE_DETAIL_PAY_TYPE_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        ProductWorkflowProp.S.PAY_TYPE);
                    String[] values = getArgs(j.payTypeCd, j.payTypeName);

```

```

String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
    .toString(j.tupleId), masterName, values, message));
}

// 支払方法マスタチェック
if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_PAY_METHOD_CD, LODGE_DETAIL_PAY_METHOD_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_METHOD);
    String[] values = getArgs(j.payMethodCd, j.payMethodName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 経費区分マスタチェック
if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_EXP_TYPE_CD, LODGE_DETAIL_EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(j.expTypeCd, j.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        j.accountCd = null;
        j.accountName = null;
        j.taxTypeCd = null;
        j.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 勘定科目マスタチェック
if (isNotEmpty(j.accountCd) && isEmpty(j.accountName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_ACCOUNT_CD, LODGE_DETAIL_ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(j.accountCd, j.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 税区分マスタチェック
if (isNotEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_TAX_TYPE_CD, LODGE_DETAIL_TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}
}
}
}
return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Deprecated
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();

```

```

list.add(SYSTEM_MATTER_ID);
list.add(USER_DATA_ID);
list.add(GADGET_CLASS);
list.add(GADGET_INSTANCE);
return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccCountIF#getClaimExpenccCount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccCount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccCountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG));
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG));
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);
}

```



```

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            list.add(new BigDecimal(strAmnt));
        }
    }
    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細テーブルの対象データを過去案件回避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    @SuppressWarnings("unchecked")
    List<LodgeDetailManagerDto> list = (List<LodgeDetailManagerDto>) dataMap.get(BLOCK_KEY_LODGE_DETAIL_TABLE);
    int rv = lodgeDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(lodgeDetailManager);
    return list;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/lodgedetail/LodgeDetailV01Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

/**
 * 宿泊明細(標準)ガジェット (Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV01Logic extends AbstractLodgeDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public LodgeDetailV01Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します。<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力</td>
     * </tr>
     * </table>
     * </p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        super.defineValidateForNormal(params, tmpFlg);
    }

```

```

// 取引通貨額
addValidateOnlyInput(LODGE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
// 会社通貨額
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

if (!tmpFlg) {
    // 組織コード
    addValidateOnlyInput(LODGE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, LODGE_DETAIL_DEPARTMENT_CD,
        LODGE_DETAIL_PROJECT_CD);
    // プロジェクトコード
    addValidateOnlyInput(LODGE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, LODGE_DETAIL_DEPARTMENT_CD,
        LODGE_DETAIL_PROJECT_CD);
}
}

/**
 * {@@inheritDoc}
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>取引通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>会社通貨額: 取引通貨額の値をコピー</li>
 * <li>会社通貨換算レート: 固定値: 1を設定</li>
 * <li>数量: 固定値: 1を設定</li>
 * <li>立替フラグ: 支払分類コードに紐づく立替フラグを支払分類マスクから取得して設定</li>
 * </ul>
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(LODGE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 取引通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(LODGE_DETAIL_TRANS_CURRENCY_CD, companyCurrencyCd);

            // 会社通貨換算レート
            j.put(LODGE_DETAIL_COMPANY_RATE, "1");

            // 数量
            j.put(LODGE_DETAIL_QUANTITY, "1");

            // 立替フラグ
            if (StringUtil.isNotEmpty((String) j.get(LODGE_DETAIL_PAY_TYPE_CD))) {
                PayTypeManagerDto criteria = new PayTypeManagerDto();
                criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
                criteria.payTypeCd = (String) j.get(LODGE_DETAIL_PAY_TYPE_CD);
                criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                    .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
                List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                    criteria, new ArrayList<OrderDto>());
                if (CollectionUtil.isNotEmpty(retList)) {
                    j.put(LODGE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
                }
            }
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの)。
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>

```

```

</tr>
* <tr>
* <td>200</td>
* <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
* </tr>
* </table>
* </p>
*
* @return グローバルデータ補完処理
*/

@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * </p>
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(LODGE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(LODGE_DETAIL_TRANS_AMOUNT),
                (String) j.get(LODGE_DETAIL_COMPANY_RATE)));
            j.put(LODGE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(LODGE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(LODGE_DETAIL_QUANTITY)));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>支払方法コード:ガジェット間連携項目. 支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード
            if (gcPayMethodCd == null) {
                gcPayMethodCd = getGadgetCouplingPayMethod();
            }
            j.put(LODGE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
        }
    }
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenccCount(Map<String, Object> params) {
    return super.putClaimExpenccCount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

```

```

}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/lodgedetail/LodgeDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 宿泊明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class LodgeDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected LodgeDetailFieldName() {

        /** 開始日. */
        public static final String LODGE_DETAIL_START_DATE = "startDate";
        /** 終了日. */
        public static final String LODGE_DETAIL_END_DATE = "endDate";
        /** 宿泊先名称. */
        public static final String LODGE_DETAIL_LODGE_NAME = "lodgeName";
        /** 宿泊先tel. */
        public static final String LODGE_DETAIL_LODGE_TELEPHONE_NUMBER = "lodgeTelephoneNumber";
        /** 宿泊先住所. */
        public static final String LODGE_DETAIL_LODGE_ADDRESS = "lodgeAddress";
        /** 経費区分コード. */
        public static final String LODGE_DETAIL_EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String LODGE_DETAIL_ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String LODGE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String LODGE_DETAIL_TAX_RATE = "taxRate";
        /** 数量. */
        public static final String LODGE_DETAIL_QUANTITY = "quantity";
        /** 取引通貨コード. */
        public static final String LODGE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String LODGE_DETAIL_TRANS_AMOUNT = "transAmount";
        /** 会社通貨コード. */
        public static final String LODGE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
        /** 会社通貨換算レート. */
        public static final String LODGE_DETAIL_COMPANY_RATE = "companyRate";
        /** 会社通貨額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT = "companyAmount";
        /** 会社通貨税抜額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
        /** 会社通貨税額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
        /** 会社通貨額合計. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
        /** 支払分類コード. */
        public static final String LODGE_DETAIL_PAY_TYPE_CD = "payTypeCd";
        /** 支払方法コード. */
        public static final String LODGE_DETAIL_PAY_METHOD_CD = "payMethodCd";
        /** 支払先コード. */
        public static final String LODGE_DETAIL_PAYEE_CD = "payeeCd";
    }
}

```

```

public static final String LODGE_DETAIL_PAYEE_CD = "payeeCd";
/** 支払先名. */
public static final String LODGE_DETAIL_PAYEE_NAME = "payeeName";
/** 組織セットコード. */
public static final String LODGE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
/** 組織コード. */
public static final String LODGE_DETAIL_DEPARTMENT_CD = "departmentCd";
/** プロジェクトコード. */
public static final String LODGE_DETAIL_PROJECT_CD = "projectCd";
/** 会社名. */
public static final String LODGE_DETAIL_COMPANY_NAME = "companyName";
/** 経費区分名称. */
public static final String LODGE_DETAIL_EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String LODGE_DETAIL_ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String LODGE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
/** 取引通貨名称. */
public static final String LODGE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String LODGE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 支払分類名称. */
public static final String LODGE_DETAIL_PAY_TYPE_NAME = "payTypeName";
/** 支払方法名称. */
public static final String LODGE_DETAIL_PAY_METHOD_NAME = "payMethodName";
/** 組織セット名称. */
public static final String LODGE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
/** 組織名称. */
public static final String LODGE_DETAIL_DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String LODGE_DETAIL_PROJECT_NAME = "projectName";
/** 立替フラグ. */
public static final String LODGE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/LodgeDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 宿泊明細テーブルマネージャクラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface LodgeDetailManager extends PluginGadgetManager<LodgeDetailManagerDto> {

    /** {@link LodgeDetailManager#selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link LodgeDetailManager#selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link LodgeDetailManager#selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager(実装)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/impl/LodgeDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;

```

```

import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tLodgeDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tLodgeDetailService;
import org.seasar.util.beans.util.BeanMap;

/**
 * 宿泊明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行います、より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link LodgeDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link LodgeDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>
 * <td>主キー検索</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>表示案件検索</td>
 * <td>selectView</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>コピー案件検索</td>
 * <td>selectViewForCopy</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件検索</td>
 * <td>selectViewArchive</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括挿入</td>
 * <td>entry</td>
 * <td>登録</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括更新</td>
 * <td>renew</td>
 * <td>更新</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>削除</td>
 * <td>remove</td>
 * <td>削除</td>
 * <td><b>&nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括削除</td>
 * <td>removeAll</td>
 * <td>削除</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>過去案件登録</td>
 * <td>matterArchiveRegister</td>
 * <td>登録</td>

```

```
* <td><b>&nbsp</b></td>  
* </tr>  
* <tr>  
* <td>過去案件処理時削除</td>  
* <td>matterArchivePostDelete</td>  
* <td>削除</td>  
* <td><b>&nbsp</b></td>  
* </tr>  
* <tr>  
* <td>過去案件一括削除</td>  
* <td>removeAllArchive</td>  
* <td>削除</td>  
* <td>&nbsp</td>  
* </tr>  
* </table>  
* <br>  
* <br>  
* <b>機能別DTOクラスのI/Oマトリクス</b>  
* <table border="1">  
* <tr>  
* <td colspan="2" rowspan="2" align="center"><b>DTO</b></td>  
* <td colspan="2" align="center"><b>主キー</b>  
* 検索</b></td>  
* <td colspan="2" align="center"><b>リスト</b>  
* 検索</b></td>  
* <td colspan="2" align="center"><b>表示</b>  
* 案件<br>  
* 検索</b></td>  
* <td colspan="2" align="center"><b>コピー</b>  
* 案件<br>  
* 検索</b></td>  
* <td colspan="2" align="center"><b>過去</b>  
* 案件<br>  
* 検索</b></td>  
* <td colspan="2" align="center"><b>一括</b>  
* 挿入</b></td>  
* <td colspan="2" align="center"><b>一括</b>  
* 更新</b></td>  
* <td colspan="2" align="center"><b>削除</b></td>  
* <td colspan="2" align="center"><b>一括</b>  
* 削除</b></td>  
* <td colspan="2" align="center"><b>過去</b>  
* 案件<br>  
* 登録</b></td>  
* <td colspan="2" align="center"><b>過去</b>  
* 案件<br>  
* 処理時<br>  
* 削除</b></td>  
* <td colspan="2" align="center"><b>過去</b>  
* 案件<br>  
* 一括<br>  
* 削除</b></td>  
* </tr>  
* <tr>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* <td align="center"><b>l</b></td>  
* <td align="center"><b>o</b></td>  
* </tr>  
* <tr>  
* <td colspan="2">{@link CriteriaForSqlDto}</td>  
* <td align="center">o</td>  
* <td align="center">-</td>  
* <td align="center">-</td>  
* <td align="center">-</td>  
* <td align="center">o</td>
```



```
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*<td align="center"></td>  
*</tr>  
*<br>  
*<br>  
*<td colspan="2">[@link LodgeDetailManagerDto]</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center">O</td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*<td align="center">O</td>  
*<td align="center"></td>  
*</tr>  
*</table>  
*<br>  
  
*@author Sumitomo Life Information Systems Co.,Ltd.  
*/
```

```
public class LodgeDetailManagerImpl extends AbstractWfManager<LodgeDetailManagerDto> implements LodgeDetailManager {
```

```

/** K30tLodgeDetailsService. */
@Resource
public K30tLodgeDetailsService k30tLodgeDetailsService;

/** K30tLodgeDetailsService. */
@Resource
public Ka30tLodgeDetailsService ka30tLodgeDetailsService;

/**
 * {@inheritDoc}
 */
@Override
public LodgeDetailManagerDto selectByPk(LodgeDetailManagerDto criteria) {
    return super.selectByPkIgnoreAuthority(k30tLodgeDetailsService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<LodgeDetailManagerDto> selectList(LodgeDetailManagerDto criteria) {
    return super.selectPkUsedUserDataId(k30tLodgeDetailsService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<LodgeDetailManagerDto> selectView(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForView(k30tLodgeDetailsService, SELECT_SQLFILE_NM, criteria, order);
}

```

```

}

/** {@inheritDoc} */
@Override
public List<LodgeDetailManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectBySqlForViewForCopy(k30tLodgeDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<LodgeDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectListForViewForArchive(ka30tLodgeDetailService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<LodgeDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.insertBatch(k30tLodgeDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int renew(List<LodgeDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.renewBatch(k30tLodgeDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(LodgeDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tLodgeDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(LodgeDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tLodgeDetailService, data);
}

/**
 * {@inheritDoc}
 * <p>
 * ■ソート仕様<br>
 * <table border="1">
 * <tr>
 * <td>フィールド論理名</td>
 * <td>フィールド名</td>
 * <td>ソート順</td>
 * <td>行番号</td>
 * <td>rowNumber</td>
 * <td>ASC</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(CriteriaForSqlDto criteria, List<LodgeDetailManagerDto> dataList) {
    int rv = super.entryArchive(ka30tLodgeDetailService, criteria, dataList);
    return rv;
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
    int rv = super.deleteNoUsedUserDataId(k30tLodgeDetailService, criteria);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAllArchive(CriteriaForSqlDto criteria) {
    int rv = super.deleteNoUsedUserDataIdForArchive(ka30tLodgeDetailService, criteria);
    return rv;
}
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/LodgeDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 宿泊明細テーブルマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザーデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstanceId;
    /** タブルid. */
    public Long tupleId;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバリエーション. */
    public String gadgetVariation;
    /** 行番号. */
    public Long rowNumber;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 宿泊先名称. */
    public String lodgeName;
    /** 宿泊先tel. */
    public String lodgeTelephoneNumber;
    /** 宿泊先住所. */
    public String lodgeAddress;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** 税率. */
    public BigDecimal taxRate;
    /** 数量. */
    public BigDecimal quantity;
    /** 取引通貨コード. */
    public String transCurrencyCd;
    /** 取引通貨額. */
    public BigDecimal transAmount;
    /** 会社通貨コード. */
    public String companyCurrencyCd;
    /** 会社通貨換算レート. */
    public BigDecimal companyRate;
    /** 会社通貨額. */
    public BigDecimal companyAmount;
    /** 会社通貨換算税額. */

```

```

    /** 会社通貨税額. */
    public BigDecimal companyAmountExcTax;
    /** 会社通貨税額. */
    public BigDecimal companyAmountTax;
    /** 会社通貨額合計. */
    public BigDecimal companyAmountSum;
    /** 支払分類コード. */
    public String payTypeCd;
    /** 支払方法コード. */
    public String payMethodCd;
    /** 支払先コード. */
    public String payeeCd;
    /** 支払先名. */
    public String payeeName;
    /** 組織セットコード. */
    public String departmentSetCd;
    /** 組織コード. */
    public String departmentCd;
    /** プロジェクトコード. */
    public String projectCd;

    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 検索基準会社コード. */
    public String searchCriteriaCompany;
    /** 取引通貨名称. */
    public String transCurrencyName;
    /** 会社通貨名称. */
    public String companyCurrencyName;
    /** 組織名称. */
    public String departmentName;
    /** プロジェクトマスタプロジェクト名称. */
    public String projectName;
    /** 経費区分名称. */
    public String expTypeName;
    /** 勘定科目名称. */
    public String accountName;
    /** 税区分名称. */
    public String taxTypeName;
    /** 支払分類名称. */
    public String payTypeName;
    /** 支払方法名称. */
    public String payMethodName;
    /** 立替フラグ. */
    public String paymentFlag;
}

```

Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tLodgeDetail.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;
```

```

package jp.co.slcs.kaiden2.base.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 宿泊明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_lodge_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tLodgeDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;

    /** 行番号. */
    @Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long rowNumber;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp endDate;

    /** 宿泊先名称. */
    @Column(name = "lodge_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeName;

    /** 宿泊先tel. */
    @Column(name = "lodge_telephone_number", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeTelephoneNumber;

    /** 宿泊先住所. */
    @Column(name = "lodge_address", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeAddress;

    /** 経費区分コード. */
    @Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String expTypeCd;

    /** 勘定科目コード. */

```

```

@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

```

```

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tLodgeDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件)宿泊明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_lodge_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tLodgeDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ロケールid. */
    @Id

```

```

@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** アーカイブ年月. */
@Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
public String archiveMonth;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 開始日. */
@Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp endDate;

/** 宿泊先名称. */
@Column(name = "lodge_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeName;

/** 宿泊先tel. */
@Column(name = "lodge_telephone_number", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeTelephoneNumber;

/** 宿泊先住所. */
@Column(name = "lodge_address", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeAddress;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

```



```

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tLodgeDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tLodgeDetail} の名称クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tLodgeDetailNames {

    /**
     * デフォルトコンストラクタ。
     */
    private K30tLodgeDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却。
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却。
     *
     * @return userDataId のプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClass のプロパティ名を返却。
     *
     * @return gadgetClass のプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstance のプロパティ名を返却。
     *
     * @return gadgetInstance のプロパティ名
     */
}

```

```

    * @return gadgetInstanceのプロパティ名
    */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却。
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * startDateのプロパティ名を返却。
     *
     * @return startDateのプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDateのプロパティ名を返却。
     *
     * @return endDateのプロパティ名
     */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }

    /**
     * lodgeNameのプロパティ名を返却。
     *
     * @return lodgeNameのプロパティ名
     */
    public static PropertyName<String> lodgeName() {
        return new PropertyName<String>("lodgeName");
    }

    /**
     * lodgeTelephoneNumberのプロパティ名を返却。
     *
     * @return lodgeTelephoneNumberのプロパティ名
     */
    public static PropertyName<String> lodgeTelephoneNumber() {
        return new PropertyName<String>("lodgeTelephoneNumber");
    }

    /**
     * lodgeAddressのプロパティ名を返却。
     *
     * @return lodgeAddressのプロパティ名
     */
    public static PropertyName<String> lodgeAddress() {
        return new PropertyName<String>("lodgeAddress");
    }
}

```

```

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */

```

```

    public static PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>("companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>("companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**

```

```

 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

 /**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

 /**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

 /**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

 /**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

 /**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

 /**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

 /**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

 /**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

 /**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {

```

```

    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却。
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却。
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却。
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30tLodgeDetailNames extends PropertyName<K30tLodgeDetail> {

    /**
     * インスタンスを構築。
     */
    public _K30tLodgeDetailNames() {
    }

    /**
     * インスタンスを構築。
     *
     * @param name 名前
     */
    public _K30tLodgeDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築。
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30tLodgeDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }
}

```

```

/**
 * systemMatterIdのプロパティ名を返却.
 *
 * @return systemMatterIdのプロパティ名
 */
public PropertyName<String> systemMatterId() {
    return new PropertyName<String>(this, "systemMatterId");
}

/**
 * userDataIdのプロパティ名を返却.
 *
 * @return userDataIdのプロパティ名
 */
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名

```



```

*/
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * lodgeNameのプロパティ名を返却.
 *
 * @return lodgeNameのプロパティ名
 */
public PropertyName<String> lodgeName() {
    return new PropertyName<String>(this, "lodgeName");
}

/**
 * lodgeTelephoneNumberのプロパティ名を返却.
 *
 * @return lodgeTelephoneNumberのプロパティ名
 */
public PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>(this, "lodgeTelephoneNumber");
}

/**
 * lodgeAddressのプロパティ名を返却.
 *
 * @return lodgeAddressのプロパティ名
 */
public PropertyName<String> lodgeAddress() {
    return new PropertyName<String>(this, "lodgeAddress");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

```

```

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */

```

```

public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */

```

```

* extensionFのプロパティ名を返却.
*
* @return extensionFのプロパティ名
*/
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
* extensionGのプロパティ名を返却.
*
* @return extensionGのプロパティ名
*/
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
* extensionHのプロパティ名を返却.
*
* @return extensionHのプロパティ名
*/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
* extensionIのプロパティ名を返却.
*
* @return extensionIのプロパティ名
*/
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
* entryUserCdのプロパティ名を返却.
*
* @return entryUserCdのプロパティ名
*/
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
* renewCntのプロパティ名を返却.
*
* @return renewCntのプロパティ名
*/
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
* renewTsのプロパティ名を返却.
*
* @return renewTsのプロパティ名
*/
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
* renewUserCdのプロパティ名を返却.
*
* @return renewUserCdのプロパティ名
*/
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}

```

```

        return new PropertyName<String>(this, renewUserCode);
    }
}
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tLodgeDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tLodgeDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tLodgeDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tLodgeDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClass のプロパティ名を返却.
     *
     * @return gadgetClass のプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstance のプロパティ名を返却.
     *
     * @return gadgetInstance のプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleId のプロパティ名を返却.
     *
     * @return tupleId のプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * localeId のプロパティ名を返却.
     *
     * @return localeId のプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonth のプロパティ名を返却.

```

```

*
* @return archiveMonthのプロパティ名
*/
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
* gadgetIdのプロパティ名を返却.
*
* @return gadgetIdのプロパティ名
*/
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
* gadgetVariationのプロパティ名を返却.
*
* @return gadgetVariationのプロパティ名
*/
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
* rowNumberのプロパティ名を返却.
*
* @return rowNumberのプロパティ名
*/
public static PropertyName<Long> rowNumber() {
    return new PropertyName<Long>("rowNumber");
}

/**
* startDateのプロパティ名を返却.
*
* @return startDateのプロパティ名
*/
public static PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>("startDate");
}

/**
* endDateのプロパティ名を返却.
*
* @return endDateのプロパティ名
*/
public static PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>("endDate");
}

/**
* lodgeNameのプロパティ名を返却.
*
* @return lodgeNameのプロパティ名
*/
public static PropertyName<String> lodgeName() {
    return new PropertyName<String>("lodgeName");
}

/**
* lodgeTelephoneNumberのプロパティ名を返却.
*
* @return lodgeTelephoneNumberのプロパティ名
*/
public static PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>("lodgeTelephoneNumber");
}

/**
* lodgeAddressのプロパティ名を返却.
*
* @return lodgeAddressのプロパティ名
*/
public static PropertyName<String> lodgeAddress() {
    return new PropertyName<String>("lodgeAddress");
}

/**
* expTypeCdのプロパティ名を返却.
*
* @return expTypeCdのプロパティ名
*/
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

```

```

}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *

```

```

    * @return companyAmountExcTaxのプロパティ名
    */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>("companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }
}

```



```

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public static PropertyName<String> expTypeName() {
    return new PropertyName<String>("expTypeName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public static PropertyName<String> accountName() {
    return new PropertyName<String>("accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public static PropertyName<String> taxTypeName() {
    return new PropertyName<String>("taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public static PropertyName<String> transCurrencyName() {
    return new PropertyName<String>("transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public static PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>("companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public static PropertyName<String> payTypeName() {
    return new PropertyName<String>("payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public static PropertyName<String> paymentFlag() {
    return new PropertyName<String>("paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public static PropertyName<String> payMethodName() {
    return new PropertyName<String>("payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public static PropertyName<String> departmentName() {
    return new PropertyName<String>("departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */

```

```

    */
    public static PropertyName<String> projectName() {
        return new PropertyName<String>("projectName");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }
}

```

```

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _Ka30tLodgeDetailNames extends PropertyName<Ka30tLodgeDetail> {

        /**
         * インスタンスを構築.
         */
        public _Ka30tLodgeDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _Ka30tLodgeDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _Ka30tLodgeDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *

```

```

    * @return systemMatterIdのプロパティ名
    */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>((this, "systemMatterId"));
    }

    /**
     * userDataIdのプロパティ名を返却。
     *
     * @return userDataIdのプロパティ名
     */
    public PropertyName<String> userDataId() {
        return new PropertyName<String>((this, "userDataId"));
    }

    /**
     * gadgetClassのプロパティ名を返却。
     *
     * @return gadgetClassのプロパティ名
     */
    public PropertyName<String> gadgetClass() {
        return new PropertyName<String>((this, "gadgetClass"));
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
    public PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>((this, "gadgetInstance"));
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public PropertyName<Long> tupleId() {
        return new PropertyName<Long>((this, "tupleId"));
    }

    /**
     * localeIdのプロパティ名を返却。
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>((this, "localeId"));
    }

    /**
     * archiveMonthのプロパティ名を返却。
     *
     * @return archiveMonthのプロパティ名
     */
    public PropertyName<String> archiveMonth() {
        return new PropertyName<String>((this, "archiveMonth"));
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public PropertyName<String> gadgetId() {
        return new PropertyName<String>((this, "gadgetId"));
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public PropertyName<String> gadgetVariation() {
        return new PropertyName<String>((this, "gadgetVariation"));
    }

    /**
     * rowNumのプロパティ名を返却。
     *
     * @return rowNumのプロパティ名
     */
    public PropertyName<Long> rowNum() {
        return new PropertyName<Long>((this, "rowNum"));
    }
}

```

```

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * lodgeNameのプロパティ名を返却.
 *
 * @return lodgeNameのプロパティ名
 */
public PropertyName<String> lodgeName() {
    return new PropertyName<String>(this, "lodgeName");
}

/**
 * lodgeTelephoneNumberのプロパティ名を返却.
 *
 * @return lodgeTelephoneNumberのプロパティ名
 */
public PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>(this, "lodgeTelephoneNumber");
}

/**
 * lodgeAddressのプロパティ名を返却.
 *
 * @return lodgeAddressのプロパティ名
 */
public PropertyName<String> lodgeAddress() {
    return new PropertyName<String>(this, "lodgeAddress");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名

```

```

    */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }
}

```

```

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public PropertyName<String> expTypeName() {
    return new PropertyName<String>(this, "expTypeName");
}

/**
 * accountNameのプロパティ名を返却.
 *
 * @return accountNameのプロパティ名
 */
public PropertyName<String> accountName() {
    return new PropertyName<String>(this, "accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public PropertyName<String> taxTypeName() {
    return new PropertyName<String>(this, "taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */

```

```

public PropertyName<String> transCurrencyName() {
    return new PropertyName<String>(this, "transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>(this, "companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**

```



```

    * extensionDのプロパティ名を返却。
    *
    * @return extensionDのプロパティ名
    */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却。
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却。
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {

```

```

        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tLodgeDetail;

/**
 * {@link K30tLodgeDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tLodgeDetailService extends GenerateService<K30tLodgeDetail> {
    //
}

```

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tLodgeDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tLodgeDetail;

/**
 * {@link Ka30tLodgeDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tLodgeDetailService extends GenerateService<Ka30tLodgeDetail> {
    //
}

```

Condition

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30tLodgeDetail}の条件クラス.
 *

```

```

* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tLodgeDetailCondition extends
    AbstractEntityCondition<K30tLodgeDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public K30tLodgeDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tLodgeDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullStringCondition<K30tLodgeDetailCondition> systemMatterId =
        new NotNullStringCondition<K30tLodgeDetailCondition>("systemMatterId", this);

    /** userDataIdの条件. */
    public NotNullStringCondition<K30tLodgeDetailCondition> userDataId =
        new NotNullStringCondition<K30tLodgeDetailCondition>("userDataId", this);

    /** gadgetClassの条件. */
    public NotNullStringCondition<K30tLodgeDetailCondition> gadgetClass =
        new NotNullStringCondition<K30tLodgeDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NotNullCondition<K30tLodgeDetailCondition, Long> gadgetInstance =
        new NotNullCondition<K30tLodgeDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NotNullCondition<K30tLodgeDetailCondition, Long> tupleId =
        new NotNullCondition<K30tLodgeDetailCondition, Long>("tupleId", this);

    /** gadgetIdの条件. */
    public NotNullStringCondition<K30tLodgeDetailCondition> gadgetId =
        new NotNullStringCondition<K30tLodgeDetailCondition>("gadgetId", this);

    /** gadgetVariationの条件. */
    public NotNullStringCondition<K30tLodgeDetailCondition> gadgetVariation =
        new NotNullStringCondition<K30tLodgeDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */
    public NotNullCondition<K30tLodgeDetailCondition, Long> rowNumber =
        new NotNullCondition<K30tLodgeDetailCondition, Long>("rowNumber", this);

    /** startDateの条件. */
    public NullableCondition<K30tLodgeDetailCondition, Timestamp> startDate =
        new NullableCondition<K30tLodgeDetailCondition, Timestamp>("startDate", this);

    /** endDateの条件. */
    public NullableCondition<K30tLodgeDetailCondition, Timestamp> endDate =
        new NullableCondition<K30tLodgeDetailCondition, Timestamp>("endDate", this);

    /** lodgeNameの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> lodgeName =
        new NullableStringCondition<K30tLodgeDetailCondition>("lodgeName", this);

    /** lodgeTelephoneNumberの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> lodgeTelephoneNumber =
        new NullableStringCondition<K30tLodgeDetailCondition>("lodgeTelephoneNumber", this);

    /** lodgeAddressの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> lodgeAddress =
        new NullableStringCondition<K30tLodgeDetailCondition>("lodgeAddress", this);

    /** expTypeCdの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> expTypeCd =
        new NullableStringCondition<K30tLodgeDetailCondition>("expTypeCd", this);

    /** accountCdの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> accountCd =
        new NullableStringCondition<K30tLodgeDetailCondition>("accountCd", this);

    /** taxTypeCdの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> taxTypeCd =
        new NullableStringCondition<K30tLodgeDetailCondition>("taxTypeCd", this);

```

```

/** taxRateの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payTypeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payMethodCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payeeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payeeName =
    new NullableStringCondition<K30tLodgeDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> departmentSetCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> departmentCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> projectCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("projectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionA =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionB =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionC =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionD =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionE =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionE", this);

```

```

    new NullableStringCondition<K30tLodgeDetailCondition>( extensionF , this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionF =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionG =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionH =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionI =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionJ =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tLodgeDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tLodgeDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tLodgeDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Long> renewCnt =
        new NotNullableCondition<K30tLodgeDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<K30tLodgeDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<K30tLodgeDetailCondition> renewUserCd =
        new NotNullableStringCondition<K30tLodgeDetailCondition>("renewUserCd", this);
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tLodgeDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;

/**
 * {@link Ka30tLodgeDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tLodgeDetailCondition extends
    AbstractEntityCondition<Ka30tLodgeDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tLodgeDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tLodgeDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullableStringCondition<Ka30tLodgeDetailCondition> systemMatterId =
        new NotNullableStringCondition<Ka30tLodgeDetailCondition>("systemMatterId", this);
}

```

```

/** userDataIdの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> userDataId =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("userDataId", this);

/** gadgetClassの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> gadgetClass =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("gadgetClass", this);

/** gadgetInstanceの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Long> gadgetInstance =
    new NotNullCondition<Ka30tLodgeDetailCondition, Long>("gadgetInstance", this);

/** tupleIdの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Long> tupleId =
    new NotNullCondition<Ka30tLodgeDetailCondition, Long>("tupleId", this);

/** localeIdの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> localeId =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("localeId", this);

/** archiveMonthの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> archiveMonth =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("archiveMonth", this);

/** gadgetIdの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> gadgetId =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NotNullStringCondition<Ka30tLodgeDetailCondition> gadgetVariation =
    new NotNullStringCondition<Ka30tLodgeDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Long> rowNumber =
    new NotNullCondition<Ka30tLodgeDetailCondition, Long>("rowNumber", this);

/** startDateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Timestamp> startDate =
    new NullableCondition<Ka30tLodgeDetailCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Timestamp> endDate =
    new NullableCondition<Ka30tLodgeDetailCondition, Timestamp>("endDate", this);

/** lodgeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeName", this);

/** lodgeTelephoneNumberの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeTelephoneNumber =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeTelephoneNumber", this);

/** lodgeAddressの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeAddress =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeAddress", this);

/** expTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> accountCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */

```

```

public NullableStringCondition<Ka30tLodgeDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payeeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> projectCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> accountName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentName", this);

```

```

/** projectNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> projectName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionA =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionB =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionC =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionD =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionE =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionF =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionG =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionH =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionI =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionJ =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Timestamp> entryTs =
    new NotNullCondition<Ka30tLodgeDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tLodgeDetailCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tLodgeDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Long> renewCnt =
    new NotNullCondition<Ka30tLodgeDetailCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<Ka30tLodgeDetailCondition, Timestamp> renewTs =
    new NotNullCondition<Ka30tLodgeDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<Ka30tLodgeDetailCondition> renewUserCd =
    new NotNullableStringCondition<Ka30tLodgeDetailCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.start_date,
    t.end_date,

```



```

t.lodge_name,
t.lodge_telephone_number,
t.lodge_address,
t.exp_type_cd,
t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_currency_cd,
t.company_rate,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
m7.exp_type_name,
m8.account_name,
m9.tax_type_name,
m4.payment_flag
from
k30t_lodge_detail t
inner join
  k20c_matter_control c1
on
  t.system_matter_id = c1.system_matter_id
and
  t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
  c1.search_criteria_company = m1.company_cd
and
  t.trans_currency_cd = m1.currency_cd
and
  /*'localeId'*/ja' = m1.locale_id
and
  m1.delete_flag = '0'
left outer join
  imm_department m2
on
  c1.search_criteria_company = m2.company_cd
and
  t.department_set_cd = m2.department_set_cd
and
  t.department_cd = m2.department_cd
and
  /*'localeId'*/ja' = m2.locale_id
and
  c1.search_criteria_date >= m2.start_date
and
  c1.search_criteria_date < m2.end_date
and
  m2.delete_flag = '0'
left outer join
  k30m_project m3
on
  c1.search_criteria_company = m3.company_cd

```

```

and
  t.project_cd = m3.project_cd
and
  c1.search_criteria_date >= m3.start_date
and
  c1.search_criteria_date < m3.end_date
and
  /*localeld*/ja' = m3.locale_id
and
  m3.delete_flag = '0'
left outer join
  k30m_pay_type m4
on
  c1.search_criteria_company = m4.company_cd
and
  t.pay_type_cd = m4.pay_type_cd
and
  /*localeld*/ja' = m4.locale_id
and
  c1.search_criteria_date >= m4.start_date
and
  c1.search_criteria_date < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeld*/ja' = m5.locale_id
and
  c1.search_criteria_date >= m5.start_date
and
  c1.search_criteria_date < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeld*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_exp_type m7
on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeld*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeld*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeld*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date

```

```

c1.search_criteria_date >= m9.start_date
and
c1.search_criteria_date < m9.end_date
and
m9.delete_flag = '0'
where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

コピー

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailService/selectViewForCopy.sql

```

select
t.system_matter_id,
t.user_data_id,
t.gadget_class,
t.gadget_instance,
t.tuple_id,
t.gadget_id,
t.gadget_variation,
t.row_number,
t.start_date,
t.end_date,
t.lodge_name,
t.lodge_telephone_number,
t.lodge_address,
t.exp_type_cd,
t.account_cd,
t.tax_type_cd,
t.tax_rate,
t.quantity,
t.trans_currency_cd,
t.trans_amount,
t.company_currency_cd,
t.company_rate,
t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,

```

```

c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
ma.exp_type_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> "" then m8.account_name else mb.account_name end as account_name,
case when t.exp_type_cd is not null and t.exp_type_cd <> "" then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
m4.payment_flg
from
k30t_lodge_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeId*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeId*/ja' = m2.locale_id
and
/*searchCriteriaDate*/" >= m2.start_date
and
/*searchCriteriaDate*/" < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
/*searchCriteriaDate*/" >= m3.start_date
and
/*searchCriteriaDate*/" < m3.end_date
and
/*localeId*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4
on
c1.search_criteria_company = m4.company_cd
and
t.pay_type_cd = m4.pay_type_cd
and
/*localeId*/ja' = m4.locale_id
and
/*searchCriteriaDate*/" >= m4.start_date
and
/*searchCriteriaDate*/" < m4.end_date
and
m4.delete_flag = '0'
left outer join
k30m_pay_method m5
on
c1.search_criteria_company = m5.company_cd
and
t.pay_method_cd = m5.pay_method_cd
and
/*localeId*/ja' = m5.locale_id
and
/*searchCriteriaDate*/" >= m5.start_date
and
/*searchCriteriaDate*/" < m5.end_date
and
m5.delete_flag = '0'
left outer join imm_currency m6

```

```

left outer join mmm_currency mo
on
    c1.search_criteria_company = m6.company_cd
and
    t.company_currency_cd = m6.currency_cd
and
    /*localeId*/ja' = m6.locale_id
and
    m6.delete_flag = '0'
left outer join
k30m_account m8
on
    c1.search_criteria_company = m8.company_cd
and
    t.account_cd = m8.account_cd
and
    /*localeId*/ja' = m8.locale_id
and
    /*searchCriteriaDate*/" >= m8.start_date
and
    /*searchCriteriaDate*/" < m8.end_date
and
    m8.delete_flag = '0'
left outer join
k30m_tax_type m9
on
    c1.search_criteria_company = m9.company_cd
and
    t.tax_type_cd = m9.tax_type_cd
and
    /*localeId*/ja' = m9.locale_id
and
    /*searchCriteriaDate*/" >= m9.start_date
and
    /*searchCriteriaDate*/" < m9.end_date
and
    m9.delete_flag = '0'

left outer join
k30m_exp_type ma
on
    c1.search_criteria_company = ma.company_cd
and
    t.exp_type_cd = ma.exp_type_cd
and
    /*localeId*/ja' = ma.locale_id
and
    /*searchCriteriaDate*/" >= ma.start_date
and
    /*searchCriteriaDate*/" < ma.end_date
and
    ma.delete_flag = '0'
left outer join
k30m_account mb
on
    ma.company_cd = mb.company_cd
and
    ma.account_cd = mb.account_cd
and
    /*localeId*/ja' = mb.locale_id
and
    /*searchCriteriaDate*/" >= mb.start_date
and
    /*searchCriteriaDate*/" < mb.end_date
and
    m8.delete_flag = '0'
left outer join
k30m_tax_type mc
on
    ma.company_cd = mc.company_cd
and
    ma.tax_type_cd = mc.tax_type_cd
and
    /*localeId*/ja' = mc.locale_id
and
    /*searchCriteriaDate*/" >= mc.start_date
and
    /*searchCriteriaDate*/" < mc.end_date
and
    mc.delete_flag = '0'
where
    t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
    t.user_data_id = /*userDataId*/
/*END*/
/*IF gadnetClass != null*/

```

```

/* IF gadgetClass != null */
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/impl/LodgeDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
  t.system_matter_id as system_matter_id,
  t.user_data_id as user_data_id,
  t.gadget_class as gadget_class,
  t.gadget_instance as gadget_instance,
  t.tuple_id as tuple_id,
  t.gadget_id as gadget_id,
  t.gadget_variation as gadget_variation,
  i1.contents_id as contents_id,
  i1.contents_version_id as contents_version_id,
  null as journal_data_type_cd,
  t.row_number as row_number,
  t.exp_type_cd as exp_type_cd,
  t.account_cd as account_cd,
  t.tax_type_cd as tax_type_cd,
  mt.tax_rate as tax_rate,
  t.quantity as quantity,
  t.trans_currency_cd as trans_currency_cd,
  t.trans_amount as trans_amount,
  t.company_currency_cd as company_currency_cd,
  t.company_rate as company_rate,
  t.company_amount as company_amount,
  t.company_amount_exc_tax as company_amount_exc_tax,
  t.company_amount_tax as company_amount_tax,
  t.company_amount_sum as company_amount_sum,
  t.payee_cd as customer_cd,
  t.payee_name as customer_name,
  t.department_set_cd as department_set_cd,
  t.department_cd as department_cd,
  t.project_cd as project_cd,
  t.pay_method_cd as pay_method_cd,
  t.pay_type_cd as pay_type_cd,
  null as adv_matter_number,
  null as adv_matter_name,
  t.start_date as detail_date1,
  t.end_date as detail_date2,
  t.lodge_name as detail_summary1,
  null as detail_summary2,
  null as detail_summary3,
  null as detail_summary4,
  null as detail_summary5,
  null as detail_note,
  t.extension_a as detail_extension_a,
  t.extension_b as detail_extension_b,
  t.extension_c as detail_extension_c,
  t.extension_d as detail_extension_d,
  t.extension_e as detail_extension_e,
  t.extension_f as detail_extension_f,
  t.extension_g as detail_extension_g,
  t.extension_h as detail_extension_h,
  t.extension_i as detail_extension_i,
  t.extension_j as detail_extension_j,
  '0' as journal_status,
  null as wish_date,
  c1.search_criteria_company,
  c1.search_criteria_date
from
  k30t_lodge_detail t
  inner join
    k20c_matter_info i1
on

```

```

on
  t.system_matter_id = i1.system_matter_id
inner join
  k20c_matter_control c1
on
  t.system_matter_id = c1.system_matter_id
and
  t.user_data_id = c1.user_data_id
left outer join
  k30m_tax_type mt
on
  c1.search_criteria_company = mt.company_cd
and
  t.tax_type_cd = mt.tax_type_cd
and
  c1.search_criteria_date >= mt.start_date
and
  c1.search_criteria_date < mt.end_date
and
  i1.locale_id_apply = mt.locale_id
and
  mt.delete_flag = '0'

where
  t.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
  and
    t.user_data_id = /*userDataId*/
  /*END*/
  /*IF gadgetClass != null*/
  and
    t.gadget_class = /*gadgetClass*/
  /*END*/
  /*IF gadgetInstance != null*/
  and
    t.gadget_instance = /*gadgetInstance*/
  /*END*/
  /*IF gadgetId != null*/
  and
    t.gadget_id = /*gadgetId*/
  /*END*/
  /*IF searchCriteriaCompany != null*/
  and
    c1.search_criteria_company = /*searchCriteriaCompany*/
  /*END*/
  /*IF orderBy != null*/
  order by /*$orderBy*/.row_number
  /*END*/

```

JSP

PC

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/lodgeDetail/lodgeDetailV01.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wkf" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- lodgeDetail v01 --%>
<c:set var="gadgetClass">lodgeDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/lodgeDetailV01</c:set>
<c:set var="gadgetBlockId">lodgeDetailTable</c:set>
<c:set var="ttableId">lodgeDetailTable</c:set>
<c:set var="ttableControllerBlockId">ttableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks[lodgeDetailTable]}" />

```

```

<!-- tmtable --%>
<c:set var="allowAddDetail" value="{1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="{1' == detail.allowDelFlag}" />

<!-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceLodgeTerm">${gadget.surfaces["lodgeTerm"]}</c:set>
<c:set var="surfaceLodgeStartDate">${gadget.surfaces["startDate"]}</c:set>
<c:set var="surfaceConcatChar">${gadget.surfaces["concatChar"]}</c:set>
<c:set var="surfaceLodgeEndDate">${gadget.surfaces["endDate"]}</c:set>
<c:set var="surfaceLodge">${gadget.surfaces["lodgeName"]}</c:set>
<c:set var="surfaceLodgeTelephoneNumber">${gadget.surfaces["lodgeTelephoneNumber"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>

<!-- Mode --%>
<c:set var="modeStartDate">${detail.fields["startDate"].inputType}</c:set>
<c:set var="modeEndDate">${detail.fields["endDate"].inputType}</c:set>
<c:set var="modeLodge">${detail.fields["lodgeName"].inputType}</c:set>
<c:set var="modeLodgeTelephoneNumber">${detail.fields["lodgeTelephoneNumber"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>

<!-- RequiredMark --%>
<c:set var="reqStartDate" >${detail.fields["startDate"].requiredFlag}</c:set>
<c:set var="reqEndDate" >${detail.fields["endDate"].requiredFlag}</c:set>
<c:set var="reqLodge" >${detail.fields["lodgeName"].requiredFlag}</c:set>
<c:set var="reqLodgeTelephoneNumber" >${detail.fields["lodgeTelephoneNumber"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>

<!-- Validator --%>
<c:set var="validStartDate_def">{k_date:[""]}</c:set>
<c:set var="validStartDate_apply">${"true" == reqStartDate ? "k_required:[""] : ""}</c:set>
<c:set var="validStartDate_tempsave"></c:set>
<c:set var="validStartDate_reapply">${"true" == reqStartDate ? "k_required:[""] : ""}</c:set>
<c:set var="validStartDate_process">${"true" == reqStartDate ? "k_required:[""] : ""}</c:set>

<c:set var="validEndDate_def">{k_date:[""], k_dateMin:['startDate']}</c:set>
<c:set var="validEndDate_apply">${"true" == reqEndDate ? "k_required:[""] : ""}</c:set>
<c:set var="validEndDate_tempsave"></c:set>
<c:set var="validEndDate_reapply">${"true" == reqEndDate ? "k_required:[""] : ""}</c:set>
<c:set var="validEndDate_process">${"true" == reqEndDate ? "k_required:[""] : ""}</c:set>

<c:set var="validLodge_def">{k_maxLength:['250']}</c:set>
<c:set var="validLodge_apply">${"true" == reqLodge ? "k_required:[""] : ""}</c:set>
<c:set var="validLodge_tempsave"></c:set>
<c:set var="validLodge_reapply">${"true" == reqLodge ? "k_required:[""] : ""}</c:set>
<c:set var="validLodge_process">${"true" == reqLodge ? "k_required:[""] : ""}</c:set>

<c:set var="validLodgeTelephoneNumber_def">{k_maxLength:['250']}</c:set>
<c:set var="validLodgeTelephoneNumber_apply">${"true" == reqLodgeTelephoneNumber ? "k_required:[""] : ""}</c:set>
<c:set var="validLodgeTelephoneNumber_tempsave"></c:set>
<c:set var="validLodgeTelephoneNumber_reapply">${"true" == reqLodgeTelephoneNumber ? "k_required:[""] : ""}</c:set>
<c:set var="validLodgeTelephoneNumber_process">${"true" == reqLodgeTelephoneNumber ? "k_required:[""] : ""}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:['expTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validExpTypeCd_apply">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>
<c:set var="validExpTypeCd_process">${"true" == reqExpTypeCd ? "k_required:[""] : ""}</c:set>

<c:set var="validAccountCd_def">{k_accountExists:['accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>
<c:set var="validAccountCd_process">${"true" == reqAccountCd ? "k_required:[""] : ""}</c:set>

```



```

<c:set var="validTaxType_def">{k_taxTypeExists:['taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">{{"true" == reqTaxTypeCd ? "k_required:[''] : ""}}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{{"true" == reqTaxTypeCd ? "k_required:[''] : ""}}</c:set>
<c:set var="validTaxType_process">{{"true" == reqTaxTypeCd ? "k_required:[''] : ""}}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:['payTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">{{"true" == reqPayTypeCd ? "k_required:[''] : ""}}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPayTypeCd_reapply">{{"true" == reqPayTypeCd ? "k_required:[''] : ""}}</c:set>
<c:set var="validPayTypeCd_process">{{"true" == reqPayTypeCd ? "k_required:[''] : ""}}</c:set>

<c:set var="validPayeeName_def">{k_maxLength:['250']}</c:set>
<c:set var="validPayeeName_apply">{{"true" == reqPayeeName ? "k_required:[''] : ""}}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">{{"true" == reqPayeeName ? "k_required:[''] : ""}}</c:set>
<c:set var="validPayeeName_process">{{"true" == reqPayeeName ? "k_required:[''] : ""}}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:['searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:['projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<k:gadgetContainer gadgetClass="{gadgetClass}"
    gadgetVariation="{gadgetVariation}"
    gadgetInstance="{gadgetInstance}"
    gadgetTitle="{surfaceChapterTitle}"
    gadgetId="{gadgetId}"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:ttable id="{ttableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumberTarget="rowNumber">
<colgroup>
<col style="width:40px;overflow:hidden">
<col style="width:235px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
</colgroup>
<thead>
<tr>
<td rowspan="3" style="vertical-align:middle;">
<k:ttableOperationPanel
    position="header"
    allowAdd="{allowAddDetail}"
    allowDel="{allowDelDetail}"
/>
</td>
<th>
<k:surface
    name="lodgeTermSurface"
    surfaceKey="{surfaceLodgeTerm}"
    requiredMark="{reqStartDate}"
/>
</th>
<th colspan="2">
<k:surface
    name="lodgeNameSurface"
    surfaceKey="{surfaceLodge}"
    requiredMark="{reqLodge}"
/>
</th>
<th>
<k:surface
    name="lodgeTelephoneSurface"
    surfaceKey="{surfaceLodgeTelephone}"
    requiredMark="{reqLodgeTelephone}"
/>
</th>
</tr>
<tr>
<th colspan="2">
<k:surface
    name="expTypeSurface"
    surfaceKey="{surfaceExpType}"
    requiredMark="{reqExpTypeCd}"
    style="display:inline;"
/>
</th>
</tr>

```

```

<k:surface
  name="accountSurface"
  surfaceKey="{surfaceAccount}"
  style="display:inline;"
/>
<k:surface
  name="taxTypeSurface"
  surfaceKey="{surfaceTaxType}"
  style="display:inline;"
/>
</th>
<th>
  <k:surface
    name="departmentSurface"
    surfaceKey="{surfaceDepartment}"
    requiredMark="{reqDepartmentCd}"
  />
</th>
<th>
  <k:surface
    name="projectSurface"
    surfaceKey="{surfaceProject}"
    requiredMark="{reqProjectCd}"
  />
</th>
</tr>
<tr>
<th>
  <k:surface
    name="payTypeSurface"
    surfaceKey="{surfacePayType}"
    requiredMark="{reqPayTypeCd}"
  />
</th>
<th>
  <k:surface
    name="payeeSurface"
    surfaceKey="{surfacePayee}"
    requiredMark="{reqPayeeName}"
  />
</th>
<th colspan="2">
  <k:surface
    name="transAmountSurface"
    surfaceKey="{surfaceTransAmount}"
    requiredMark="{reqTransAmount}"
  />
</th>
</tr>
</thead>
<tbody>
<tr>
  <td rowspan="3">
    <k:tmtableOperationPanel
      position="detail"
      allowAdd="{allowAddDetail}"
      allowDel="{allowDelDetail}"
      name="rowNumber"
    />
  </td>
  <td>
    <k:date
      name="startDate"
      mode="{modeStartDate}"
      surfaceKey="{surfaceLodgeStartDate}"
      validationDefault="{validStartDate_def}"
      validationApply="{validStartDate_apply}"
      validationTempsave="{validStartDate_tempsave}"
      validationReapply="{validStartDate_reapply}"
      validationProcess="{validStartDate_process}"
    />
    <k:surface
      name="concatCharSurface"
      surfaceKey="{surfaceConcatChar}"
    />
    <k:date
      name="endDate"
      mode="{modeEndDate}"
      surfaceKey="{surfaceLodgeEndDate}"
      validationDefault="{validEndDate_def}"
      validationApply="{validEndDate_apply}"
      validationTempsave="{validEndDate_tempsave}"
      validationReapply="{validEndDate_reapply}"
      validationProcess="{validEndDate_process}"
    />
  </td>
</tr>

```

```

<td colspan="2">
  <k:text
    name="lodgeName"
    mode="{modeLodge}"
    imeMode="on"
    surfaceKey="{surfaceLodge}"
    style="width:350px;"
    maxLength="250"
    validationDefault="{validLodge_def}"
    validationApply="{validLodge_apply}"
    validationTempsave="{validLodge_tempsave}"
    validationReapply="{validLodge_reapply}"
    validationProcess="{validLodge_process}"
  />
</td>
<td>
  <k:text
    name="lodgeTelephoneNumber"
    mode="{modeLodgeTelephoneNumber}"
    imeMode="off"
    surfaceKey="{surfaceLodgeTelephoneNumber}"
    style="width:165px;"
    maxLength="250"
    validationDefault="{validLodgeTelephoneNumber_def}"
    validationApply="{validLodgeTelephoneNumber_apply}"
    validationTempsave="{validLodgeTelephoneNumber_tempsave}"
    validationReapply="{validLodgeTelephoneNumber_reapply}"
    validationProcess="{validLodgeTelephoneNumber_process}"
  />
</td>
</tr>
<tr>
<td colspan="2">
  <k:hidden
    name="expTypeCd"
    mode="{modeExpTypeCd}"
  />
  <kmsw:expType
    name="expTypeName"
    mode="{modeExpTypeCd}"
    imeMode="on"
    surfaceKey="{surfaceExpType}"
    resultValue="expTypeName"
    resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
    primaryTarget="expTypeCd"
    searchCriteriaDate="searchCriteriaDate"
    searchCriteriaCompany="searchCriteriaCompany"
    searchCriteriaExpTypeCd="expTypeCd"
    style="width:150px;"
    validationDefault="{validExpTypeCd_def}"
    validationApply="{validExpTypeCd_apply}"
    validationTempsave="{validExpTypeCd_tempsave}"
    validationReapply="{validExpTypeCd_reapply}"
    validationProcess="{validExpTypeCd_process}"
  />
  <k:hidden
    name="accountCd"
    mode="{modeAccountCd}"
    surfaceKey="{surfaceAccount}"
    validationDefault="{validAccountCd_def}"
    validationApply="{validAccountCd_apply}"
    validationTempsave="{validAccountCd_tempsave}"
    validationReapply="{validAccountCd_reapply}"
    validationProcess="{validAccountCd_process}"
  />
  <k:text
    name="accountName"
    mode="1"
    class="kaiden-ui-mst-search-chain-item"
    style="width:115px;"
  />
  <k:hidden
    name="taxTypeCd"
    mode="{modeTaxTypeCd}"
    surfaceKey="{surfaceTaxType}"
    validationDefault="{validTaxType_def}"
    validationApply="{validTaxType_apply}"
    validationTempsave="{validTaxType_tempsave}"
    validationReapply="{validTaxType_reapply}"
    validationProcess="{validTaxType_process}"
  />
  <k:text
    name="taxTypeName"
    mode="1"
    class="kaiden-ui-mst-search-chain-item"
    style="width:115px;"

```

```

/>
</td>
<td>
<k:hidden
  name="departmentSetCd"
  mode="{modeDepartmentCd}"
/>
<k:hidden
  name="departmentCd"
  mode="{modeDepartmentCd}"
  surfaceKey="{surfaceDepartment}"
/>
<kmsb:department
  name="departmentName"
  mode="{modeDepartmentCd}"
  imeMode="on"
  surfaceKey="{surfaceDepartment}"
  resultValue="departmentName"
  resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd}"
  primaryTarget="departmentCd,departmentSetCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaDepartmentCd="departmentCd"
  searchCriteriaDepartmentSetCd="departmentSetCd"
  style="width:150px;"
  validationDefault="{validDepartmentCd_def}"
  validationApply="{validDepartmentCd_apply}"
  validationTempsave="{validDepartmentCd_tempsave}"
  validationReapply="{validDepartmentCd_reapply}"
  validationProcess="{validDepartmentCd_process}"
/>
</td>
<td>
<k:hidden
  name="projectCd"
  mode="{modeProjectCd}"
  surfaceKey="{surfaceProject}"
/>
<kmsw:project
  name="projectName"
  mode="{modeProjectCd}"
  imeMode="on"
  surfaceKey="{surfaceProject}"
  resultValue="projectName"
  resultChain="{projectCd:'projectCd'"
  primaryTarget="projectCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaProjectCd="projectCd"
  style="width:150px;"
  validationDefault="{validProjectCd_def}"
  validationApply="{validProjectCd_apply}"
  validationTempsave="{validProjectCd_tempsave}"
  validationReapply="{validProjectCd_reapply}"
  validationProcess="{validProjectCd_process}"
/>
</td>
</tr>
<tr>
<td>
<klist_wkf:listDataPayType
  blank="false"
  name="payType"
  companyCd="{param.searchCriteriaCompany}"
  searchCriteriaDate="{param.searchCriteriaDate}"
/>
<k:select
  name="payTypeCd"
  mode="{modePayTypeCd}"
  surfaceKey="{surfacePayType}"
  viewModeName="payTypeName"
  dataName="payType"
  selectChain="{paymentFlag:'paymentFlag'"
  style="width:230px;"
  validationDefault="{validPayTypeCd_def}"
  validationApply="{validPayTypeCd_apply}"
  validationTempsave="{validPayTypeCd_tempsave}"
  validationReapply="{validPayTypeCd_reapply}"
  validationProcess="{validPayTypeCd_process}"
/>
<k:hidden
  name="paymentFlag"
  mode="{modePayTypeCd}"
/>
</td>
<td>

```

```

<k:text
  name="payeeName"
  mode="{modePayeeName}"
  imeMode="on"
  surfaceKey="{surfacePayee}"
  style="width:160px;"
  maxLength="250"
  validationDefault="{validPayeeName_def}"
  validationApply="{validPayeeName_apply}"
  validationTempsave="{validPayeeName_tempsave}"
  validationReapply="{validPayeeName_reapply}"
  validationProcess="{validPayeeName_process}"
/>
</td>
<td colspan="2">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericAmountInputParts.jsp">
<jsp:param name="gadgetId" value="{gadgetId}" />
<jsp:param name="tmtableId" value="{tmtableId}" />
<jsp:param name="searchCriteriaCompany" value="{searchCriteriaCompany}" />
<%-- 金額 --%>
<jsp:param name="modeTransAmnt" value="{modeTransAmount}" />
<jsp:param name="surfaceTransAmnt" value="{surfaceTransAmount}" />
<jsp:param name="requiredTransAmntApply" value="{reqTransAmount}" />
<jsp:param name="requiredTransAmntReapply" value="{reqTransAmount}" />
<jsp:param name="requiredTransAmntProcess" value="{reqTransAmount}" />
</jsp:include>
</td>
</tr>
</tbody>
</k:tmtable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){

  var eventMan = KAIDEN.getGadget("{gadgetId}").eventManager
  , blockAccessor = KAIDEN.getGadget("{gadgetId}").blockAccessors["{gadgetBlockHeaderId}"]
  , tmt = KAIDEN.getGadget("{gadgetId}").tmtables["{tmtableId}"]
  , tma = tmt.tmtableAccessor;

  /*****
  * 内部イベント
  *****/

  //明細行:行挿入(データ設定前)
  tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

    //デフォルト部門を要求
    var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "{gadgetId}");
    defaultDept = defaultDept[0].result; //先頭行のみ利用
    tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
    tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
    tma("departmentName", tupleId).setValue(defaultDept.departmentName);

    //デフォルトプロジェクトを要求
    var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "{gadgetId}");
    defaultPrj = defaultPrj[0].result; //先頭行のみ利用
    tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
    tma("projectName", tupleId).setValue(defaultPrj.projectName);

  });

  //明細行:行挿入(データ設定後)
  tmt.addEventPostNewRow(function(tupleId, tma, data) {
    //マスタ検索部品の整合性確認。
    KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

    //初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
    if (tma("payTypeCd", tupleId).elem(true).length > 0) {
      KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
    }

    //事前申請書取込では、会社通貨額が設定されているため、取引通貨額に会社通貨額を設定する
    //※外貨版から標準版にコピーする際、会社通貨額(レート換算後)の金額を設定する
    var compAmnt = tma("companyAmount", tupleId).getValue();
    if (compAmnt) {
      tma("transAmount", tupleId).setValue(compAmnt);
      tma("companyAmount", tupleId).setValue("");
    }

  });

  /*****
  * 内部イベント
  *****/

```

```

* 受信イベント (push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
  //明細行のクリア
  tmt.deleteAll();
  return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
  return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
  return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

/*****
* 受信イベント (pull)
*****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
  return tmt.getColumnValues(["startDate", "endDate"]);
});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues("transAmount");
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
  return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["transAmount", "paymentFlag"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value[0]) && value[1] == "1") {
      result = result.add(new BigDecimal(value[0]));
    }
  });
  return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
  var amounts = tmt.getColumnValues(["transAmount", "paymentFlag"]);
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value[0]) && value[1] == "0") {
      result = result.add(new BigDecimal(value[0]));
    }
  });
  return result.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
  return tmt.getColumnValues("transAmount");
});

//claimExpenCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenCount", function(**Object*/arg) {
  return tmt.getCount();
});

});
</script>

```

SP

- **Path**
src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/lodgeDetail/lodgeDetailV01_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>

```

```

<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<%-- product_workflow 共通インクルード --%>
<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- 宿泊明細(標準)ガジェット-->
<c:set var="gadgetClass">lodgeDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">lodgeDetailV01Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">lodgeDetailTable</c:set>

<%-- ガジェットデータの読み込み --%>
<c:set var="gadgetResponse" value="${kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>

<%-- サーフェスの読み込み --%>
<c:set var="gadget" value="{appDto.gadgets[gadgetId]}" />

<%-- サーフェス --%>
<%--タイトル --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<%-- 宿泊期間 --%>
<c:set var="surfaceLodgeTerm">${gadget.surfaces["lodgeTerm"]}</c:set>
<%-- 宿泊先 --%>
<c:set var="surfaceLodge">${gadget.surfaces["lodgeName"]}</c:set>
<%-- 宿泊先TEL --%>
<c:set var="surfaceLodgeTelephoneNumber">${gadget.surfaces["lodgeTelephoneNumber"]}</c:set>
<%-- 経費区分 --%>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<%-- 勘定科目 --%>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<%-- 税区分 --%>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<%-- 金額 --%>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<%-- 支払分類 --%>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<%-- 支払先 --%>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<%-- 部門 --%>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<%-- プロジェクト --%>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<%-- 戻る --%>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconpc
<h3>
  <span>${surfaceChapterTitle}</span>
  <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="{fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>

<ul data-role="listview" data-divider-theme="b">
  <c:forEach items="{gadgetResponse.data[tmtableId]}" var="detailData">
    <li>
      <span class="ui-li-heading">
        <c:out value="{detailData.startDate}"></c:out>
        <c:out value="-"></c:out>
        <c:out value="{detailData.endDate}"></c:out>
      </span>
      <span class="ui-li-aside"><c:out value="{detailData.companyAmount[1]}"></c:out></span>
      <div class="indent">
        <p class="kaiden-ui-overview-contents"><c:out value="{detailData.lodgeName}"></c:out></p>
        <p class="kaiden-ui-overview-contents-sub"><c:out value="{detailData.payTypeName}"></c:out></p>
      </div>
      <ul>
        <li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="{surfaceBack}"></c:out></a></li>
      </ul>
      <%-- 宿泊期間 --%>
      <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceLodgeTerm}"></c:out></p>
      <div class="indent">
        <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.startDate}"></c:out><c:out value="-"></c:out><c:out value="{detailData.endDate}"></c:out></label>

```

```

</div>

<%-- 宿泊先 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceLodge}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.lodgeName}"></c:out></label>
</div>

<%-- 宿泊先TEL --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceLodgeTelephoneNumber}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.lodgeTelephoneNumber}"></c:out></label>
</div>

<%-- 経費区分 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceExpType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.expTypeName}"></c:out></label>
</div>

<%-- 勘定科目 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceAccount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.accountName}"></c:out></label>
</div>

<%-- 税区分 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTaxType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.taxTypeName}"></c:out></label>
</div>

<%-- 金額 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTransAmount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.transAmount[1]}"></c:out></strong></label>
</div>

<%-- 支払分類 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.payTypeName}"></c:out></label>
</div>

<%-- 支払先 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayee}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="\${detailData.payeeName}"></c:out></label>
</div>

<%-- 部門 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceDepartment}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.departmentName}"></c:out></label>
</div>

<%-- プロジェクト --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.projectName}"></c:out></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>

```


宿泊明細 (複数通貨)

宿泊明細 (複数通貨) のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   └── slcs
│   │   │       └── kaiden2
│   │           ├── product_workflow
│   │           │   ├── foundation
│   │           │   │   └── model
│   │           │       ├── entity
│   │           │       │   ├── K30tLodgeDetail.java
│   │           │       │   ├── K30tLodgeDetailNames.java
│   │           │       │   ├── Ka30tLodgeDetail.java
│   │           │       │   └── Ka30tLodgeDetailNames.java
│   │           │       └── service
│   │           │           ├── K30tLodgeDetailCondition.java
│   │           │           ├── K30tLodgeDetailsService.java
│   │           │           ├── Ka30tLodgeDetailCondition.java
│   │           │           └── Ka30tLodgeDetailsService.java
│   │   └── java
│   │       ├── jp
│   │       │   ├── co
│   │       │   │   └── slcs
│   │       │       └── kaiden2
│   │           ├── product_workflow
│   │           │   ├── feature
│   │           │   │   ├── action
│   │           │   │   │   ├── kaiden
│   │           │   │   │   │   └── gadget
│   │           │   │   │       ├── AbstractLodgeDetailActionBase.java
│   │           │   │   │       └── LodgeDetailV02Action.java
│   │           │   │   ├── conf
│   │           │   │   │   ├── kaiden
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── lodgedetail
│   │           │   │   │           └── LodgeDetailFieldName.java
│   │           │   │   ├── form
│   │           │   │   │   ├── kaiden
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── LodgeDetailV02Form.java
│   │           │   │   ├── logic
│   │           │   │   │   ├── kaiden
│   │           │   │   │   │   └── gadget
│   │           │   │   │       └── lodgedetail
│   │           │   │   │           ├── AbstractLodgeDetailLogicBase.java
│   │           │   │   │           └── LodgeDetailV02Logic.java
│   │           │   └── foundation
│   │           │       ├── manager
│   │           │       │   └── lodgedetail
│   │           │       │       ├── LodgeDetailManager.java
│   │           │       │       ├── LodgeDetailManagerDto.java
│   │           │       │       └── impl
│   │           │       │           └── LodgeDetailManagerImpl.java
│   │   └── resources
│   │       ├── META-INF
│   │       │   └── sql
│   │       │       ├── jp
│   │       │       │   ├── co
│   │       │       │   │   └── slcs
│   │       │       │       └── kaiden2
│   │       │       │   ├── product_workflow
│   │       │       │   │   ├── foundation
│   │       │       │   │   │   ├── manager
│   │       │       │   │   │   │   ├── lodgedetail
│   │       │       │   │   │   │   │   └── impl
│   │       │       │   │   │       └── LodgeDetailManagerImpl
│   │       │       │   │   │           selectForJournalSourceDtl.sql
│   │       │       │   │   ├── model
│   │       │       │   │   │   └── service
│   │       │       │   │       └── K30tLodgeDetailsService
│   │       │       │   │           ├── selectView.sql
│   │       │       │   │           └── selectViewForCopy.sql
│   │   └── webapp
│   │       ├── WEB-INF
│   │       │   └── view
│   │       │       └── kaiden2
│   │           ├── product_workflow
│   │           │   └── gadget
│   │           │       └── lodgeDetail
│   │           │           ├── lodgeDetailV02.jsp
│   │           │           └── lodgeDetailV02_sp.isp

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractLodgeDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 宿泊明細がジェットActionの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractLodgeDetailActionBase implements GadgetActionIF {

    /**
     * 対象ロジッククラスをサブクラスより返却。
     *
     * @return ロジッククラス
     */
    @Override
    public abstract PluginLogicIF getLogic();

    /**
     * Fromをサブクラスより返却。
     *
     * @return From
     */
    @Override
    public abstract AbstractImwPageForm getForm();

    /**
     * 申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード。
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード。
     *
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
    }

```

```

    return null;
}

/**
 * 処理画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcess() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirm() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 処理詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadProcessDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 画面表示用データを返却.
 *
 * @return Managerからの返却値
 */
@Override
public LogicResultDto getViewData() {
    return getLogic().view(getForm(), getConvertor());
}

/**
 * 参照詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadReferenceDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

/**
 * 確認詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadConfirmDetail() {
    ResponseWriter.writeJson(JSON.encode(getViewData()));
    return null;
}

```

```

}

/**
 * 申請書コピーのデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadCopy() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConvertor())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #LodgeDetailConvertor 宿泊明細(標準)ガジェットの書式変換}を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected ConvertForJsonSource getConvertor() {
    return new LodgeDetailConvertor(getForm());
}

/**
 * 宿泊明細ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class LodgeDetailConvertor extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;
    /**
     * NumberHelper.
     */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public LodgeDetailConvertor(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     */

```

```

* <tr>
* <td>換算額</td>
* <td>スケールを会社通貨のスケールに変換</td>
* </tr>
* </table>
* </p>
*/

@Override
protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

    Map<String, Object> result = new HashMap<String, Object>(map);

    // 取引通貨額の書式変換
    if (result.containsKey(LODGE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) result.get(LODGE_DETAIL_TRANS_AMOUNT);
        String currencyCd = (String) result.get(LODGE_DETAIL_TRANS_CURRENCY_CD);
        result.put(LODGE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, currencyCd));
    }
    // レートの書式変換
    if (result.containsKey(LODGE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(LODGE_DETAIL_COMPANY_RATE);
        result.put(LODGE_DETAIL_COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }
    // 換算額の書式変換
    if (result.containsKey(LODGE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(LODGE_DETAIL_COMPANY_AMOUNT);
        String companyCurrencyCd = (String) result.get(LODGE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(LODGE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>換算額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(LODGE_DETAIL_TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(LODGE_DETAIL_TRANS_AMOUNT);
        Object currencyCd = map.get(LODGE_DETAIL_TRANS_CURRENCY_CD);
        result.put(LODGE_DETAIL_TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }
    // レートの書式変換
    if (map.containsKey(LODGE_DETAIL_COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(LODGE_DETAIL_COMPANY_RATE);
        result.put(LODGE_DETAIL_COMPANY_RATE, numberHelper.formatRate4View(rate));
    }
    // 換算額の書式変換
    if (map.containsKey(LODGE_DETAIL_COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(LODGE_DETAIL_COMPANY_AMOUNT);
        Object companyCurrencyCd = result.get(LODGE_DETAIL_COMPANY_CURRENCY_CD);
        result.put(LODGE_DETAIL_COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, (String) companyCurrencyCd));
    }
    return result;
}
}
}

```

Action

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/LodgeDetailV02Action.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.LodgeDetailV02Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail.LodgeDetailV02Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 宿泊明細(標準・外貨)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV02Action extends AbstractLodgeDetailActionBase {

    /**
     * LodgeDetailV02Logic.
     */
    @Resource
    public LodgeDetailV02Logic lodgeDetailV02Logic;

    /**
     * LodgeDetailV02Form.
     */
    @ActionForm
    @Resource
    public LodgeDetailV02Form lodgeDetailV02Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return lodgeDetailV02Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return lodgeDetailV02Form;
    }
}
```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/LodgeDetailV02Form.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 宿泊明細(標準・外貨)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV02Form extends AbstractImwPageForm {

}
```

Logic(基底)

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/lodgedetail/AbstractLodgeDetailLogicBase.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
```

```

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_CLASS;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.GADGET_INSTANCE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SYSTEM_MATTER_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.USER_DATA_ID;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_ADDRESS;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_LODGE_TELEPHONE_NUMBER;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAYEE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.CompanySettingsHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DateValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.MaxLengthValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.common.ClaimExpenseAmountIF;

```



```

import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beat;
import org.seasar.util.beans.util.BeanMap;

/**
 * 宿泊明細ガジェット用Logicの抽象クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractLodgeDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロッカー: 宿泊明細ガジェット。 */
    protected static final String BLOCK_KEY_LODGE_DETAIL_TABLE = "lodgeDetailTable";

    /** 宿泊先名称。 */
    public static final int LODGE_NAME_LEN = 250;
    /** 宿泊先tel。 */
    public static final int LODGE_TELEPHONE_NUMBER_LEN = 250;
    /** 宿泊先住所。 */
    public static final int LODGE_ADDRESS_LEN = 250;
    /** 税率(整数部)。 */
    public static final int TAX_RATE_INT_LEN = 18;
    /** 税率(小数部)。 */
    public static final int TAX_RATE_DEC_LEN = 10;
    /** 会社通貨換算レート(整数部)。 */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 会社通貨換算レート(小数部)。 */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 支払先。 */
    public static final int PAYEE_NAME_LEN = 250;
    /** 数量(整数部)。 */
    public static final int QUANTITY_INT_LEN = 18;
    /** 数量(小数部)。 */
    public static final int QUANTITY_DEC_LEN = 10;

    /** LodgeDetailManager。 */
    @Resource
    protected LodgeDetailManager lodgeDetailManager;

    /** JournalSourceDtlManager。 */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** CompanySettingsHelper。 */
    @Resource
    protected CompanySettingsHelper companySettingsHelper;

    /** コンストラクタ。 */
    public AbstractLodgeDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 宿泊明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる宿泊明細テーブルを削除後に、宿泊明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
     * </p>
     */
    @Override

```

```

public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override

```

```

public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、宿泊明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細を登録します。
 * </p>
 */
@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細テーブルの対象データを削除します。<br>

```

```

* 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
* <p>
*/
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる宿泊明細テーブルを削除後に、宿泊明細テーブルを登録します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * <p>
 */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * <p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * <p>
 */
@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.imwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status) || LAST_RESULT_STATUS_APPROVEEND.equals(status)) {

        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 宿泊明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userId = pluginGadgetDTO.userId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 宿泊明細データを取得
            List<BeanMap> beanMapList = lodgeDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * <p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}

```

```

    * {@inheritDoc}
    * <p>
    * 対象となる宿泊明細の履歴を含めた全てのデータを削除します。
    * </p>
    */
    @Override
    public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる(過去案件)宿泊明細テーブルを一括削除します。
    * </p>
    */
    @Override
    public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = lodgeDetailManager
            .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().imwPluginParameterDto));
        return rv;
    }

    /**
    * {@inheritDoc}
    * <p>
    * 対象となる宿泊明細の履歴を含めた全てのデータを削除します。
    * </p>
    */
    @Override
    public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return deleteTableIncludeBackground();
    }

    /**
    * {@inheritDoc}
    *
    * @see #viewMapFormatted
    */
    @Override
    protected Map<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
        return viewMapFormatted(conditionDto, null);
    }

    /**
    * {@inheritDoc}
    * <p>
    * データの取得は、{@link LodgeDetailManager#selectList(LodgeDetailManagerDto)}を利用します。
    * </p>
    */
    @Override
    protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
        throws KaidenSystemException {

        LodgeDetailManagerDto dto = Beans.createAndCopy(LodgeDetailManagerDto.class, conditionDto).execute();
        List<LodgeDetailManagerDto> listSelect = lodgeDetailManager.selectList(dto);
        LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();
        List<Map<String, Object>> listDtoMap = new ArrayList<Map<String, Object>>();
        for (LodgeDetailManagerDto src : listSelect) {
            if (converter == null) {
                listDtoMap.add(BeansUtil.toMap(src, true));
            } else {
                listDtoMap.add(converter.mapConvert(BeansUtil.toMap(src, true)));
            }
        }
        rv.put(BLOCK_KEY_LODGE_DETAIL_TABLE, listDtoMap);
        return rv;
    }

    /**
    * 宿泊明細テーブルデータ登録処理。
    *
    * @return 更新カウント
    * @throws KaidenSystemException システムエラー
    */
    protected int insertTable() throws KaidenSystemException {
        // 宿泊明細ガジェットデータをMap形式で取得
        List<LodgeDetailManagerDto> dtos = JsonUtil.getDtoList(LodgeDetailManagerDto.class,
            getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE));

        // 宿泊明細テーブル登録
        return lodgeDetailManager.entry(dtos);
    }

    /**
    * 仕訳連携用明細テーブルデータ登録処理。
    *

```

```

    * @param srcList 宿泊明細テーブルデータリスト
    * @param matterEndProcessDto 案件終了処理用DTO
    * @return 更新カウント
    * @throws KaidenSystemException システムエラー
    */
    protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
        MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
        for (JournalSourceDtlManagerDto srcDto : srcList) {
            // 仕訳データ区分をセット
            srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
        }

        // 仕訳連携用明細テーブル登録
        return journalSourceDtlManager.entry(srcList);
    }

    /**
     * 宿泊明細テーブルデータ削除処理.
     *
     * @param tempFlg 一時保存データフラグ
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
        LodgeDetailManagerDto mngDto = new LodgeDetailManagerDto();
        mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
        mngDto.userDataId = super.getGadgetDto().userDataId;
        mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
        mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
        if (tempFlg) {
            mngDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
        }
        return lodgeDetailManager.remove(mngDto);
    }

    /**
     * 宿泊明細テーブルデータ削除処理(履歴を含む).
     *
     * @return 更新カウント
     * @throws KaidenSystemException システムエラー
     */
    protected int deleteTableIncludeBackground() throws KaidenSystemException {
        LodgeDetailManagerDto mngDto = new LodgeDetailManagerDto();
        mngDto.systemMatterId = super.getGadgetDto().systemMatterId;
        mngDto.userDataId = null;
        mngDto.gadgetClass = super.getGadgetDto().gadgetClass;
        mngDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
        return lodgeDetailManager.removeAll(mngDto);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *

```

```

    * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
    */
    @Override
    protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, false);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
            defineValidateForNormal(params, false);
        }
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override
    protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
        KaidenApplicationException {
        defineValidateForNormal(params, true);
    }

    /**
     * {@inheritDoc}
     *
     * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
     */
    @Override

```

```

protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>{@link DateValidator 日付型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">終了日</td>
 * <td>{@link DateValidator 日付型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link DateMinValidator 最小日付チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先名称</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先tel</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>宿泊先住所</td>
 * <td>{@link MaxLengthValidator 最大文字数チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>取引通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td rowspan="2">会社通貨換算レート</td>
 * <td>{@link DigitsValidator 小数型チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericMinValidator 最小数値チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
 * <td align="center">入力</td>

```



```

* </tr>
* <tr>
* <td>支払分類コード</td>
* <td>{@link PayTypeExistsValidator 支払分類マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払方法コード</td>
* <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>支払先名</td>
* <td>{@link MaxLengthValidator 最大文字数チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>数量</td>
* <td>{@link DigitsValidator 小数量チェック}</td>
* <td align="center">入力</td>
* </tr>
* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
KaidenApplicationException {

if (fieldDefsMap.containsKey(BLOCK_KEY_LODGE_DETAIL_TABLE)) {
addValidateDefsRequired(BLOCK_KEY_LODGE_DETAIL_TABLE, tmpFlg);
// 開始日
addValidateOnlyInput(LODGE_DETAIL_START_DATE, VALID_DATE);
// 終了日
addValidateOnlyInput(LODGE_DETAIL_END_DATE, VALID_DATE);
addValidateOnlyInput(LODGE_DETAIL_END_DATE, VALID_DATE_MIN, LODGE_DETAIL_START_DATE);
// 宿泊先名称
addValidateOnlyInput(LODGE_DETAIL_LODGE_NAME, VALID_MAX_LENGTH, getArgs(LODGE_NAME_LEN));
// 宿泊先tel
addValidateOnlyInput(LODGE_DETAIL_LODGE_TELEPHONE_NUMBER, VALID_MAX_LENGTH,
getArgs(LODGE_TELEPHONE_NUMBER_LEN));
// 宿泊先住所
addValidateOnlyInput(LODGE_DETAIL_LODGE_ADDRESS, VALID_MAX_LENGTH, getArgs(LODGE_ADDRESS_LEN));
// 経費区分コード
addValidate(LODGE_DETAIL_EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, LODGE_DETAIL_EXP_TYPE_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 勘定科目コード
addValidate(LODGE_DETAIL_ACCOUNT_CD, VALID_ACCOUNT_EXISTS, LODGE_DETAIL_ACCOUNT_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 税区分コード
addValidate(LODGE_DETAIL_TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, LODGE_DETAIL_TAX_TYPE_CD,
getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 取引通貨額
addValidateOnlyInput(LODGE_DETAIL_TRANS_AMOUNT, VALID_CURRENCY_AMNT,
getArgs(LODGE_DETAIL_TRANS_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
// 取引通貨コード
addValidate(LODGE_DETAIL_TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, LODGE_DETAIL_TRANS_CURRENCY_CD,
getGadgetDto().searchCriteriaCompany);
// 会社通貨換算レート
addValidateOnlyInput(LODGE_DETAIL_COMPANY_RATE, VALID_DIGITS,
getArgs(COMpany_RATE_INT_LEN, COMPANY_RATE_DEC_LEN));
addValidateOnlyInput(LODGE_DETAIL_COMPANY_RATE, VALID_NUMERIC_MIN, getArgs(0));
// 会社通貨コード
addValidate(LODGE_DETAIL_COMPANY_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS, LODGE_DETAIL_COMPANY_CURRENCY_CD,
getGadgetDto().searchCriteriaCompany);
// 会社通貨額
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));

```

```

// 支払分類コード
addValidate(LODGE_DETAIL_PAY_TYPE_CD, VALID_PAY_TYPE_EXISTS, LODGE_DETAIL_PAY_TYPE_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 支払方法コード
addValidate(LODGE_DETAIL_PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, LODGE_DETAIL_PAY_METHOD_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 支払先名
addValidateOnlyInput(LODGE_DETAIL_PAYEE_NAME, VALID_MAX_LENGTH, getArgs(PAYEE_NAME_LEN));
// 組織コード
addValidate(LODGE_DETAIL_DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS, getGadgetDto().searchCriteriaCompany,
    LODGE_DETAIL_DEPARTMENT_SET_CD, LODGE_DETAIL_DEPARTMENT_CD,
    dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// プロジェクトコード
addValidate(LODGE_DETAIL_PROJECT_CD, VALID_PROJECT_EXISTS, LODGE_DETAIL_PROJECT_CD,
    getGadgetDto().searchCriteriaCompany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
// 数量
addValidateOnlyInput(LODGE_DETAIL_QUANTITY, VALID_DIGITS, getArgs(QUANTITY_INT_LEN, QUANTITY_DEC_LEN));
// 会社通貨額合計
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
    getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, SEARCH_CRITERIA_COMPANY));
}
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidate(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMap(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * ブロックの共通シングルバリデート定義MAPにエントリを追加.
 * <p>
 * 当該フィールドが入力モードの時のみ、指定されたバリデート定義を追加します。
 * </p>
 */
* @param fieldKey フィールドキー
* @param validationId バリデートキー
* @param args バリデート実行時アーギュメント
*/
protected void addValidateOnlyInput(String fieldKey, String validationId, String... args) {
    super.addValidateDefsMapOnlyInput(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldKey, validationId, args);
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
 * 設定されている場合は、{@link LodgeDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, new ArrayList<LodgeDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectView(criteria));
    }
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link LodgeDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link LodgeDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>

```

```

*/
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_LODGE_DETAIL_TABLE, lodgeDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した宿泊明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_depaertment)</li>
 * <li>プロジェクトマスタ</li>
 * <li>支払分類マスタ</li>
 * <li>支払方法マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * </p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_LODGE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<LodgeDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), LodgeDetailManagerDto.class);
            for (LodgeDetailManagerDto j : sqlResultList) {
                // 取引通貨チェック
                if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_TRANS_CURRENCY_CD, LODGE_DETAIL_TRANS_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(j.transCurrencyCd, j.transCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 会社通貨チェック
                if (isNotEmpty(j.companyCurrencyCd) && isEmpty(j.companyCurrencyName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_COMPANY_CURRENCY_CD, LODGE_DETAIL_COMPANY_CURRENCY_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.CURRENCY);
                    String[] values = getArgs(j.companyCurrencyCd, j.companyCurrencyName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 組織チェック
                if (isNotEmpty(j.departmentSetCd) && isEmpty(j.departmentCd) && isEmpty(j.departmentName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_DEPARTMENT_SET_CD, LODGE_DETAIL_DEPARTMENT_CD,
                        LODGE_DETAIL_DEPARTMENT_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        BaseProp.S.DEPARTMENT);
                    String[] values = getArgs(j.departmentSetCd, j.departmentCd, j.departmentName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // プロジェクトチェック
                if (isNotEmpty(j.projectCd) && isEmpty(j.projectName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_PROJECT_CD, LODGE_DETAIL_PROJECT_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        ProductWorkflowProp.S.PROJECT);
                    String[] values = getArgs(j.projectCd, j.projectName);
                    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
                    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
                        .toString(j.tupleId), masterName, values, message));
                }

                // 支払分類マスタチェック
                if (isNotEmpty(j.payTypeCd) && isEmpty(j.payTypeName)) {
                    String[] fieldId = getArgs(LODGE_DETAIL_PAY_TYPE_CD, LODGE_DETAIL_PAY_TYPE_NAME);
                    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
                        ProductWorkflowProp.S.PAY_TYPE);
                    String[] values = getArgs(j.payTypeCd, j.payTypeName);

```

```

String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
    .toString(j.tupleId), masterName, values, message));
}

// 支払方法マスタチェック
if (isNotEmpty(j.payMethodCd) && isEmpty(j.payMethodName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_PAY_METHOD_CD, LODGE_DETAIL_PAY_METHOD_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PAY_METHOD);
    String[] values = getArgs(j.payMethodCd, j.payMethodName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 経費区分マスタチェック
if (isNotEmpty(j.expTypeCd) && isEmpty(j.expTypeName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_EXP_TYPE_CD, LODGE_DETAIL_EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(j.expTypeCd, j.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        j.accountCd = null;
        j.accountName = null;
        j.taxTypeCd = null;
        j.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 勘定科目マスタチェック
if (isNotEmpty(j.accountCd) && isEmpty(j.accountName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_ACCOUNT_CD, LODGE_DETAIL_ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(j.accountCd, j.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}

// 税区分マスタチェック
if (isNotEmpty(j.taxTypeCd) && isEmpty(j.taxTypeName)) {
    String[] fieldId = getArgs(LODGE_DETAIL_TAX_TYPE_CD, LODGE_DETAIL_TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(j.taxTypeCd, j.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 税率、税抜金額、税額をクリア
        j.taxRate = null;
        j.companyAmountExcTax = null;
        j.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_LODGE_DETAIL_TABLE, fieldId, StringUtil
        .toString(j.tupleId), masterName, values, message));
}
}
}
}
return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 使用していません
 * </p>
 *
 * @deprecated
 */
@Deprecated
@Override
protected List<String> getDtoPkDefList(String key) {
    List<String> list = new ArrayList<String>();

```

```

list.add(SYSTEM_MATTER_ID);
list.add(USER_DATA_ID);
list.add(GADGET_CLASS);
list.add(GADGET_INSTANCE);
return list;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccCountIF#getClaimExpenccCount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccCount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccCountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG));
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOff(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strPaymentFlag = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG));
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (GenericFlag.isOn(strPaymentFlag)) {
            if (StringUtil.isNotEmpty(strAmnt)) {
                sum = sum.add(new BigDecimal(strAmnt));
            }
        }
    }

    retMap.put(ClaimTemporaryPayAmountIF.RTN_KEY_CLAIM_TEMPORARY_PAY_AMOUNT, sum);
}

```

```

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();

    List<BigDecimal> list = new ArrayList<BigDecimal>();

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            list.add(new BigDecimal(strAmnt));
        }
    }
    retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 宿泊明細テーブルの対象データを過去案件回避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    @SuppressWarnings("unchecked")
    List<LodgeDetailManagerDto> list = (List<LodgeDetailManagerDto>) dataMap.get(BLOCK_KEY_LODGE_DETAIL_TABLE);
    int rv = lodgeDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(lodgeDetailManager);
    return list;
}
}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/lodgedetail/LodgeDetailV02Logic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.lodgedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;

```

```

import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAYMENT_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PAY_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail.LodgeDetailFieldName.LODGE_DETAIL_TRANS_CURRENCY_CD;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManagerDto;

/**
 * 宿泊明細(標準・外貨)ガジェット(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailV02Logic extends AbstractLodgeDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public LodgeDetailV02Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します。<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力</td>
     * </tr>
     * </table>
     * </p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {
        super.defineValidateForNormal(params, tmpFlg);
    }

```

```

// 取引通貨額
addValidateOnlyInput(LODGE_DETAIL_TRANS_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));
// 会社通貨額
addValidateOnlyInput(LODGE_DETAIL_COMPANY_AMOUNT, VALID_NUMERIC_MIN, getArgs(0));

if (!tmpFlg) {
    // 組織コード
    addValidateOnlyInput(LODGE_DETAIL_DEPARTMENT_CD, VALID_REQUIRED_ANY, LODGE_DETAIL_DEPARTMENT_CD,
        LODGE_DETAIL_PROJECT_CD);
    // プロジェクトコード
    addValidateOnlyInput(LODGE_DETAIL_PROJECT_CD, VALID_REQUIRED_ANY, LODGE_DETAIL_DEPARTMENT_CD,
        LODGE_DETAIL_PROJECT_CD);
}
}

/**
 * {@inheritDoc}
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード:空の場合に{@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>支払方法コード:空の場合にガジェット関連携項目. 支払方法コードを設定</li>
 * <li>数量:固定値1を設定</li>
 * </ul>
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(LODGE_DETAIL_COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 数量
            j.put(LODGE_DETAIL_QUANTITY, "1");

            // 立替フラグ
            if (StringUtil.isNotEmpty((String) j.get(LODGE_DETAIL_PAY_TYPE_CD))) {
                PayTypeManagerDto criteria = new PayTypeManagerDto();
                criteria.companyCd = (String) j.get(SEARCH_CRITERIA_COMPANY);
                criteria.payTypeCd = (String) j.get(LODGE_DETAIL_PAY_TYPE_CD);
                criteria.searchCriteriaDate = dateHelper.toTimestamp((String) j
                    .get(ImwEntityCommonFieldNames.SEARCH_CRITERIA_DATE));
                List<PayTypeManagerDto> retList = payTypeManager.selectListIncludesSearchDateIgnoreAuthCompany(
                    criteria, new ArrayList<OrderDto>());
                if (CollectionUtil.isNotEmpty(retList)) {
                    j.put(LODGE_DETAIL_PAYMENT_FLAG, retList.get(0).paymentFlag);
                }
            }
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();

```



```

List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * </p>
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount(((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(LODGE_DETAIL_TRANS_CURRENCY_CD), (String) j.get(LODGE_DETAIL_TRANS_AMOUNT),
                (String) j.get(LODGE_DETAIL_COMPANY_RATE))));
            j.put(LODGE_DETAIL_COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(LODGE_DETAIL_COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum(((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(LODGE_DETAIL_QUANTITY))));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>支払方法コード:ガジェット関連連携項目. 支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_LODGE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード
            if (gcPayMethodCd == null) {
                gcPayMethodCd = getGadgetCouplingPayMethod();
            }
            j.put(LODGE_DETAIL_PAY_METHOD_CD, gcPayMethodCd);
        }
    }
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenccCount(Map<String, Object> params) {
    return super.putClaimExpenccCount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

```

```

}

/** {@inheritDoc} */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/lodgedetail/LodgeDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.lodgedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 宿泊明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class LodgeDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected LodgeDetailFieldName() {

        /** 開始日. */
        public static final String LODGE_DETAIL_START_DATE = "startDate";
        /** 終了日. */
        public static final String LODGE_DETAIL_END_DATE = "endDate";
        /** 宿泊先名称. */
        public static final String LODGE_DETAIL_LODGE_NAME = "lodgeName";
        /** 宿泊先tel. */
        public static final String LODGE_DETAIL_LODGE_TELEPHONE_NUMBER = "lodgeTelephoneNumber";
        /** 宿泊先住所. */
        public static final String LODGE_DETAIL_LODGE_ADDRESS = "lodgeAddress";
        /** 経費区分コード. */
        public static final String LODGE_DETAIL_EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String LODGE_DETAIL_ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String LODGE_DETAIL_TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String LODGE_DETAIL_TAX_RATE = "taxRate";
        /** 数量. */
        public static final String LODGE_DETAIL_QUANTITY = "quantity";
        /** 取引通貨コード. */
        public static final String LODGE_DETAIL_TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String LODGE_DETAIL_TRANS_AMOUNT = "transAmount";
        /** 会社通貨コード. */
        public static final String LODGE_DETAIL_COMPANY_CURRENCY_CD = "companyCurrencyCd";
        /** 会社通貨換算レート. */
        public static final String LODGE_DETAIL_COMPANY_RATE = "companyRate";
        /** 会社通貨額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT = "companyAmount";
        /** 会社通貨税抜額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
        /** 会社通貨税額. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_TAX = "companyAmountTax";
        /** 会社通貨額合計. */
        public static final String LODGE_DETAIL_COMPANY_AMOUNT_SUM = "companyAmountSum";
        /** 支払分類コード. */
        public static final String LODGE_DETAIL_PAY_TYPE_CD = "payTypeCd";
        /** 支払方法コード. */
        public static final String LODGE_DETAIL_PAY_METHOD_CD = "payMethodCd";
        /** 支払先コード. */
        public static final String LODGE_DETAIL_PAYEE_CD = "payeeCd";
        /** 支払先名. */
        public static final String LODGE_DETAIL_PAYEE_NAME = "payeeName";
        /** 組織セットコード. */
        public static final String LODGE_DETAIL_DEPARTMENT_SET_CD = "departmentSetCd";
        /** 組織コード. */
        public static final String LODGE_DETAIL_DEPARTMENT_CD = "departmentCd";
        /** プロジェクトコード. */
        public static final String LODGE_DETAIL_PROJECT_CD = "projectCd";
        /** 会社名. */
        public static final String LODGE_DETAIL_COMPANY_NAME = "companyName";
        /** 経費区分名称. */
        public static final String LODGE_DETAIL_EXP_TYPE_NAME = "expTypeName";
    }
}

```

```

public static final String LODGE_DETAIL_EXP_TYPE_NAME = expTypevname ;
/** 勘定科目名称. */
public static final String LODGE_DETAIL_ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String LODGE_DETAIL_TAX_TYPE_NAME = "taxTypeName";
/** 取引通貨名称. */
public static final String LODGE_DETAIL_TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String LODGE_DETAIL_COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 支払分類名称. */
public static final String LODGE_DETAIL_PAY_TYPE_NAME = "payTypeName";
/** 支払方法名称. */
public static final String LODGE_DETAIL_PAY_METHOD_NAME = "payMethodName";
/** 組織セット名称. */
public static final String LODGE_DETAIL_DEPARTMENT_SET_NAME = "departmentSetName";
/** 組織名称. */
public static final String LODGE_DETAIL_DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String LODGE_DETAIL_PROJECT_NAME = "projectName";
/** 立替フラグ. */
public static final String LODGE_DETAIL_PAYMENT_FLAG = "paymentFlag";
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/LodgeDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 宿泊明細テーブルマネージャクラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface LodgeDetailManager extends PluginGadgetManager<LodgeDetailManagerDto> {

    /** {@link LodgeDetailManager#selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link LodgeDetailManager#selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link LodgeDetailManager#selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager(実装)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/impl/LodgeDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail.LodgeDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tLodgeDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tLodgeDetailService;
import org.seasar.util.beans.util.BeanMap;

/**

```

```

* 宿泊明細テーブルへの各種操作を提供するAPI.
* 本APIは、主にデータベース操作を行います、より業務的な機能を付加した操作を提供します。<br>
* 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
* {@link LodgeDetailManagerDto}リストが戻り値として返却されます。<br>
* <br>
* 更新系機能を使用する際には、{@link LodgeDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
* 処理件数を数値型で返却します。<br>
* <br>
* <b>機能一覧(共通)</b>
* <table border="1">
* <tr>
* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>主キー検索</td>
* <td>selectByPk</td>
* <td>主キー検索</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>リスト検索</td>
* <td>selectList</td>
* <td>主キー検索</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>表示案件検索</td>
* <td>selectView</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>コピー案件検索</td>
* <td>selectViewForCopy</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件検索</td>
* <td>selectViewArchive</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括挿入</td>
* <td>entry</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括更新</td>
* <td>renew</td>
* <td>更新</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括削除</td>
* <td>removeAll</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>過去案件登録</td>
* <td>matterArchiveRegister</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件処理時削除</td>
* <td>matterArchivePostDelete</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件一括削除</td>
* <td>removeAllArchive</td>
* <td>削除</td>

```

[illegible]

[illegible]

```

/
@Override
public List<LodgeDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
    return super.selectListForViewForArchive(ka30tLodgeDetailService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<LodgeDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.insertBatch(k30tLodgeDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int renew(List<LodgeDetailManagerDto> dataList) {
    return CollectionUtil.summaryInt(super.renewBatch(k30tLodgeDetailService, dataList));
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(LodgeDetailManagerDto data) {
    return super.deleteUsedUserDataId(k30tLodgeDetailService, data);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeAll(LodgeDetailManagerDto data) {
    return super.deleteNoUsedUserDataId(k30tLodgeDetailService, data);
}

/**
 * {@inheritDoc}
 * <p>
 * ■ソート仕様<br>
 * <table border="1">
 * <tr>
 * <td>フィールド論理名</td>
 * <td>フィールド名</td>
 * <td>ソート順</td>
 * <td>行番号</td>
 * <td>rowNumber</td>
 * <td>ASC</td>
 * </tr>
 * </table>
 * </p>
 */
@Override
public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
    List<OrderDto> order = new ArrayList<OrderDto>();
    order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

    return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchiveRegister(CriteriaForSqlDto criteria, List<LodgeDetailManagerDto> dataList) {
    int rv = super.entryArchive(ka30tLodgeDetailService, criteria, dataList);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
    int rv = super.deleteNoUsedUserDataId(k30tLodgeDetailService, criteria);
    return rv;
}

/**
 * {@inheritDoc}

```

```

    {commentDoc}
    */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        int rv = super.deleteNoUsedUserDataIdForArchive(ka30tLodgeDetailService, criteria);
        return rv;
    }
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/LodgeDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.lodgedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 宿泊明細テーブルマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class LodgeDetailManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザーデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstanceId;
    /** タブルid. */
    public Long tupleId;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバリエーション. */
    public String gadgetVariation;
    /** 行番号. */
    public Long rowNumber;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 宿泊先名称. */
    public String lodgeName;
    /** 宿泊先tel. */
    public String lodgeTelephoneNumber;
    /** 宿泊先住所. */
    public String lodgeAddress;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** 税率. */
    public BigDecimal taxRate;
    /** 数量. */
    public BigDecimal quantity;
    /** 取引通貨コード. */
    public String transCurrencyCd;
    /** 取引通貨額. */
    public BigDecimal transAmount;
    /** 会社通貨コード. */
    public String companyCurrencyCd;
    /** 会社通貨換算レート. */
    public BigDecimal companyRate;
    /** 会社通貨額. */
    public BigDecimal companyAmount;
    /** 会社通貨税抜額. */
    public BigDecimal companyAmountExcTax;
    /** 会社通貨税額. */
    public BigDecimal companyAmountTax;
    /** 会社通貨額合計. */
    public BigDecimal companyAmountSum;
    /** 支払分類コード. */
    public String payTypeCd;
    /** 支払方法コード. */
    public String payMethodCd;
    /** 支払先コード. */
    public String payeeCd;
    /** 支払先名称. */

```



```

    /** 支払元名. */
    public String payeeName;
    /** 組織セットコード. */
    public String departmentSetCd;
    /** 組織コード. */
    public String departmentCd;
    /** プロジェクトコード. */
    public String projectCd;

    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 検索基準会社コード. */
    public String searchCriteriaCompany;
    /** 取引通貨名称. */
    public String transCurrencyName;
    /** 会社通貨名称. */
    public String companyCurrencyName;
    /** 組織名称. */
    public String departmentName;
    /** プロジェクトマスタプロジェクト名称. */
    public String projectName;
    /** 経費区分名称. */
    public String expTypeName;
    /** 勘定科目名称. */
    public String accountName;
    /** 税区分名称. */
    public String taxTypeName;
    /** 支払分類名称. */
    public String payTypeName;
    /** 支払方法名称. */
    public String payMethodName;
    /** 立替フラグ. */
    public String paymentFlag;
}

```

Java (S2JDBC)

Entity

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tLodgeDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

```

```
/**
```

```

* 宿泊明細テーブル。
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Entity
@Table(name = "k30t_lodge_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tLodgeDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;

    /** 行番号. */
    @Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long rowNumber;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
    public Timestamp endDate;

    /** 宿泊先名称. */
    @Column(name = "lodge_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeName;

    /** 宿泊先tel. */
    @Column(name = "lodge_telephone_number", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeTelephoneNumber;

    /** 宿泊先住所. */
    @Column(name = "lodge_address", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String lodgeAddress;

    /** 経費区分コード. */
    @Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String expTypeCd;

    /** 勘定科目コード. */
    @Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String accountCd;

    /** 税区分コード. */
    @Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String taxTypeCd;

    /** 税率. */
    @Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal taxRate;

    /** 数量. */

```

```

@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

```

```

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tLodgeDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件) 宿泊明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_lodge_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tLodgeDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** アーカイブ年月. */
    @Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
    public String archiveMonth;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバージョン. */

```

```

@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 開始日. */
@Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = true, unique = false)
public Timestamp endDate;

/** 宿泊先名称. */
@Column(name = "lodge_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeName;

/** 宿泊先tel. */
@Column(name = "lodge_telephone_number", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeTelephoneNumber;

/** 宿泊先住所. */
@Column(name = "lodge_address", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String lodgeAddress;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 数量. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

```

```

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tLodgeDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tLodgeDetail}の名称クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tLodgeDetailNames {

    /**
     * デフォルトコンストラクタ。
     */
    private K30tLodgeDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却。
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却。
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却。
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }
}

```

```

    return new PropertyName<Long>("tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public static PropertyName<Long> rowNumber() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public static PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>("startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public static PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>("endDate");
}

/**
 * lodgeNameのプロパティ名を返却.
 *
 * @return lodgeNameのプロパティ名
 */
public static PropertyName<String> lodgeName() {
    return new PropertyName<String>("lodgeName");
}

/**
 * lodgeTelephoneNumberのプロパティ名を返却.
 *
 * @return lodgeTelephoneNumberのプロパティ名
 */
public static PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>("lodgeTelephoneNumber");
}

/**
 * lodgeAddressのプロパティ名を返却.
 *
 * @return lodgeAddressのプロパティ名
 */
public static PropertyName<String> lodgeAddress() {
    return new PropertyName<String>("lodgeAddress");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *

```



```

    * @return accountCdのプロパティ名
    */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public static PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>("taxRate");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public static PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>("quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public static PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>("transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>("transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public static PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>("companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public static PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>("companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>("companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }
}

```

```

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名

```

```

    */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }
}

```

```

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30tLodgeDetailNames extends PropertyName<K30tLodgeDetail> {

    /**
     * インスタンスを構築.
     */
    public _K30tLodgeDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30tLodgeDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30tLodgeDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>(this, "systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.

```

```

*
* @return userDataIdのプロパティ名
*/
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
* gadgetClassのプロパティ名を返却.
*
* @return gadgetClassのプロパティ名
*/
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
* gadgetInstanceのプロパティ名を返却.
*
* @return gadgetInstanceのプロパティ名
*/
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
* tupleIdのプロパティ名を返却.
*
* @return tupleIdのプロパティ名
*/
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
* gadgetIdのプロパティ名を返却.
*
* @return gadgetIdのプロパティ名
*/
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
* gadgetVariationのプロパティ名を返却.
*
* @return gadgetVariationのプロパティ名
*/
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
* rowNumのプロパティ名を返却.
*
* @return rowNumのプロパティ名
*/
public PropertyName<Long> rowNum() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
* startDateのプロパティ名を返却.
*
* @return startDateのプロパティ名
*/
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
* endDateのプロパティ名を返却.
*
* @return endDateのプロパティ名
*/
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
* lodgeNameのプロパティ名を返却.
*
* @return lodgeNameのプロパティ名
*/
public PropertyName<String> lodgeName() {
    return new PropertyName<String>(this, "lodgeName");
}

```

```

}

/**
 * lodgeTelephoneNumberのプロパティ名を返却.
 *
 * @return lodgeTelephoneNumberのプロパティ名
 */
public PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>(this, "lodgeTelephoneNumber");
}

/**
 * lodgeAddressのプロパティ名を返却.
 *
 * @return lodgeAddressのプロパティ名
 */
public PropertyName<String> lodgeAddress() {
    return new PropertyName<String>(this, "lodgeAddress");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *

```

```

    * @return companyCurrencyCdのプロパティ名
    */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>(this, "payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>(this, "payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>(this, "payeeName");
    }
}

```

```

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */

```



```

    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(<this>, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(<this>, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(<this>, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(<this>, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(<this>, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(<this>, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(<this>, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(<this>, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(<this>, "renewUserCd");
    }
}

```

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tLodgeDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;

```

```

import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tLodgeDetail}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tLodgeDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tLodgeDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却.
     *
     * @return gadgetInstanceのプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却.
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * archiveMonthのプロパティ名を返却.
     *
     * @return archiveMonthのプロパティ名
     */
    public static PropertyName<String> archiveMonth() {
        return new PropertyName<String>("archiveMonth");
    }

    /**
     * gadgetIdのプロパティ名を返却.
     *
     * @return gadgetIdのプロパティ名
     */

```

```

public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumnerのプロパティ名を返却.
 *
 * @return rowNumnerのプロパティ名
 */
public static PropertyName<Long> rowNumner() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public static PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>("startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public static PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>("endDate");
}

/**
 * lodgeNameのプロパティ名を返却.
 *
 * @return lodgeNameのプロパティ名
 */
public static PropertyName<String> lodgeName() {
    return new PropertyName<String>("lodgeName");
}

/**
 * lodgeTelephoneNumberのプロパティ名を返却.
 *
 * @return lodgeTelephoneNumberのプロパティ名
 */
public static PropertyName<String> lodgeTelephoneNumber() {
    return new PropertyName<String>("lodgeTelephoneNumber");
}

/**
 * lodgeAddressのプロパティ名を返却.
 *
 * @return lodgeAddressのプロパティ名
 */
public static PropertyName<String> lodgeAddress() {
    return new PropertyName<String>("lodgeAddress");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**

```

```

    * taxTypeCdのプロパティ名を返却.
    *
    * @return taxTypeCdのプロパティ名
    */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public static PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>("taxRate");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public static PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>("quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public static PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>("transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>("transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public static PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>("companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public static PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>("companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>("companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountTax() {

```

```

    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public static PropertyName<String> expTypeName() {
    return new PropertyName<String>("expTypeName");
}

/**
 * accountNameのプロパティ名を返却.

```

```

 *
 * @return accountNameのプロパティ名
 */
public static PropertyName<String> accountName() {
    return new PropertyName<String>("accountName");
}

/**
 * taxTypeNameのプロパティ名を返却.
 *
 * @return taxTypeNameのプロパティ名
 */
public static PropertyName<String> taxTypeName() {
    return new PropertyName<String>("taxTypeName");
}

/**
 * transCurrencyNameのプロパティ名を返却.
 *
 * @return transCurrencyNameのプロパティ名
 */
public static PropertyName<String> transCurrencyName() {
    return new PropertyName<String>("transCurrencyName");
}

/**
 * companyCurrencyNameのプロパティ名を返却.
 *
 * @return companyCurrencyNameのプロパティ名
 */
public static PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>("companyCurrencyName");
}

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public static PropertyName<String> payTypeName() {
    return new PropertyName<String>("payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public static PropertyName<String> paymentFlag() {
    return new PropertyName<String>("paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public static PropertyName<String> payMethodName() {
    return new PropertyName<String>("payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public static PropertyName<String> departmentName() {
    return new PropertyName<String>("departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public static PropertyName<String> projectName() {
    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

```

```

}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名

```

```

    @return entryTsのプロパティ名
    */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _Ka30tLodgeDetailNames extends PropertyName<Ka30tLodgeDetail> {

        /**
         * インスタンスを構築.
         */
        public _Ka30tLodgeDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _Ka30tLodgeDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _Ka30tLodgeDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *
         * @return systemMatterIdのプロパティ名
         */
        public PropertyName<String> systemMatterId() {
            return new PropertyName<String>("systemMatterId");
        }

        /**
         * userDataIdのプロパティ名を返却.
         *
         * @return userDataIdのプロパティ名
         */
        public PropertyName<String> userDataId() {

```



```

public PropertyName<String> userData() {
    return new PropertyName<String>(this, "userData");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>(this, "archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumのプロパティ名を返却.
 *
 * @return rowNumのプロパティ名
 */
public PropertyName<Long> rowNum() {
    return new PropertyName<Long>(this, "rowNum");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.

```

```

    *
    * @return endDateのプロパティ名
    */
    public PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>(this, "endDate");
    }

    /**
     * lodgeNameのプロパティ名を返却.
     *
     * @return lodgeNameのプロパティ名
     */
    public PropertyName<String> lodgeName() {
        return new PropertyName<String>(this, "lodgeName");
    }

    /**
     * lodgeTelephoneNumberのプロパティ名を返却.
     *
     * @return lodgeTelephoneNumberのプロパティ名
     */
    public PropertyName<String> lodgeTelephoneNumber() {
        return new PropertyName<String>(this, "lodgeTelephoneNumber");
    }

    /**
     * lodgeAddressのプロパティ名を返却.
     *
     * @return lodgeAddressのプロパティ名
     */
    public PropertyName<String> lodgeAddress() {
        return new PropertyName<String>(this, "lodgeAddress");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public PropertyName<String> accountCd() {
        return new PropertyName<String>(this, "accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public PropertyName<String> taxTypeCd() {
        return new PropertyName<String>(this, "taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

```

```

}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *

```

```

    * @return payeeCdのプロパティ名
    */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>(this, "payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>(this, "payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.
     *
     * @return departmentSetCdのプロパティ名
     */
    public PropertyName<String> departmentSetCd() {
        return new PropertyName<String>(this, "departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却.
     *
     * @return departmentCdのプロパティ名
     */
    public PropertyName<String> departmentCd() {
        return new PropertyName<String>(this, "departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>(this, "projectCd");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public PropertyName<String> expTypeName() {
        return new PropertyName<String>(this, "expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却.
     *
     * @return accountNameのプロパティ名
     */
    public PropertyName<String> accountName() {
        return new PropertyName<String>(this, "accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却.
     *
     * @return taxTypeNameのプロパティ名
     */
    public PropertyName<String> taxTypeName() {
        return new PropertyName<String>(this, "taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却.
     *
     * @return transCurrencyNameのプロパティ名
     */
    public PropertyName<String> transCurrencyName() {
        return new PropertyName<String>(this, "transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却.
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>(this, "companyCurrencyName");
    }
}

```

```

/**
 * payTypeNameのプロパティ名を返却.
 *
 * @return payTypeNameのプロパティ名
 */
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
 * paymentFlagのプロパティ名を返却.
 *
 * @return paymentFlagのプロパティ名
 */
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名

```

```

*/
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却。
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却。
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

```

```

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>("this, "renewUserCd");
}
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tLodgeDetail;

/**
 * {@link K30tLodgeDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tLodgeDetailService extends GenerateService<K30tLodgeDetail> {
    //
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tLodgeDetailService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tLodgeDetail;

/**
 * {@link Ka30tLodgeDetail}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tLodgeDetailService extends GenerateService<Ka30tLodgeDetail> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30tLodgeDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tLodgeDetailCondition extends AbstractEntityCondition<K30tLodgeDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public K30tLodgeDetailCondition() {
    }
}

```

```

/**
 * インスタンスを構築.
 *
 * @param prefix プレフィックス
 * @param where 検索条件
 */
public K30tLodgeDetailCondition(String prefix, ComplexWhere where) {
    super(prefix, where);
}

/** systemMatterIdの条件. */
public NotNullableStringCondition<K30tLodgeDetailCondition> systemMatterId =
    new NotNullableStringCondition<K30tLodgeDetailCondition>("systemMatterId", this);

/** userDataIdの条件. */
public NotNullableStringCondition<K30tLodgeDetailCondition> userDataId =
    new NotNullableStringCondition<K30tLodgeDetailCondition>("userDataId", this);

/** gadgetClassの条件. */
public NotNullableStringCondition<K30tLodgeDetailCondition> gadgetClass =
    new NotNullableStringCondition<K30tLodgeDetailCondition>("gadgetClass", this);

/** gadgetInstanceの条件. */
public NotNullableCondition<K30tLodgeDetailCondition, Long> gadgetInstance =
    new NotNullableCondition<K30tLodgeDetailCondition, Long>("gadgetInstance", this);

/** tupleIdの条件. */
public NotNullableCondition<K30tLodgeDetailCondition, Long> tupleId =
    new NotNullableCondition<K30tLodgeDetailCondition, Long>("tupleId", this);

/** gadgetIdの条件. */
public NotNullableStringCondition<K30tLodgeDetailCondition> gadgetId =
    new NotNullableStringCondition<K30tLodgeDetailCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NotNullableStringCondition<K30tLodgeDetailCondition> gadgetVariation =
    new NotNullableStringCondition<K30tLodgeDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NotNullableCondition<K30tLodgeDetailCondition, Long> rowNumber =
    new NotNullableCondition<K30tLodgeDetailCondition, Long>("rowNumber", this);

/** startDateの条件. */
public NullableCondition<K30tLodgeDetailCondition, Timestamp> startDate =
    new NullableCondition<K30tLodgeDetailCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NullableCondition<K30tLodgeDetailCondition, Timestamp> endDate =
    new NullableCondition<K30tLodgeDetailCondition, Timestamp>("endDate", this);

/** lodgeNameの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> lodgeName =
    new NullableStringCondition<K30tLodgeDetailCondition>("lodgeName", this);

/** lodgeTelephoneNumberの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> lodgeTelephoneNumber =
    new NullableStringCondition<K30tLodgeDetailCondition>("lodgeTelephoneNumber", this);

/** lodgeAddressの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> lodgeAddress =
    new NullableStringCondition<K30tLodgeDetailCondition>("lodgeAddress", this);

/** expTypeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> expTypeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> accountCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> taxTypeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> taxRate =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> quantity =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> transCurrencyCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("transCurrencyCd", this);

```



```

/** transAmountの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> transAmount =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> companyCurrencyCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyRate =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<K30tLodgeDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<K30tLodgeDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payTypeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payMethodCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payeeCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> payeeName =
    new NullableStringCondition<K30tLodgeDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> departmentSetCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> departmentCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> projectCd =
    new NullableStringCondition<K30tLodgeDetailCondition>("projectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionA =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionB =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionC =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionD =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionE =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionF =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionG =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30tLodgeDetailCondition> extensionH =
    new NullableStringCondition<K30tLodgeDetailCondition>("extensionH", this);

```

```

    new NullableStringCondition<K30tLodgeDetailCondition>( "extensionI", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionI =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tLodgeDetailCondition> extensionJ =
        new NullableStringCondition<K30tLodgeDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Timestamp> entryTs =
        new NotNullableCondition<K30tLodgeDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullableStringCondition<K30tLodgeDetailCondition> entryUserCd =
        new NotNullableStringCondition<K30tLodgeDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Long> renewCnt =
        new NotNullableCondition<K30tLodgeDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullableCondition<K30tLodgeDetailCondition, Timestamp> renewTs =
        new NotNullableCondition<K30tLodgeDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullableStringCondition<K30tLodgeDetailCondition> renewUserCd =
        new NotNullableStringCondition<K30tLodgeDetailCondition>("renewUserCd", this);
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tLodgeDetailCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;

/**
 * {@link Ka30tLodgeDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl"})
public final class Ka30tLodgeDetailCondition extends
    AbstractEntityCondition<Ka30tLodgeDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tLodgeDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tLodgeDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullableStringCondition<Ka30tLodgeDetailCondition> systemMatterId =
        new NotNullableStringCondition<Ka30tLodgeDetailCondition>("systemMatterId", this);

    /** userDataIdの条件. */
    public NotNullableStringCondition<Ka30tLodgeDetailCondition> userDataId =
        new NotNullableStringCondition<Ka30tLodgeDetailCondition>("userDataId", this);

    /** gadgetClassの条件. */
    public NotNullableStringCondition<Ka30tLodgeDetailCondition> gadgetClass =
        new NotNullableStringCondition<Ka30tLodgeDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NotNullableCondition<Ka30tLodgeDetailCondition, Long> gadgetInstance =
        new NotNullableCondition<Ka30tLodgeDetailCondition, Long>("gadgetInstance", this);
}

```

```

/** tupleIdの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Long> tupleId =
    new NullableCondition<Ka30tLodgeDetailCondition, Long>("tupleId", this);

/** localeIdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> localeId =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("localeId", this);

/** archiveMonthの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> archiveMonth =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("archiveMonth", this);

/** gadgetIdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> gadgetId =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("gadgetId", this);

/** gadgetVariationの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> gadgetVariation =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("gadgetVariation", this);

/** rowNumberの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Long> rowNumber =
    new NullableCondition<Ka30tLodgeDetailCondition, Long>("rowNumber", this);

/** startDateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Timestamp> startDate =
    new NullableCondition<Ka30tLodgeDetailCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, Timestamp> endDate =
    new NullableCondition<Ka30tLodgeDetailCondition, Timestamp>("endDate", this);

/** lodgeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeName", this);

/** lodgeTelephoneNumberの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeTelephoneNumber =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeTelephoneNumber", this);

/** lodgeAddressの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> lodgeAddress =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("lodgeAddress", this);

/** expTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> accountCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("taxRate", this);

/** quantityの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("quantity", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */

```

```

public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tLodgeDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tLodgeDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payeeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> projectCd =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("expTypeName", this);

/** accountNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> accountName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> departmentName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("departmentName", this);

/** projectNameの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> projectName =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionA =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionB =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionB", this);

```

```

/** extensionCの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionC =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionD =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionE =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionF =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionG =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionH =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionI =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tLodgeDetailCondition> extensionJ =
    new NullableStringCondition<Ka30tLodgeDetailCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<Ka30tLodgeDetailCondition, Timestamp> entryTs =
    new NotNullableCondition<Ka30tLodgeDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tLodgeDetailCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tLodgeDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<Ka30tLodgeDetailCondition, Long> renewCnt =
    new NotNullableCondition<Ka30tLodgeDetailCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullableCondition<Ka30tLodgeDetailCondition, Timestamp> renewTs =
    new NotNullableCondition<Ka30tLodgeDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<Ka30tLodgeDetailCondition> renewUserCd =
    new NotNullableStringCondition<Ka30tLodgeDetailCondition>("renewUserCd", this);
}

```

SQL

検索

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailService/selectView.sql

```

select
    t.system_matter_id,
    t.user_data_id,
    t.gadget_class,
    t.gadget_instance,
    t.tuple_id,
    t.gadget_id,
    t.gadget_variation,
    t.row_number,
    t.start_date,
    t.end_date,
    t.lodge_name,
    t.lodge_telephone_number,
    t.lodge_address,
    t.exp_type_cd,
    t.account_cd,
    t.tax_type_cd,
    t.tax_rate,
    t.quantity,
    t.trans_currency_cd,
    t.trans_amount,
    t.company_currency_cd,
    t.company_rate,

```

```

t.company_amount,
t.company_amount_exc_tax,
t.company_amount_tax,
t.company_amount_sum,
t.pay_type_cd,
t.pay_method_cd,
t.payee_cd,
t.payee_name,
t.department_set_cd,
t.department_cd,
t.project_cd,
t.extension_a,
t.extension_b,
t.extension_c,
t.extension_d,
t.extension_e,
t.extension_f,
t.extension_g,
t.extension_h,
t.extension_i,
t.extension_j,
t.entry_ts,
t.entry_user_cd,
t.renew_cnt,
t.renew_ts,
t.renew_user_cd,
c1.search_criteria_company,
m1.currency_name as trans_currency_name,
m2.department_name,
m3.project_name,
m4.pay_type_name,
m5.pay_method_name,
m6.currency_name as company_currency_name,
m7.exp_type_name,
m8.account_name,
m9.tax_type_name,
m4.payment_flag
from
k30t_lodge_detail t
inner join
k20c_matter_control c1
on
t.system_matter_id = c1.system_matter_id
and
t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
c1.search_criteria_company = m1.company_cd
and
t.trans_currency_cd = m1.currency_cd
and
/*localeld*/ja' = m1.locale_id
and
m1.delete_flag = '0'
left outer join
imm_department m2
on
c1.search_criteria_company = m2.company_cd
and
t.department_set_cd = m2.department_set_cd
and
t.department_cd = m2.department_cd
and
/*localeld*/ja' = m2.locale_id
and
c1.search_criteria_date >= m2.start_date
and
c1.search_criteria_date < m2.end_date
and
m2.delete_flag = '0'
left outer join
k30m_project m3
on
c1.search_criteria_company = m3.company_cd
and
t.project_cd = m3.project_cd
and
c1.search_criteria_date >= m3.start_date
and
c1.search_criteria_date < m3.end_date
and
/*localeld*/ja' = m3.locale_id
and
m3.delete_flag = '0'
left outer join
k30m_pay_type m4

```

```

on
  c1.search_criteria_company = m4.company_cd
and
  t.pay_type_cd = m4.pay_type_cd
and
  /*localeld*/ja' = m4.locale_id
and
  c1.search_criteria_date >= m4.start_date
and
  c1.search_criteria_date < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeld*/ja' = m5.locale_id
and
  c1.search_criteria_date >= m5.start_date
and
  c1.search_criteria_date < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeld*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_exp_type m7
on
  c1.search_criteria_company = m7.company_cd
and
  t.exp_type_cd = m7.exp_type_cd
and
  /*localeld*/ja' = m7.locale_id
and
  c1.search_criteria_date >= m7.start_date
and
  c1.search_criteria_date < m7.end_date
and
  m7.delete_flag = '0'

left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd
and
  t.account_cd = m8.account_cd
and
  /*localeld*/ja' = m8.locale_id
and
  c1.search_criteria_date >= m8.start_date
and
  c1.search_criteria_date < m8.end_date
and
  m8.delete_flag = '0'

left outer join
  k30m_tax_type m9
on
  c1.search_criteria_company = m9.company_cd
and
  t.tax_type_cd = m9.tax_type_cd
and
  /*localeld*/ja' = m9.locale_id
and
  c1.search_criteria_date >= m9.start_date
and
  c1.search_criteria_date < m9.end_date
and
  m9.delete_flag = '0'
where
  t.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
  /*END*/
  /*IF gadgetId != null*/
and

```

```

and
  t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

コピー

- Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tLodgeDetailsService/selectViewForCopy.sql

```

select
  t.system_matter_id,
  t.user_data_id,
  t.gadget_class,
  t.gadget_instance,
  t.tuple_id,
  t.gadget_id,
  t.gadget_variation,
  t.row_number,
  t.start_date,
  t.end_date,
  t.lodge_name,
  t.lodge_telephone_number,
  t.lodge_address,
  t.exp_type_cd,
  t.account_cd,
  t.tax_type_cd,
  t.tax_rate,
  t.quantity,
  t.trans_currency_cd,
  t.trans_amount,
  t.company_currency_cd,
  t.company_rate,
  t.company_amount,
  t.company_amount_exc_tax,
  t.company_amount_tax,
  t.company_amount_sum,
  t.pay_type_cd,
  t.pay_method_cd,
  t.payee_cd,
  t.payee_name,
  t.department_set_cd,
  t.department_cd,
  t.project_cd,
  t.extension_a,
  t.extension_b,
  t.extension_c,
  t.extension_d,
  t.extension_e,
  t.extension_f,
  t.extension_g,
  t.extension_h,
  t.extension_i,
  t.extension_j,
  t.entry_ts,
  t.entry_user_cd,
  t.renew_cnt,
  t.renew_ts,
  t.renew_user_cd,
  c1.search_criteria_company,
  m1.currency_name as trans_currency_name,
  m2.department_name,
  m3.project_name,
  m4.pay_type_name,
  m5.pay_method_name,
  m6.currency_name as company_currency_name,
  ma.exp_type_name,
  case when t.exp_type_cd is not null and t.exp_type_cd <> '' then m8.account_name else mb.account_name end as account_name,
  case when t.exp_type_cd is not null and t.exp_type_cd <> '' then m9.tax_type_name else mc.tax_type_name end as tax_type_name,
  m4.payment_flag
from
  lodge_detail

```



```

k30t_lodge_detail t
inner join
  k20c_matter_control c1
on
  t.system_matter_id = c1.system_matter_id
and
  t.user_data_id = c1.user_data_id
left outer join imm_currency m1
on
  c1.search_criteria_company = m1.company_cd
and
  t.trans_currency_cd = m1.currency_cd
and
  /*localeId*/ja' = m1.locale_id
and
  m1.delete_flag = '0'
left outer join
  imm_department m2
on
  c1.search_criteria_company = m2.company_cd
and
  t.department_set_cd = m2.department_set_cd
and
  t.department_cd = m2.department_cd
and
  /*localeId*/ja' = m2.locale_id
and
  /*searchCriteriaDate*/" >= m2.start_date
and
  /*searchCriteriaDate*/" < m2.end_date
and
  m2.delete_flag = '0'
left outer join
  k30m_project m3
on
  c1.search_criteria_company = m3.company_cd
and
  t.project_cd = m3.project_cd
and
  /*searchCriteriaDate*/" >= m3.start_date
and
  /*searchCriteriaDate*/" < m3.end_date
and
  /*localeId*/ja' = m3.locale_id
and
  m3.delete_flag = '0'
left outer join
  k30m_pay_type m4
on
  c1.search_criteria_company = m4.company_cd
and
  t.pay_type_cd = m4.pay_type_cd
and
  /*localeId*/ja' = m4.locale_id
and
  /*searchCriteriaDate*/" >= m4.start_date
and
  /*searchCriteriaDate*/" < m4.end_date
and
  m4.delete_flag = '0'
left outer join
  k30m_pay_method m5
on
  c1.search_criteria_company = m5.company_cd
and
  t.pay_method_cd = m5.pay_method_cd
and
  /*localeId*/ja' = m5.locale_id
and
  /*searchCriteriaDate*/" >= m5.start_date
and
  /*searchCriteriaDate*/" < m5.end_date
and
  m5.delete_flag = '0'
left outer join imm_currency m6
on
  c1.search_criteria_company = m6.company_cd
and
  t.company_currency_cd = m6.currency_cd
and
  /*localeId*/ja' = m6.locale_id
and
  m6.delete_flag = '0'
left outer join
  k30m_account m8
on
  c1.search_criteria_company = m8.company_cd

```

```

c1.search_criteria_company = m8.company_cd
and
t.account_cd = m8.account_cd
and
/*localeId*/ja' = m8.locale_id
and
/*searchCriteriaDate*/" >= m8.start_date
and
/*searchCriteriaDate*/" < m8.end_date
and
m8.delete_flag = '0'
left outer join
k30m_tax_type m9
on
c1.search_criteria_company = m9.company_cd
and
t.tax_type_cd = m9.tax_type_cd
and
/*localeId*/ja' = m9.locale_id
and
/*searchCriteriaDate*/" >= m9.start_date
and
/*searchCriteriaDate*/" < m9.end_date
and
m9.delete_flag = '0'

left outer join
k30m_exp_type ma
on
c1.search_criteria_company = ma.company_cd
and
t.exp_type_cd = ma.exp_type_cd
and
/*localeId*/ja' = ma.locale_id
and
/*searchCriteriaDate*/" >= ma.start_date
and
/*searchCriteriaDate*/" < ma.end_date
and
ma.delete_flag = '0'
left outer join
k30m_account mb
on
ma.company_cd = mb.company_cd
and
ma.account_cd = mb.account_cd
and
/*localeId*/ja' = mb.locale_id
and
/*searchCriteriaDate*/" >= mb.start_date
and
/*searchCriteriaDate*/" < mb.end_date
and
m8.delete_flag = '0'
left outer join
k30m_tax_type mc
on
ma.company_cd = mc.company_cd
and
ma.tax_type_cd = mc.tax_type_cd
and
/*localeId*/ja' = mc.locale_id
and
/*searchCriteriaDate*/" >= mc.start_date
and
/*searchCriteriaDate*/" < mc.end_date
and
mc.delete_flag = '0'
where
t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/

```

```

/* If searchCriteriaCompany != null */
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/lodgedetail/impl/LodgeDetailManagerImpl/selectForJournalSourceDtl.sql

```

select
  t.system_matter_id as system_matter_id,
  t.user_data_id as user_data_id,
  t.gadget_class as gadget_class,
  t.gadget_instance as gadget_instance,
  t.tuple_id as tuple_id,
  t.gadget_id as gadget_id,
  t.gadget_variation as gadget_variation,
  i1.contents_id as contents_id,
  i1.contents_version_id as contents_version_id,
  null as journal_data_type_cd,
  t.row_number as row_number,
  t.exp_type_cd as exp_type_cd,
  t.account_cd as account_cd,
  t.tax_type_cd as tax_type_cd,
  mt.tax_rate as tax_rate,
  t.quantity as quantity,
  t.trans_currency_cd as trans_currency_cd,
  t.trans_amount as trans_amount,
  t.company_currency_cd as company_currency_cd,
  t.company_rate as company_rate,
  t.company_amount as company_amount,
  t.company_amount_exc_tax as company_amount_exc_tax,
  t.company_amount_tax as company_amount_tax,
  t.company_amount_sum as company_amount_sum,
  t.payee_cd as customer_cd,
  t.payee_name as customer_name,
  t.department_set_cd as department_set_cd,
  t.department_cd as department_cd,
  t.project_cd as project_cd,
  t.pay_method_cd as pay_method_cd,
  t.pay_type_cd as pay_type_cd,
  null as adv_matter_number,
  null as adv_matter_name,
  t.start_date as detail_date1,
  t.end_date as detail_date2,
  t.lodge_name as detail_summary1,
  null as detail_summary2,
  null as detail_summary3,
  null as detail_summary4,
  null as detail_summary5,
  null as detail_note,
  t.extension_a as detail_extension_a,
  t.extension_b as detail_extension_b,
  t.extension_c as detail_extension_c,
  t.extension_d as detail_extension_d,
  t.extension_e as detail_extension_e,
  t.extension_f as detail_extension_f,
  t.extension_g as detail_extension_g,
  t.extension_h as detail_extension_h,
  t.extension_i as detail_extension_i,
  t.extension_j as detail_extension_j,
  '0' as journal_status,
  null as wish_date,
  c1.search_criteria_company,
  c1.search_criteria_date
from
  k30t_lodge_detail t
  inner join
    k20c_matter_info i1
  on
    t.system_matter_id = i1.system_matter_id
  inner join
    k20c_matter_control c1
  on
    t.system_matter_id = c1.system_matter_id
and
  t.user_data_id = c1.user_data_id
left outer join
  k30m_tax_type mt
on
  c1.search_criteria_company = mt.company_cd
and

```

```

and
  t.tax_type_cd = mt.tax_type_cd
and
  c1.search_criteria_date >= mt.start_date
and
  c1.search_criteria_date < mt.end_date
and
  i1.locale_id_apply = mt.locale_id
and
  mt.delete_flag = '0'

where
  t.system_matter_id = /*systemMatterId*/
  /*IF userDataId != null*/
and
  t.user_data_id = /*userDataId*/
  /*END*/
  /*IF gadgetClass != null*/
and
  t.gadget_class = /*gadgetClass*/
  /*END*/
  /*IF gadgetInstance != null*/
and
  t.gadget_instance = /*gadgetInstance*/
  /*END*/
  /*IF gadgetId != null*/
and
  t.gadget_id = /*gadgetId*/
  /*END*/
  /*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
  /*END*/
  /*IF orderBy != null*/
order by /*$orderBy*/t.row_number
  /*END*/

```

JSP

PC

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/lodgeDetail/lodgeDetailV02.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wkf" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>

<!-- product_workflow 共通インクルード --%>
<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- lodgeDetail v02 --%>
<c:set var="gadgetClass">lodgeDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="accessUrl">kaiden/gadget/lodgeDetailV02</c:set>
<c:set var="gadgetBlockId">lodgeDetailTable</c:set>
<c:set var="gadgetBlockHeaderId">lodgeDetailHeader</c:set>
<c:set var="ttableId">lodgeDetailTable</c:set>
<c:set var="ttableControllerBlockId">ttableController</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>

<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="${gadget.blocks[lodgeDetailTable]}" />

<!-- ttable --%>
<c:set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<c:set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<!-- Message --%>
<c:set var="msgRateReget">%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0008) %></c:set>

```

```

<%-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</c:set>
<c:set var="surfaceLodgeTerm">${gadget.surfaces["lodgeTerm"]}</c:set>
<c:set var="surfaceLodgeStartDate">${gadget.surfaces["startDate"]}</c:set>
<c:set var="surfaceConcatChar">${gadget.surfaces["concatChar"]}</c:set>
<c:set var="surfaceLodgeEndDate">${gadget.surfaces["endDate"]}</c:set>
<c:set var="surfaceLodge">${gadget.surfaces["lodgeName"]}</c:set>
<c:set var="surfaceLodgeTelephoneNumber">${gadget.surfaces["lodgeTelephoneNumber"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceTransCurrency">${gadget.surfaces["amount_transCurrency"]}</c:set>
<c:set var="surfaceCompanyRate">${gadget.surfaces["amount_transRate"]}</c:set>
<c:set var="surfaceCompanyAmount">${gadget.surfaces["amount_companyAmount"]}</c:set>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>

<%-- Mode --%>
<c:set var="modeStartDate">${detail.fields["startDate"].inputType}</c:set>
<c:set var="modeEndDate">${detail.fields["endDate"].inputType}</c:set>
<c:set var="modeLodge">${detail.fields["lodgeName"].inputType}</c:set>
<c:set var="modeLodgeTelephoneNumber">${detail.fields["lodgeTelephoneNumber"].inputType}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeTransAmount">${detail.fields["transAmount"].inputType}</c:set>
<c:set var="modeTransCurrency">${detail.fields["transCurrencyCd"].inputType}</c:set>
<c:set var="modeCompanyRate">${detail.fields["companyRate"].inputType}</c:set>
<c:set var="modeCompanyAmount">${detail.fields["companyAmount"].inputType}</c:set>
<c:set var="modePayTypeCd">${detail.fields["payTypeCd"].inputType}</c:set>
<c:set var="modePayeeName">${detail.fields["payeeName"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>

<%-- RequiredMark --%>
<c:set var="reqStartDate" >${detail.fields["startDate"].requiredFlag}</c:set>
<c:set var="reqEndDate" >${detail.fields["endDate"].requiredFlag}</c:set>
<c:set var="reqLodge" >${detail.fields["lodgeName"].requiredFlag}</c:set>
<c:set var="reqLodgeTelephoneNumber" >${detail.fields["lodgeTelephoneNumber"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd" >${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd" >${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd" >${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount" >${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqTransCurrency" >${detail.fields["transCurrencyCd"].requiredFlag}</c:set>
<c:set var="reqCompanyRate" >${detail.fields["companyRate"].requiredFlag}</c:set>
<c:set var="reqCompanyAmount" >${detail.fields["companyAmount"].requiredFlag}</c:set>
<c:set var="reqPayTypeCd" >${detail.fields["payTypeCd"].requiredFlag}</c:set>
<c:set var="reqPayeeName" >${detail.fields["payeeName"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd" >${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd" >${detail.fields["projectCd"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validStartDate_def">{k_date:[""]}</c:set>
<c:set var="validStartDate_apply">${"true" == reqStartDate ? "k_required:[""]" : ""}</c:set>
<c:set var="validStartDate_tempsave"></c:set>
<c:set var="validStartDate_reapply">${"true" == reqStartDate ? "k_required:[""]" : ""}</c:set>
<c:set var="validStartDate_process">${"true" == reqStartDate ? "k_required:[""]" : ""}</c:set>

<c:set var="validEndDate_def">{k_date:[""], k_dateMin:["startDate"]}</c:set>
<c:set var="validEndDate_apply">${"true" == reqEndDate ? "k_required:[""]" : ""}</c:set>
<c:set var="validEndDate_tempsave"></c:set>
<c:set var="validEndDate_reapply">${"true" == reqEndDate ? "k_required:[""]" : ""}</c:set>
<c:set var="validEndDate_process">${"true" == reqEndDate ? "k_required:[""]" : ""}</c:set>

<c:set var="validLodge_def">{k_maxLength:["250"]}</c:set>
<c:set var="validLodge_apply">${"true" == reqLodge ? "k_required:[""]" : ""}</c:set>
<c:set var="validLodge_tempsave"></c:set>
<c:set var="validLodge_reapply">${"true" == reqLodge ? "k_required:[""]" : ""}</c:set>
<c:set var="validLodge_process">${"true" == reqLodge ? "k_required:[""]" : ""}</c:set>

<c:set var="validLodgeTelephoneNumber_def">{k_maxLength:["250"]}</c:set>
<c:set var="validLodgeTelephoneNumber_apply">${"true" == reqLodgeTelephoneNumber ? "k_required:[""]" : ""}</c:set>
<c:set var="validLodgeTelephoneNumber_tempsave"></c:set>
<c:set var="validLodgeTelephoneNumber_reapply">${"true" == reqLodgeTelephoneNumber ? "k_required:[""]" : ""}</c:set>
<c:set var="validLodgeTelephoneNumber_process">${"true" == reqLodgeTelephoneNumber ? "k_required:[""]" : ""}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:["expTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validExpTypeCd_apply">${"true" == reqExpTypeCd ? "k_required:[""]" : ""}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">${"true" == reqExpTypeCd ? "k_required:[""]" : ""}</c:set>
<c:set var="validExpTypeCd_process">${"true" == reqExpTypeCd ? "k_required:[""]" : ""}</c:set>

```

```

<c:set var="validAccountCd_def">{k_accountExists:['accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">{"true" == reqAccountCd ? "k_required:[''] : ''}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">{"true" == reqAccountCd ? "k_required:[''] : ''}</c:set>
<c:set var="validAccountCd_process">{"true" == reqAccountCd ? "k_required:[''] : ''}</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:['taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">{"true" == reqTaxTypeCd ? "k_required:[''] : ''}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{"true" == reqTaxTypeCd ? "k_required:[''] : ''}</c:set>
<c:set var="validTaxType_process">{"true" == reqTaxTypeCd ? "k_required:[''] : ''}</c:set>

<c:set var="validPayTypeCd_def">{k_payTypeExists:['payTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validPayTypeCd_apply">{"true" == reqPayTypeCd ? "k_required:[''] : ''}</c:set>
<c:set var="validPayTypeCd_tempsave"></c:set>
<c:set var="validPayTypeCd_reapply">{"true" == reqPayTypeCd ? "k_required:[''] : ''}</c:set>
<c:set var="validPayTypeCd_process">{"true" == reqPayTypeCd ? "k_required:[''] : ''}</c:set>

<c:set var="validPayeeName_def">{k_maxLength:['250']}</c:set>
<c:set var="validPayeeName_apply">{"true" == reqPayeeName ? "k_required:[''] : ''}</c:set>
<c:set var="validPayeeName_tempsave"></c:set>
<c:set var="validPayeeName_reapply">{"true" == reqPayeeName ? "k_required:[''] : ''}</c:set>
<c:set var="validPayeeName_process">{"true" == reqPayeeName ? "k_required:[''] : ''}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:['searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate', 'false']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:['projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<k:gadgetContainer gadgetClass="{gadgetClass}"
    gadgetVariation="{gadgetVariation}"
    gadgetInstance="{gadgetInstance}"
    gadgetTitle="{surfaceChapterTitle}"
    gadgetId="{gadgetId}"
    hideGadgetTitle="true"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="{gadgetBlockHeaderId}">
<div class="imui-chapter-title">
<h2 style="display:inline-block;width:50%">
<k:surface
    surfaceKey="{surfaceChapterTitle}"
/>
</h2>
<k:a
    class="kaiden-ui-icon-refresh-gray16_nosize"
    name="rateReget"
    mode="{modeTransCurrency}"
    text="{surfaceRateReget}"
    style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem"
/>
</div>
</k:blockContainer>
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:tmtable id="{tmtableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumTarget="rowNumber">
<colgroup>
<col style="width:40px;overflow:hidden">
<col style="width:235px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
<col style="width:185px;overflow:hidden">
</colgroup>
<thead>
<tr>
<td rowspan="3" style="vertical-align:middle;">
<k:tmtableOperationPanel
    position="header"
    allowAdd="{allowAddDetail}"
    allowDel="{allowDelDetail}"
/>
</td>
<th>
<k:surface
    name="lodgeTermSurface"
    surfaceKey="{surfaceLodgeTerm}"
    requiredMark="{reqStartDate}"
/>
</th>
<th colspan="2">

```

```

<k:surface
  name="lodgeNameSurface"
  surfaceKey="{surfaceLodge}"
  requiredMark="{reqLodge}"
/>
</th>
<th>
<k:surface
  name="lodgeTelephoneSurface"
  surfaceKey="{surfaceLodgeTelephone}"
  requiredMark="{reqLodgeTelephone}"
/>
</th>
</tr>
<tr>
<th colspan="2">
<k:surface
  name="expTypeSurface"
  surfaceKey="{surfaceExpType}"
  requiredMark="{reqExpTypeCd}"
  style="display:inline;"
/>
<k:surface
  name="accountSurface"
  surfaceKey="{surfaceAccount}"
  style="display:inline;"
/>
<k:surface
  name="taxTypeSurface"
  surfaceKey="{surfaceTaxType}"
  style="display:inline;"
/>
</th>
<th>
<k:surface
  name="departmentSurface"
  surfaceKey="{surfaceDepartment}"
  requiredMark="{reqDepartmentCd}"
/>
</th>
<th>
<k:surface
  name="projectSurface"
  surfaceKey="{surfaceProject}"
  requiredMark="{reqProjectCd}"
/>
</th>
</tr>
<tr>
<th>
<k:surface
  name="payTypeSurface"
  surfaceKey="{surfacePayType}"
  requiredMark="{reqPayTypeCd}"
/>
</th>
<th>
<k:surface
  name="payeeSurface"
  surfaceKey="{surfacePayee}"
  requiredMark="{reqPayeeName}"
/>
</th>
<th colspan="2">
<k:surface
  name="transAmountSurface"
  surfaceKey="{surfaceTransAmount}"
  requiredMark="{reqTransAmount}"
/>
</th>
</tr>
</thead>
<tbody>
<tr>
<td rowspan="3">
<k:tmtableOperationPanel
  position="detail"
  allowAdd="{allowAddDetail}"
  allowDel="{allowDelDetail}"
  name="rowNumber"
/>
</td>
<td>
<k:date
  name="startDate"
  mode="{modeStartDate}"

```

```

        surfaceKey="${surfaceLodgeStartDate}"
        validationDefault="${validStartDate_def}"
        validationApply="${validStartDate_apply}"
        validationTempsave="${validStartDate_tempsave}"
        validationReapply="${validStartDate_reapply}"
        validationProcess="${validStartDate_process}"
    />
    <k:surface
        name="concatCharSurface"
        surfaceKey="${surfaceConcatChar}"
    />
    <k:date
        name="endDate"
        mode="${modeEndDate}"
        surfaceKey="${surfaceLodgeEndDate}"
        validationDefault="${validEndDate_def}"
        validationApply="${validEndDate_apply}"
        validationTempsave="${validEndDate_tempsave}"
        validationReapply="${validEndDate_reapply}"
        validationProcess="${validEndDate_process}"
    />
</td>
<td colspan="2">
    <k:text
        name="lodgeName"
        mode="${modeLodge}"
        imeMode="on"
        surfaceKey="${surfaceLodge}"
        style="width:350px;"
        maxlength="250"
        validationDefault="${validLodge_def}"
        validationApply="${validLodge_apply}"
        validationTempsave="${validLodge_tempsave}"
        validationReapply="${validLodge_reapply}"
        validationProcess="${validLodge_process}"
    />
</td>
<td>
    <k:text
        name="lodgeTelephoneNumber"
        mode="${modeLodgeTelephoneNumber}"
        imeMode="off"
        surfaceKey="${surfaceLodgeTelephoneNumber}"
        style="width:165px;"
        maxlength="250"
        validationDefault="${validLodgeTelephoneNumber_def}"
        validationApply="${validLodgeTelephoneNumber_apply}"
        validationTempsave="${validLodgeTelephoneNumber_tempsave}"
        validationReapply="${validLodgeTelephoneNumber_reapply}"
        validationProcess="${validLodgeTelephoneNumber_process}"
    />
</td>
</tr>
<tr>
    <td colspan="2">
        <k:hidden
            name="expTypeCd"
            mode="${modeExpTypeCd}"
        />
        <kmsw:expType
            name="expTypeName"
            mode="${modeExpTypeCd}"
            imeMode="on"
            surfaceKey="${surfaceExpType}"
            resultValue="expTypeName"
            resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
            primaryTarget="expTypeCd"
            searchCriteriaDate="searchCriteriaDate"
            searchCriteriaCompany="searchCriteriaCompany"
            searchCriteriaExpTypeCd="expTypeCd"
            style="width:150px;"
            validationDefault="${validExpTypeCd_def}"
            validationApply="${validExpTypeCd_apply}"
            validationTempsave="${validExpTypeCd_tempsave}"
            validationReapply="${validExpTypeCd_reapply}"
            validationProcess="${validExpTypeCd_process}"
        />
        <k:hidden
            name="accountCd"
            mode="${modeAccountCd}"
            surfaceKey="${surfaceAccount}"
            validationDefault="${validAccountCd_def}"
            validationApply="${validAccountCd_apply}"
            validationTempsave="${validAccountCd_tempsave}"
            validationReapply="${validAccountCd_reapply}"
            validationProcess="${validAccountCd_process}"

```



```

/>
<k:text
  name="accountName"
  mode="1"
  class="kaiden-ui-mst-search-chain-item"
  style="width:115px;"
/>
<k:hidden
  name="taxTypeCd"
  mode="{modeTaxTypeCd}"
  validationDefault="{validTaxType_def}"
  validationApply="{validTaxType_apply}"
  validationTempsave="{validTaxType_tempsave}"
  validationReapply="{validTaxType_reapply}"
  validationProcess="{validTaxType_process}"
  surfaceKey="{surfaceTaxType}"
/>
<k:text
  name="taxTypeName"
  mode="1"
  class="kaiden-ui-mst-search-chain-item"
  style="width:115px;"
/>
</td>
<td>
<k:hidden
  name="departmentSetCd"
  mode="{modeDepartmentCd}"
/>
<k:hidden
  name="departmentCd"
  mode="{modeDepartmentCd}"
  surfaceKey="{surfaceDepartment}"
/>
<kmsb:department
  name="departmentName"
  mode="{modeDepartmentCd}"
  imeMode="on"
  surfaceKey="{surfaceDepartment}"
  resultValue="departmentName"
  resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd}"
  primaryTarget="departmentCd,departmentSetCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaDepartmentCd="departmentCd"
  searchCriteriaDepartmentSetCd="departmentSetCd"
  style="width:150px;"
  validationDefault="{validDepartmentCd_def}"
  validationApply="{validDepartmentCd_apply}"
  validationTempsave="{validDepartmentCd_tempsave}"
  validationReapply="{validDepartmentCd_reapply}"
  validationProcess="{validDepartmentCd_process}"
/>
</td>
<td>
<k:hidden
  name="projectCd"
  mode="{modeProjectCd}"
  surfaceKey="{surfaceProject}"
/>
<kmsw:project
  name="projectName"
  mode="{modeProjectCd}"
  imeMode="on"
  surfaceKey="{surfaceProject}"
  resultValue="projectName"
  resultChain="{projectCd:'projectCd}"
  primaryTarget="projectCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaProjectCd="projectCd"
  style="width:150px;"
  validationDefault="{validProjectCd_def}"
  validationApply="{validProjectCd_apply}"
  validationTempsave="{validProjectCd_tempsave}"
  validationReapply="{validProjectCd_reapply}"
  validationProcess="{validProjectCd_process}"
/>
</td>
</tr>
<tr>
<td>
<klist_wkf:listDataPayType
  blank="false"
  name="payType"
  companyCd="{param.searchCriteriaCompany}"

```

```

        searchCriteriaDate="${param.searchCriteriaDate}"
    />
    <k:select
        name="payTypeCd"
        mode="${modePayTypeCd}"
        surfaceKey="${surfacePayType}"
        viewModeName="payTypeName"
        dataName="payType"
        selectChain="{paymentFlag;paymentFlag}"
        style="width:230px;"
        validationDefault="${validPayTypeCd_def}"
        validationApply="${validPayTypeCd_apply}"
        validationTempsave="${validPayTypeCd_tempsave}"
        validationReapply="${validPayTypeCd_reapply}"
        validationProcess="${validPayTypeCd_process}"
    />
    <k:hidden
        name="paymentFlag"
        mode="${modePayTypeCd}"
    />
</td>
<td>
    <k:text
        name="payeeName"
        mode="${modePayeeName}"
        imeMode="on"
        surfaceKey="${surfacePayee}"
        style="width:160px;"
        maxlength="250"
        validationDefault="${validPayeeName_def}"
        validationApply="${validPayeeName_apply}"
        validationTempsave="${validPayeeName_tempsave}"
        validationReapply="${validPayeeName_reapply}"
        validationProcess="${validPayeeName_process}"
    />
</td>
<td colspan="2">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericCurrencyInputParts.jsp">
        <jsp:param name="gadgetId" value="${gadgetId}" />
        <jsp:param name="tmtableId" value="${tmtableId}" />
        <jsp:param name="searchCriteriaCompany" value="${searchCriteriaCompany}" />
        <jsp:param name="companyCurrencyCd" value="${companyCurrencyCd}" />
        <%-- 金額 --%>
        <jsp:param name="modeTransAmnt" value="${modeTransAmount}" />
        <jsp:param name="surfaceTransAmnt" value="${surfaceTransAmount}" />
        <jsp:param name="requiredTransAmntApply" value="${reqTransAmount}" />
        <jsp:param name="requiredTransAmntReapply" value="${reqTransAmount}" />
        <jsp:param name="requiredTransAmntProcess" value="${reqTransAmount}" />
        <%-- 通貨 --%>
        <jsp:param name="modeTransCryCd" value="${modeTransCurrency}" />
        <jsp:param name="surfaceTransCryCd" value="${surfaceTransCurrency}" />
        <jsp:param name="requiredTransCryCdApply" value="${reqTransCurrency}" />
        <jsp:param name="requiredTransCryCdReapply" value="${reqTransCurrency}" />
        <jsp:param name="requiredTransCryCdProcess" value="${reqTransCurrency}" />
        <%-- レート --%>
        <jsp:param name="modeCmpRate" value="${modeCompanyRate}" />
        <jsp:param name="surfaceCmpRate" value="${surfaceCompanyRate}" />
        <jsp:param name="requiredCmpRateApply" value="${reqTransCurrency}" />
        <jsp:param name="requiredCmpRateReapply" value="${reqTransCurrency}" />
        <jsp:param name="requiredCmpRateProcess" value="${reqTransCurrency}" />
        <%-- 換算額 --%>
        <jsp:param name="modeCmpAmnt" value="${modeCompanyAmount}" />
        <jsp:param name="surfaceCmpAmnt" value="${surfaceCompanyAmount}" />
        <jsp:param name="requiredCmpAmntApply" value="${reqCompanyAmount}" />
        <jsp:param name="requiredCmpAmntReapply" value="${reqCompanyAmount}" />
        <jsp:param name="requiredCmpAmntProcess" value="${reqCompanyAmount}" />
    </jsp:include>
</td>

</tr>
</tbody>
</k:tmtable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){

    var eventMan = KAIDEN.getGadget("${gadgetId").eventManager
    , blockAccessor = KAIDEN.getGadget("${gadgetId").blockAccessors["${gadgetBlockHeaderId}"]
    , tmt = KAIDEN.getGadget("${gadgetId").tmtables["${tmtableId}"]
    , tma = tmt.tmtableAccessor;

    /*****
    * 内部イベント
    *****/

```

```

//明細行:行挿入(データ設定前)
tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

//デフォルト部門を要求
var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "${gadgetId}");
defaultDept = defaultDept[0].result; //先頭行のみ利用
tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
tma("departmentName", tupleId).setValue(defaultDept.departmentName);

//デフォルトプロジェクトを要求
var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "${gadgetId}");
defaultPrj = defaultPrj[0].result; //先頭行のみ利用
tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
tma("projectName", tupleId).setValue(defaultPrj.projectName);

});

//明細行:行挿入(データ設定後)
tmt.addEventPostNewRow(function(tupleId, tma, data) {
//マスタ検索部品の整合性確認。
KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

//レートボックスのreadonly化
if (tma("transCurrencyCd", tupleId).elem(true).length > 0) {
var currencyCd = tma("transCurrencyCd", tupleId).getValue();
if (currencyCd == "${companyCurrencyCd}") {
tma("companyRate", tupleId).readonly();
} else {
tma("companyRate", tupleId).removeReadonly();
}
}

//初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
if (tma("payTypeCd", tupleId).elem(true).length > 0) {
KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
}
});
//レート再取得ロジック
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll = function() {
tma("transCurrencyCd", tupleId).elem(true).each(function() {
tmt.variable.loadRate(this);
});
}
//レート再取得:ボタン押下
blockAccessor("rateReget").elem(true).on("click", function(e){
imuiConfirm("${msgRateReget}", "", function() {
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
}, undefined, false);
});

/*****
* 受信イベント(push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
//明細行のクリア
tmt.deleteAll();
return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//reloadAllCurrencyRate レート再取得 イベントリスナの登録
eventMan.registPushListener("reloadAllCurrencyRate", function(**Object*/arg) {
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
});

/*****
* 受信イベント(pull)
*****/

//claimDateDetail 明細日付を要求 イベントリスナの登録
eventMan.registPullListener("claimDateDetail", function(**Object*/arg) {
return tmt.getColumnValues(["startDate", "endDate"]);
});

```

```

});

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues("companyAmount");
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value)) {
            result = result.add(new BigDecimal(value));
        }
    });
    return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues(["companyAmount", "paymentFlag"]);
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value[0]) && value[1] == "1") {
            result = result.add(new BigDecimal(value[0]));
        }
    });
    return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues(["companyAmount", "paymentFlag"]);
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value[0]) && value[1] == "0") {
            result = result.add(new BigDecimal(value[0]));
        }
    });
    return result.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
    return tmt.getColumnValues("companyAmount");
});

//claimExpenCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenCount", function(**Object*/arg) {
    return tmt.getCount();
});

});
</script>

```

SP

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/lodgeDetail/lodgeDetailV02_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<!-- product_workflow 共通インクルード --%>
<c:if test="{empty pwkfCommonIncluded}" scope="request">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
    <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- 宿泊明細(標準・外貨)ガジェット-->
<c:set var="gadgetClass">lodgeDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_ ${gadgetVariation}_ ${gadgetInstance}</c:set>
<c:set var="componentName">lodgeDetailV02Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">lodgeDetailTable</c:set>

<!-- ガジェットデータの読み込み --%>

```

```

<c:set var="gadgetResponse" value="{kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>

<%-- サーフェスの読み込み --%>
<c:set var="gadget" value="{appDto.gadgets[gadgetId]}" />
<c:set var="detail" value="{gadget.blocks['lodgeDetailTable']}" />

<%-- サーフェス --%>
<%--タイトル --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<%-- レート再取得--%>
<c:set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</c:set>
<%-- 宿泊期間--%>
<c:set var="surfaceLodgeTerm">${gadget.surfaces["lodgeTerm"]}</c:set>
<%-- 宿泊開始日--%>
<c:set var="surfaceLodgeStartDate">${gadget.surfaces["startDate"]}</c:set>
<%-- 宿泊終了日--%>
<c:set var="surfaceLodgeEndDate">${gadget.surfaces["endDate"]}</c:set>
<%-- 宿泊先--%>
<c:set var="surfaceLodge">${gadget.surfaces["lodgeName"]}</c:set>
<%-- 宿泊先TEL--%>
<c:set var="surfaceLodgeTelephoneNumber">${gadget.surfaces["lodgeTelephoneNumber"]}</c:set>
<%-- 経費区分--%>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<%-- 勘定科目--%>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<%-- 税区分--%>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<%-- 金額--%>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<%-- 金額(通貨) --%>
<c:set var="surfaceTransCurrency">${gadget.surfaces["transCurrencyCd"]}</c:set>
<%-- 金額(レート) --%>
<c:set var="surfaceCompanyRate">${gadget.surfaces["companyRate"]}</c:set>
<%-- 金額(換算額) --%>
<c:set var="surfaceCompanyAmount">${gadget.surfaces["companyAmount"]}</c:set>
<%-- 支払分類--%>
<c:set var="surfacePayType">${gadget.surfaces["payType"]}</c:set>
<%-- 支払先--%>
<c:set var="surfacePayee">${gadget.surfaces["payee"]}</c:set>
<%-- 部門--%>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<%-- プロジェクト--%>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<%-- 戻る --%>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconpc
<h3>
    <span>${surfaceChapterTitle}</span>
    <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="{fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>

<ul data-role="listview" data-divder-theme="b">
<c:forEach items="${gadgetResponse.data[tmtableId]}" var="detailData">
<li>
    <span class="ui-li-heading">
        <c:out value="{detailData.startDate}"></c:out>
        <c:out value="-"></c:out>
        <c:out value="{detailData.endDate}"></c:out>
    </span>
    <span class="ui-li-aside"><c:out value="{detailData.companyAmount[1]}"></c:out></span>
    <div class="indent">
        <p class="kaiden-ui-overview-contents"><c:out value="{detailData.lodgeName}"></c:out></p>
        <p class="kaiden-ui-overview-contents-sub"><c:out value="{detailData.payTypeName}"></c:out></p>
    </div>
    <ul>
<li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="{surfaceBack}"></c:out></a></li>
<li>
        <%-- 宿泊期間 --%>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceLodgeTerm}"></c:out></p>
        <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.startDate}"></c:out><c:out value="- "></c:out><c:out value="{detailData.endDate}"></c:out></label>
        </div>

        <%-- 宿泊先 --%>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceLodge}"></c:out></p>
        <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.lodgeName}"></c:out></label>
        </div>

        <%-- 宿泊先TEL --%>
        <p class="kaiden-ui-detail-item-title"><c:out value="{surfaceLodgeTelephoneNumber}"></c:out></p>
        <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.lodgeTelephoneNumber}"></c:out></label>
        </div>
    </li>
    </ul>
</li>
</c:forEach>

```

```

<%-- 経費区分 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceExpType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.expTypeName}"></c:out></label>
</div>

<%-- 勘定科目 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceAccount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.accountName}"></c:out></label>
</div>

<%-- 税区分 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTaxType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.taxTypeName}"></c:out></label>
</div>

<%-- 金額 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTransAmount}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents-sub"><c:out value="\${detailData.transAmount[1]}"></c:out></label>
  <label class="kaiden-ui-detail-item-contents-sub" style="margin-left: 1em"><c:out value="\${detailData.transCurrencyName}"></c:out></label>
</div>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents-sub kaiden-ui-cmpRate"><c:out value="\${detailData.companyRate[1]}"></c:out></label>
</div>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><strong><c:out value="\${detailData.companyAmount[1]}"></c:out></strong></label>
</div>

<%-- 支払分類 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.payTypeName}"></c:out></label>
</div>

<%-- 支払先 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfacePayee}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents" style="word-wrap: break-word;"><c:out value="\${detailData.payeeName}"></c:out></label>
</div>

<%-- 部門 --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceDepartment}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.departmentName}"></c:out></label>
</div>

<%-- プロジェクト --%>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.projectName}"></c:out></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>

```

日当明細のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   ├── K30tAllowanceDetail.java
│   │   │   │   │   │   │   │   │   ├── K30tAllowanceDetailNames.java
│   │   │   │   │   │   │   │   │   ├── Ka30tAllowanceDetail.java
│   │   │   │   │   │   │   │   │   ├── Ka30tAllowanceDetailNames.java
│   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │       ├── K30tAllowanceDetailCondition.java
│   │   │   │   │   │   │   │       ├── K30tAllowanceDetailService.java
│   │   │   │   │   │   │   │       ├── Ka30tAllowanceDetailCondition.java
│   │   │   │   │   │   │   │       └── Ka30tAllowanceDetailService.java
│   │   │   │   │   │   │   └── java
│   │   │   │   │   │   │       ├── jp
│   │   │   │   │   │   │       │   ├── co
│   │   │   │   │   │   │       │   │   ├── slcs
│   │   │   │   │   │   │       │   │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │   ├── feature
│   │   │   │   │   │   │       │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │       │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── AbstractAllowanceDetailActionBase.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   └── AllowanceDetailV01Action.java
│   │   │   │   │   │   │       │   │   │   │   │   │   ├── conf
│   │   │   │   │   │   │       │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   └── AllowanceDetailFieldName.java
│   │   │   │   │   │   │       │   │   │   │   │   │   ├── form
│   │   │   │   │   │   │       │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   └── AllowanceDetailV01Form.java
│   │   │   │   │   │   │       │   │   │   │   │   │   ├── logic
│   │   │   │   │   │   │       │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── AbstractAllowanceDetailLogicBase.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailV01Logic.java
│   │   │   │   │   │   │       │   │   │   │   │   │   └── foundation
│   │   │   │   │   │   │       │   │   │   │   │   │       ├── manager
│   │   │   │   │   │   │       │   │   │   │   │   │       │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │       │   │   ├── AllowanceDetailManager.java
│   │   │   │   │   │   │       │   │   │   │   │   │       │   │   ├── AllowanceDetailManagerDto.java
│   │   │   │   │   │   │       │   │   │   │   │   │       │   └── impl
│   │   │   │   │   │   │       │   │   │   │   │   │       │       └── AllowanceDetailManagerImpl.java
│   │   │   │   │   │   │       │   │   │   │   │   └── resources
│   │   │   │   │   │   │       │   │   │   │   │       ├── META-INF
│   │   │   │   │   │   │       │   │   │   │   │       │   ├── sql
│   │   │   │   │   │   │       │   │   │   │   │       │   │   ├── jp
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   ├── co
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   ├── slcs
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── impl
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   ├── AllowanceDetailManagerImpl
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   └── selectForJournalSourceDtl.sql
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── K30tAllowanceDetailService
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── selectView.sql
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   └── selectViewForCopy.sql
│   │   │   │   │   │   │       │   │   │   │   │   └── webapp
│   │   │   │   │   │   │       │   │   │   │   │       ├── WEB-INF
│   │   │   │   │   │   │       │   │   │   │   │       │   ├── view
│   │   │   │   │   │   │       │   │   │   │   │       │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   ├── allowanceDetailV01.jsp
│   │   │   │   │   │   │       │   │   │   │   │       │   │   │   │   │   │   └── allowanceDetailV01_sp.jsp

```


Action (基底)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractAllowanceDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 日当明細ガジェット用Actionの抽象クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractAllowanceDetailActionBase implements GadgetActionIF {

    /**
     * 申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadProcess() {

```

```

        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadConfirm() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadProcessDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 画面表示用データを返却.
     *
     * @return Managerからの返却値
     */
    @Override
    public LogicResultDto getViewData() {
        return getLogic().view(getForm(), getConvertor());
    }

    /**
     * 参照詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadReferenceDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadConfirmDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 申請書コピーのデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadCopy() {
        ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    }

```

```

    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #AllowanceDetailConverter 日当明細(標準)ガジェット}の書式変換を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected AllowanceDetailConverter getConverter() {
    return new AllowanceDetailConverter(getForm());
}

/**
 * 日当明細(標準)ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class AllowanceDetailConverter extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;

    /** NumberHelper. */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public AllowanceDetailConverter(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨換算レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>日数</td>
     * <td>整数表示に変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨額合計</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * </table>
     * </p>
     */

```

```

@Override
protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

    Map<String, Object> result = new HashMap<String, Object>(map);

    // 取引通貨額の書式変換
    if (result.containsKey(TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) result.get(TRANS_AMOUNT);
        Object currencyCd = result.get(TRANS_CURRENCY_CD);
        result.put(TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }
    // レートの書式変換
    if (result.containsKey(COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(COMPANY_RATE);
        result.put(COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }

    String companyCurrencyCd = (String) result.get(COMPANY_CURRENCY_CD);

    // 換算額の書式変換
    if (result.containsKey(COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(COMPANY_AMOUNT);
        result.put(COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    // 日数の書式変換
    if (result.containsKey(QUANTITY)) {
        BigDecimal quantity = (BigDecimal) map.get(QUANTITY);
        if (quantity == null) {
            result.put(QUANTITY, "");
        } else {
            result.put(QUANTITY, StringUtil.toString(quantity.setScale(0)));
        }
    }

    // 金額(換算額*日数)の書式変換
    if (result.containsKey(COMPANY_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(COMPANY_AMOUNT_SUM);
        result.put(COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4Input(companyAmountSum,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨換算レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>日数</td>
 * <td>整数表示に変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額合計</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * <p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(TRANS_AMOUNT);
    }

```

```

        BigDecimal amount = (BigDecimal) map.get(TRANS_AMOUNT);
        Object currencyCd = map.get(TRANS_CURRENCY_CD);
        result.put(TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(COMpany_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(COMpany_RATE);
        result.put(COMpany_RATE, numberHelper.formatRate4View(rate));
    }

    String companyCurrencyCd = result.get(COMpany_CURRENCY_CD);
    // 換算額の書式変換
    if (map.containsKey(COMpany_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(COMpany_AMOUNT);
        result.put(COMpany_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    // 日数の書式変換
    if (map.containsKey(QUANTITY)) {
        BigDecimal quantity = (BigDecimal) map.get(QUANTITY);
        if (quantity == null) {
            result.put(QUANTITY, "");
        } else {
            result.put(QUANTITY, StringUtil.toString(quantity.setScale(0)));
        }
    }

    // 金額(換算額*回数)の書式変換
    if (map.containsKey(COMpany_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(COMpany_AMOUNT_SUM);
        result.put(COMpany_AMOUNT_SUM, numberHelper.formatCurrencyAmount4View(companyAmountSum,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    return result;
}
}
}

```

Action

- Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AllowanceDetailV01 Action.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.AllowanceDetailV01Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail.AllowanceDetailV01Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 日当明細(標準)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV01Action extends AbstractAllowanceDetailActionBase {

    /**
     * AllowancedetailV01Logic.
     */
    @Resource
    public AllowanceDetailV01Logic allowanceDetailV01Logic;

    /**
     * AllowanceDetailV01Form.
     */
    @ActionForm
    @Resource
    public AllowanceDetailV01Form allowanceDetailV01Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return allowanceDetailV01Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return allowanceDetailV01Form;
    }
}

```

Form

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/AllowanceDetailV01Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 日当明細(標準)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV01Form extends AbstractImwPageForm {

}

```

Logic(基底)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/allowancedetail/AbstractAllowanceDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_RANGE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;

```

```

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ALLOWANCE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericRangeValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AllowanceExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

```

```
/**
```

```
 * 日当明細ガジェット用Logicの抽象クラス。
```

```
 *
```

```
 * @author Sumitomo Life Information Systems Co.,Ltd.
```

```

*/
public abstract class AbstractAllowanceDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 日当明細ガジェット・日当明細テーブル. */
    protected static final String BLOCK_KEY_ALLOWANCE_DETAIL_TABLE = "allowanceDetailTable";

    /** 金額、整数部桁数. */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 金額、小数部桁数. */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 日数 最大値. */
    public static final int QUANTITY_MAX = 999;

    /** AllowanceDetailManager. */
    @Resource(name = "allowanceGadgetManager")
    protected AllowanceDetailManager allowanceDetailManager;

    /** JournalSourceDtlManager. */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** コンストラクタ. */
    public AbstractAllowanceDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 日当明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる日当明細テーブルを削除後に、日当明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM\_MATTER\_ID\_TEMPORARY} temporary"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 日当明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {

```



```

        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
}

```

```

    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細を登録します。
 * </p>
 */
@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細テーブルの対象データを削除します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * </p>
 */
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細テーブルを削除後に、日当明細テーブルを登録します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * </p>
 */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * </p>
 */
@Override

```

```

@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.imwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_APPROVEEND.equals(status) || LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status)) {
        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 日当明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userDataId = pluginGadgetDTO.userDataId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 日当明細データを取得
            List<BeanMap> beanMapList = allowanceDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる(過去案件)日当明細テーブルを一括削除します。
 * </p>
 */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = allowanceDetailManager
        .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().imwPluginParameterDto));
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * データの取得は、{@link AllowanceDetailManager#selectList(AllowanceDetailManagerDto)}を利用します。
 * </p>
 */
@Override
protected LinkedHashMap<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();

```

```

LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();

// 日当明細ガジェットテーブル取得
AllowanceDetailManagerDto allowanceDetailCriteria = Beans.createAndCopy(AllowanceDetailManagerDto.class,
    conditionDto).execute();
List<AllowanceDetailManagerDto> allowanceDetailDtoList = allowanceDetailManager
    .selectList(allowanceDetailCriteria);
List<Map<String, Object>> allowanceDetailMapList = CollectionUtil.toMapList(allowanceDetailDtoList);
rv.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailMapList);

return rv;
}

/**
 * {@inheritDoc}
 *
 * @see #viewMap
 */
@Override
protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
    throws KaidenSystemException {
    LinkedHashMap<String, Object> map = viewMap(conditionDto);
    if (map == null) {
        return null;
    }
    return converter.mapConvert(map);
}

/**
 * 仕訳連携用ヘッダテーブルへのデータ登録処理.
 * <p>
 * 当該処理は、処理結果ステータスが"approveEnd", "matterComplete" の場合のみ行う。
 * </p>
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 * @deprecated<br>
 *      {@link #insertTableJournalSourceDtl(List, MatterEndProcessDto)} を利用してください。
 */
@Deprecated
protected int insertk30tJournalSourceHdr() throws KaidenSystemException {
    return journalSourceDtlManager.entry(makeInsertJshMngDto());
}

/**
 * 登録用仕訳連携用明細DTOを作成.
 *
 * @return 仕訳連携用ヘッダDTO
 * @deprecated
 */
@Deprecated
protected List<JournalSourceDtlManagerDto> makeInsertJshMngDto() {
    CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();

    criteriaDto.systemMatterId = super.getGadgetDto().systemMatterId;
    criteriaDto.userDataId = super.getGadgetDto().userDataId;
    criteriaDto.gadgetClass = super.getGadgetDto().gadgetClass;
    criteriaDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    List<BeanMap> beanMapList = allowanceDetailManager.selectForJournalSourceDtl(criteriaDto);

    List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
        beanMapList);

    return insList;
}

/**
 * 日当明細データ登録処理.
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int insertTable() throws KaidenSystemException {
    int retCnt = 0;

    // 日当明細ガジェットテーブル登録
    List<AllowanceDetailManagerDto> allowanceDetailManagerDtoList = JsonUtil.getDtlList(
        AllowanceDetailManagerDto.class, getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE));
    retCnt += allowanceDetailManager.entry(allowanceDetailManagerDtoList);

    return retCnt;
}

/**
 * 仕訳連携用明細テーブルデータ登録処理.
 *

```

```

* @param srcList 日当明細テーブルデータリスト
* @param matterEndProcessDto 案件終了処理用DTO
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
    MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
    for (JournalSourceDtlManagerDto srcDto : srcList) {
        // 仕訳データ区分をセット
        srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
    }

    // 仕訳連携用明細テーブル登録
    return journalSourceDtlManager.entry(srcList);
}

/**
* 日当明細データ削除処理.
*
* @param tempFlg 一時保存データフラグ
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    int retCnt = 0;
    // 日当明細ガジェットテーブル削除
    AllowanceDetailManagerDto allowanceDetailManagerDto = new AllowanceDetailManagerDto();
    allowanceDetailManagerDto.systemMatterId = super.getGadgetDto().systemMatterId;
    allowanceDetailManagerDto.userDataId = super.getGadgetDto().userDataId;
    allowanceDetailManagerDto.gadgetClass = super.getGadgetDto().gadgetClass;
    allowanceDetailManagerDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        allowanceDetailManagerDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    retCnt += allowanceDetailManager.remove(allowanceDetailManagerDto);

    return retCnt;
}

/**
* 日当明細テーブルデータ削除処理(履歴を含む).
*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/
protected int deleteTableIncludeBackground() throws KaidenSystemException {
    int retCnt = 0;

    // 日当明細ガジェットテーブル削除(履歴を含む)
    AllowanceDetailManagerDto allowanceDetailManagerDto = new AllowanceDetailManagerDto();
    allowanceDetailManagerDto.systemMatterId = super.getGadgetDto().systemMatterId;
    allowanceDetailManagerDto.userDataId = null;
    allowanceDetailManagerDto.gadgetClass = super.getGadgetDto().gadgetClass;
    allowanceDetailManagerDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    retCnt += allowanceDetailManager.remove(allowanceDetailManagerDto);

    return retCnt;
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/
@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/
@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/

```

```

@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {

```

```

defineValidateForNormal(params, true);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 *
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>
 * </tr>
 * <tr>
 * <td>日当コード</td>
 * <td>{@link AllowanceExistsValidator 日当存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>取引通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">会社通貨換算レート</td>
 * <td>{@link DigitsValidator 小数字チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericMinValidator 最小数値チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">日数</td>
 * <td>{@link NumericValidator 半角数字チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericRangeValidator 数値範囲チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>支払方法コード</td>
 * <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>

```

```

* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    // 日当明細ガジェットテーブル部
    if (fieldDefsMap.containsKey(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, tmpFlg);
        // 日当コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ALLOWANCE_CD, VALID_ALLOWANCE_EXISTS, ALLOWANCE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            TRANS_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 取引通貨コード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS,
            TRANS_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨換算レート
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_RATE, VALID_DIGITS,
            String.valueOf(COMPANY_RATE_INT_LEN), String.valueOf(COMPANY_RATE_DEC_LEN));
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_RATE, VALID_NUMERIC_MIN, "0");
        // 会社通貨額
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨コード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_CURRENCY_CD,
            VALID_IMM_CURRENCY_EXISTS, COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 日数
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, QUANTITY, VALID_NUMERIC);
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, QUANTITY, VALID_NUMERIC_RANGE,
            getArgs(1, QUANTITY_MAX));
        // 経費区分コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 組織コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS,
            getGadgetDto().searchCriteriaCampany, DEPARTMENT_SET_CD, DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PROJECT_CD, VALID_PROJECT_EXISTS, PROJECT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 会社通貨額合計
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
    }
}

/**
* {@inheritDoc}
* <p>
* 常に空のListを返却。
* </p>
*/
@Override
protected List<KaidenValidateResult> postSingleValidate(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {

```



```

    return new ArrayList<KaidenValidateResult>();
}

/**
 * {@inheritDoc}
 * <p>
 * 常に空のListを返却。
 * </p>
 */
@Override
protected List<KaidenValidateResult> globalValidateMain(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {
    return new ArrayList<KaidenValidateResult>();
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合(初回表示時など)は、空のリストを返却します。<br>
 * 設定されている場合は、{@link AllowanceDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, new ArrayList<AllowanceDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectView(criteria));
    }
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link AllowanceDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link AllowanceDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した日当明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_depaertment)</li>
 * <li>プロジェクトマスタ</li>
 * <li>日当マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * </p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_ALLOWANCE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<AllowanceDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), AllowanceDetailManagerDto.class);
            for (AllowanceDetailManagerDto dto : sqlResultList) {

```

```

// 取引通貨コードチェック
if (isNotEmpty(dto.transCurrencyCd) && isEmpty(dto.transCurrencyName)) {
    String[] fieldId = getArgs(TRANS_CURRENCY_CD, TRANS_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(dto.transCurrencyCd, dto.transCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 会社通貨コードチェック
if (isNotEmpty(dto.companyCurrencyCd) && isEmpty(dto.companyCurrencyName)) {
    String[] fieldId = getArgs(COMPANY_CURRENCY_CD, COMPANY_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(dto.companyCurrencyCd, dto.companyCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 組織チェック
if (isNotEmpty(dto.departmentSetCd) && isNotEmpty(dto.departmentCd) && isEmpty(dto.departmentName)) {
    String[] fieldId = getArgs(DEPARTMENT_SET_CD, DEPARTMENT_CD, DEPARTMENT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.DEPARTMENT);
    String[] values = getArgs(dto.departmentSetCd, dto.departmentCd, dto.departmentName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(StringUtil.toString(dto.tupleId)), masterName, values, message));
}

// プロジェクトチェック
if (isNotEmpty(dto.projectCd) && isEmpty(dto.projectName)) {
    String[] fieldId = getArgs(PROJECT_CD, PROJECT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PROJECT);
    String[] values = getArgs(dto.projectCd, dto.projectName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 日当チェック
if (isNotEmpty(dto.allowanceCd) && isEmpty(dto.allowanceName)) {
    String[] fieldId = getArgs(ALLOWANCE_CD, ALLOWANCE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ALLOWANCE);
    String[] values = getArgs(dto.allowanceCd, dto.allowanceName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 経費区分チェック
if (isNotEmpty(dto.expTypeCd) && isEmpty(dto.expTypeName)) {
    String[] fieldId = getArgs(EXP_TYPE_CD, EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(dto.expTypeCd, dto.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        dto.accountCd = null;
        dto.accountName = null;
        dto.taxTypeCd = null;
        dto.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        dto.taxRate = null;
        dto.companyAmountExcTax = null;
        dto.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 勘定科目チェック
if (isNotEmpty(dto.accountCd) && isEmpty(dto.accountName)) {
    String[] fieldId = getArgs(ACCOUNT_CD, ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(dto.accountCd, dto.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 税区分チェック
if (isNotEmpty(dto.taxTypeCd) && isEmpty(dto.taxTypeName)) {
    String[] fieldId = getArgs(TAX_TYPE_CD, TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(dto.taxTypeCd, dto.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

```

```

        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.TAX_TYPE);
        String[] values = getArgs(dto.taxTypeCd, dto.taxTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
            .toString(dto.tupleId), masterName, values, message));
    }
}
}
}
return result;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccountIF#getClaimExpenccount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, BigDecimal.ZERO);
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
        if (StringUtil.isNotEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

```

```

List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
Map<String, Object> retMap = new HashMap<String, Object>();

List<BigDecimal> list = new ArrayList<BigDecimal>();

for (Map<String, Object> j : blocjData) {
    String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
    if (StringUtil.isEmpty(strAmnt)) {
        list.add(new BigDecimal(strAmnt));
    }
}
retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細テーブルの対象データを過去案件退避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    // 日当明細
    @SuppressWarnings("unchecked")
    List<AllowanceDetailManagerDto> list = (List<AllowanceDetailManagerDto>) dataMap
        .get(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    int rv = allowanceDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(allowanceDetailManager);
    return list;
}
}

```

Logic

■ Path

```
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/allowancedetail/AllowanceDetailV01Logic.java
```

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;

```

```

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ALLOWANCE_AMOUNT_EXISTS;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.gadget.AbstractPluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.pay_type.PayTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AllowanceAmountValidator;

/**
 * 日当明細(標準)ガジェット(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV01Logic extends AbstractAllowanceDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public AllowanceDetailV01Logic() {
        super();
    }

    /** PayTypeManager. */
    @Resource
    protected PayTypeManager payTypeManager;

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します。<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>日当金額</td>
     * <td>{@link AllowanceAmountValidator 日当金額妥当性チェック}</td>
     * <td align="center">入力/表示</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額合計</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * </table>
     * </p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        super.defineValidateForNormal(params, tmpFlg);

        if (!tmpFlg) {
            // 日当金額
            addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ALLOWANCE_CD, VALID_ALLOWANCE_AMOUNT_EXISTS,
                ALLOWANCE_CD, getGadgetDto().searchCriteriaCompany, getGadgetDto().applyAuthUserCd,
                TRANS_CURRENCY_CD, TRANS_AMOUNT, StringUtil.toStrings(getGadgetDto().searchCriteriaDetail));
        }
    }

```

```

        TRANS_CURRENCY_CD, TRANS_AMOUNT, StringUtil.toString(getGadgetId().searchCriteriaDate)),
    }

    // 取引通貨額
    addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_AMOUNT, VALID_NUMERIC_MIN, "0");
    if (tmpFig) {
        // 組織コード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, DEPARTMENT_CD, VALID_REQUIRED_ANY,
            DEPARTMENT_CD, PROJECT_CD);
        // プロジェクトコード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PROJECT_CD, VALID_REQUIRED_ANY,
            DEPARTMENT_CD, PROJECT_CD);
    }
    // 会社通貨額合計
    addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT_SUM, VALID_NUMERIC_MIN, "0");
}

/**
 * データ補完処理.
 *
 * <p>
 * {@link AbstractPluginGadgetLogic#preProcess(GadgetMediatorParams) 個別前処理}の最後に実行されます.<br>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード:{@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>取引通貨コード:{@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>会社通貨換算レート:固定値:1を設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {

    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(COMPANY_CURRENCY_CD, companyCurrencyCd);

            // 取引通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(TRANS_CURRENCY_CD, companyCurrencyCd);

            // 会社通貨換算レート
            j.put(COMPANY_RATE, "1");
        }
    }

}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 *
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * </p>
 *
 * @return グローバルデータ補完処理
 */
@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

```

```

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * </p>
 * <ul>
 * <li>会社通貨額:取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計:会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = StringUtil.toString(calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(TRANS_CURRENCY_CD), (String) j.get(TRANS_AMOUNT), (String) j.get(COMPANY_RATE)));
            j.put(COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(QUANTITY)));
        }
    }
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>支払方法コード:ガジェット間連携項目. 支払方法コードを設定</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
public void globalComplementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

    String gcPayMethodCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 支払方法コード
            if (gcPayMethodCd == null) {
                gcPayMethodCd = getGadgetCouplingPayMethod();
            }
            j.put(PAY_METHOD_CD, gcPayMethodCd);
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> getClaimExpenceCount(Map<String, Object> params) {
    return super.putClaimExpenceCount(params);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
    return super.putClaimCompanyPayAmount(params);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
    return super.putClaimTemporaryPayAmount(params);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
    return super.putClaimExpenseDetail(params);
}

```

```

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
    return super.putClaimExpenseAmount(params);
}
}

```

FieldName

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/allowancedetail/AllowanceDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 日当明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class AllowanceDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected AllowanceDetailFieldName() {

        /** システム案件id. */
        public static final String SYSTEM_MATTER_ID = "systemMatterId";
        /** ユーザーデータid. */
        public static final String USER_DATA_ID = "userDataId";
        /** ガジェットクラス. */
        public static final String GADGET_CLASS = "gadgetClass";
        /** ガジェットインスタンス. */
        public static final String GADGET_INSTANCE = "gadgetInstance";
        /** タプルid. */
        public static final String TUPLE_ID = "tupleId";
        /** ガジェットid. */
        public static final String GADGET_ID = "gadgetId";
        /** ガジェットバリエーション. */
        public static final String GADGET_VARIATION = "gadgetVariation";
        /** 行番号. */
        public static final String ROW_NUMBER = "rowNumber";
        /** 日当コード. */
        public static final String ALLOWANCE_CD = "allowanceCd";
        /** 日数. */
        public static final String QUANTITY = "quantity";
        /** 経費区分コード. */
        public static final String EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String TAX_RATE = "taxRate";
        /** 取引通貨コード. */
        public static final String TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String TRANS_AMOUNT = "transAmount";
        /** 会社通貨コード. */
        public static final String COMPANY_CURRENCY_CD = "companyCurrencyCd";
        /** 会社通貨換算レート. */
        public static final String COMPANY_RATE = "companyRate";
        /** 会社通貨額. */
        public static final String COMPANY_AMOUNT = "companyAmount";
        /** 会社通貨税抜額. */
        public static final String COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
        /** 会社通貨税額. */
        public static final String COMPANY_AMOUNT_TAX = "companyAmountTax";
        /** 会社通貨額合計. */
        public static final String COMPANY_AMOUNT_SUM = "companyAmountSum";
        /** 支払分類コード. */
        public static final String PAY_TYPE_CD = "payTypeCd";
        /** 支払方法コード. */
        public static final String PAY_METHOD_CD = "payMethodCd";
        /** 支払先コード. */
        public static final String PAYEE_CD = "payeeCd";
        /** 支払先名. */
        public static final String PAYEE_NAME = "payeeName";
        /** 組織セットコード. */
        public static final String DEPARTMENT_SET_CD = "departmentSetCd";

```



```

public static final String DEPARTMENT_CD = "departmentCd";
/** 組織コード. */
public static final String DEPARTMENT_CD = "departmentCd";
/** プロジェクトコード. */
public static final String PROJECT_CD = "projectCd";

/** 取引通貨名称. */
public static final String TRANS_CURRENCY_NAME = "transCurrencyName";
/** 会社通貨名称. */
public static final String COMPANY_CURRENCY_NAME = "companyCurrencyName";
/** 部門名称. */
public static final String DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String PROJECT_NAME = "projectName";
/** 日当名称. */
public static final String ALLOWANCE_NAME = "allowanceName";
/** 経費区分名称. */
public static final String EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String TAX_TYPE_NAME = "taxTypeName";
}

```

Manager

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/AllowanceDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 日当明細がジェットマネージャクラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface AllowanceDetailManager extends PluginGadgetManager<AllowanceDetailManagerDto> {

    /** {@link #selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link #selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link #selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細登録データ検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager(実装)

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/impl/AllowanceDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tAllowanceDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tAllowanceDetailService;
import org.seasar.util.beans.util.BeanMap;

```

```

/**
 * 日当明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link AllowanceDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link AllowanceDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>selectByPk</td>
 * <td>主キー検索</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>リスト検索</td>
 * <td>selectList</td>
 * <td>主キー検索</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>表示案件検索</td>
 * <td>selectView</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>コピー案件検索</td>
 * <td>selectViewForCopy</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件検索</td>
 * <td>selectViewArchive</td>
 * <td>リスト検索(マスタ情報付与)</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括挿入</td>
 * <td>entry</td>
 * <td>登録</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括更新</td>
 * <td>renew</td>
 * <td>更新</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>削除</td>
 * <td>remove</td>
 * <td>削除</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>一括削除</td>
 * <td>removeAll</td>
 * <td>削除</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件登録</td>
 * <td>matterArchiveRegister</td>
 * <td>登録</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件処理時削除</td>
 * <td>matterArchivePostDelete</td>
 * <td>削除</td>
 * <td><b><nbsp;</b></td>
 * </tr>
 * <tr>
 * <td>過去案件一括削除</td>
 * <td>removeAllArchive</td>
 * <td>削除</td>
 * <td><b><nbsp;</b></td>
 * </tr>

```

[illegible]

```

<td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center"></td>
* </tr>
* <tr>
* <td colspan="2">{@link AllowanceDetailManagerDto}</td>
* <td align="center">O</td>
* <td align="center">O</td>
* <td align="center">O</td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* <td align="center">O</td>
* <td align="center"></td>
* </tr>
* </table>
* <br>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

```

```

public class AllowanceDetailManagerImpl extends AbstractWfManager<AllowanceDetailManagerDto> implements

```

```

    AllowanceDetailManager {

```

```

    /** K30tAllowanceDetailService. */

```

```

    @Resource

```

```

    protected K30tAllowanceDetailService k30tAllowanceDetailService;

```

```

    /** Ka30tAllowanceDetailService. */

```

```

    @Resource

```

```

    protected Ka30tAllowanceDetailService ka30tAllowanceDetailService;

```

```

    /**

```

```

     * {@inheritDoc}

```

```

     */

```

```

    @Override

```

```

    public AllowanceDetailManagerDto selectByPk(AllowanceDetailManagerDto criteria) {

```

```

        return super.selectByPkIgnoreAuthority(k30tAllowanceDetailService, criteria);

```

```

    }

```

```

    /**

```

```

     * {@inheritDoc}

```

```

     */

```

```

    @Override

```

```

    public List<AllowanceDetailManagerDto> selectList(AllowanceDetailManagerDto criteria) {

```

```

        return super.selectPkUsedUserDataId(k30tAllowanceDetailService, criteria);

```

```

    }

```

```

    /**

```

```

     * {@inheritDoc}

```

```

     */

```

```

    @Override

```

```

    public List<AllowanceDetailManagerDto> selectView(CriteriaForSqlDto criteria) {

```

```

        List<OrderDto> order = new ArrayList<OrderDto>();

```

```

        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

```

```

        return super.selectBySqlForView(k30tAllowanceDetailService, SELECT_SQLFILE_NM, criteria, order);

```

```

    }

```

```

    /** {@inheritDoc} */

```

```

    @Override

```

```

    public List<AllowanceDetailManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {

```

```

        List<OrderDto> order = new ArrayList<OrderDto>();

```

```

        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

```

```

        return super.selectBySqlForViewForCopy(k30tAllowanceDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);

```

```

    }

```

```

    /**

```

```

    /**
     * {@inheritDoc}
     */
    @Override
    public List<AllowanceDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectListForViewForArchive(ka30tAllowanceDetailService, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

        return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<AllowanceDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.insertBatch(k30tAllowanceDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<AllowanceDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.renewBatch(k30tAllowanceDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int remove(AllowanceDetailManagerDto data) {
        return super.deleteUsedUserDataId(k30tAllowanceDetailService, data);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAll(AllowanceDetailManagerDto data) {
        return super.deleteNoUsedUserDataId(k30tAllowanceDetailService, data);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<AllowanceDetailManagerDto> dataList) {
        return super.entryArchive(ka30tAllowanceDetailService, criteria, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataId(k30tAllowanceDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataIdForArchive(ka30tAllowanceDetailService, criteria);
    }
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/AllowanceDetailManagerDto.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail;
```

```
import java.math.BigDecimal;
```

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 日当明細ガジェットマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailManagerDto extends KaidenDto {

    /** システム案件id. */
    public String systemMatterId;
    /** ユーザーデータid. */
    public String userDataId;
    /** ガジェットクラス. */
    public String gadgetClass;
    /** ガジェットインスタンス. */
    public Long gadgetInstanceId;
    /** タブルid. */
    public Long tupleId;
    /** ガジェットid. */
    public String gadgetId;
    /** ガジェットバージョン. */
    public String gadgetVariation;
    /** 行番号. */
    public Long rowNumber;
    /** 日当コード. */
    public String allowanceCd;
    /** 日数. */
    public BigDecimal quantity;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** 税率. */
    public BigDecimal taxRate;
    /** 取引通貨コード. */
    public String transCurrencyCd;
    /** 取引通貨額. */
    public BigDecimal transAmount;
    /** 会社通貨コード. */
    public String companyCurrencyCd;
    /** 会社通貨換算レート. */
    public BigDecimal companyRate;
    /** 会社通貨額. */
    public BigDecimal companyAmount;
    /** 会社通貨税抜額. */
    public BigDecimal companyAmountExcTax;
    /** 会社通貨税額. */
    public BigDecimal companyAmountTax;
    /** 会社通貨額合計. */
    public BigDecimal companyAmountSum;
    /** 支払分類コード. */
    public String payTypeCd;
    /** 支払方法コード. */
    public String payMethodCd;
    /** 支払先コード. */
    public String payeeCd;
    /** 支払先名. */
    public String payeeName;
    /** 組織セットコード. */
    public String departmentSetCd;
    /** 組織コード. */
    public String departmentCd;
    /** プロジェクトコード. */
    public String projectCd;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;

```

```

    public String extensionJ;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 取引通貨名称. */
    public String transCurrencyName;
    /** 会社通貨名称. */
    public String companyCurrencyName;
    /** 部門名称. */
    public String departmentName;
    /** プロジェクト名称. */
    public String projectName;
    /** 日当名称. */
    public String allowanceName;
    /** 経費区分名称. */
    public String expTypeName;
    /** 勘定科目名称. */
    public String accountName;
    /** 税区分名称. */
    public String taxTypeName;
}

```

Java(S2JDBC)

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tAllowanceDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 日当明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_allowance_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl"})
public final class K30tAllowanceDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstance;

    /** タブルid. */
    @Id

```

```

    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;

    /** 行番号. */
    @Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long rowNum;

    /** 日当コード. */
    @Column(name = "allowance_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String allowanceCd;

    /** 日数. */
    @Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal quantity;

    /** 経費区分コード. */
    @Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String expTypeCd;

    /** 勘定科目コード. */
    @Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String accountCd;

    /** 税区分コード. */
    @Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String taxTypeCd;

    /** 税率. */
    @Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal taxRate;

    /** 取引通貨コード. */
    @Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String transCurrencyCd;

    /** 取引通貨額. */
    @Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal transAmount;

    /** 会社通貨コード. */
    @Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String companyCurrencyCd;

    /** 会社通貨換算レート. */
    @Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyRate;

    /** 会社通貨額. */
    @Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmount;

    /** 会社通貨税抜額. */
    @Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountExcTax;

    /** 会社通貨税額. */
    @Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountTax;

    /** 会社通貨額合計. */
    @Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountSum;

    /** 支払分類コード. */
    @Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payTypeCd;

    /** 支払方法コード. */
    @Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payMethodCd;

    /** 支払先コード. */
    @Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payeeCd;

    /** 支払先名. */
    @Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String payeeName;

```



```

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tAllowanceDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity; implements Serializable {

/**
 * (過去案件) 日当明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_allowance_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tAllowanceDetail extends GenerateEntity implements Serializable {

/** シリアルバージョンUID. */
private static final long serialVersionUID = 1L;

/** システム案件id. */
@Id
@Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
public String systemMatterId;

```

```

/** ユーザーデータid. */
@Id
@Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
public String userDataId;

/** ガジェットクラス. */
@Id
@Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetClass;

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** アーカイブ年月. */
@Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
public String archiveMonth;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 日当コード. */
@Column(name = "allowance_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String allowanceCd;

/** 日当名称. */
@Column(name = "allowance_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String allowanceName;

/** 日数. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */

```

```

    /** 会社通員額. */
    @Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmount;

    /** 会社通貨税抜額. */
    @Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountExcTax;

    /** 会社通貨税額. */
    @Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountTax;

    /** 会社通貨額合計. */
    @Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
    public BigDecimal companyAmountSum;

    /** 支払分類コード. */
    @Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payTypeCd;

    /** 支払方法コード. */
    @Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payMethodCd;

    /** 支払先コード. */
    @Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String payeeCd;

    /** 支払先名. */
    @Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String payeeName;

    /** 組織セットコード. */
    @Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String departmentSetCd;

    /** 組織コード. */
    @Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String departmentCd;

    /** プロジェクトコード. */
    @Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String projectCd;

    /** 経費区分名称. */
    @Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String expTypeName;

    /** 勘定科目名称. */
    @Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String accountName;

    /** 税区分名. */
    @Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String taxTypeName;

    /** 取引通貨名. */
    @Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String transCurrencyName;

    /** 会社通貨名. */
    @Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String companyCurrencyName;

    /** 支払分類名称. */
    @Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String payTypeName;

    /** 立替フラグ. */
    @Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
    public String paymentFlag;

    /** 支払方法名称. */
    @Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String payMethodName;

    /** 組織名. */
    @Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String departmentName;

    /** プロジェクト名. */
    @Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String projectName;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

```

```

public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tAllowanceDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tAllowanceDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tAllowanceDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tAllowanceDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * codeIdのプロパティ名を返却.
     */
}

```

```

    * gadgetClassのプロパティ名を返却。
    *
    * @return gadgetClassのプロパティ名
    */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却。
     *
     * @return gadgetInstanceのプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumのプロパティ名を返却。
     *
     * @return rowNumのプロパティ名
     */
    public static PropertyName<Long> rowNum() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * allowanceCdのプロパティ名を返却。
     *
     * @return allowanceCdのプロパティ名
     */
    public static PropertyName<String> allowanceCd() {
        return new PropertyName<String>("allowanceCd");
    }

    /**
     * quantityのプロパティ名を返却。
     *
     * @return quantityのプロパティ名
     */
    public static PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>("quantity");
    }

    /**
     * expTypeCdのプロパティ名を返却。
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却。
     *
     * @return accountCdのプロパティ名
     */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

```

```

    return new PropertyName<String>( accountCd );
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *

```

```

    * @return companyAmountSumのプロパティ名
    */
    public static PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>("companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却。
     *
     * @return payTypeCdのプロパティ名
     */
    public static PropertyName<String> payTypeCd() {
        return new PropertyName<String>("payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却。
     *
     * @return payMethodCdのプロパティ名
     */
    public static PropertyName<String> payMethodCd() {
        return new PropertyName<String>("payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却。
     *
     * @return payeeCdのプロパティ名
     */
    public static PropertyName<String> payeeCd() {
        return new PropertyName<String>("payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却。
     *
     * @return payeeNameのプロパティ名
     */
    public static PropertyName<String> payeeName() {
        return new PropertyName<String>("payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却。
     *
     * @return departmentSetCdのプロパティ名
     */
    public static PropertyName<String> departmentSetCd() {
        return new PropertyName<String>("departmentSetCd");
    }

    /**
     * departmentCdのプロパティ名を返却。
     *
     * @return departmentCdのプロパティ名
     */
    public static PropertyName<String> departmentCd() {
        return new PropertyName<String>("departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却。
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * extensionAのプロパティ名を返却。
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却。
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }
}

```

```

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名

```



```

*/
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30tAllowanceDetailNames extends PropertyName<K30tAllowanceDetail> {

    /**
     * インスタンスを構築.
     */
    public _K30tAllowanceDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30tAllowanceDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30tAllowanceDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>(this, "systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public PropertyName<String> userDataId() {
        return new PropertyName<String>(this, "userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public PropertyName<String> gadgetClass() {

```

```

    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * allowanceCdのプロパティ名を返却.
 *
 * @return allowanceCdのプロパティ名
 */
public PropertyName<String> allowanceCd() {
    return new PropertyName<String>(this, "allowanceCd");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.

```

```

*
* @return taxTypeCdのプロパティ名
*/
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
* taxRateのプロパティ名を返却.
*
* @return taxRateのプロパティ名
*/
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
* transCurrencyCdのプロパティ名を返却.
*
* @return transCurrencyCdのプロパティ名
*/
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
* transAmountのプロパティ名を返却.
*
* @return transAmountのプロパティ名
*/
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
* companyCurrencyCdのプロパティ名を返却.
*
* @return companyCurrencyCdのプロパティ名
*/
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
* companyRateのプロパティ名を返却.
*
* @return companyRateのプロパティ名
*/
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

/**
* companyAmountのプロパティ名を返却.
*
* @return companyAmountのプロパティ名
*/
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
* companyAmountExcTaxのプロパティ名を返却.
*
* @return companyAmountExcTaxのプロパティ名
*/
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
* companyAmountTaxのプロパティ名を返却.
*
* @return companyAmountTaxのプロパティ名
*/
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
* companyAmountSumのプロパティ名を返却.
*
* @return companyAmountSumのプロパティ名
*/
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

```

```

}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *

```

```

    * @return extensionCのプロパティ名
    */
    public PropertyName<String> extensionC() {
        return new PropertyName<String>(this, "extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

```

```

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tAllowanceDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tAllowanceDetail}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tAllowanceDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tAllowanceDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却.
     *
     * @return gadgetInstanceのプロパティ名
     */
}

```

```

*/
public static PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>("gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public static PropertyName<String> localeId() {
    return new PropertyName<String>("localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public static PropertyName<Long> rowNumber() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * allowanceCdのプロパティ名を返却.
 *
 * @return allowanceCdのプロパティ名
 */
public static PropertyName<String> allowanceCd() {
    return new PropertyName<String>("allowanceCd");
}

/**
 * allowanceNameのプロパティ名を返却.
 *
 * @return allowanceNameのプロパティ名
 */
public static PropertyName<String> allowanceName() {
    return new PropertyName<String>("allowanceName");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

```

```

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */

```



```

public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**

```

```

    * expTypeNameのプロパティ名を返却.
    *
    * @return expTypeNameのプロパティ名
    */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却.
     *
     * @return accountNameのプロパティ名
     */
    public static PropertyName<String> accountName() {
        return new PropertyName<String>("accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却.
     *
     * @return taxTypeNameのプロパティ名
     */
    public static PropertyName<String> taxTypeName() {
        return new PropertyName<String>("taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却.
     *
     * @return transCurrencyNameのプロパティ名
     */
    public static PropertyName<String> transCurrencyName() {
        return new PropertyName<String>("transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却.
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public static PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>("companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却.
     *
     * @return payTypeNameのプロパティ名
     */
    public static PropertyName<String> payTypeName() {
        return new PropertyName<String>("payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却.
     *
     * @return paymentFlagのプロパティ名
     */
    public static PropertyName<String> paymentFlag() {
        return new PropertyName<String>("paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却.
     *
     * @return payMethodNameのプロパティ名
     */
    public static PropertyName<String> payMethodName() {
        return new PropertyName<String>("payMethodName");
    }

    /**
     * departmentNameのプロパティ名を返却.
     *
     * @return departmentNameのプロパティ名
     */
    public static PropertyName<String> departmentName() {
        return new PropertyName<String>("departmentName");
    }

    /**
     * projectNameのプロパティ名を返却.
     *
     * @return projectNameのプロパティ名
     */
    public static PropertyName<String> projectName() {

```

```

    return new PropertyName<String>("projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *

```

```

    * @return extensionJのプロパティ名
    */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _Ka30tAllowanceDetailNames extends PropertyName<Ka30tAllowanceDetail> {

        /**
         * インスタンスを構築.
         */
        public _Ka30tAllowanceDetailNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _Ka30tAllowanceDetailNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _Ka30tAllowanceDetailNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * systemMatterIdのプロパティ名を返却.
         *
         * @return systemMatterIdのプロパティ名
         */

```

```

/
public PropertyName<String> systemMatterId() {
    return new PropertyName<String>(this, "systemMatterId");
}

/**
 * userDataIdのプロパティ名を返却.
 *
 * @return userDataIdのプロパティ名
 */
public PropertyName<String> userDataId() {
    return new PropertyName<String>(this, "userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>(this, "archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**

```

```

     * allowanceCdのプロパティ名を返却.
     *
     * @return allowanceCdのプロパティ名
     */
    public PropertyName<String> allowanceCd() {
        return new PropertyName<String>(this, "allowanceCd");
    }

    /**
     * allowanceNameのプロパティ名を返却.
     *
     * @return allowanceNameのプロパティ名
     */
    public PropertyName<String> allowanceName() {
        return new PropertyName<String>(this, "allowanceName");
    }

    /**
     * quantityのプロパティ名を返却.
     *
     * @return quantityのプロパティ名
     */
    public PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>(this, "quantity");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public PropertyName<String> accountCd() {
        return new PropertyName<String>(this, "accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public PropertyName<String> taxTypeCd() {
        return new PropertyName<String>(this, "taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {

```

```

    private PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }

    /**
     * payMethodCdのプロパティ名を返却.
     *
     * @return payMethodCdのプロパティ名
     */
    public PropertyName<String> payMethodCd() {
        return new PropertyName<String>(this, "payMethodCd");
    }

    /**
     * payeeCdのプロパティ名を返却.
     *
     * @return payeeCdのプロパティ名
     */
    public PropertyName<String> payeeCd() {
        return new PropertyName<String>(this, "payeeCd");
    }

    /**
     * payeeNameのプロパティ名を返却.
     *
     * @return payeeNameのプロパティ名
     */
    public PropertyName<String> payeeName() {
        return new PropertyName<String>(this, "payeeName");
    }

    /**
     * departmentSetCdのプロパティ名を返却.

```

```

*
* @return departmentSetCdのプロパティ名
*/
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
* departmentCdのプロパティ名を返却.
*
* @return departmentCdのプロパティ名
*/
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
* projectCdのプロパティ名を返却.
*
* @return projectCdのプロパティ名
*/
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
* expTypeNameのプロパティ名を返却.
*
* @return expTypeNameのプロパティ名
*/
public PropertyName<String> expTypeName() {
    return new PropertyName<String>(this, "expTypeName");
}

/**
* accountNameのプロパティ名を返却.
*
* @return accountNameのプロパティ名
*/
public PropertyName<String> accountName() {
    return new PropertyName<String>(this, "accountName");
}

/**
* taxTypeNameのプロパティ名を返却.
*
* @return taxTypeNameのプロパティ名
*/
public PropertyName<String> taxTypeName() {
    return new PropertyName<String>(this, "taxTypeName");
}

/**
* transCurrencyNameのプロパティ名を返却.
*
* @return transCurrencyNameのプロパティ名
*/
public PropertyName<String> transCurrencyName() {
    return new PropertyName<String>(this, "transCurrencyName");
}

/**
* companyCurrencyNameのプロパティ名を返却.
*
* @return companyCurrencyNameのプロパティ名
*/
public PropertyName<String> companyCurrencyName() {
    return new PropertyName<String>(this, "companyCurrencyName");
}

/**
* payTypeNameのプロパティ名を返却.
*
* @return payTypeNameのプロパティ名
*/
public PropertyName<String> payTypeName() {
    return new PropertyName<String>(this, "payTypeName");
}

/**
* paymentFlagのプロパティ名を返却.
*
* @return paymentFlagのプロパティ名
*/
public PropertyName<String> paymentFlag() {
    return new PropertyName<String>(this, "paymentFlag");
}

```



```

}

/**
 * payMethodNameのプロパティ名を返却.
 *
 * @return payMethodNameのプロパティ名
 */
public PropertyName<String> payMethodName() {
    return new PropertyName<String>(this, "payMethodName");
}

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *

```

```

    * @return extensionGのプロパティ名
    */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tAllowanceDetail;

/**
 * {@link K30tAllowanceDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tAllowanceDetailService extends GenerateService<K30tAllowanceDetail> {
    //
}
```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tAllowanceDetailService.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tAllowanceDetail;

/**
 * {@link Ka30tAllowanceDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tAllowanceDetailService extends GenerateService<Ka30tAllowanceDetail> {
    //
}
```

Condition

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailCondition.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30tAllowanceDetail} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tAllowanceDetailCondition extends
    AbstractEntityCondition<K30tAllowanceDetailCondition> {

    /**
     * インスタンスを構築。
     */
    public K30tAllowanceDetailCondition() {
    }

    /**
     * インスタンスを構築。
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tAllowanceDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterId の条件。 */
    public NotNulltableStringCondition<K30tAllowanceDetailCondition> systemMatterId =
        new NotNulltableStringCondition<K30tAllowanceDetailCondition>("systemMatterId", this);

    /** userDatId の条件。 */
    public NotNulltableStringCondition<K30tAllowanceDetailCondition> userDatId =
        new NotNulltableStringCondition<K30tAllowanceDetailCondition>("userDatId", this);
}
```

```

    new NullableStringCondition<K30tAllowanceDetailCondition> ( gadgetClass , this);

    /** gadgetClassの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetClass =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> gadgetInstance =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> tupleId =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("tupleId", this);

    /** gadgetIdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetId =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetId", this);

    /** gadgetVariationの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetVariation =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> rowNumber =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("rowNumber", this);

    /** allowanceCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> allowanceCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("allowanceCd", this);

    /** quantityの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> quantity =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("quantity", this);

    /** expTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> expTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("expTypeCd", this);

    /** accountCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> accountCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("accountCd", this);

    /** taxTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> taxTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("taxTypeCd", this);

    /** taxRateの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> taxRate =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("taxRate", this);

    /** transCurrencyCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> transCurrencyCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("transCurrencyCd", this);

    /** transAmountの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> transAmount =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("transAmount", this);

    /** companyCurrencyCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> companyCurrencyCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("companyCurrencyCd", this);

    /** companyRateの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyRate =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyRate", this);

    /** companyAmountの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmount =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmount", this);

    /** companyAmountExcTaxの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountExcTax =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountExcTax", this);

    /** companyAmountTaxの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountTax =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountTax", this);

    /** companyAmountSumの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountSum =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountSum", this);

    /** payTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> payTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("payTypeCd", this);

    /** payMethodCdの条件. */

```

```

public NullableStringCondition<K30tAllowanceDetailCondition> payMethodCd =
    new NullableStringCondition<K30tAllowanceDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> payeeCd =
    new NullableStringCondition<K30tAllowanceDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> payeeName =
    new NullableStringCondition<K30tAllowanceDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> departmentSetCd =
    new NullableStringCondition<K30tAllowanceDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> departmentCd =
    new NullableStringCondition<K30tAllowanceDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> projectCd =
    new NullableStringCondition<K30tAllowanceDetailCondition>("projectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionA =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionB =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionC =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionD =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionE =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionF =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionG =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionH =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionI =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30tAllowanceDetailCondition> extensionJ =
    new NullableStringCondition<K30tAllowanceDetailCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30tAllowanceDetailCondition, Timestamp> entryTs =
    new NotNullCondition<K30tAllowanceDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullStringCondition<K30tAllowanceDetailCondition> entryUserCd =
    new NotNullStringCondition<K30tAllowanceDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30tAllowanceDetailCondition, Long> renewCnt =
    new NotNullCondition<K30tAllowanceDetailCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30tAllowanceDetailCondition, Timestamp> renewTs =
    new NotNullCondition<K30tAllowanceDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30tAllowanceDetailCondition> renewUserCd =
    new NotNullStringCondition<K30tAllowanceDetailCondition>("renewUserCd", this);
}

```

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tAllowanceDetail} の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tAllowanceDetailCondition extends
    AbstractEntityCondition<Ka30tAllowanceDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tAllowanceDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tAllowanceDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterId の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> systemMatterId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("systemMatterId", this);

    /** userDataId の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> userDataId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("userDataId", this);

    /** gadgetClass の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetClass =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetClass", this);

    /** gadgetInstance の条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> gadgetInstance =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("gadgetInstance", this);

    /** tupleId の条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> tupleId =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("tupleId", this);

    /** localeId の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> localeId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("localeId", this);

    /** archiveMonth の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> archiveMonth =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("archiveMonth", this);

    /** gadgetId の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetId", this);

    /** gadgetVariation の条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetVariation =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetVariation", this);

    /** rowNumber の条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> rowNumber =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("rowNumber", this);

    /** allowanceCd の条件. */
    public NullableStringCondition<Ka30tAllowanceDetailCondition> allowanceCd =
        new NullableStringCondition<Ka30tAllowanceDetailCondition>("allowanceCd", this);

    /** allowanceName の条件. */
    public NullableStringCondition<Ka30tAllowanceDetailCondition> allowanceName =
        new NullableStringCondition<Ka30tAllowanceDetailCondition>("allowanceName", this);

```

```

/** quantityの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("quantity", this);

/** expTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> accountCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("taxRate", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payeeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> projectCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("expTypeName", this);

/** accountNameの条件. */

```

```

public NullableStringCondition<Ka30tAllowanceDetailCondition> accountName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentName", this);

/** projectNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> projectName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionA =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionB =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionC =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionD =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionE =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionF =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionG =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionH =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionI =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionJ =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp> entryTs =
    new NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tAllowanceDetailCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tAllowanceDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<Ka30tAllowanceDetailCondition, Long> renewCnt =
    new NotNullableCondition<Ka30tAllowanceDetailCondition, Long>("renewCnt", this);

```



```

/** renewTsの条件. */
public NotNullCondition<Ka30tAllowanceDetailCondition, Timestamp> renewTs =
    new NotNullCondition<Ka30tAllowanceDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<Ka30tAllowanceDetailCondition> renewUserCd =
    new NotNullStringCondition<Ka30tAllowanceDetailCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService/selectView.sql

```

select
    allowance_t.system_matter_id,
    allowance_t.user_data_id,
    allowance_t.gadget_class,
    allowance_t.gadget_instance,
    allowance_t.tuple_id,
    allowance_t.gadget_id,
    allowance_t.gadget_variation,
    allowance_t.row_number,
    allowance_t.allowance_cd,
    allowance_t.quantity,
    allowance_t.exp_type_cd,
    allowance_t.account_cd,
    allowance_t.tax_type_cd,
    allowance_t.tax_rate,
    allowance_t.trans_currency_cd,
    allowance_t.trans_amount,
    allowance_t.company_currency_cd,
    allowance_t.company_rate,
    allowance_t.company_amount,
    allowance_t.company_amount_exc_tax,
    allowance_t.company_amount_tax,
    allowance_t.company_amount_sum,
    allowance_t.pay_type_cd,
    allowance_t.pay_method_cd,
    allowance_t.payee_cd,
    allowance_t.payee_name,
    allowance_t.department_set_cd,
    allowance_t.department_cd,
    allowance_t.project_cd,
    allowance_t.extension_a,
    allowance_t.extension_b,
    allowance_t.extension_c,
    allowance_t.extension_d,
    allowance_t.extension_e,
    allowance_t.extension_f,
    allowance_t.extension_g,
    allowance_t.extension_h,
    allowance_t.extension_i,
    allowance_t.extension_j,
    allowance_t.entry_ts,
    allowance_t.entry_user_cd,
    allowance_t.renew_cnt,
    allowance_t.renew_ts,
    allowance_t.renew_user_cd,
    c1.search_criteria_company,
    cu1.currency_name as trans_currency_name,
    cu2.currency_name as company_currency_name,
    dept1.department_name,
    pro1.project_name,
    allowance_m.allowance_name,
    exp.exp_type_name,
    account.account_name,
    tax.tax_type_name
from
    k30t_allowance_detail allowance_t
inner join
    k20c_matter_control c1
on
    allowance_t.system_matter_id = c1.system_matter_id
and
    allowance_t.user_data_id = c1.user_data_id
left outer join imm_currency cu1
on
    c1.search_criteria_company = cu1.company_cd
and
    allowance_t.trans_currency_cd = cu1.currency_cd

```

```

and
  /*localeld*/ja' = cu1.locale_id
and
  cu1.delete_flag = '0'
left outer join imm_currency cu2
on
  c1.search_criteria_company = cu2.company_cd
and
  allowance_t.company_currency_cd = cu2.currency_cd
and
  /*localeld*/ja' = cu2.locale_id
and
  cu2.delete_flag = '0'
left outer join
  imm_department dept1
on
  c1.search_criteria_company = dept1.company_cd
and
  allowance_t.department_set_cd = dept1.department_set_cd
and
  allowance_t.department_cd = dept1.department_cd
and
  /*localeld*/ja' = dept1.locale_id
and
  c1.search_criteria_date >= dept1.start_date
and
  c1.search_criteria_date < dept1.end_date
and
  dept1.delete_flag = '0'
left outer join
  k30m_project pro1
on
  c1.search_criteria_company = pro1.company_cd
and
  allowance_t.project_cd = pro1.project_cd
and
  c1.search_criteria_date >= pro1.start_date
and
  c1.search_criteria_date < pro1.end_date
and
  /*localeld*/ja' = pro1.locale_id
and
  pro1.delete_flag = '0'
left outer join
  k30m_allowance allowance_m
on
  c1.search_criteria_company = allowance_m.company_cd
and
  allowance_t.allowance_cd = allowance_m.allowance_cd
and
  /*localeld*/ja' = allowance_m.locale_id
and
  c1.search_criteria_date >= allowance_m.start_date
and
  c1.search_criteria_date < allowance_m.end_date
and
  allowance_m.delete_flag = '0'
left outer join
  k30m_exp_type exp
on
  c1.search_criteria_company = exp.company_cd
and
  allowance_t.exp_type_cd = exp.exp_type_cd
and
  /*localeld*/ja' = exp.locale_id
and
  c1.search_criteria_date >= exp.start_date
and
  c1.search_criteria_date < exp.end_date
and
  exp.delete_flag = '0'
left outer join
  k30m_account account
on
  c1.search_criteria_company = account.company_cd
and
  allowance_t.account_cd = account.account_cd
and
  /*localeld*/ja' = account.locale_id
and
  c1.search_criteria_date >= account.start_date
and
  c1.search_criteria_date < account.end_date
and
  account.delete_flag = '0'
left outer join

```

```

k30m_tax_type tax
on
  c1.search_criteria_company = tax.company_cd
and
  allowance_t.tax_type_cd = tax.tax_type_cd
and
  /*localeId*ja' = tax.locale_id
and
  c1.search_criteria_date >= tax.start_date
and
  c1.search_criteria_date < tax.end_date
and
  tax.delete_flag = '0'
where
  allowance_t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  allowance_t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
  allowance_t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  allowance_t.gadget_instance = /*gadgetInstance*/0
/*END*/
/*IF gadgetId != null*/
and
  allowance_t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

コピー

- Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService/selectViewForCopy.sql

```

select
  allowance_t.system_matter_id,
  allowance_t.user_data_id,
  allowance_t.gadget_class,
  allowance_t.gadget_instance,
  allowance_t.tuple_id,
  allowance_t.gadget_id,
  allowance_t.gadget_variation,
  allowance_t.row_number,
  allowance_t.allowance_cd,
  allowance_t.quantity,
  allowance_t.exp_type_cd,
  allowance_t.account_cd,
  allowance_t.tax_type_cd,
  allowance_t.tax_rate,
  allowance_t.trans_currency_cd,
  allowance_t.trans_amount,
  allowance_t.company_currency_cd,
  allowance_t.company_rate,
  allowance_t.company_amount,
  allowance_t.company_amount_exc_tax,
  allowance_t.company_amount_tax,
  allowance_t.company_amount_sum,
  allowance_t.pay_type_cd,
  allowance_t.pay_method_cd,
  allowance_t.payee_cd,
  allowance_t.payee_name,
  allowance_t.department_set_cd,
  allowance_t.department_cd,
  allowance_t.project_cd,
  allowance_t.extension_a,
  allowance_t.extension_b,
  allowance_t.extension_c,
  allowance_t.extension_d,
  allowance_t.extension_e,
  allowance_t.extension_f,
  allowance_t.extension_g,
  allowance_t.extension_h,
  allowance_t.extension_i,
  allowance_t.extension_j,
  allowance_t.entry_ts,

```

```

allowance_t.entry_user_cd,
allowance_t.renew_cnt,
allowance_t.renew_ts,
allowance_t.renew_user_cd,
c1.search_criteria_company,
cu1.currency_name as trans_currency_name,
cu2.currency_name as company_currency_name,
dept1.department_name,
pro1.project_name,
allowance_m.allowance_name,
exp.exp_type_name,
account.account_name,
tax.tax_type_name
from
k30t_allowance_detail allowance_t
inner join
k20c_matter_control c1
on
allowance_t.system_matter_id = c1.system_matter_id
and
allowance_t.user_data_id = c1.user_data_id
left outer join imm_currency cu1
on
c1.search_criteria_company = cu1.company_cd
and
allowance_t.trans_currency_cd = cu1.currency_cd
and
/*localeld*/ja' = cu1.locale_id
and
cu1.delete_flag = '0'
left outer join imm_currency cu2
on
c1.search_criteria_company = cu2.company_cd
and
allowance_t.company_currency_cd = cu2.currency_cd
and
/*localeld*/ja' = cu2.locale_id
and
cu2.delete_flag = '0'
left outer join
imm_department dept1
on
c1.search_criteria_company = dept1.company_cd
and
allowance_t.department_set_cd = dept1.department_set_cd
and
allowance_t.department_cd = dept1.department_cd
and
/*localeld*/ja' = dept1.locale_id
and
/*searchCriteriaDate*/" >= dept1.start_date
and
/*searchCriteriaDate*/" < dept1.end_date
and
dept1.delete_flag = '0'
left outer join
k30m_project pro1
on
c1.search_criteria_company = pro1.company_cd
and
allowance_t.project_cd = pro1.project_cd
and
/*searchCriteriaDate*/" >= pro1.start_date
and
/*searchCriteriaDate*/" < pro1.end_date
and
/*localeld*/ja' = pro1.locale_id
and
pro1.delete_flag = '0'
left outer join
k30m_allowance allowance_m
on
c1.search_criteria_company = allowance_m.company_cd
and
allowance_t.allowance_cd = allowance_m.allowance_cd
and
/*localeld*/ja' = allowance_m.locale_id
and
/*searchCriteriaDate*/" >= allowance_m.start_date
and
/*searchCriteriaDate*/" < allowance_m.end_date
and
allowance_m.delete_flag = '0'
left outer join
k30m_exp_type exp
on

```

```

c1.search_criteria_company = exp.company_cd
and
allowance_t.exp_type_cd = exp.exp_type_cd
and
/*localeId*/ja' = exp.locale_id
and
/*searchCriteriaDate*/" >= exp.start_date
and
/*searchCriteriaDate*/" < exp.end_date
and
exp.delete_flag = '0'
left outer join
k30m_account account
on
c1.search_criteria_company = account.company_cd
and
allowance_t.account_cd = account.account_cd
and
/*localeId*/ja' = account.locale_id
and
/*searchCriteriaDate*/" >= account.start_date
and
/*searchCriteriaDate*/" < account.end_date
and
account.delete_flag = '0'
left outer join
k30m_tax_type tax
on
c1.search_criteria_company = tax.company_cd
and
allowance_t.tax_type_cd = tax.tax_type_cd
and
/*localeId*/ja' = tax.locale_id
and
/*searchCriteriaDate*/" >= tax.start_date
and
/*searchCriteriaDate*/" < tax.end_date
and
tax.delete_flag = '0'
where
allowance_t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
allowance_t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
allowance_t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
allowance_t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
allowance_t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/impl/AllowanceDetailManagerImpl/selectForJournalSourceDtl

```

select
ad.system_matter_id      as system_matter_id,
ad.user_data_id          as user_data_id,
ad.gadget_class          as gadget_class,
ad.gadget_instance       as gadget_instance,
ad.gadget_id             as gadget_id,
ad.tuple_id              as tuple_id,
ad.gadget_variation       as gadget_variation,
i1.contents_id           as contents_id,
i1.contents_version_id    as contents_version_id,
'30'                     as journal_data_type_cd,
ad.row_number             as row_number,
ad.exp_type_cd            as exp_type_cd,
ad.account_cd            as account_cd,
ad.tax_type_cd           as tax_type_cd,

```

```

tax.tax_rate      as tax_rate,
ad.quantity       as quantity,
ad.trans_currency_cd as trans_currency_cd,
ad.trans_amount   as trans_amount,
ad.company_currency_cd as company_currency_cd,
ad.company_rate    as company_rate,
ad.company_amount  as company_amount,
ad.company_amount_exc_tax as company_amount_exc_tax,
ad.company_amount_tax as company_amount_tax,
ad.company_amount_sum as company_amount_sum,
ad.payee_cd        as customer_cd,
ad.payee_name      as customer_name,
ad.department_set_cd as department_set_cd,
ad.department_cd    as department_cd,
ad.project_cd       as project_cd,
ad.pay_method_cd    as pay_method_cd,
at.allowance_name   as detail_summary1,
ad.extension_a      as detail_extension_a,
ad.extension_b      as detail_extension_b,
ad.extension_c      as detail_extension_c,
ad.extension_d      as detail_extension_d,
ad.extension_e      as detail_extension_e,
ad.extension_f      as detail_extension_f,
ad.extension_g      as detail_extension_g,
ad.extension_h      as detail_extension_h,
ad.extension_i      as detail_extension_i,
ad.extension_j      as detail_extension_j,
'0'                as journal_status
from
k30t_allowance_detail ad
inner join
k20c_matter_info i1
on
ad.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
ad.system_matter_id = c1.system_matter_id
and
ad.user_data_id = c1.user_data_id
left outer join
k30m_tax_type tax
on
c1.search_criteria_company = tax.company_cd
and
ad.tax_type_cd = tax.tax_type_cd
and
i1.locale_id_apply = tax.locale_id
and
c1.search_criteria_date >= tax.start_date
and
c1.search_criteria_date < tax.end_date
and
tax.delete_flag = '0'
left outer join
k30m_allowance at
on
c1.search_criteria_company = at.company_cd
and
ad.allowance_cd = at.allowance_cd
and
i1.locale_id_apply = at.locale_id
and
c1.search_criteria_date >= at.start_date
and
c1.search_criteria_date < at.end_date
and
at.delete_flag = '0'
where
ad.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
ad.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
ad.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
ad.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
ad.gadget_id = /*gadgetId*/
/*END*/

```

```

/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/ad.row_number
/*END*/

```

JSP

PC

- Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/allowanceDetail/allowanceDetailV01.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant"%>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui"%>
<%@ taglib prefix="klist_wk" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow"%>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base"%>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow"%>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil"%>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp"%>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp"%>

<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"/></jsp:include>
  <set var="pwkfCommonIncluded" scope="request">true</set>
</c:if>

<!-- allowanceDetail v01 -->
<set var="gadgetClass">allowanceDetail</set>
<set var="gadgetVariation">v01</set>
<set var="gadgetInstance">${param.gadget_instance}</set>
<set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</set>
<set var="accessUrl">kaiden/gadget/allowanceDetailV01</set>
<set var="gadgetBlockHeaderId">allowanceDetailHeader</set>
<set var="tmtableId">allowanceDetailTable</set>
<set var="tmtableControllerBlockId">tmtableController</set>
<set var="searchCriteriaCompany">${param.searchCriteriaCompany}</set>
<set var="searchCriteriaDate">${param.searchCriteriaDate}</set>
<set var="companyCurrencyCd">${param.companyCurrencyCd}</set>
<set var="applyAuthUserCode">${param.applyAuthUserCode}</set>

<set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<set var="detail" value="${gadget.blocks[allowanceDetailTable]}" />

<!-- tmtable -->
<set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<!-- Message -->
<set var="msgRateReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0008)%></set>
<set var="msgAllowanceReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0009)%></set>
<set var="messageSearchWarning"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0010)%></set>
<set var="messageSearchWarningSub"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0011)%></set>

<!-- Surface -->
<set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</set>
<set var="surfaceAllowanceAmountReget">${gadget.surfaces["allowanceAmountReget"]}</set>
<set var="surfaceAllowance">${gadget.surfaces["allowance"]}</set>
<set var="surfaceTransAmount">${gadget.surfaces["amount"]}</set>
<set var="surfaceQuantity">${gadget.surfaces["quantity"]}</set>
<set var="surfaceCompanyAmountSum">${gadget.surfaces["companyAmountSum"]}</set>
<set var="surfaceExpType">${gadget.surfaces["expType"]}</set>
<set var="surfaceAccount">${gadget.surfaces["account"]}</set>
<set var="surfaceTaxType">${gadget.surfaces["taxType"]}</set>
<set var="surfaceDepartment">${gadget.surfaces["department"]}</set>
<set var="surfaceProject">${gadget.surfaces["project"]}</set>

<!-- Mode -->
<set var="modeAllowanceAmountReget">${detail.fields["allowanceCd"].inputType}</set>
<set var="modeAllowance">${detail.fields["allowanceCd"].inputType}</set>
<!-- <set var="modeTransAmount">${detail.fields["transAmount"].inputType}</set> -->
<set var="modeTransAmount">${detail.fields["allowanceCd"].inputType}</set>
<set var="modeQuantity">${detail.fields["quantity"].inputType}</set>
<!-- <set var="modeCompanyAmountSum">${detail.fields["companyAmountSum"].inputType}</set> -->
<set var="modeCompanyAmountSum">${modeTransAmount == modeQuantity ? modeTransAmount : "0"}</set>
<set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</set>
<set var="modeAccountCd">${detail.fields["accountCd"].inputType}</set>
<set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</set>
<set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</set>
<set var="modeProjectCd">${detail.fields["projectCd"].inputType}</set>

```

```

<%-- RequiredMark --%>
<c:set var="reqAllowance">${detail.fields["allowanceCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount">${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqQuantity">${detail.fields["quantity"].requiredFlag}</c:set>
<c:set var="reqCompanyAmountSum">${detail.fields["companyAmountSum"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd">${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd">${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd">${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd">${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd">${detail.fields["projectCd"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validAllowanceCd_def">{k_allowanceExists:['allowanceCd', 'searchCriteriaCompany', 'searchCriteriaDate', 'false'], k_allowanceAmount:['allowanceCd', 'searchCriteriaComp
<c:set var="validAllowanceCd_apply">{"true" == reqAllowance ? "k_required:[''] : ""}</c:set>
<c:set var="validAllowanceCd_tempsave"></c:set>
<c:set var="validAllowanceCd_reapply">{"true" == reqAllowance ? "k_required:[''] : ""}</c:set>
<c:set var="validAllowanceCd_process">{"true" == reqAllowance ? "k_required:[''] : ""}</c:set>

<c:set var="validQuantity_def">{k_numeric:[''], k_numericRange:['1','999']}</c:set>
<c:set var="validQuantity_apply">{"true" == reqQuantity ? "k_required:[''] : ""}</c:set>
<c:set var="validQuantity_tempsave"></c:set>
<c:set var="validQuantity_reapply">{"true" == reqQuantity ? "k_required:[''] : ""}</c:set>
<c:set var="validQuantity_process">{"true" == reqQuantity ? "k_required:[''] : ""}</c:set>

<c:set var="validCompanyAmountSum_def">{k_currencyAmnt:['', 'searchCriteriaCompany']}</c:set>
<c:set var="validCompanyAmountSum_apply">{"true" == reqCompanyAmountSum ? "k_required:[''] : ""}</c:set>
<c:set var="validCompanyAmountSum_tempsave"></c:set>
<c:set var="validCompanyAmountSum_reapply">{"true" == reqCompanyAmountSum ? "k_required:[''] : ""}</c:set>
<c:set var="validCompanyAmountSum_process">{"true" == reqCompanyAmountSum ? "k_required:[''] : ""}</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:['expTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validExpTypeCd_apply">{"true" == reqExpTypeCd ? "k_required:[''] : ""}</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">{"true" == reqExpTypeCd ? "k_required:[''] : ""}</c:set>
<c:set var="validExpTypeCd_process">{"true" == reqExpTypeCd ? "k_required:[''] : ""}</c:set>

<c:set var="validAccountCd_def">{k_accountExists:['accountCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validAccountCd_apply">{"true" == reqAccountCd ? "k_required:[''] : ""}</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">{"true" == reqAccountCd ? "k_required:[''] : ""}</c:set>
<c:set var="validAccountCd_process">{"true" == reqAccountCd ? "k_required:[''] : ""}</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:['taxTypeCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validTaxType_apply">{"true" == reqTaxTypeCd ? "k_required:[''] : ""}</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">{"true" == reqTaxTypeCd ? "k_required:[''] : ""}</c:set>
<c:set var="validTaxType_process">{"true" == reqTaxTypeCd ? "k_required:[''] : ""}</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:['searchCriteriaCompany', 'departmentSetCd', 'departmentCd', 'searchCriteriaDate']}</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<c:set var="validProjectCd_def">{k_projectExists:['projectCd', 'searchCriteriaCompany', 'searchCriteriaDate']}</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:['departmentCd', 'projectCd']}</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${surfaceChapterTitle}"
    gadgetId="${gadgetId}"
    hideGadgetTitle="true"
    style="width:850px;overflow:hidden;">
<k:blockContainer id="${gadgetBlockHeaderId}">
<div class="imui-chapter-title">
<h2 style="display:inline-block;width:50%">
<k:surface
    name="chapterTitle"
    surfaceKey="${surfaceChapterTitle}"
/>
</h2>
<k:a
    class="kaiden-ui-icon-refresh-gray16_nosize"
    name="allowanceAmountReget"
    mode="${modeAllowanceAmountReget}"
    text="${surfaceAllowanceAmountReget}"
    style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem"
/>
</div>
</k:blockContainer>
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false">

```



```

<:form id="gaugel_form" onSubmit="return false;">
<:k:tmtable id="{tmtableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumTarget="rowNumber">
  <:colgroup>
    <:col style="width: 40px; overflow: hidden">
    <:col style="width: 235px; overflow: hidden">
    <:col style="width: 185px; overflow: hidden">
    <:col style="width: 185px; overflow: hidden">
    <:col style="width: 60px; overflow: hidden">
    <:col style="width: 125px; overflow: hidden">
  </colgroup>
  <:thead>
    <:tr>
      <:td rowspan="2" style="vertical-align: middle;">
        <:k:tmtableOperationPanel
          position="header"
          allowAdd="{allowAddDetail}"
          allowDel="{allowDelDetail}"
        />
      </td>
      <:th>
        <:k:surface
          name="allowanceSurface"
          surfaceKey="{surfaceAllowance}"
          requiredMark="{reqAllowance}"
        />
      </th>
      <:th colspan="2">
        <:k:surface
          name="transAmountSurface"
          surfaceKey="{surfaceTransAmount}"
          requiredMark="{reqTransAmount}"
        />
      </th>
      <:th>
        <:k:surface
          name="quantitySurface"
          surfaceKey="{surfaceQuantity}"
          requiredMark="{reqQuantity}"
        />
      </th>
      <:th>
        <:k:surface
          name="companyAmountSumSurface"
          surfaceKey="{surfaceCompanyAmouontSum}"
          requiredMark="{reqCompanyAmountSum}"
        />
      </th>
    </tr>
    <:tr>
      <:th colspan="2">
        <:k:surface
          name="expTypeSurface"
          surfaceKey="{surfaceExpType}"
          style="display:inline;"
          requiredMark="{reqExpTypeCd}"
        />
        <:k:surface
          name="accountSurface"
          surfaceKey="{surfaceAccount}"
          style="display:inline;"
        />
        <:k:surface
          name="taxTypeSurface"
          surfaceKey="{surfaceTaxType}"
          style="display:inline;"
        />
      </th>
      <:th>
        <:k:surface
          name="departmentSurface"
          surfaceKey="{surfaceDepartment}"
          requiredMark="{reqDepartmentCd}"
        />
      </th>
      <:th colspan="2">
        <:k:surface
          name="projectSurface"
          surfaceKey="{surfaceProject}"
          requiredMark="{reqProjectCd}"
        />
      </th>
    </tr>
  </thead>
  <:tbody>
    <:tr>
      <:td rowspan="2">
        <:k:tmtableOperationPanel

```

```

        position="detail"
        allowAdd="{allowAddDetail}"
        allowDel="{allowDelDetail}"
        name="rowNumber"
    />
</td>
<td>
    <k:hidden
        name="allowanceCd"
        mode="0"
    />
    <kmsw:allowance
        name="allowanceName"
        mode="{modeAllowance}"
        imeMode="on"
        surfaceKey="{surfaceAllowance}"
        resultValue="allowanceName"
        resultChain="{transAmount:'amount', 'allowanceCd':allowanceCd}"
        primaryTarget="allowanceCd"
        currencyCd="{param.companyCurrencyCd}"
        allowanceCriteriaUserCd="{applyAuthUserCode}"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaAllowanceCd="allowanceCd"
        style="width:150px;"
        validationDefault="{validAllowanceCd_def}"
        validationApply="{validAllowanceCd_apply}"
        validationTemsave="{validAllowanceCd_tempsave}"
        validationReapply="{validAllowanceCd_reapply}"
        validationProcess="{validAllowanceCd_process}"
    />
</td>
<td colspan="2">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericAmountInputParts.jsp">
        <jsp:param name="gadgetId" value="{gadgetId}" />
        <jsp:param name="tmtableId" value="{tmtableId}" />
        <jsp:param name="searchCriteriaCompany" value="{searchCriteriaCompany}" />
        <%-- 金額 --%>
        <jsp:param name="modeTransAmnt" value="{modeTransAmount}" />
        <jsp:param name="surfaceTransAmnt" value="{surfaceTransAmount}" />
        <jsp:param name="requiredTransAmntApply" value="{reqTransAmount}" />
        <jsp:param name="requiredTransAmntReapply" value="{reqTransAmount}" />
        <jsp:param name="requiredTransAmntProcess" value="{reqTransAmount}" />
    </jsp:include>
</td>
<td>
    <k:numericText
        name="quantity"
        mode="{modeQuantity}"
        surfaceKey="{surfaceQuantity}"
        style="width:40px;"
        maxlength="3"
        validationDefault="{validQuantity_def}"
        validationApply="{validQuantity_apply}"
        validationTemsave="{validQuantity_tempsave}"
        validationReapply="{validQuantity_reapply}"
        validationProcess="{validQuantity_process}"
    />
</td>
<td style="text-align:right">
    <k:numericText
        name="companyAmountSum"
        mode="{modeCompanyAmountSum}"
        formatted="view"
        surfaceKey="{surfaceCompanyAmouontSum}"
        class="kaiden-ui-cmpAmnt"
        style="width:100px;"
        tabindex=".1"
        readonly="readonly"
        validationDefault="{validCompanyAmountSum_def}"
        validationApply="{validCompanyAmountSum_apply}"
        validationTemsave="{validCompanyAmountSum_tempsave}"
        validationReapply="{validCompanyAmountSum_reapply}"
        validationProcess="{validCompanyAmountSum_process}"
    />
</td>
</tr>
<tr>
    <td colspan="2">
    <k:hidden
        name="expTypeCd"
        mode="0"
    />
    <kmsw:expType
        name="expTypeName"
        mode="{modeExpTypeCd}"

```

```

imeMode="on"
surfaceKey="${surfaceExpType}"
resultValue="expTypeName"
resultChain="[{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}]"
primaryTarget="expTypeCd"
searchCriteriaDate="searchCriteriaDate"
searchCriteriaCompany="searchCriteriaCompany"
searchCriteriaExpTypeCd="expTypeCd"
style="width:150px;"
validationDefault="${validExpTypeCd_def}"
validationApply="${validExpTypeCd_apply}"
validationTempsave="${validExpTypeCd_tempsave}"
validationReapply="${validExpTypeCd_reapply}"
validationProcess="${validExpTypeCd_process}"
/>
<k:hidden
name="accountCd"
mode="${modeAccountCd}"
surfaceKey="${surfaceAccount}"
validationDefault="${validAccountCd_def}"
validationApply="${validAccountCd_apply}"
validationTempsave="${validAccountCd_tempsave}"
validationReapply="${validAccountCd_reapply}"
validationProcess="${validAccountCd_process}"
/>
<k:text
name="accountName"
mode="1"
class="kaiden-ui-mst-search-chain-item"
style="width:115px;"
/>
<k:hidden
name="taxTypeCd"
mode="${modeTaxTypeCd}"
surfaceKey="${surfaceTaxType}"
validationDefault="${validTaxType_def}"
validationApply="${validTaxType_apply}"
validationTempsave="${validTaxType_tempsave}"
validationReapply="${validTaxType_reapply}"
validationProcess="${validTaxType_process}"
/>
<k:text
name="taxTypeName"
mode="1"
class="kaiden-ui-mst-search-chain-item"
style="width:115px;"
/>
</td>
<td>
<k:hidden
name="departmentSetCd"
mode="${modeDepartmentCd}"
/>
<k:hidden
name="departmentCd"
mode="${modeDepartmentCd}"
surfaceKey="${surfaceDepartment}"
/>
<kmsb:department
name="departmentName"
mode="${modeDepartmentCd}"
imeMode="on"
surfaceKey="${surfaceDepartment}"
resultValue="departmentName"
resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd'}"
primaryTarget="departmentCd,departmentSetCd"
searchCriteriaDate="searchCriteriaDate"
searchCriteriaCompany="searchCriteriaCompany"
searchCriteriaDepartmentCd="departmentCd"
searchCriteriaDepartmentSetCd="departmentSetCd"
style="width:150px;"
validationDefault="${validDepartmentCd_def}"
validationApply="${validDepartmentCd_apply}"
validationTempsave="${validDepartmentCd_tempsave}"
validationReapply="${validDepartmentCd_reapply}"
validationProcess="${validDepartmentCd_process}"
/>
</td>
<td colspan="2">
<k:hidden
name="projectCd"
mode="0"
surfaceKey="${surfaceProject}"
/>
<kmsw:project
name="projectName"

```

```

        mode="{modeProjectCd}"
        imeMode="on"
        surfaceKey="{surfaceProject}"
        resultValue="projectName"
        resultChain="{projectCd:projectCd}"
        primaryTarget="projectCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaProjectCd="projectCd"
        style="width:150px;"
        validationDefault="{validProjectCd_def}"
        validationApply="{validProjectCd_apply}"
        validationTempsave="{validProjectCd_tempsave}"
        validationReapply="{validProjectCd_reapply}"
        validationProcess="{validProjectCd_process}"
    />
</td>
</tr>
</tbody>
</k:tmttable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {
    var eventMan = KAIDEN.getGadget("${gadgetId").eventManager
    , blockAccessor = KAIDEN.getGadget("${gadgetId").blockAccessors["${gadgetBlockHeaderId}"]
    , tmt = KAIDEN.getGadget("${gadgetId").tmtables["${tmttableId}"]
    , tma = tmt.tmttableAccessor;

    /*****
    * 内部イベント
    *****/

    //明細行:行挿入(データ設定前)
    tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

        //デフォルト部門を要求
        var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "${gadgetId}");
        defaultDept = defaultDept[0].result; //先頭行のみ利用
        tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
        tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
        tma("departmentName", tupleId).setValue(defaultDept.departmentName);

        //デフォルトプロジェクトを要求
        var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "${gadgetId}");
        defaultPrj = defaultPrj[0].result; //先頭行のみ利用
        tma("projectCd", tupleId).setValue(defaultPrj.projectCd);
        tma("projectName", tupleId).setValue(defaultPrj.projectName);

    });

    //明細行:行挿入(データ設定後)
    tmt.addEventPostNewRow(function(tupleId, tma, data) {
        //マスタ検索部品の整合性確認。
        KAIDEN.MasterSearch.refresh(tma("allowanceName", tupleId).elem().get(0));
        KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
        KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
        KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

        //初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
        if (tma("payTypeCd", tupleId).elem(true).length > 0) {
            KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
        }

        //事前申請書取込では、会社通貨額が設定されているため、取引通貨額に会社通貨額を設定する
        //※外貨版から標準版にコピーする際、会社通貨額(レート換算後)の金額を設定する
        var compAmnt = tma("companyAmount", tupleId).getValue();
        if (compAmnt) {
            tma("transAmount", tupleId).setValue(compAmnt);
            tma("companyAmount", tupleId).setValue("");
        }

        //取引通貨額、通貨を編集不可
        tma("transAmount", tupleId).readonly().elem().attr("tabindex", "-1");

    });

    //日当金額再取得:ボタン押下
    blockAccessor("allowanceAmountReget").elem(true).on("click",function(e) {
        imuiConfirm("${msgAllowanceReget}", "", function() {
            var ajax = KAIDEN.Ajax(), criteria = {}, Accessor = KAIDEN.base.foundation.Accessor, ret;

            // 検索基準会社
            ret = KAIDEN.gadgetMan.execPull("claimSearchCriteriaCompany");
            criteria.searchCriteriaCompany = ret[0].result;
            // 検索基準日

```

```

ret = KAIDEN.gadgetMan.execPull("claimSearchCriteriaDate");
criteria.searchCriteriaDate = ret[0].result;
// 申請権限者
ret = KAIDEN.gadgetMan.execPull("claimApplyAuthUserCode");
criteria.userCd = ret[0].result;

tma("allowanceCd").elem(true).each(function() {
    var allowance = "";
    criteria.allowanceCd = ""
    criteria.allowanceCd = Accessor($(this)).getValue();

    if (criteria.allowanceCd) {
        allowance = ajax.postSync("kaiden/master/allowanceAmount", criteria);
    } else {
        return;
    }
    KAIDEN.gadgetMan.getNeighboringAccessor(this, "transAmount").setValue(allowance.amount).elem().trigger("change");
});
}, undefined, false);
});

//日数*金額
tmt.variable.multiplyDays = function(target) {
    var transAmount = KAIDEN.gadgetMan.getNeighboringAccessor(target, "transAmount").getValue();
    var quantity = KAIDEN.gadgetMan.getNeighboringAccessor(target, "quantity").getValue();

    if (($.isNumeric(transAmount) && $.isNumeric(quantity))) {
        //金額項目に日数 * 換算額を設定する
        var total = new BigDecimal(transAmount).multiply(new BigDecimal(quantity));
        KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue(total.toString());
    } else {
        KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue("");
    }
};

//日数入力後処理
tma("quantity").elem(true).live("change", function(e) {
    tmt.variable.multiplyDays(this);
});

//取引通貨額入力後処理
tma("transAmount").elem(true).live("change", function(e) {
    tmt.variable.multiplyDays(this);
});

//マスタ検索確定時処理
tma("allowanceName").elem(true).live("choice", function(e, data) {
    tmt.variable.multiplyDays(this);
});

//マスタ検索クリア時処理
tma("allowanceName").elem(true).live("clear", function(e) {
    //金額(換算額*日数)をクリアする
    KAIDEN.gadgetMan.getNeighboringAccessor(this, "companyAmountSum").setValue("");
});

/*****
* 受信イベント(push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
    //明細行のクリア
    tmt.deleteAll();
    return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});

//postLoadData データ読み込み後処理 イベントリスナの登録
eventMan.registPushListener("postLoadData", function(**Object*/arg) {
    //金額項目に「マスタチェック: 日当金額」を実行する
    var rv = true;
    tma("transAmount").elem(true).each(function() {
        KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
    });
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
    return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//postLoadData4Copy 申請書コピー後処理 イベントリスナの登録
eventMan.registPushListener("postLoadData4Copy", function(**Object*/arg) {
    var rv = true;
    tma("transAmount").elem(true).each(function() {
        KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
    });
});

```

```

});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
    return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//postData4Settle 事前申請書取り込み後処理 イベントリスナの登録
eventMan.registPushListener("postData4Settle", function(**Object*/arg) {
    var rv = true;
    tma("transAmount").elem(true).each(function() {
        KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
    });
});

KAIDEN.getGadget("${gadgetId").variable.validateTransAmount = function (**HtmlElement*/target) {
    var rv = true;
    var val = KAIDEN.base.foundation.Accessor($(target)).getValue();
    var param = [ 'allowanceCd', 'searchCriteriaCompany', '$${applyAuthUserCode}', '$${companyCurrencyCd}', 'transAmount', 'searchCriteriaDate' ];
    rv = KAIDEN.validator.validateFunction("k_allowanceAmount")(val, target, param);
    if (!rv) {
        imuiShowWarningMessage("${messageSearchWarning}", "${messageSearchWarningSub}");
    }
};
/*****
* 受信イベント (pull)
*****/

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues("companyAmountSum");
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value)) {
            result = result.add(new BigDecimal(value));
        }
    });
    return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(**Object*/arg) {
    var amounts = tmt.getColumnValues("companyAmountSum");
    var result = new BigDecimal().ZERO;
    $.each(amounts, function(index, value) {
        if ($.isNumeric(value)) {
            result = result.add(new BigDecimal(value));
        }
    });
    return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(**Object*/arg) {
    return new BigDecimal().ZERO.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(**Object*/arg) {
    return tmt.getColumnValues("companyAmountSum");
});

//claimExpenccCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenccCount", function(**Object*/arg) {
    return tmt.getCount();
});

});
</script>

```

SP

- Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/allowanceDetail/allowanceDetailV01_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

```

```

<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<%-- allowanceDetail v01 --%>
<c:set var="gadgetClass">allowanceDetail</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="componentName">allowanceDetailV01 Action</c:set>
<c:set var="imwSystemMatterId">${param.imwSystemMatterId}</c:set>
<c:set var="imwUserDataId">${param.userDataId}</c:set>
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
<c:set var="companyCurrencyCd">${param.companyCurrencyCd}</c:set>
<c:set var="tmtableId">allowanceDetailTable</c:set>

<c:set var="gadgetResponse" value="${kimw:load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany)}"></c:set>
<c:set var="gadget" value="${appDto.gadgets[gadgetId]}" />

<%-- Surface --%>
<c:set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</c:set>
<c:set var="surfaceAllowance">${gadget.surfaces["allowance"]}</c:set>
<c:set var="surfaceTransAmount">${gadget.surfaces["amount"]}</c:set>
<c:set var="surfaceQuantity">${gadget.surfaces["quantity"]}</c:set>
<c:set var="surfaceCompanyAmouontSum">${gadget.surfaces["companyAmountSum"]}</c:set>
<c:set var="surfaceExpType">${gadget.surfaces["expType"]}</c:set>
<c:set var="surfaceAccount">${gadget.surfaces["account"]}</c:set>
<c:set var="surfaceTaxType">${gadget.surfaces["taxType"]}</c:set>
<c:set var="surfaceDepartment">${gadget.surfaces["department"]}</c:set>
<c:set var="surfaceProject">${gadget.surfaces["project"]}</c:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconp
  <h3>
    <span>${surfaceChapterTitle}</span>
    <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="${fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
  </h3>
  <ul data-role="listview" data-divider-theme="b">
    <c:forEach items="${gadgetResponse.data[tmtableId]}" var="detailData">
      <li>
        <span class="ui-li-heading"><c:out value="${detailData.allowanceName}"></c:out></span>
        <span class="ui-li-aside"><c:out value="${detailData.companyAmountSum[1]}"></c:out></span>
        <div class="indent">
          <p class="kaiden-ui-overview-contents">
            <c:out value="${surfaceTransAmount}" .:></c:out>
            <c:out value="${detailData.companyAmount[1]}"></c:out>
          </p>
          <p class="kaiden-ui-overview-contents">
            <c:out value="${surfaceQuantity}" .:></c:out>
            <c:out value="${detailData.quantity[1]}"></c:out>
          </p>
        </div>
        <ul>
          <li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="${surfaceBack}"></c:out></a></li>
          <li>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceAllowance}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.allowanceName}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceTransAmount}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.transAmount[1]}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceQuantity}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.quantity[1]}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceCompanyAmouontSum}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><strong><c:out value="${detailData.companyAmountSum[1]}"></c:out></strong></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceExpType}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.expTypeName}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceAccount}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.accountName}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceTaxType}"></c:out></p>
            <div class="indent">
              <label class="kaiden-ui-detail-item-contents"><c:out value="${detailData.taxTypeName}"></c:out></label>
            </div>
            <p class="kaiden-ui-detail-item-title"><c:out value="${surfaceDepartment}"></c:out></p>

```

```
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.departmentName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="{surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="{detailData.projectName}"></c:out></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>
```


日当明細 (複数通貨)

日当明細 (複数通貨) のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Action (基底)
■	Action
■	Form
■	Logic (基底)
■	Logic
■	FieldName
■	Manager
■	Manager (実装)
■	ManagerDto
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	コピー
■	仕訳連携
■	JSP
■	PC
■	SP

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   │   ├── K30tAllowanceDetail.java
│   │   │   │   │   │   │   │   │   │   ├── K30tAllowanceDetailNames.java
│   │   │   │   │   │   │   │   │   │   ├── Ka30tAllowanceDetail.java
│   │   │   │   │   │   │   │   │   │   ├── Ka30tAllowanceDetailNames.java
│   │   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │   │       ├── K30tAllowanceDetailCondition.java
│   │   │   │   │   │   │   │   │       ├── K30tAllowanceDetailService.java
│   │   │   │   │   │   │   │   │       ├── Ka30tAllowanceDetailCondition.java
│   │   │   │   │   │   │   │   │       └── Ka30tAllowanceDetailService.java
│   │   │   │   │   │   │   └── java
│   │   │   │   │   │   │       ├── jp
│   │   │   │   │   │   │       │   ├── co
│   │   │   │   │   │   │       │   │   ├── slcs
│   │   │   │   │   │   │       │   │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │   ├── feature
│   │   │   │   │   │   │       │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │       │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── AbstractAllowanceDetailActionBase.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   └── AllowanceDetailV02Action.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   └── conf
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailFieldName.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   └── form
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailV02Form.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   └── logic
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   ├── AbstractAllowanceDetailLogicBase.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailV02Logic.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   └── foundation
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── AllowanceDetailManager.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── AllowanceDetailManagerDto.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   └── impl
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailManagerImpl.java
│   │   │   │   │   │   │       │   │   │   │   │   │   │   └── resources
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── META-INF
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── sql
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── jp
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   ├── co
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── impl
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── AllowanceDetailManagerImpl
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── selectForJournalSourceDtl.sql
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── model
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── K30tAllowanceDetailService
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── selectView.sql
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── selectViewForCopy.sql
│   │   │   │   │   │   │       │   │   │   │   │   │   │   └── webapp
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   ├── WEB-INF
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   ├── view
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   ├── gadget
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   ├── allowanceDetail
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   ├── allowanceDetailV02.jsp
│   │   │   │   │   │   │       │   │   │   │   │   │   │   │   │   │   │   │   │   │   └── allowanceDetailV02_sp.jsp

```

Action (基底)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AbstractAllowanceDetailActionBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;
import jp.co.slcs.kaiden2.base.foundation.helper.NumberHelper;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.GadgetActionIF;
import net.arnx.jsonic.JSON;
import org.seasar.framework.container.SingletonS2Container;
import org.seasar.struts.annotation.Execute;

/**
 * 日当明細ガジェット用Actionの抽象クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public abstract class AbstractAllowanceDetailActionBase implements GadgetActionIF {

    /**
     * 申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadApply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 再申請画面のデータロード.
     *
     * @return 検索結果
     */
    @Override
    @Execute(validator = false)
    public String loadReapply() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 一時保存画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadTempsave() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理画面のデータロード.
     * <p>レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力</p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    public String loadProcess() {

```

```

        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadConfirm() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 処理詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadProcessDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 画面表示用データを返却.
     *
     * @return Managerからの返却値
     */
    @Override
    public LogicResultDto getViewData() {
        return getLogic().view(getForm(), getConvertor());
    }

    /**
     * 参照詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadReferenceDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 確認詳細画面のデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadConfirmDetail() {
        ResponseWriter.writeJson(JSON.encode(getViewData()));
        return null;
    }

    /**
     * 申請書コピーのデータロード.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute validator = false
    public String loadCopy() {
        ResponseWriter.writeJson(JSON.encode(getLogic().viewForCopy(getForm(), getConvertor())));
    }

```

```

    return null;
}

/**
 * 過去案件詳細画面のデータロード.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
public String loadArchiveMatterDetail() {
    ResponseWriter.writeJson(JSON.encode(getLogic().viewArchive(getForm(), getConverter())));
    return null;
}

/**
 * 書式変換設定を返却.
 * <p>
 * 標準では{@link #AllowanceDetailConverter 日当明細(標準)ガジェット}の書式変換を返却します。<br>
 * 書式を変更する場合は、サブクラスでオーバーライドしてください
 * </p>
 *
 * @return 書式変換後データ
 */
protected AllowanceDetailConverter getConverter() {
    return new AllowanceDetailConverter(getForm());
}

/**
 * 日当明細(標準)ガジェットの書式変換.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
private static class AllowanceDetailConverter extends ConvertForJsonSource {

    /** リクエストパラメータ. */
    private AbstractImwPageForm frm;

    /** NumberHelper. */
    private NumberHelper numberHelper;

    /**
     * コンストラクタ.
     *
     * @param frm リクエストパラメータ
     */
    public AllowanceDetailConverter(AbstractImwPageForm frm) {
        this.frm = frm;
        numberHelper = SingletonS2Container.getComponent(NumberHelper.class);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 本サブクラスでは、以下の書式変換を行う。<br>
     * <table border="1">
     * <tr>
     * <td>項目名</td>
     * <td>書式</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>スケールを通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨換算レート</td>
     * <td>スケールをレートスケールに変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * <tr>
     * <td>日数</td>
     * <td>整数表示に変換</td>
     * </tr>
     * <tr>
     * <td>会社通貨額合計</td>
     * <td>スケールを会社通貨のスケールに変換</td>
     * </tr>
     * </table>
     * </p>
     */

```

```

@Override
protected Map<String, Object> formatPlainValue(Map<String, Object> map, Stack<String> fldNameStack) {

    Map<String, Object> result = new HashMap<String, Object>(map);

    // 取引通貨額の書式変換
    if (result.containsKey(TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) result.get(TRANS_AMOUNT);
        Object currencyCd = result.get(TRANS_CURRENCY_CD);
        result.put(TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }
    // レートの書式変換
    if (result.containsKey(COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) result.get(COMPANY_RATE);
        result.put(COMPANY_RATE, numberHelper.formatRate4Input(rate));
    }

    String companyCurrencyCd = (String) result.get(COMPANY_CURRENCY_CD);

    // 換算額の書式変換
    if (result.containsKey(COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) result.get(COMPANY_AMOUNT);
        result.put(COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4Input(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    // 日数の書式変換
    if (result.containsKey(QUANTITY)) {
        BigDecimal quantity = (BigDecimal) map.get(QUANTITY);
        if (quantity == null) {
            result.put(QUANTITY, "");
        } else {
            result.put(QUANTITY, StringUtil.toString(quantity.setScale(0)));
        }
    }

    // 金額(換算額*日数)の書式変換
    if (result.containsKey(COMPANY_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(COMPANY_AMOUNT_SUM);
        result.put(COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4Input(companyAmountSum,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 本サブクラスでは、以下の書式変換を行う。<br>
 * <table border="1">
 * <tr>
 * <td>項目名</td>
 * <td>書式</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>スケールを通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨換算レート</td>
 * <td>スケールをレートスケールに変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * <tr>
 * <td>日数</td>
 * <td>整数表示に変換</td>
 * </tr>
 * <tr>
 * <td>会社通貨額合計</td>
 * <td>スケールを会社通貨のスケールに変換</td>
 * </tr>
 * </table>
 * <p>
 */
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 取引通貨額の書式変換
    if (map.containsKey(TRANS_AMOUNT)) {
        BigDecimal amount = (BigDecimal) map.get(TRANS_AMOUNT);
        Object currencyCd = map.get(TRANS_CURRENCY_CD);
        result.put(TRANS_AMOUNT, numberHelper.formatCurrencyAmount4Input(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

```

```

        BigDecimal amount = (BigDecimal) map.get(TRANS_AMOUNT);
        Object currencyCd = map.get(TRANS_CURRENCY_CD);
        result.put(TRANS_AMOUNT,
            numberHelper.formatCurrencyAmount4View(amount, frm.searchCriteriaCompany, (String) currencyCd));
    }

    // レートの書式変換
    if (map.containsKey(COMPANY_RATE)) {
        BigDecimal rate = (BigDecimal) map.get(COMPANY_RATE);
        result.put(COMPANY_RATE, numberHelper.formatRate4View(rate));
    }

    String companyCurrencyCd = result.get(COMPANY_CURRENCY_CD);
    // 換算額の書式変換
    if (map.containsKey(COMPANY_AMOUNT)) {
        BigDecimal companyAmount = (BigDecimal) map.get(COMPANY_AMOUNT);
        result.put(COMPANY_AMOUNT, numberHelper.formatCurrencyAmount4View(companyAmount,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    // 日数の書式変換
    if (map.containsKey(QUANTITY)) {
        BigDecimal quantity = (BigDecimal) map.get(QUANTITY);
        if (quantity == null) {
            result.put(QUANTITY, "");
        } else {
            result.put(QUANTITY, StringUtil.toString(quantity.setScale(0)));
        }
    }

    // 金額(換算額*回数)の書式変換
    if (map.containsKey(COMPANY_AMOUNT_SUM)) {
        BigDecimal companyAmountSum = (BigDecimal) map.get(COMPANY_AMOUNT_SUM);
        result.put(COMPANY_AMOUNT_SUM, numberHelper.formatCurrencyAmount4View(companyAmountSum,
            frm.searchCriteriaCompany, companyCurrencyCd));
    }

    return result;
}
}
}

```

Action

- Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/gadget/AllowanceDetailV02Action.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.gadget;

import javax.annotation.Resource;
import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginLogicIF;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget.AllowanceDetailV02Form;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail.AllowanceDetailV02Logic;
import org.seasar.struts.annotation.ActionForm;

/**
 * 日当明細(標準・外貨)ガジェット(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV02Action extends AbstractAllowanceDetailActionBase {

    /**
     * AllowancedetailV02Logic.
     */
    @Resource
    public AllowanceDetailV02Logic allowanceDetailV02Logic;

    /**
     * AllowanceDetailV02Form.
     */
    @ActionForm
    @Resource
    public AllowanceDetailV02Form allowanceDetailV02Form;

    /**
     * {@inheritDoc}
     */
    @Override
    public PluginLogicIF getLogic() {
        return allowanceDetailV02Logic;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public AbstractImwPageForm getForm() {
        return allowanceDetailV02Form;
    }
}

```

Form

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/gadget/AllowanceDetailV02Form.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden2.extension_imw.foundation.imw_page.AbstractImwPageForm;

/**
 * 日当明細(標準・外貨)ガジェット(Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV02Form extends AbstractImwPageForm {

}

```

Logic(基底)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/allowancedetail/AbstractAllowanceDetailLogicBase.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.StringUtil.isNotEmpty;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_CURRENCY_AMNT;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DEPARTMENT_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DIGITS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_IMM_CURRENCY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_RANGE;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_APPROVEEND;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants.LAST_RESULT_STATUS_MATTERCOMPLETE;

```



```

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_SET_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ALLOWANCE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_EXP_TYPE_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PAY_METHOD_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.model.dto.ErrorDetailDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.CurrencyAmtValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DepartmentExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.DigitsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.ImmCurrencyExistsValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericRangeValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.WorkflowConstants;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.MatterEndProcessDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.model.dto.PluginGadgetDTO;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimPayMethodIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.JournalDataTypeCd;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.gadget.AbstractExpensePluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.journalsourcedtl.JournalSourceDtlManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AccountExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AllowanceExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ExpTypeExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.PayMethodExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.ProjectExistsValidator;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.TaxTypeExistsValidator;
import org.seasar.framework.beans.util.Beans;
import org.seasar.util.beans.util.BeanMap;

```

```
/**
```

```
 * 日当明細ガジェット用Logicの抽象クラス。
```

```
 *
```

```
 * @author Sumitomo Life Information Systems Co.,Ltd.
```

```

*/
public abstract class AbstractAllowanceDetailLogicBase extends AbstractExpensePluginGadgetLogic {

    /** ガジェットブロックキー: 日当明細ガジェット・日当明細テーブル. */
    protected static final String BLOCK_KEY_ALLOWANCE_DETAIL_TABLE = "allowanceDetailTable";

    /** 金額、整数部桁数. */
    public static final int COMPANY_RATE_INT_LEN = 18;
    /** 金額、小数部桁数. */
    public static final int COMPANY_RATE_DEC_LEN = 10;
    /** 日数 最大値. */
    public static final int QUANTITY_MAX = 999;

    /** AllowanceDetailManager. */
    @Resource(name = "allowanceGadgetManager")
    protected AllowanceDetailManager allowanceDetailManager;

    /** JournalSourceDtlManager. */
    @Resource
    protected JournalSourceDtlManager journalSourceDtlManager;

    /** コンストラクタ. */
    public AbstractAllowanceDetailLogicBase() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 日当明細を登録します。
     * </p>
     */
    @Override
    public int apply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 対象となる日当明細テーブルを削除後に、日当明細テーブルを登録します。<br>
     * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM\_MATTER\_ID\_TEMPORARY} temporary"を利用します。
     * </p>
     */
    @Override
    public int applyFromTempSave(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        deleteTable(true);
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 日当明細を登録します。
     * </p>
     */
    @Override
    public int applyFromUnapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        return insertTable();
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
            rv = insertTable();
        }
        return rv;
    }

    /**
     * {@inheritDoc}
     * <p>
     * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
     * </p>
     */
    @Override
    public int approveEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
        int rv = 0;
        if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {

```

```

        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int deny(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int discontinue(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterHandle(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int pullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細を登録します。
 * </p>
 */
@Override
public int reapply(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int reserve(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int reserveCancel(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
}

```

```

    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 直前のユーザーデータIDが設定されている場合に、日当明細テーブルを登録します。
 * </p>
 */
@Override
public int sendBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = 0;
    if (StringUtil.isEmpty(super.getGadgetDto().beforeUserDataId)) {
        rv = insertTable();
    }
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int sendBackToPullBack(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細を登録します。
 * </p>
 */
@Override
public int tempSaveCreate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細テーブルの対象データを削除します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * </p>
 */
@Override
public int tempSaveDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTable(true);
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細テーブルを削除後に、日当明細テーブルを登録します。<br>
 * 削除条件のシステム案件IDは、"{@link WorkflowConstants#SYSTEM_MATTER_ID_TEMPORARY temporary}"を利用します。
 * </p>
 */
@Override
public int tempSaveUpdate(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    deleteTable(true);
    return insertTable();
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int matterStart(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 処理結果ステータスが[承認終了]、[最終承認]の場合に、仕訳連携用明細テーブルへのデータ登録処理を行います。<br>
 * ただし上記処理結果ステータスの場合でも、{@link #isRequiredCreateJournalData(JournalDataCreateType)}の判定結果によっては、<br>
 * 仕訳連携用明細テーブルへのデータ登録処理を行わない場合があります。<br>
 * 詳細は、{@link #isRequiredCreateJournalData(JournalDataCreateType)}を参照ください。
 * </p>
 */
@Override

```

```

@Override
public int matterEnd(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {

    PluginGadgetDTO pluginGadgetDTO = BeanUtil.toCast(PluginGadgetDTO.class, dto);
    MatterEndProcessDto matterEndProcessDto = BeanUtil.toCast(MatterEndProcessDto.class,
        pluginGadgetDTO.imwPluginParameterDto);

    String status = matterEndProcessDto.lastResultStatus;

    int rv = 0;

    if (LAST_RESULT_STATUS_APPROVEEND.equals(status) || LAST_RESULT_STATUS_MATTERCOMPLETE.equals(status)) {
        if (isRequiredCreateJournalData(JournalDataCreateType.DETAIL_COST)) {
            // 日当明細データ取得キー
            CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();
            criteriaDto.systemMatterId = pluginGadgetDTO.systemMatterId;
            criteriaDto.userDataId = pluginGadgetDTO.userDataId;
            criteriaDto.gadgetClass = pluginGadgetDTO.gadgetClass;
            criteriaDto.gadgetInstance = pluginGadgetDTO.gadgetInstance;
            // 日当明細データを取得
            List<BeanMap> beanMapList = allowanceDetailManager.selectForJournalSourceDtl(criteriaDto);
            List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtlList(JournalSourceDtlManagerDto.class,
                beanMapList);

            // 仕訳連携用明細テーブル登録処理
            rv += insertTableJournalSourceDtl(insList, matterEndProcessDto);
        }
    }

    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 本機能では何も処理を行わず、常に0を返却します。
 * </p>
 */
@Override
public int arrive(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return 0;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int actvMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる(過去案件)日当明細テーブルを一括削除します。
 * </p>
 */
@Override
public int arcMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    int rv = allowanceDetailManager
        .removeAllArchive(super.convCriteriaForSqlDtoForArchive(super.getGadgetDto().imwPluginParameterDto));
    return rv;
}

/**
 * {@inheritDoc}
 * <p>
 * 対象となる日当明細の履歴を含めた全てのデータを削除します。
 * </p>
 */
@Override
public int cplMatterDelete(KaidenDto dto) throws KaidenSystemException, KaidenApplicationException {
    return deleteTableIncludeBackground();
}

/**
 * {@inheritDoc}
 * <p>
 * データの取得は、{@link AllowanceDetailManager#selectList(AllowanceDetailManagerDto)}を利用します。
 * </p>
 */
@Override
protected LinkedHashMap<String, Object> viewMap(PluginGadgetDTO conditionDto) throws KaidenSystemException {
    LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();

```

```

LinkedHashMap<String, Object> rv = new LinkedHashMap<String, Object>();

// 日当明細ガジェットテーブル取得
AllowanceDetailManagerDto allowanceDetailCriteria = Beans.createAndCopy(AllowanceDetailManagerDto.class,
    conditionDto).execute();
List<AllowanceDetailManagerDto> allowanceDetailDtoList = allowanceDetailManager
    .selectList(allowanceDetailCriteria);
List<Map<String, Object>> allowanceDetailMapList = CollectionUtil.toMapList(allowanceDetailDtoList);
rv.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailMapList);

return rv;
}

/**
 * {@inheritDoc}
 *
 * @see #viewMap
 */
@Override
protected Map<String, Object> viewMapFormatted(PluginGadgetDTO conditionDto, ConvertForJsonSource converter)
    throws KaidenSystemException {
    LinkedHashMap<String, Object> map = viewMap(conditionDto);
    if (map == null) {
        return null;
    }
    return converter.mapConvert(map);
}

/**
 * 仕訳連携用ヘッダテーブルへのデータ登録処理.
 * <p>
 * 当該処理は、処理結果ステータスが"approveEnd", "matterComplete" の場合のみ行う。
 * </p>
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 * @deprecated<br>
 *      {@link #insertTableJournalSourceDtl(List, MatterEndProcessDto)} を利用してください。
 */
@Deprecated
protected int insertk30tJournalSourceHdr() throws KaidenSystemException {
    return journalSourceDtlManager.entry(makeInsertJshMngDto());
}

/**
 * 登録用仕訳連携用明細DTOを作成.
 *
 * @return 仕訳連携用ヘッダDTO
 * @deprecated
 */
@Deprecated
protected List<JournalSourceDtlManagerDto> makeInsertJshMngDto() {
    CriteriaForSqlDto criteriaDto = new CriteriaForSqlDto();

    criteriaDto.systemMatterId = super.getGadgetDto().systemMatterId;
    criteriaDto.userDataId = super.getGadgetDto().userDataId;
    criteriaDto.gadgetClass = super.getGadgetDto().gadgetClass;
    criteriaDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    List<BeanMap> beanMapList = allowanceDetailManager.selectForJournalSourceDtl(criteriaDto);

    List<JournalSourceDtlManagerDto> insList = CollectionUtil.toDtoList(JournalSourceDtlManagerDto.class,
        beanMapList);

    return insList;
}

/**
 * 日当明細データ登録処理.
 *
 * @return 更新カウント
 * @throws KaidenSystemException システムエラー
 */
protected int insertTable() throws KaidenSystemException {
    int retCnt = 0;

    // 日当明細ガジェットテーブル登録
    List<AllowanceDetailManagerDto> allowanceDetailManagerDtoList = JsonUtil.getDtoList(
        AllowanceDetailManagerDto.class, getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE));
    retCnt += allowanceDetailManager.entry(allowanceDetailManagerDtoList);

    return retCnt;
}

/**
 * 仕訳連携用明細テーブルデータ登録処理.
 *

```

```

* @param srcList 日当明細テーブルデータリスト
* @param matterEndProcessDto 案件終了処理用DTO
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int insertTableJournalSourceDtl(List<JournalSourceDtlManagerDto> srcList,
    MatterEndProcessDto matterEndProcessDto) throws KaidenSystemException {
    for (JournalSourceDtlManagerDto srcDto : srcList) {
        // 仕訳データ区分をセット
        srcDto.journalDataTypeCd = JournalDataTypeCd.JOURNAL_DATA_TYPE_CD_COST.getCode();
    }

    // 仕訳連携用明細テーブル登録
    return journalSourceDtlManager.entry(srcList);
}

/**
* 日当明細データ削除処理.
*
* @param tempFlg 一時保存データフラグ
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int deleteTable(boolean tempFlg) throws KaidenSystemException {
    int retCnt = 0;
    // 日当明細ガジェットテーブル削除
    AllowanceDetailManagerDto allowanceDetailManagerDto = new AllowanceDetailManagerDto();
    allowanceDetailManagerDto.systemMatterId = super.getGadgetDto().systemMatterId;
    allowanceDetailManagerDto.userDataId = super.getGadgetDto().userDataId;
    allowanceDetailManagerDto.gadgetClass = super.getGadgetDto().gadgetClass;
    allowanceDetailManagerDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    if (tempFlg) {
        allowanceDetailManagerDto.systemMatterId = WorkflowConstants.SYSTEM_MATTER_ID_TEMPORARY;
    }
    retCnt += allowanceDetailManager.remove(allowanceDetailManagerDto);

    return retCnt;
}

/**
* 日当明細テーブルデータ削除処理(履歴を含む).
*
* @return 更新カウント
* @throws KaidenSystemException システムエラー
*/

protected int deleteTableIncludeBackground() throws KaidenSystemException {
    int retCnt = 0;

    // 日当明細ガジェットテーブル削除(履歴を含む)
    AllowanceDetailManagerDto allowanceDetailManagerDto = new AllowanceDetailManagerDto();
    allowanceDetailManagerDto.systemMatterId = super.getGadgetDto().systemMatterId;
    allowanceDetailManagerDto.userDataId = null;
    allowanceDetailManagerDto.gadgetClass = super.getGadgetDto().gadgetClass;
    allowanceDetailManagerDto.gadgetInstance = super.getGadgetDto().gadgetInstance;
    retCnt += allowanceDetailManager.remove(allowanceDetailManagerDto);

    return retCnt;
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/

@Override
protected void defineValidateForApply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/

@Override
protected void defineValidateForApplyFromTempSave(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
* {@inheritDoc}
*
* @see #defineValidateForNormal(GadgetMediatorParams, boolean)
*/

```

```

@Override
protected void defineValidateForApplyFromUnapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApprove(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForApproveEnd(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReapply(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, false);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserve(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForReserveCancel(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForSendBack(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    if (StringUtil.isNotEmpty(super.getGadgetDto().beforeUserDataId)) {
        defineValidateForNormal(params, false);
    }
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveCreate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {

```



```

defineValidateForNormal(params, true);
}

/**
 * {@inheritDoc}
 *
 * @see #defineValidateForNormal(GadgetMediatorParams, boolean)
 */
@Override
protected void defineValidateForTempSaveUpdate(GadgetMediatorParams params) throws KaidenSystemException,
    KaidenApplicationException {
    defineValidateForNormal(params, true);
}

/**
 * フィールドバリデート定義.
 *
 * <p>
 * <table border="1">
 * <tr>
 * <td align="center">対象フィールド名</td>
 * <td align="center">検証内容</td>
 * <td align="center">モード</td>
 * </tr>
 * <tr>
 * <td>日当コード</td>
 * <td>{@link AllowanceExistsValidator 日当存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>取引通貨額</td>
 * <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>取引通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">会社通貨換算レート</td>
 * <td>{@link DigitsValidator 小数字チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericMinValidator 最小数値チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨額</td>
 * <td>{@link CurrencyAmntValidator 通貨額チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>会社通貨コード</td>
 * <td>{@link ImmCurrencyExistsValidator 通貨存在チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td rowspan="2">日数</td>
 * <td>{@link NumericValidator 半角数字チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>{@link NumericRangeValidator 数値範囲チェック}</td>
 * <td align="center">入力</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>{@link ExpTypeExistsValidator 経費区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>{@link AccountExistsValidator 勘定科目マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>{@link TaxTypeExistsValidator 税区分マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>
 * <tr>
 * <td>支払方法コード</td>
 * <td>{@link PayMethodExistsValidator 支払方法マスタ存在チェック}</td>
 * <td align="center">入力/表示</td>
 * </tr>

```

```

* <tr>
* <td>組織コード</td>
* <td>{@link DepartmentExistsValidator 組織コード存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>{@link ProjectExistsValidator プロジェクトマスタ存在チェック}</td>
* <td align="center">入力/表示</td>
* </tr>
* <tr>
* <td>会社通貨額合計</td>
* <td>{@link CurrencyAmtValidator 通貨額チェック}</td>
* <td align="center">入力</td>
* </tr>
* </table>
* </p>
*
* @param params ガジェットメディエータ処理情報クラス
* @param tmpFlg 一時保存識別 (true:一時保存)
* @throws KaidenSystemException システムエラーが発生
* @throws KaidenApplicationException アプリケーションエラーが発生
*/
protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
    KaidenApplicationException {

    // 日当明細ガジェットテーブル部
    if (fieldDefsMap.containsKey(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE)) {
        addValidateDefsRequired(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, tmpFlg);
        // 日当コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ALLOWANCE_CD, VALID_ALLOWANCE_EXISTS, ALLOWANCE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 取引通貨額
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_AMOUNT, VALID_CURRENCY_AMNT,
            TRANS_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 取引通貨コード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_CURRENCY_CD, VALID_IMM_CURRENCY_EXISTS,
            TRANS_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨換算レート
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_RATE, VALID_DIGITS,
            String.valueOf(COMPANY_RATE_INT_LEN), String.valueOf(COMPANY_RATE_DEC_LEN));
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_RATE, VALID_NUMERIC_MIN, "0");
        // 会社通貨額
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT, VALID_CURRENCY_AMNT,
            COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 会社通貨コード
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_CURRENCY_CD,
            VALID_IMM_CURRENCY_EXISTS, COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
        // 日数
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, QUANTITY, VALID_NUMERIC);
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, QUANTITY, VALID_NUMERIC_RANGE,
            getArgs(1, QUANTITY_MAX));
        // 経費区分コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, EXP_TYPE_CD, VALID_EXP_TYPE_EXISTS, EXP_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 勘定科目コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 税区分コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 支払方法コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PAY_METHOD_CD, VALID_PAY_METHOD_EXISTS, PAY_METHOD_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 組織コード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, DEPARTMENT_CD, VALID_DEPARTMENT_EXISTS,
            getGadgetDto().searchCriteriaCampany, DEPARTMENT_SET_CD, DEPARTMENT_CD,
            dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // プロジェクトコード
        addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PROJECT_CD, VALID_PROJECT_EXISTS, PROJECT_CD,
            getGadgetDto().searchCriteriaCampany, dateHelper.toYYYYMMDD(getGadgetDto().searchCriteriaDate));
        // 会社通貨額合計
        addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT_SUM, VALID_CURRENCY_AMNT,
            COMPANY_CURRENCY_CD, getGadgetDto().searchCriteriaCampany);
    }
}

/**
* {@inheritDoc}
* <p>
* 常に空のListを返却。
* </p>
*/
@Override
protected List<KaidenValidateResult> postSingleValidate(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {

```

```

    return new ArrayList<KaidenValidateResult>();
}

/**
 * {@inheritDoc}
 * <p>
 * 常に空のListを返却。
 * </p>
 */
@Override
protected List<KaidenValidateResult> globalValidateMain(String eventKey, GadgetMediatorParams params,
    boolean isAuto, boolean isLump) throws KaidenSystemException, KaidenApplicationException {
    return new ArrayList<KaidenValidateResult>();
}

/**
 * {@inheritDoc}
 * <p>
 * システム案件idが未設定の場合（初回表示時など）は、空のリストを返却します。<br>
 * 設定されている場合は、{@link AllowanceDetailManager#selectView(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForView(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();

    if (StringUtil.isEmpty(criteria.systemMatterId)
        || (StringUtil.isEmpty(criteria.userDataId) && !criteria.searchFromArchiveFlag)) {
        result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, new ArrayList<AllowanceDetailManagerDto>());
    } else {
        result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectView(criteria));
    }
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link AllowanceDetailManager#selectViewForCopy(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForCopy(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectViewForCopy(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * {@link AllowanceDetailManager#selectViewArchive(CriteriaForSqlDto)}の取得結果を返却します。
 * </p>
 */
@Override
protected Map<String, Object> selectBlockDatasForArchive(CriteriaForSqlDto criteria) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, allowanceDetailManager.selectViewArchive(criteria));
    return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 取得した日当明細のデータに対して、以下マスタの整合性チェックを行います。
 * <ul>
 * <li>[*1]通貨マスタ(imm_currency)</li>
 * <li>組織マスタ(imm_depaertment)</li>
 * <li>プロジェクトマスタ</li>
 * <li>日当マスタ</li>
 * <li>経費区分マスタ</li>
 * <li>勘定科目マスタ</li>
 * <li>税区分マスタ</li>
 * </ul>
 * *1:取引通貨と会社通貨
 * </p>
 */
@Override
protected List<ErrorDetailDto> checkSelectedData(Map<String, Object> dataMap, ProcessType processType) {
    List<ErrorDetailDto> result = new ArrayList<ErrorDetailDto>();
    for (Entry<String, Object> dataEs : dataMap.entrySet()) {
        if (BLOCK_KEY_ALLOWANCE_DETAIL_TABLE.equals(dataEs.getKey())) {
            @SuppressWarnings("unchecked")
            List<AllowanceDetailManagerDto> sqlResultList = CollectionUtil.castListContent(
                (List<? extends KaidenDto>) dataEs.getValue(), AllowanceDetailManagerDto.class);
            for (AllowanceDetailManagerDto dto : sqlResultList) {

```

```

// 取引通貨コードチェック
if (isNotEmpty(dto.transCurrencyCd) && isEmpty(dto.transCurrencyName)) {
    String[] fieldId = getArgs(TRANS_CURRENCY_CD, TRANS_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(dto.transCurrencyCd, dto.transCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 会社通貨コードチェック
if (isNotEmpty(dto.companyCurrencyCd) && isEmpty(dto.companyCurrencyName)) {
    String[] fieldId = getArgs(COMPANY_CURRENCY_CD, COMPANY_CURRENCY_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.CURRENCY);
    String[] values = getArgs(dto.companyCurrencyCd, dto.companyCurrencyName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 組織チェック
if (isNotEmpty(dto.departmentSetCd) && isNotEmpty(dto.departmentCd) && isEmpty(dto.departmentName)) {
    String[] fieldId = getArgs(DEPARTMENT_SET_CD, DEPARTMENT_CD, DEPARTMENT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        BaseProp.S.DEPARTMENT);
    String[] values = getArgs(dto.departmentSetCd, dto.departmentCd, dto.departmentName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(StringUtil.toString(dto.tupleId)), masterName, values, message));
}

// プロジェクトチェック
if (isNotEmpty(dto.projectCd) && isEmpty(dto.projectName)) {
    String[] fieldId = getArgs(PROJECT_CD, PROJECT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.PROJECT);
    String[] values = getArgs(dto.projectCd, dto.projectName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 日当チェック
if (isNotEmpty(dto.allowanceCd) && isEmpty(dto.allowanceName)) {
    String[] fieldId = getArgs(ALLOWANCE_CD, ALLOWANCE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ALLOWANCE);
    String[] values = getArgs(dto.allowanceCd, dto.allowanceName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 経費区分チェック
if (isNotEmpty(dto.expTypeCd) && isEmpty(dto.expTypeName)) {
    String[] fieldId = getArgs(EXP_TYPE_CD, EXP_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.EXP_TYPE);
    String[] values = getArgs(dto.expTypeCd, dto.expTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);

    if (!ProcessType.VIEW_FOR_MATTER_ARCHIVE.equals(processType)) {
        // 勘定科目、税区分をクリア
        dto.accountCd = null;
        dto.accountName = null;
        dto.taxTypeCd = null;
        dto.taxTypeName = null;
        // 税率、税抜金額、税額をクリア
        dto.taxRate = null;
        dto.companyAmountExcTax = null;
        dto.companyAmountTax = null;
    }

    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 勘定科目チェック
if (isNotEmpty(dto.accountCd) && isEmpty(dto.accountName)) {
    String[] fieldId = getArgs(ACCOUNT_CD, ACCOUNT_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.ACCOUNT);
    String[] values = getArgs(dto.accountCd, dto.accountName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

// 税区分チェック
if (isNotEmpty(dto.taxTypeCd) && isEmpty(dto.taxTypeName)) {
    String[] fieldId = getArgs(TAX_TYPE_CD, TAX_TYPE_NAME);
    String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
        ProductWorkflowProp.S.TAX_TYPE);
    String[] values = getArgs(dto.taxTypeCd, dto.taxTypeName);
    String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
    result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
        .toString(dto.tupleId), masterName, values, message));
}

```

```

        String masterName = propertyHelper.getSurface(contextHelper.getLoginUserLocale(),
            ProductWorkflowProp.S.TAX_TYPE);
        String[] values = getArgs(dto.taxTypeCd, dto.taxTypeName);
        String message = propertyHelper.getMessage(ProductWorkflowProp.M.W0005, masterName);
        result.add(new ErrorDetailDto(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, fieldId, StringUtil
            .toString(dto.tupleId), masterName, values, message));
    }
}
}
return result;
}

/**
 * 支払方法をガジェット間通信で取得.
 *
 * @return 支払方法
 */
protected String getGadgetCouplingPayMethod() {
    // 支払方法をガジェット間通信で取得
    Map<String, Object> args = new HashMap<String, Object>();
    String rv = StringUtil.nullToBlank(getSingleValue(String.class, args, ClaimPayMethodIF.class,
        ClaimPayMethodIF.METHOD_NM_GET_CLAIM_PAY_METHOD, ClaimPayMethodIF.RTN_KEY_PAY_METHOD_CD));
    return rv;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenccountIF#getClaimExpenccount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細の件数
 */
public Map<String, Object> putClaimExpenccount(Map<String, Object> params) {

    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimExpenccountIF.RTN_KEY_CLAIM_EXPENCE_COUNT, blocjData.size());
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF#getClaimCompanyPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 会社支払額
 */
public Map<String, Object> putClaimCompanyPayAmount(Map<String, Object> params) {
    Map<String, Object> retMap = new HashMap<String, Object>();
    retMap.put(ClaimCompanyPayAmountIF.RTN_KEY_CLAIM_COMPANY_PAY_AMOUNT, BigDecimal.ZERO);
    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF#getClaimTemporaryPayAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 従業員立替金額
 */
public Map<String, Object> putClaimTemporaryPayAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
        if (StringUtil.isNotEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF#getClaimExpenseDetail(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用明細
 */
public Map<String, Object> putClaimExpenseDetail(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

```

```

List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
Map<String, Object> retMap = new HashMap<String, Object>();

List<BigDecimal> list = new ArrayList<BigDecimal>();

for (Map<String, Object> j : blocjData) {
    String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
    if (StringUtil.isEmpty(strAmnt)) {
        list.add(new BigDecimal(strAmnt));
    }
}
retMap.put(ClaimExpenseDetailIF.RTN_KEY_CLAIM_EXPENSE_DETAIL, list);

return retMap;
}

/**
 * {@link jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF#getClaimExpenseAmount(Map)}
 * の値を設定.
 *
 * @param params アーギュメント
 * @return 費用合計
 */
public Map<String, Object> putClaimExpenseAmount(Map<String, Object> params) {
    List<Map<String, Object>> blocjData = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    Map<String, Object> retMap = new HashMap<String, Object>();
    BigDecimal sum = BigDecimal.ZERO;

    for (Map<String, Object> j : blocjData) {
        String strAmnt = StringUtil.toString(j.get(COMPANY_AMOUNT_SUM));
        if (StringUtil.isEmpty(strAmnt)) {
            sum = sum.add(new BigDecimal(strAmnt));
        }
    }

    retMap.put(ClaimExpenseAmountIF.RTN_KEY_CLAIM_EXPENSE_AMOUNT, sum);

    return retMap;
}

/**
 * {@inheritDoc}
 * <p>
 * 日当明細テーブルの対象データを過去案件退避します。
 * </p>
 */
@Override
protected int matterArchiveRegister(KaidenDto dto, CriteriaForSqlDto criteria, Map<String, Object> dataMap) {
    // 日当明細
    @SuppressWarnings("unchecked")
    List<AllowanceDetailManagerDto> list = (List<AllowanceDetailManagerDto>) dataMap
        .get(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);
    int rv = allowanceDetailManager.matterArchiveRegister(criteria, list);
    return rv;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<PluginGadgetManager<? extends KaidenDto>> getDeleteTargetsForArchiveManagers() {
    List<PluginGadgetManager<? extends KaidenDto>> list = new ArrayList<PluginGadgetManager<? extends KaidenDto>>();
    list.add(allowanceDetailManager);
    return list;
}
}

```

Logic

■ Path

```
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/gadget/allowancedetail/AllowanceDetailV02Logic.java
```

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.allowancedetail;

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_NUMERIC_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED_ANY;
import static jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames.SEARCH_CRITERIA_COMPANY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.ALLOWANCE_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_AMOUNT_SUM;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.COMPANY_RATE;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.DEPARTMENT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PAY_METHOD_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;

```

```

import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.QUANTITY;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_AMOUNT;
import static jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail.AllowanceDetailFieldName.TRANS_CURRENCY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ALLOWANCE_AMOUNT_EXISTS;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.event.GadgetMediatorParams;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.gadget.GlobalComplementJsonDataProcess;
import jp.co.slcs.kaiden2.base.foundation.helper.CurrencyHelper;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.NumericMinValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.RequiredAnyValidator;
import jp.co.slcs.kaiden2.extension_imw.foundation.gadget.AbstractPluginGadgetLogic;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimCompanyPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseCountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseAmountIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimExpenseDetailIF;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.gadget.common.ClaimTemporaryPayAmountIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.AllowanceAmountValidator;

/**
 * 日当明細(標準・外貨)ガジェット(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailV02Logic extends AbstractAllowanceDetailLogicBase implements ClaimExpenseAmountIF,
    ClaimExpenseDetailIF, ClaimTemporaryPayAmountIF, ClaimCompanyPayAmountIF, ClaimExpenseCountIF {

    /** コンストラクタ. */
    public AllowanceDetailV02Logic() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * サブクラスで追加定義します.<br>
     * <table border="1">
     * <tr>
     * <td align="center">対象フィールド名</td>
     * <td align="center">検証内容</td>
     * <td align="center">モード</td>
     * </tr>
     * <tr>
     * <td>日当金額</td>
     * <td>{@link AllowanceAmountValidator 日当金額妥当性チェック}</td>
     * <td align="center">入力/表示</td>
     * </tr>
     * <tr>
     * <td>取引通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>組織コード/プロジェクトコード</td>
     * <td>{@link RequiredAnyValidator 必須チェック(複数項目用)}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * <tr>
     * <td>会社通貨額合計</td>
     * <td>{@link NumericMinValidator 最小数値チェック}</td>
     * <td align="center">入力</td>
     * </tr>
     * </table>
     * <p>
     */
    @Override
    protected void defineValidateForNormal(GadgetMediatorParams params, boolean tmpFlg) throws KaidenSystemException,
        KaidenApplicationException {

        super.defineValidateForNormal(params, tmpFlg);

        if (!tmpFlg) {
            // 日当金額
            addValidateDefsMap(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, ALLOWANCE_CD, VALID_ALLOWANCE_AMOUNT_EXISTS,
                ALLOWANCE_CD, getGadgetDto().searchCriteriaCompany, getGadgetDto().applyAuthUserCd,
                TRANS_CURRENCY_CD, TRANS_AMOUNT, StringUtil.toString(getGadgetDto().searchCriteriaDate));
        }
    }

```

```

}
// 取引通貨額
addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, TRANS_AMOUNT, VALID_NUMERIC_MIN, "0");
if (!tmpFlg) {
    // 組織コード
    addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, DEPARTMENT_CD, VALID_REQUIRED_ANY,
        DEPARTMENT_CD, PROJECT_CD);
    // プロジェクトコード
    addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, PROJECT_CD, VALID_REQUIRED_ANY,
        DEPARTMENT_CD, PROJECT_CD);
}
// 会社通貨額
addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT, VALID_NUMERIC_MIN, "0");
// 会社通貨額合計
addValidateDefsMapOnlyInput(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE, COMPANY_AMOUNT_SUM, VALID_NUMERIC_MIN, "0");
}

/**
 * データ補完処理.
 * <p>
 * {@link AbstractPluginGadgetLogic#preProcess(GadgetMediatorParams) 個別前処理}の最後に実行されます。<br>
 * 下記補完を行います。
 * </p>
 * <ul>
 * <li>会社通貨コード: {@link CurrencyHelper#getCompanyCurrencyCd(String)}を設定</li>
 * <li>会社通貨額: 取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計: 会社通貨額 * 数量の計算結果</li>
 * </ul>
 *
 * @param params ガジェットメディエータ処理情報
 */
@Override
protected void complementJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

    String companyCurrencyCd = null;

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨コード
            if (companyCurrencyCd == null) {
                companyCurrencyCd = currencyHelper.getCompanyCurrencyCd((String) j.get(SEARCH_CRITERIA_COMPANY));
            }
            j.put(COMPANY_CURRENCY_CD, companyCurrencyCd);
        }
    }
}

/**
 * グローバルデータ補完処理をセット(ガジェット間通信を行うもの).
 * <p>
 * 以下の補完処理をセットしています。
 * <table border="1">
 * <tr>
 * <th>処理順</th>
 * <th>補完処理</th>
 * </tr>
 * <tr>
 * <td>100</td>
 * <td>{@link #globalComplementJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * <tr>
 * <td>200</td>
 * <td>{@link #calculateJsonData(GadgetMediatorParams)}</td>
 * </tr>
 * </table>
 * </p>
 *
 * @return グローバルデータ補完処理
 */
@Override
protected List<GlobalComplementJsonDataProcess> setGlobalComplementJsonDataProcess() {
    List<GlobalComplementJsonDataProcess> methodList = new ArrayList<GlobalComplementJsonDataProcess>();
    methodList.add(new GlobalComplementJsonDataProcess(100, "globalComplementJsonData"));
    methodList.add(new GlobalComplementJsonDataProcess(200, "calculateJsonData"));
    return methodList;
}

/**
 * グローバルデータ補完処理.
 * <p>
 * 下記計算を行います。
 * </p>
 * <ul>
 * <li>会社通貨額: 取引通貨額 * 通貨レートの計算結果</li>
 * <li>会社通貨額合計: 会社通貨額 * 数量の計算結果</li>
 * </ul>

```



```

*
* @param params ガジェットメディエータ処理情報
*/
public void calculateJsonData(GadgetMediatorParams params) {
    List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

    if (CollectionUtil.isEmpty(blockDat)) {
        for (Map<String, Object> j : blockDat) {
            // 会社通貨額
            String companyAmount = calcCompanyAmount((String) j.get(SEARCH_CRITERIA_COMPANY),
                (String) j.get(TRANS_CURRENCY_CD), (String) j.get(TRANS_AMOUNT), (String) j.get(COMPANY_RATE))
                .toString();
            j.put(COMPANY_AMOUNT, companyAmount);

            // 会社通貨額合計
            j.put(COMPANY_AMOUNT_SUM,
                calcCompanyAmountSum((String) j.get(SEARCH_CRITERIA_COMPANY), companyAmount,
                    (String) j.get(QUANTITY)));
        }
    }

    /**
    * グローバルデータ補完処理.
    * <p>
    * 下記補完を行います。
    * </p>
    * <ul>
    * <li>支払方法コード:ガジェット関連連携項目、支払方法コードを設定</li>
    * </ul>
    */
    * @param params ガジェットメディエータ処理情報
    */
    public void globalComplementJsonData(GadgetMediatorParams params) {
        List<Map<String, Object>> blockDat = getJsonDataDetailBlock(BLOCK_KEY_ALLOWANCE_DETAIL_TABLE);

        String gcPayMethodCd = null;

        if (CollectionUtil.isEmpty(blockDat)) {
            for (Map<String, Object> j : blockDat) {
                // 支払方法コード
                if (gcPayMethodCd == null) {
                    gcPayMethodCd = getGadgetCouplingPayMethod();
                }
                j.put(PAY_METHOD_CD, gcPayMethodCd);
            }
        }

        /**
        * {@inheritDoc}
        */
        @Override
        public Map<String, Object> getClaimExpenCount(Map<String, Object> params) {
            return super.putClaimExpenCount(params);
        }

        /** {@inheritDoc} */
        @Override
        public Map<String, Object> getClaimCompanyPayAmount(Map<String, Object> params) {
            return super.putClaimCompanyPayAmount(params);
        }

        /** {@inheritDoc} */
        @Override
        public Map<String, Object> getClaimTemporaryPayAmount(Map<String, Object> params) {
            return super.putClaimTemporaryPayAmount(params);
        }

        /**
        * {@inheritDoc}
        */
        @Override
        public Map<String, Object> getClaimExpenseDetail(Map<String, Object> params) {
            return super.putClaimExpenseDetail(params);
        }

        /**
        * {@inheritDoc}
        */
        @Override
        public Map<String, Object> getClaimExpenseAmount(Map<String, Object> params) {
            return super.putClaimExpenseAmount(params);
        }
    }
}

```

FieldName

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/conf/kaiden/gadget/allowancedetail/AllowanceDetailFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.conf.kaiden.gadget.allowancedetail;

import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;

/**
 * 日当明細ガジェット用 フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class AllowanceDetailFieldName extends ImwEntityCommonFieldNames {

    /**
     * コンストラクタ.
     */
    protected AllowanceDetailFieldName() {

        /** システム案件id. */
        public static final String SYSTEM_MATTER_ID = "systemMatterId";
        /** ユーザーデータid. */
        public static final String USER_DATA_ID = "userDataId";
        /** ガジェットクラス. */
        public static final String GADGET_CLASS = "gadgetClass";
        /** ガジェットインスタンス. */
        public static final String GADGET_INSTANCE = "gadgetInstance";
        /** タプルid. */
        public static final String TUPLE_ID = "tupleId";
        /** ガジェットid. */
        public static final String GADGET_ID = "gadgetId";
        /** ガジェットバリエーション. */
        public static final String GADGET_VARIATION = "gadgetVariation";
        /** 行番号. */
        public static final String ROW_NUMBER = "rowNumber";
        /** 日当コード. */
        public static final String ALLOWANCE_CD = "allowanceCd";
        /** 日数. */
        public static final String QUANTITY = "quantity";
        /** 経費区分コード. */
        public static final String EXP_TYPE_CD = "expTypeCd";
        /** 勘定科目コード. */
        public static final String ACCOUNT_CD = "accountCd";
        /** 税区分コード. */
        public static final String TAX_TYPE_CD = "taxTypeCd";
        /** 税率. */
        public static final String TAX_RATE = "taxRate";
        /** 取引通貨コード. */
        public static final String TRANS_CURRENCY_CD = "transCurrencyCd";
        /** 取引通貨額. */
        public static final String TRANS_AMOUNT = "transAmount";
        /** 会社通貨コード. */
        public static final String COMPANY_CURRENCY_CD = "companyCurrencyCd";
        /** 会社通貨換算レート. */
        public static final String COMPANY_RATE = "companyRate";
        /** 会社通貨額. */
        public static final String COMPANY_AMOUNT = "companyAmount";
        /** 会社通貨税抜額. */
        public static final String COMPANY_AMOUNT_EXC_TAX = "companyAmountExcTax";
        /** 会社通貨税額. */
        public static final String COMPANY_AMOUNT_TAX = "companyAmountTax";
        /** 会社通貨額合計. */
        public static final String COMPANY_AMOUNT_SUM = "companyAmountSum";
        /** 支払分類コード. */
        public static final String PAY_TYPE_CD = "payTypeCd";
        /** 支払方法コード. */
        public static final String PAY_METHOD_CD = "payMethodCd";
        /** 支払先コード. */
        public static final String PAYEE_CD = "payeeCd";
        /** 支払先名. */
        public static final String PAYEE_NAME = "payeeName";
        /** 組織セットコード. */
        public static final String DEPARTMENT_SET_CD = "departmentSetCd";
        /** 組織コード. */
        public static final String DEPARTMENT_CD = "departmentCd";
        /** プロジェクトコード. */
        public static final String PROJECT_CD = "projectCd";

        /** 取引通貨名称. */
        public static final String TRANS_CURRENCY_NAME = "transCurrencyName";
        /** 会社通貨名称. */
        public static final String COMPANY_CURRENCY_NAME = "companyCurrencyName";
        /** 部門名称. */

```

```

public static final String DEPARTMENT_NAME = "departmentName";
/** プロジェクト名称. */
public static final String PROJECT_NAME = "projectName";
/** 日当名称. */
public static final String ALLOWANCE_NAME = "allowanceName";
/** 経費区分名称. */
public static final String EXP_TYPE_NAME = "expTypeName";
/** 勘定科目名称. */
public static final String ACCOUNT_NAME = "accountName";
/** 税区分名称. */
public static final String TAX_TYPE_NAME = "taxTypeName";
}

```

Manager

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/AllowanceDetailManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail;

import java.util.List;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.PluginGadgetManager;
import org.seasar.util.beans.util.BeanMap;

/**
 * 日当明細ガジェットマネージャクラスのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface AllowanceDetailManager extends PluginGadgetManager<AllowanceDetailManagerDto> {

    /** {@link #selectView(CriteriaForSqlDto) リスト検索(マスタ情報付与)}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM = "selectView.sql";

    /** {@link #selectViewForCopy(CriteriaForSqlDto) 申請書コピー用検索}用SQLファイル名. */
    static final String SELECT_SQLFILE_NM_FOR_COPY = "selectViewForCopy.sql";

    /** {@link #selectForJournalSourceDtl(CriteriaForSqlDto) 仕訳連携用明細登録データ検索}用SQLファイル名. */
    static final String SELECT_FOR_JOURNAL_SOURCE_DTL = "selectForJournalSourceDtl.sql";

    /**
     * 仕訳連携用明細テーブル登録用データ取得.
     *
     * @param criteria 検索条件
     * @return 検索結果
     */
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria);
}

```

Manager(実装)

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/impl/AllowanceDetailManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.impl;

import java.util.ArrayList;
import java.util.List;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.OrderType;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.extension_imw.foundation.conf.ImwEntityCommonFieldNames;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.AbstractWfManager;
import jp.co.slcs.kaiden2.extension_imw.foundation.manager.CriteriaForSqlDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail.AllowanceDetailManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30tAllowanceDetailService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.Ka30tAllowanceDetailService;
import org.seasar.util.beans.util.BeanMap;

/**
 * 日当明細テーブルへの各種操作を提供するAPI.
 * 本APIは、主にデータベース操作を行いますが、より業務的な機能を付加した操作を提供します。<br>
 * 検索系機能を使用する際には、{@link CriteriaForSqlDto}クラスをパラメータとして渡し、<br>
 * {@link AllowanceDetailManagerDto}リストが戻り値として返却されます。<br>
 * <br>
 * 更新系機能を使用する際には、{@link AllowanceDetailManagerDto}クラスまたはそのリストをパラメータとして渡し、<br>
 * 処理件数を数値型で返却します。<br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>

```

```

* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>主キー検索</td>
* <td>selectByPk</td>
* <td>主キー検索</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>リスト検索</td>
* <td>selectList</td>
* <td>主キー検索</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>表示案件検索</td>
* <td>selectView</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>コピー案件検索</td>
* <td>selectViewForCopy</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件検索</td>
* <td>selectViewArchive</td>
* <td>リスト検索(マスタ情報付与)</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括挿入</td>
* <td>entry</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括更新</td>
* <td>renew</td>
* <td>更新</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>一括削除</td>
* <td>removeAll</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>過去案件登録</td>
* <td>matterArchiveRegister</td>
* <td>登録</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件処理時削除</td>
* <td>matterArchivePostDelete</td>
* <td>削除</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>過去案件一括削除</td>
* <td>removeAllArchive</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能別DTOクラスのI/Oマトリクス</b>
* <table border="1">
* <tr>
* <td colspan="2" rowspan="2" align="center"><b>DTO</b></td>
* <td colspan="2" align="center"><b>主キー</b><br>
* 検索</td>

```

[illegible]

```
* <td align="center"> O</td>
* <td align="center"> O</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* <td align="center"> O</td>
* <td align="center"> -</td>
* </tr>
* </table>
* <br>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
```

```

public class AllowanceDetailManagerImpl extends AbstractWfManager<AllowanceDetailManagerDto> implements
    AllowanceDetailManager {

    /** K30tAllowanceDetailService. */
    @Resource
    protected K30tAllowanceDetailService k30tAllowanceDetailService;

    /** Ka30tAllowanceDetailService. */
    @Resource
    protected Ka30tAllowanceDetailService ka30tAllowanceDetailService;

    /**
     * {@@inheritDoc}
     */
    @Override
    public AllowanceDetailManagerDto selectByPk(AllowanceDetailManagerDto criteria) {
        return super.selectByPkIgnoreAuthority(k30tAllowanceDetailService, criteria);
    }

    /**
     * {@@inheritDoc}
     */
    @Override
    public List<AllowanceDetailManagerDto> selectList(AllowanceDetailManagerDto criteria) {
        return super.selectPkUsedUserDataId(k30tAllowanceDetailService, criteria);
    }

    /**
     * {@@inheritDoc}
     */
    @Override
    public List<AllowanceDetailManagerDto> selectView(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForView(k30tAllowanceDetailService, SELECT_SQLFILE_NM, criteria, order);
    }

    /** {@@inheritDoc} */
    @Override
    public List<AllowanceDetailManagerDto> selectViewForCopy(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectBySqlForViewForCopy(k30tAllowanceDetailService, SELECT_SQLFILE_NM_FOR_COPY, criteria, order);
    }

    /**
     * {@@inheritDoc}
     */
    @Override
    public List<AllowanceDetailManagerDto> selectViewArchive(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));
        return super.selectListForViewForArchive(ka30tAllowanceDetailService, criteria, order);
    }

    /**

```

```

    * {@inheritDoc}
    */
    @Override
    public List<BeanMap> selectForJournalSourceDtl(CriteriaForSqlDto criteria) {
        List<OrderDto> order = new ArrayList<OrderDto>();
        order.add(new OrderDto(OrderType.ASC, ImwEntityCommonFieldNames.ROW_NUMBER));

        return super.selectBySqlFileUsedBeanMap(SELECT_FOR_JOURNAL_SOURCE_DTL, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<AllowanceDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.insertBatch(k30tAllowanceDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<AllowanceDetailManagerDto> dataList) {
        return CollectionUtil.summaryInt(super.renewBatch(k30tAllowanceDetailService, dataList));
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int remove(AllowanceDetailManagerDto data) {
        return super.deleteUsedUserDataId(k30tAllowanceDetailService, data);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAll(AllowanceDetailManagerDto data) {
        return super.deleteNoUsedUserDataId(k30tAllowanceDetailService, data);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchiveRegister(CriteriaForSqlDto criteria, List<AllowanceDetailManagerDto> dataList) {
        return super.entryArchive(ka30tAllowanceDetailService, criteria, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int matterArchivePostDelete(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataId(k30tAllowanceDetailService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeAllArchive(CriteriaForSqlDto criteria) {
        return super.deleteNoUsedUserDataIdForArchive(ka30tAllowanceDetailService, criteria);
    }
}

```

ManagerDto

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/AllowanceDetailManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.allowancedetail;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;

/**
 * 日当明細ガジェットマネージャDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class AllowanceDetailManagerDto extends KaidenDto {

```

```

/** システム案件id. */
public String systemMatterId;
/** ユーザーデータid. */
public String userDataId;
/** ガジェットクラス. */
public String gadgetClass;
/** ガジェットインスタンス. */
public Long gadgetInstance;
/** タブルid. */
public Long tupleId;
/** ガジェットid. */
public String gadgetId;
/** ガジェットバージョン. */
public String gadgetVariation;
/** 行番号. */
public Long rowNumber;
/** 日当コード. */
public String allowanceCd;
/** 日数. */
public BigDecimal quantity;
/** 経費区分コード. */
public String expTypeCd;
/** 勘定科目コード. */
public String accountCd;
/** 税区分コード. */
public String taxTypeCd;
/** 税率. */
public BigDecimal taxRate;
/** 取引通貨コード. */
public String transCurrencyCd;
/** 取引通貨額. */
public BigDecimal transAmount;
/** 会社通貨コード. */
public String companyCurrencyCd;
/** 会社通貨換算レート. */
public BigDecimal companyRate;
/** 会社通貨額. */
public BigDecimal companyAmount;
/** 会社通貨税抜額. */
public BigDecimal companyAmountExcTax;
/** 会社通貨税額. */
public BigDecimal companyAmountTax;
/** 会社通貨額合計. */
public BigDecimal companyAmountSum;
/** 支払分類コード. */
public String payTypeCd;
/** 支払方法コード. */
public String payMethodCd;
/** 支払先コード. */
public String payeeCd;
/** 支払先名. */
public String payeeName;
/** 組織セットコード. */
public String departmentSetCd;
/** 組織コード. */
public String departmentCd;
/** プロジェクトコード. */
public String projectCd;
/** 予備項目a. */
public String extensionA;
/** 予備項目b. */
public String extensionB;
/** 予備項目c. */
public String extensionC;
/** 予備項目d. */
public String extensionD;
/** 予備項目e. */
public String extensionE;
/** 予備項目f. */
public String extensionF;
/** 予備項目g. */
public String extensionG;
/** 予備項目h. */
public String extensionH;
/** 予備項目i. */
public String extensionI;
/** 予備項目j. */
public String extensionJ;
/** 登録日時. */
public Timestamp entryTs;
/** 登録者コード. */
public String entryUserCd;
/** 更新カウンタ. */
public Long renewCnt;
/** 更新日時. */
public Timestamp renewTs;

```



```

/** 更新者コード. */
public String renewUserCd;

/** 取引通貨名称. */
public String transCurrencyName;
/** 会社通貨名称. */
public String companyCurrencyName;
/** 部門名称. */
public String departmentName;
/** プロジェクト名称. */
public String projectName;
/** 日当名称. */
public String allowanceName;
/** 経費区分名称. */
public String expTypeName;
/** 勘定科目名称. */
public String accountName;
/** 税区分名称. */
public String taxTypeName;
}

```

Java (S2JDBC)

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tAllowanceDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 日当明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30t_allowance_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30tAllowanceDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;

    /** ガジェットインスタンス. */
    @Id
    @Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long gadgetInstanceId;

    /** タプルid. */
    @Id
    @Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
    public Long tupleId;

    /** ガジェットid. */
    @Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
    public String gadgetId;

    /** ガジェットバリエーション. */
    @Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetVariation;
}

```

```

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 日当コード. */
@Column(name = "allowance_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String allowanceCd;

/** 日数. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */

```

```

@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tAllowanceDetail.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * (過去案件) 日当明細テーブル.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "ka30t_allowance_detail")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class Ka30tAllowanceDetail extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** システム案件id. */
    @Id
    @Column(name = "system_matter_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String systemMatterId;

    /** ユーザーデータid. */
    @Id
    @Column(name = "user_data_id", columnDefinition = "varchar2(20)", nullable = false, unique = false)
    public String userDataId;

    /** ガジェットクラス. */
    @Id
    @Column(name = "gadget_class", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String gadgetClass;
}

```

```

/** ガジェットインスタンス. */
@Id
@Column(name = "gadget_instance", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long gadgetInstance;

/** タプルid. */
@Id
@Column(name = "tuple_id", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long tupleId;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** アーカイブ年月. */
@Column(name = "archive_month", columnDefinition = "varchar2(6)", nullable = false, unique = false)
public String archiveMonth;

/** ガジェットid. */
@Column(name = "gadget_id", columnDefinition = "varchar2(220)", nullable = false, unique = false)
public String gadgetId;

/** ガジェットバリエーション. */
@Column(name = "gadget_variation", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String gadgetVariation;

/** 行番号. */
@Column(name = "row_number", columnDefinition = "number(18,0)", nullable = false, unique = false)
public Long rowNumber;

/** 日当コード. */
@Column(name = "allowance_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String allowanceCd;

/** 日当名称. */
@Column(name = "allowance_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String allowanceName;

/** 日数. */
@Column(name = "quantity", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal quantity;

/** 経費区分コード. */
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String expTypeCd;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String taxTypeCd;

/** 税率. */
@Column(name = "tax_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal taxRate;

/** 取引通貨コード. */
@Column(name = "trans_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String transCurrencyCd;

/** 取引通貨額. */
@Column(name = "trans_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal transAmount;

/** 会社通貨コード. */
@Column(name = "company_currency_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String companyCurrencyCd;

/** 会社通貨換算レート. */
@Column(name = "company_rate", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyRate;

/** 会社通貨額. */
@Column(name = "company_amount", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmount;

/** 会社通貨税抜額. */
@Column(name = "company_amount_exc_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountExcTax;

/** 会社通貨税額. */
@Column(name = "company_amount_tax", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountTax;

```

```

/** 会社通貨額合計. */
@Column(name = "company_amount_sum", columnDefinition = "number(28,10)", nullable = true, unique = false)
public BigDecimal companyAmountSum;

/** 支払分類コード. */
@Column(name = "pay_type_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payTypeCd;

/** 支払方法コード. */
@Column(name = "pay_method_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payMethodCd;

/** 支払先コード. */
@Column(name = "payee_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String payeeCd;

/** 支払先名. */
@Column(name = "payee_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payeeName;

/** 組織セットコード. */
@Column(name = "department_set_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentSetCd;

/** 組織コード. */
@Column(name = "department_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String departmentCd;

/** プロジェクトコード. */
@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String projectCd;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeName;

/** 勘定科目名称. */
@Column(name = "account_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String accountName;

/** 税区分名. */
@Column(name = "tax_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String taxTypeName;

/** 取引通貨名. */
@Column(name = "trans_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String transCurrencyName;

/** 会社通貨名. */
@Column(name = "company_currency_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String companyCurrencyName;

/** 支払分類名称. */
@Column(name = "pay_type_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payTypeName;

/** 立替フラグ. */
@Column(name = "payment_flag", columnDefinition = "varchar2(1)", nullable = true, unique = false)
public String paymentFlag;

/** 支払方法名称. */
@Column(name = "pay_method_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String payMethodName;

/** 組織名. */
@Column(name = "department_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String departmentName;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String projectName;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */

```

```

@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30tAllowanceDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30tAllowanceDetail} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30tAllowanceDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30tAllowanceDetailNames() {
        super();
    }

    /**
     * systemMatterId のプロパティ名を返却.
     *
     * @return systemMatterId のプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataId のプロパティ名を返却.
     *
     * @return userDataId のプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClass のプロパティ名を返却.
     *
     * @return gadgetClass のプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstance のプロパティ名を返却.
     *

```

```

    * @return gadgetInstanceのプロパティ名
    */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却。
     *
     * @return tupleIdのプロパティ名
     */
    public static PropertyName<Long> tupleId() {
        return new PropertyName<Long>("tupleId");
    }

    /**
     * gadgetIdのプロパティ名を返却。
     *
     * @return gadgetIdのプロパティ名
     */
    public static PropertyName<String> gadgetId() {
        return new PropertyName<String>("gadgetId");
    }

    /**
     * gadgetVariationのプロパティ名を返却。
     *
     * @return gadgetVariationのプロパティ名
     */
    public static PropertyName<String> gadgetVariation() {
        return new PropertyName<String>("gadgetVariation");
    }

    /**
     * rowNumberのプロパティ名を返却。
     *
     * @return rowNumberのプロパティ名
     */
    public static PropertyName<Long> rowNumber() {
        return new PropertyName<Long>("rowNumber");
    }

    /**
     * allowanceCdのプロパティ名を返却。
     *
     * @return allowanceCdのプロパティ名
     */
    public static PropertyName<String> allowanceCd() {
        return new PropertyName<String>("allowanceCd");
    }

    /**
     * quantityのプロパティ名を返却。
     *
     * @return quantityのプロパティ名
     */
    public static PropertyName<BigDecimal> quantity() {
        return new PropertyName<BigDecimal>("quantity");
    }

    /**
     * expTypeCdのプロパティ名を返却。
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * accountCdのプロパティ名を返却。
     *
     * @return accountCdのプロパティ名
     */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却。
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }
}

```

```

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名
 */
public static PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>("taxRate");
}

/**
 * transCurrencyCdのプロパティ名を返却.
 *
 * @return transCurrencyCdのプロパティ名
 */
public static PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>("transCurrencyCd");
}

/**
 * transAmountのプロパティ名を返却.
 *
 * @return transAmountのプロパティ名
 */
public static PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>("transAmount");
}

/**
 * companyCurrencyCdのプロパティ名を返却.
 *
 * @return companyCurrencyCdのプロパティ名
 */
public static PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>("companyCurrencyCd");
}

/**
 * companyRateのプロパティ名を返却.
 *
 * @return companyRateのプロパティ名
 */
public static PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>("companyRate");
}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>("companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>("companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>("companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名

```



```

*/
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却。
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却。
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却。
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却。
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却。
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却。
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * extensionAのプロパティ名を返却。
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却。
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

```

```

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却。
 *
 * @return renewCntのプロパティ名
 */

```

```

public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30tAllowanceDetailNames extends PropertyName<K30tAllowanceDetail> {

    /**
     * インスタンスを構築.
     */
    public _K30tAllowanceDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30tAllowanceDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30tAllowanceDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却.
     *
     * @return gadgetInstanceのプロパティ名
     */
    public PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }
}

```

```

}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * allowanceCdのプロパティ名を返却.
 *
 * @return allowanceCdのプロパティ名
 */
public PropertyName<String> allowanceCd() {
    return new PropertyName<String>(this, "allowanceCd");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * taxRateのプロパティ名を返却.
 *
 * @return taxRateのプロパティ名

```

```

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>(this, "taxRate");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>(this, "transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>(this, "transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>(this, "companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>(this, "companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>(this, "companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>(this, "companyAmountTax");
    }

    /**
     * companyAmountSumのプロパティ名を返却.
     *
     * @return companyAmountSumのプロパティ名
     */
    public PropertyName<BigDecimal> companyAmountSum() {
        return new PropertyName<BigDecimal>(this, "companyAmountSum");
    }

    /**
     * payTypeCdのプロパティ名を返却.
     *
     * @return payTypeCdのプロパティ名
     */
    public PropertyName<String> payTypeCd() {
        return new PropertyName<String>(this, "payTypeCd");
    }

```

```

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public PropertyName<String> departmentCd() {
    return new PropertyName<String>(this, "departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */

```

```

public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**

```

```

    * renewTsのプロパティ名を返却.
    *
    * @return renewTsのプロパティ名
    */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/Ka30tAllowanceDetailNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link Ka30tAllowanceDetail}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class Ka30tAllowanceDetailNames {

    /**
     * デフォルトコンストラクタ.
     */
    private Ka30tAllowanceDetailNames() {
        super();
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public static PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataIdのプロパティ名を返却.
     *
     * @return userDataIdのプロパティ名
     */
    public static PropertyName<String> userDataId() {
        return new PropertyName<String>("userDataId");
    }

    /**
     * gadgetClassのプロパティ名を返却.
     *
     * @return gadgetClassのプロパティ名
     */
    public static PropertyName<String> gadgetClass() {
        return new PropertyName<String>("gadgetClass");
    }

    /**
     * gadgetInstanceのプロパティ名を返却.
     *
     * @return gadgetInstanceのプロパティ名
     */
    public static PropertyName<Long> gadgetInstance() {
        return new PropertyName<Long>("gadgetInstance");
    }

    /**
     * tupleIdのプロパティ名を返却.
     *
     * @return tupleIdのプロパティ名
     */

```



```

public static PropertyName<Long> tupleId() {
    return new PropertyName<Long>("tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public static PropertyName<String> localeId() {
    return new PropertyName<String>("localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public static PropertyName<String> archiveMonth() {
    return new PropertyName<String>("archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public static PropertyName<String> gadgetId() {
    return new PropertyName<String>("gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public static PropertyName<String> gadgetVariation() {
    return new PropertyName<String>("gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public static PropertyName<Long> rowNumber() {
    return new PropertyName<Long>("rowNumber");
}

/**
 * allowanceCdのプロパティ名を返却.
 *
 * @return allowanceCdのプロパティ名
 */
public static PropertyName<String> allowanceCd() {
    return new PropertyName<String>("allowanceCd");
}

/**
 * allowanceNameのプロパティ名を返却.
 *
 * @return allowanceNameのプロパティ名
 */
public static PropertyName<String> allowanceName() {
    return new PropertyName<String>("allowanceName");
}

/**
 * quantityのプロパティ名を返却.
 *
 * @return quantityのプロパティ名
 */
public static PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>("quantity");
}

/**
 * expTypeCdのプロパティ名を返却.
 *
 * @return expTypeCdのプロパティ名
 */
public static PropertyName<String> expTypeCd() {
    return new PropertyName<String>("expTypeCd");
}

/**
 * expTypeCdのプロパティ名を返却.

```

```

    * accountCdのプロパティ名を返却.
    *
    * @return accountCdのプロパティ名
    */
    public static PropertyName<String> accountCd() {
        return new PropertyName<String>("accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public static PropertyName<String> taxTypeCd() {
        return new PropertyName<String>("taxTypeCd");
    }

    /**
     * taxRateのプロパティ名を返却.
     *
     * @return taxRateのプロパティ名
     */
    public static PropertyName<BigDecimal> taxRate() {
        return new PropertyName<BigDecimal>("taxRate");
    }

    /**
     * transCurrencyCdのプロパティ名を返却.
     *
     * @return transCurrencyCdのプロパティ名
     */
    public static PropertyName<String> transCurrencyCd() {
        return new PropertyName<String>("transCurrencyCd");
    }

    /**
     * transAmountのプロパティ名を返却.
     *
     * @return transAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> transAmount() {
        return new PropertyName<BigDecimal>("transAmount");
    }

    /**
     * companyCurrencyCdのプロパティ名を返却.
     *
     * @return companyCurrencyCdのプロパティ名
     */
    public static PropertyName<String> companyCurrencyCd() {
        return new PropertyName<String>("companyCurrencyCd");
    }

    /**
     * companyRateのプロパティ名を返却.
     *
     * @return companyRateのプロパティ名
     */
    public static PropertyName<BigDecimal> companyRate() {
        return new PropertyName<BigDecimal>("companyRate");
    }

    /**
     * companyAmountのプロパティ名を返却.
     *
     * @return companyAmountのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmount() {
        return new PropertyName<BigDecimal>("companyAmount");
    }

    /**
     * companyAmountExcTaxのプロパティ名を返却.
     *
     * @return companyAmountExcTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountExcTax() {
        return new PropertyName<BigDecimal>("companyAmountExcTax");
    }

    /**
     * companyAmountTaxのプロパティ名を返却.
     *
     * @return companyAmountTaxのプロパティ名
     */
    public static PropertyName<BigDecimal> companyAmountTax() {
        return new PropertyName<BigDecimal>("companyAmountTax");
    }

```

```

    return new PropertyName<BigDecimal>( companyAmountTax );
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public static PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>("companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public static PropertyName<String> payTypeCd() {
    return new PropertyName<String>("payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public static PropertyName<String> payMethodCd() {
    return new PropertyName<String>("payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public static PropertyName<String> payeeCd() {
    return new PropertyName<String>("payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public static PropertyName<String> payeeName() {
    return new PropertyName<String>("payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public static PropertyName<String> departmentSetCd() {
    return new PropertyName<String>("departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *
 * @return departmentCdのプロパティ名
 */
public static PropertyName<String> departmentCd() {
    return new PropertyName<String>("departmentCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public static PropertyName<String> projectCd() {
    return new PropertyName<String>("projectCd");
}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public static PropertyName<String> expTypeName() {
    return new PropertyName<String>("expTypeName");
}

/**
 * accountNameのプロパティ名を返却.
 *

```

```

    * @return accountNameのプロパティ名
    */
    public static PropertyName<String> accountName() {
        return new PropertyName<String>("accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却.
     *
     * @return taxTypeNameのプロパティ名
     */
    public static PropertyName<String> taxTypeName() {
        return new PropertyName<String>("taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却.
     *
     * @return transCurrencyNameのプロパティ名
     */
    public static PropertyName<String> transCurrencyName() {
        return new PropertyName<String>("transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却.
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public static PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>("companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却.
     *
     * @return payTypeNameのプロパティ名
     */
    public static PropertyName<String> payTypeName() {
        return new PropertyName<String>("payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却.
     *
     * @return paymentFlagのプロパティ名
     */
    public static PropertyName<String> paymentFlag() {
        return new PropertyName<String>("paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却.
     *
     * @return payMethodNameのプロパティ名
     */
    public static PropertyName<String> payMethodName() {
        return new PropertyName<String>("payMethodName");
    }

    /**
     * departmentNameのプロパティ名を返却.
     *
     * @return departmentNameのプロパティ名
     */
    public static PropertyName<String> departmentName() {
        return new PropertyName<String>("departmentName");
    }

    /**
     * projectNameのプロパティ名を返却.
     *
     * @return projectNameのプロパティ名
     */
    public static PropertyName<String> projectName() {
        return new PropertyName<String>("projectName");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }
}

```

```

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名

```

```

*/
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _Ka30tAllowanceDetailNames extends PropertyName<Ka30tAllowanceDetail> {

    /**
     * インスタンスを構築.
     */
    public _Ka30tAllowanceDetailNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _Ka30tAllowanceDetailNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _Ka30tAllowanceDetailNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * systemMatterIdのプロパティ名を返却.
     *
     * @return systemMatterIdのプロパティ名
     */
    public PropertyName<String> systemMatterId() {
        return new PropertyName<String>("systemMatterId");
    }

    /**
     * userDataldのプロパティ名を返却.
     *
     * @return userDataldのプロパティ名
     */
    public PropertyName<String> userDatald() {

```

```

    return new PropertyName<String>(this, "userDataId");
}

/**
 * gadgetClassのプロパティ名を返却.
 *
 * @return gadgetClassのプロパティ名
 */
public PropertyName<String> gadgetClass() {
    return new PropertyName<String>(this, "gadgetClass");
}

/**
 * gadgetInstanceのプロパティ名を返却.
 *
 * @return gadgetInstanceのプロパティ名
 */
public PropertyName<Long> gadgetInstance() {
    return new PropertyName<Long>(this, "gadgetInstance");
}

/**
 * tupleIdのプロパティ名を返却.
 *
 * @return tupleIdのプロパティ名
 */
public PropertyName<Long> tupleId() {
    return new PropertyName<Long>(this, "tupleId");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * archiveMonthのプロパティ名を返却.
 *
 * @return archiveMonthのプロパティ名
 */
public PropertyName<String> archiveMonth() {
    return new PropertyName<String>(this, "archiveMonth");
}

/**
 * gadgetIdのプロパティ名を返却.
 *
 * @return gadgetIdのプロパティ名
 */
public PropertyName<String> gadgetId() {
    return new PropertyName<String>(this, "gadgetId");
}

/**
 * gadgetVariationのプロパティ名を返却.
 *
 * @return gadgetVariationのプロパティ名
 */
public PropertyName<String> gadgetVariation() {
    return new PropertyName<String>(this, "gadgetVariation");
}

/**
 * rowNumberのプロパティ名を返却.
 *
 * @return rowNumberのプロパティ名
 */
public PropertyName<Long> rowNumber() {
    return new PropertyName<Long>(this, "rowNumber");
}

/**
 * allowanceCdのプロパティ名を返却.
 *
 * @return allowanceCdのプロパティ名
 */
public PropertyName<String> allowanceCd() {
    return new PropertyName<String>(this, "allowanceCd");
}

/**
 * allowanceNameのプロパティ名を返却.

```

```

*
* @return allowanceNameのプロパティ名
*/
public PropertyName<String> allowanceName() {
    return new PropertyName<String>(this, "allowanceName");
}

/**
* quantityのプロパティ名を返却.
*
* @return quantityのプロパティ名
*/
public PropertyName<BigDecimal> quantity() {
    return new PropertyName<BigDecimal>(this, "quantity");
}

/**
* expTypeCdのプロパティ名を返却.
*
* @return expTypeCdのプロパティ名
*/
public PropertyName<String> expTypeCd() {
    return new PropertyName<String>(this, "expTypeCd");
}

/**
* accountCdのプロパティ名を返却.
*
* @return accountCdのプロパティ名
*/
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
* taxTypeCdのプロパティ名を返却.
*
* @return taxTypeCdのプロパティ名
*/
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
* taxRateのプロパティ名を返却.
*
* @return taxRateのプロパティ名
*/
public PropertyName<BigDecimal> taxRate() {
    return new PropertyName<BigDecimal>(this, "taxRate");
}

/**
* transCurrencyCdのプロパティ名を返却.
*
* @return transCurrencyCdのプロパティ名
*/
public PropertyName<String> transCurrencyCd() {
    return new PropertyName<String>(this, "transCurrencyCd");
}

/**
* transAmountのプロパティ名を返却.
*
* @return transAmountのプロパティ名
*/
public PropertyName<BigDecimal> transAmount() {
    return new PropertyName<BigDecimal>(this, "transAmount");
}

/**
* companyCurrencyCdのプロパティ名を返却.
*
* @return companyCurrencyCdのプロパティ名
*/
public PropertyName<String> companyCurrencyCd() {
    return new PropertyName<String>(this, "companyCurrencyCd");
}

/**
* companyRateのプロパティ名を返却.
*
* @return companyRateのプロパティ名
*/
public PropertyName<BigDecimal> companyRate() {
    return new PropertyName<BigDecimal>(this, "companyRate");
}

```



```

}

/**
 * companyAmountのプロパティ名を返却.
 *
 * @return companyAmountのプロパティ名
 */
public PropertyName<BigDecimal> companyAmount() {
    return new PropertyName<BigDecimal>(this, "companyAmount");
}

/**
 * companyAmountExcTaxのプロパティ名を返却.
 *
 * @return companyAmountExcTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountExcTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountExcTax");
}

/**
 * companyAmountTaxのプロパティ名を返却.
 *
 * @return companyAmountTaxのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountTax() {
    return new PropertyName<BigDecimal>(this, "companyAmountTax");
}

/**
 * companyAmountSumのプロパティ名を返却.
 *
 * @return companyAmountSumのプロパティ名
 */
public PropertyName<BigDecimal> companyAmountSum() {
    return new PropertyName<BigDecimal>(this, "companyAmountSum");
}

/**
 * payTypeCdのプロパティ名を返却.
 *
 * @return payTypeCdのプロパティ名
 */
public PropertyName<String> payTypeCd() {
    return new PropertyName<String>(this, "payTypeCd");
}

/**
 * payMethodCdのプロパティ名を返却.
 *
 * @return payMethodCdのプロパティ名
 */
public PropertyName<String> payMethodCd() {
    return new PropertyName<String>(this, "payMethodCd");
}

/**
 * payeeCdのプロパティ名を返却.
 *
 * @return payeeCdのプロパティ名
 */
public PropertyName<String> payeeCd() {
    return new PropertyName<String>(this, "payeeCd");
}

/**
 * payeeNameのプロパティ名を返却.
 *
 * @return payeeNameのプロパティ名
 */
public PropertyName<String> payeeName() {
    return new PropertyName<String>(this, "payeeName");
}

/**
 * departmentSetCdのプロパティ名を返却.
 *
 * @return departmentSetCdのプロパティ名
 */
public PropertyName<String> departmentSetCd() {
    return new PropertyName<String>(this, "departmentSetCd");
}

/**
 * departmentCdのプロパティ名を返却.
 *

```

```

    * @return departmentCdのプロパティ名
    */
    public PropertyName<String> departmentCd() {
        return new PropertyName<String>(this, "departmentCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>(this, "projectCd");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public PropertyName<String> expTypeName() {
        return new PropertyName<String>(this, "expTypeName");
    }

    /**
     * accountNameのプロパティ名を返却.
     *
     * @return accountNameのプロパティ名
     */
    public PropertyName<String> accountName() {
        return new PropertyName<String>(this, "accountName");
    }

    /**
     * taxTypeNameのプロパティ名を返却.
     *
     * @return taxTypeNameのプロパティ名
     */
    public PropertyName<String> taxTypeName() {
        return new PropertyName<String>(this, "taxTypeName");
    }

    /**
     * transCurrencyNameのプロパティ名を返却.
     *
     * @return transCurrencyNameのプロパティ名
     */
    public PropertyName<String> transCurrencyName() {
        return new PropertyName<String>(this, "transCurrencyName");
    }

    /**
     * companyCurrencyNameのプロパティ名を返却.
     *
     * @return companyCurrencyNameのプロパティ名
     */
    public PropertyName<String> companyCurrencyName() {
        return new PropertyName<String>(this, "companyCurrencyName");
    }

    /**
     * payTypeNameのプロパティ名を返却.
     *
     * @return payTypeNameのプロパティ名
     */
    public PropertyName<String> payTypeName() {
        return new PropertyName<String>(this, "payTypeName");
    }

    /**
     * paymentFlagのプロパティ名を返却.
     *
     * @return paymentFlagのプロパティ名
     */
    public PropertyName<String> paymentFlag() {
        return new PropertyName<String>(this, "paymentFlag");
    }

    /**
     * payMethodNameのプロパティ名を返却.
     *
     * @return payMethodNameのプロパティ名
     */
    public PropertyName<String> payMethodName() {
        return new PropertyName<String>(this, "payMethodName");
    }

```

```

/**
 * departmentNameのプロパティ名を返却.
 *
 * @return departmentNameのプロパティ名
 */
public PropertyName<String> departmentName() {
    return new PropertyName<String>(this, "departmentName");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */

```

```

/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30tAllowanceDetail;

/**
 * {@link K30tAllowanceDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30tAllowanceDetailService extends GenerateService<K30tAllowanceDetail> {
    //
}
```

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/Ka30tAllowanceDetailService.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.Ka30tAllowanceDetail;

/**
 * {@link Ka30tAllowanceDetail} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class Ka30tAllowanceDetailService extends GenerateService<Ka30tAllowanceDetail> {
    //
}
```

Condition

- Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailCondition.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30tAllowanceDetail} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30tAllowanceDetailCondition extends
    AbstractEntityCondition<K30tAllowanceDetailCondition> {

    /**
     * インスタンスを構築。
     */
    public K30tAllowanceDetailCondition() {
    }

    /**
     * インスタンスを構築。
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30tAllowanceDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterId の条件。 */
    public NotNullableViewCondition<K30tAllowanceDetailCondition> systemMatterId =
        new NotNullableViewCondition<K30tAllowanceDetailCondition>("systemMatterId", this);

    /** userDatId の条件。 */
    public NotNullableViewCondition<K30tAllowanceDetailCondition> userDatId =
        new NotNullableViewCondition<K30tAllowanceDetailCondition>("userDatId", this);
}
```

```

    new NullableStringCondition<K30tAllowanceDetailCondition> ( gadgetClass , this);

    /** gadgetClassの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetClass =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> gadgetInstance =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> tupleId =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("tupleId", this);

    /** gadgetIdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetId =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetId", this);

    /** gadgetVariationの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> gadgetVariation =
        new NullableStringCondition<K30tAllowanceDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, Long> rowNumber =
        new NullableCondition<K30tAllowanceDetailCondition, Long>("rowNumber", this);

    /** allowanceCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> allowanceCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("allowanceCd", this);

    /** quantityの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> quantity =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("quantity", this);

    /** expTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> expTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("expTypeCd", this);

    /** accountCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> accountCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("accountCd", this);

    /** taxTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> taxTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("taxTypeCd", this);

    /** taxRateの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> taxRate =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("taxRate", this);

    /** transCurrencyCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> transCurrencyCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("transCurrencyCd", this);

    /** transAmountの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> transAmount =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("transAmount", this);

    /** companyCurrencyCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> companyCurrencyCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("companyCurrencyCd", this);

    /** companyRateの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyRate =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyRate", this);

    /** companyAmountの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmount =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmount", this);

    /** companyAmountExcTaxの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountExcTax =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountExcTax", this);

    /** companyAmountTaxの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountTax =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountTax", this);

    /** companyAmountSumの条件. */
    public NullableCondition<K30tAllowanceDetailCondition, BigDecimal> companyAmountSum =
        new NullableCondition<K30tAllowanceDetailCondition, BigDecimal>("companyAmountSum", this);

    /** payTypeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> payTypeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("payTypeCd", this);

    /** payMethodCdの条件. */

```

```

    public NullableStringCondition<K30tAllowanceDetailCondition> payMethodCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("payMethodCd", this);

    /** payeeCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> payeeCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("payeeCd", this);

    /** payeeNameの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> payeeName =
        new NullableStringCondition<K30tAllowanceDetailCondition>("payeeName", this);

    /** departmentSetCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> departmentSetCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("departmentSetCd", this);

    /** departmentCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> departmentCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("departmentCd", this);

    /** projectCdの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> projectCd =
        new NullableStringCondition<K30tAllowanceDetailCondition>("projectCd", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionA =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionB =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionC =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionD =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionD", this);

    /** extensionEの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionE =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionE", this);

    /** extensionFの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionF =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionG =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionH =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionI =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30tAllowanceDetailCondition> extensionJ =
        new NullableStringCondition<K30tAllowanceDetailCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullCondition<K30tAllowanceDetailCondition, Timestamp> entryTs =
        new NotNullCondition<K30tAllowanceDetailCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullStringCondition<K30tAllowanceDetailCondition> entryUserCd =
        new NotNullStringCondition<K30tAllowanceDetailCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullCondition<K30tAllowanceDetailCondition, Long> renewCnt =
        new NotNullCondition<K30tAllowanceDetailCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullCondition<K30tAllowanceDetailCondition, Timestamp> renewTs =
        new NotNullCondition<K30tAllowanceDetailCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullStringCondition<K30tAllowanceDetailCondition> renewUserCd =
        new NotNullStringCondition<K30tAllowanceDetailCondition>("renewUserCd", this);
}

```

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.math.BigDecimal;
import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link Ka30tAllowanceDetail}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class Ka30tAllowanceDetailCondition extends
    AbstractEntityCondition<Ka30tAllowanceDetailCondition> {

    /**
     * インスタンスを構築.
     */
    public Ka30tAllowanceDetailCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public Ka30tAllowanceDetailCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** systemMatterIdの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> systemMatterId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("systemMatterId", this);

    /** userDataIdの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> userDataId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("userDataId", this);

    /** gadgetClassの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetClass =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetClass", this);

    /** gadgetInstanceの条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> gadgetInstance =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("gadgetInstance", this);

    /** tupleIdの条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> tupleId =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("tupleId", this);

    /** localeIdの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> localeId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("localeId", this);

    /** archiveMonthの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> archiveMonth =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("archiveMonth", this);

    /** gadgetIdの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetId =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetId", this);

    /** gadgetVariationの条件. */
    public NotNullStringCondition<Ka30tAllowanceDetailCondition> gadgetVariation =
        new NotNullStringCondition<Ka30tAllowanceDetailCondition>("gadgetVariation", this);

    /** rowNumberの条件. */
    public NotNullCondition<Ka30tAllowanceDetailCondition, Long> rowNumber =
        new NotNullCondition<Ka30tAllowanceDetailCondition, Long>("rowNumber", this);

    /** allowanceCdの条件. */
    public NullableStringCondition<Ka30tAllowanceDetailCondition> allowanceCd =
        new NullableStringCondition<Ka30tAllowanceDetailCondition>("allowanceCd", this);

    /** allowanceNameの条件. */
    public NullableStringCondition<Ka30tAllowanceDetailCondition> allowanceName =
        new NullableStringCondition<Ka30tAllowanceDetailCondition>("allowanceName", this);

```



```

/** quantityの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> quantity =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("quantity", this);

/** expTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> expTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("expTypeCd", this);

/** accountCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> accountCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> taxTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("taxTypeCd", this);

/** taxRateの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> taxRate =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("taxRate", this);

/** transCurrencyCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> transCurrencyCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("transCurrencyCd", this);

/** transAmountの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> transAmount =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("transAmount", this);

/** companyCurrencyCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> companyCurrencyCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("companyCurrencyCd", this);

/** companyRateの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyRate =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyRate", this);

/** companyAmountの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmount =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmount", this);

/** companyAmountExcTaxの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountExcTax =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountExcTax", this);

/** companyAmountTaxの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountTax =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountTax", this);

/** companyAmountSumの条件. */
public NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal> companyAmountSum =
    new NullableCondition<Ka30tAllowanceDetailCondition, BigDecimal>("companyAmountSum", this);

/** payTypeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payTypeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payTypeCd", this);

/** payMethodCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payMethodCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payMethodCd", this);

/** payeeCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payeeCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payeeCd", this);

/** payeeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payeeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payeeName", this);

/** departmentSetCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentSetCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentSetCd", this);

/** departmentCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentCd", this);

/** projectCdの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> projectCd =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("projectCd", this);

/** expTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> expTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("expTypeName", this);

/** accountNameの条件. */

```

```

public NullableStringCondition<Ka30tAllowanceDetailCondition> accountName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("accountName", this);

/** taxTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> taxTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("taxTypeName", this);

/** transCurrencyNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> transCurrencyName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("transCurrencyName", this);

/** companyCurrencyNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> companyCurrencyName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("companyCurrencyName", this);

/** payTypeNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payTypeName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payTypeName", this);

/** paymentFlagの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> paymentFlag =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("paymentFlag", this);

/** payMethodNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> payMethodName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("payMethodName", this);

/** departmentNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> departmentName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("departmentName", this);

/** projectNameの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> projectName =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("projectName", this);

/** extensionAの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionA =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionB =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionC =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionD =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionE =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionF =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionG =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionH =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionI =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<Ka30tAllowanceDetailCondition> extensionJ =
    new NullableStringCondition<Ka30tAllowanceDetailCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp> entryTs =
    new NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<Ka30tAllowanceDetailCondition> entryUserCd =
    new NotNullableStringCondition<Ka30tAllowanceDetailCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<Ka30tAllowanceDetailCondition, Long> renewCnt =
    new NotNullableCondition<Ka30tAllowanceDetailCondition, Long>("renewCnt", this);

```

```

/** renewTsの条件. */
public NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp> renewTs =
    new NotNullableCondition<Ka30tAllowanceDetailCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<Ka30tAllowanceDetailCondition> renewUserCd =
    new NotNullableStringCondition<Ka30tAllowanceDetailCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService/selectView.sql

```

select
    allowance_t.system_matter_id,
    allowance_t.user_data_id,
    allowance_t.gadget_class,
    allowance_t.gadget_instance,
    allowance_t.tuple_id,
    allowance_t.gadget_id,
    allowance_t.gadget_variation,
    allowance_t.row_number,
    allowance_t.allowance_cd,
    allowance_t.quantity,
    allowance_t.exp_type_cd,
    allowance_t.account_cd,
    allowance_t.tax_type_cd,
    allowance_t.tax_rate,
    allowance_t.trans_currency_cd,
    allowance_t.trans_amount,
    allowance_t.company_currency_cd,
    allowance_t.company_rate,
    allowance_t.company_amount,
    allowance_t.company_amount_exc_tax,
    allowance_t.company_amount_tax,
    allowance_t.company_amount_sum,
    allowance_t.pay_type_cd,
    allowance_t.pay_method_cd,
    allowance_t.payee_cd,
    allowance_t.payee_name,
    allowance_t.department_set_cd,
    allowance_t.department_cd,
    allowance_t.project_cd,
    allowance_t.extension_a,
    allowance_t.extension_b,
    allowance_t.extension_c,
    allowance_t.extension_d,
    allowance_t.extension_e,
    allowance_t.extension_f,
    allowance_t.extension_g,
    allowance_t.extension_h,
    allowance_t.extension_i,
    allowance_t.extension_j,
    allowance_t.entry_ts,
    allowance_t.entry_user_cd,
    allowance_t.renew_cnt,
    allowance_t.renew_ts,
    allowance_t.renew_user_cd,
    c1.search_criteria_company,
    cu1.currency_name as trans_currency_name,
    cu2.currency_name as company_currency_name,
    dept1.department_name,
    pro1.project_name,
    allowance_m.allowance_name,
    exp.exp_type_name,
    account.account_name,
    tax.tax_type_name
from
    k30t_allowance_detail allowance_t
inner join
    k20c_matter_control c1
on
    allowance_t.system_matter_id = c1.system_matter_id
and
    allowance_t.user_data_id = c1.user_data_id
left outer join imm_currency cu1
on
    c1.search_criteria_company = cu1.company_cd
and
    allowance_t.trans_currency_cd = cu1.currency_cd

```

```

and
  /*localeld*/ja' = cu1.locale_id
and
  cu1.delete_flag = '0'
left outer join imm_currency cu2
on
  c1.search_criteria_company = cu2.company_cd
and
  allowance_t.company_currency_cd = cu2.currency_cd
and
  /*localeld*/ja' = cu2.locale_id
and
  cu2.delete_flag = '0'
left outer join
  imm_department dept1
on
  c1.search_criteria_company = dept1.company_cd
and
  allowance_t.department_set_cd = dept1.department_set_cd
and
  allowance_t.department_cd = dept1.department_cd
and
  /*localeld*/ja' = dept1.locale_id
and
  c1.search_criteria_date >= dept1.start_date
and
  c1.search_criteria_date < dept1.end_date
and
  dept1.delete_flag = '0'
left outer join
  k30m_project pro1
on
  c1.search_criteria_company = pro1.company_cd
and
  allowance_t.project_cd = pro1.project_cd
and
  c1.search_criteria_date >= pro1.start_date
and
  c1.search_criteria_date < pro1.end_date
and
  /*localeld*/ja' = pro1.locale_id
and
  pro1.delete_flag = '0'
left outer join
  k30m_allowance allowance_m
on
  c1.search_criteria_company = allowance_m.company_cd
and
  allowance_t.allowance_cd = allowance_m.allowance_cd
and
  /*localeld*/ja' = allowance_m.locale_id
and
  c1.search_criteria_date >= allowance_m.start_date
and
  c1.search_criteria_date < allowance_m.end_date
and
  allowance_m.delete_flag = '0'
left outer join
  k30m_exp_type exp
on
  c1.search_criteria_company = exp.company_cd
and
  allowance_t.exp_type_cd = exp.exp_type_cd
and
  /*localeld*/ja' = exp.locale_id
and
  c1.search_criteria_date >= exp.start_date
and
  c1.search_criteria_date < exp.end_date
and
  exp.delete_flag = '0'
left outer join
  k30m_account account
on
  c1.search_criteria_company = account.company_cd
and
  allowance_t.account_cd = account.account_cd
and
  /*localeld*/ja' = account.locale_id
and
  c1.search_criteria_date >= account.start_date
and
  c1.search_criteria_date < account.end_date
and
  account.delete_flag = '0'
left outer join

```

```

k30m_tax_type tax
on
  c1.search_criteria_company = tax.company_cd
and
  allowance_t.tax_type_cd = tax.tax_type_cd
and
  /*localeId*ja' = tax.locale_id
and
  c1.search_criteria_date >= tax.start_date
and
  c1.search_criteria_date < tax.end_date
and
  tax.delete_flag = '0'
where
  allowance_t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
  allowance_t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
  allowance_t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
  allowance_t.gadget_instance = /*gadgetInstance*/0
/*END*/
/*IF gadgetId != null*/
and
  allowance_t.gadget_id = /*gadgetId*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
  c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

コピー

- **Path**
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30tAllowanceDetailService/selectViewForCopy.sql

```

select
  allowance_t.system_matter_id,
  allowance_t.user_data_id,
  allowance_t.gadget_class,
  allowance_t.gadget_instance,
  allowance_t.tuple_id,
  allowance_t.gadget_id,
  allowance_t.gadget_variation,
  allowance_t.row_number,
  allowance_t.allowance_cd,
  allowance_t.quantity,
  allowance_t.exp_type_cd,
  allowance_t.account_cd,
  allowance_t.tax_type_cd,
  allowance_t.tax_rate,
  allowance_t.trans_currency_cd,
  allowance_t.trans_amount,
  allowance_t.company_currency_cd,
  allowance_t.company_rate,
  allowance_t.company_amount,
  allowance_t.company_amount_exc_tax,
  allowance_t.company_amount_tax,
  allowance_t.company_amount_sum,
  allowance_t.pay_type_cd,
  allowance_t.pay_method_cd,
  allowance_t.payee_cd,
  allowance_t.payee_name,
  allowance_t.department_set_cd,
  allowance_t.department_cd,
  allowance_t.project_cd,
  allowance_t.extension_a,
  allowance_t.extension_b,
  allowance_t.extension_c,
  allowance_t.extension_d,
  allowance_t.extension_e,
  allowance_t.extension_f,
  allowance_t.extension_g,
  allowance_t.extension_h,
  allowance_t.extension_i,
  allowance_t.extension_j,
  allowance_t.entry_ts,

```

```

allowance_t.entry_user_cd,
allowance_t.renew_cnt,
allowance_t.renew_ts,
allowance_t.renew_user_cd,
c1.search_criteria_company,
cu1.currency_name as trans_currency_name,
cu2.currency_name as company_currency_name,
dept1.department_name,
pro1.project_name,
allowance_m.allowance_name,
exp.exp_type_name,
account.account_name,
tax.tax_type_name
from
k30t_allowance_detail allowance_t
inner join
k20c_matter_control c1
on
allowance_t.system_matter_id = c1.system_matter_id
and
allowance_t.user_data_id = c1.user_data_id
left outer join imm_currency cu1
on
c1.search_criteria_company = cu1.company_cd
and
allowance_t.trans_currency_cd = cu1.currency_cd
and
/*localeId*/ja' = cu1.locale_id
and
cu1.delete_flag = '0'
left outer join imm_currency cu2
on
c1.search_criteria_company = cu2.company_cd
and
allowance_t.company_currency_cd = cu2.currency_cd
and
/*localeId*/ja' = cu2.locale_id
and
cu2.delete_flag = '0'
left outer join
imm_department dept1
on
c1.search_criteria_company = dept1.company_cd
and
allowance_t.department_set_cd = dept1.department_set_cd
and
allowance_t.department_cd = dept1.department_cd
and
/*localeId*/ja' = dept1.locale_id
and
/*searchCriteriaDate*/" >= dept1.start_date
and
/*searchCriteriaDate*/" < dept1.end_date
and
dept1.delete_flag = '0'
left outer join
k30m_project pro1
on
c1.search_criteria_company = pro1.company_cd
and
allowance_t.project_cd = pro1.project_cd
and
/*searchCriteriaDate*/" >= pro1.start_date
and
/*searchCriteriaDate*/" < pro1.end_date
and
/*localeId*/ja' = pro1.locale_id
and
pro1.delete_flag = '0'
left outer join
k30m_allowance allowance_m
on
c1.search_criteria_company = allowance_m.company_cd
and
allowance_t.allowance_cd = allowance_m.allowance_cd
and
/*localeId*/ja' = allowance_m.locale_id
and
/*searchCriteriaDate*/" >= allowance_m.start_date
and
/*searchCriteriaDate*/" < allowance_m.end_date
and
allowance_m.delete_flag = '0'
left outer join
k30m_exp_type exp
on

```

```

c1.search_criteria_company = exp.company_cd
and
allowance_t.exp_type_cd = exp.exp_type_cd
and
/*localeId*/ja' = exp.locale_id
and
/*searchCriteriaDate*/" >= exp.start_date
and
/*searchCriteriaDate*/" < exp.end_date
and
exp.delete_flag = '0'
left outer join
k30m_account account
on
c1.search_criteria_company = account.company_cd
and
allowance_t.account_cd = account.account_cd
and
/*localeId*/ja' = account.locale_id
and
/*searchCriteriaDate*/" >= account.start_date
and
/*searchCriteriaDate*/" < account.end_date
and
account.delete_flag = '0'
left outer join
k30m_tax_type tax
on
c1.search_criteria_company = tax.company_cd
and
allowance_t.tax_type_cd = tax.tax_type_cd
and
/*localeId*/ja' = tax.locale_id
and
/*searchCriteriaDate*/" >= tax.start_date
and
/*searchCriteriaDate*/" < tax.end_date
and
tax.delete_flag = '0'
where
allowance_t.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
allowance_t.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetId != null*/
and
allowance_t.gadget_id = /*gadgetId*/
/*END*/
/*IF gadgetClass != null*/
and
allowance_t.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
allowance_t.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/system_matter_id
/*END*/

```

仕訳連携

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/manager/allowancedetail/impl/AllowanceDetailManagerImpl/selectForJournalSourceDtl

```

select
ad.system_matter_id      as system_matter_id,
ad.user_data_id          as user_data_id,
ad.gadget_class          as gadget_class,
ad.gadget_instance       as gadget_instance,
ad.gadget_id             as gadget_id,
ad.tuple_id              as tuple_id,
ad.gadget_variation      as gadget_variation,
i1.contents_id           as contents_id,
i1.contents_version_id   as contents_version_id,
'30'                     as journal_data_type_cd,
ad.row_number            as row_number,
ad.exp_type_cd           as exp_type_cd,
ad.account_cd            as account_cd,
ad.tax_type_cd           as tax_type_cd,

```

```

tax.tax_rate      as tax_rate,
ad.quantity       as quantity,
ad.trans_currency_cd as trans_currency_cd,
ad.trans_amount   as trans_amount,
ad.company_currency_cd as company_currency_cd,
ad.company_rate    as company_rate,
ad.company_amount  as company_amount,
ad.company_amount_exc_tax as company_amount_exc_tax,
ad.company_amount_tax as company_amount_tax,
ad.company_amount_sum as company_amount_sum,
ad.payee_cd        as customer_cd,
ad.payee_name      as customer_name,
ad.department_set_cd as department_set_cd,
ad.department_cd    as department_cd,
ad.project_cd       as project_cd,
ad.pay_method_cd    as pay_method_cd,
at.allowance_name   as detail_summary1,
ad.extension_a      as detail_extension_a,
ad.extension_b      as detail_extension_b,
ad.extension_c      as detail_extension_c,
ad.extension_d      as detail_extension_d,
ad.extension_e      as detail_extension_e,
ad.extension_f      as detail_extension_f,
ad.extension_g      as detail_extension_g,
ad.extension_h      as detail_extension_h,
ad.extension_i      as detail_extension_i,
ad.extension_j      as detail_extension_j,
'0'                as journal_status
from
k30t_allowance_detail ad
inner join
k20c_matter_info i1
on
ad.system_matter_id = i1.system_matter_id
inner join
k20c_matter_control c1
on
ad.system_matter_id = c1.system_matter_id
and
ad.user_data_id = c1.user_data_id
left outer join
k30m_tax_type tax
on
c1.search_criteria_company = tax.company_cd
and
ad.tax_type_cd = tax.tax_type_cd
and
i1.locale_id_apply = tax.locale_id
and
c1.search_criteria_date >= tax.start_date
and
c1.search_criteria_date < tax.end_date
and
tax.delete_flag = '0'
left outer join
k30m_allowance at
on
c1.search_criteria_company = at.company_cd
and
ad.allowance_cd = at.allowance_cd
and
i1.locale_id_apply = at.locale_id
and
c1.search_criteria_date >= at.start_date
and
c1.search_criteria_date < at.end_date
and
at.delete_flag = '0'
where
ad.system_matter_id = /*systemMatterId*/
/*IF userDataId != null*/
and
ad.user_data_id = /*userDataId*/
/*END*/
/*IF gadgetClass != null*/
and
ad.gadget_class = /*gadgetClass*/
/*END*/
/*IF gadgetInstance != null*/
and
ad.gadget_instance = /*gadgetInstance*/
/*END*/
/*IF gadgetId != null*/
and
ad.gadget_id = /*gadgetId*/
/*END*/

```



```

/*IF searchCriteriaCompany != null*/
and
c1.search_criteria_company = /*searchCriteriaCompany*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/ad.row_number
/*END*/

```

JSP

PC

- Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/allowanceDetail/allowanceDetailV02.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_wk" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <set var="pwkfCommonIncluded" scope="request">true</set>
</c:if>

<!-- allowanceDetail v02 -->
<set var="gadgetClass">allowanceDetail</set>
<set var="gadgetVariation">v02</set>
<set var="gadgetInstance">${param.gadget_instance}</set>
<set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</set>
<set var="accessUrl">kaiden/gadget/allowanceDetailV02</set>
<set var="gadgetBlockId">allowanceDetailTable</set>
<set var="gadgetBlockHeaderId">allowanceDetailHeader</set>
<set var="ttableId">allowanceDetailTable</set>
<set var="ttableControllerBlockId">ttableController</set>
<set var="searchCriteriaCompany">${param.searchCriteriaCompany}</set>
<set var="searchCriteriaDate">${param.searchCriteriaDate}</set>
<set var="companyCurrencyCd">${param.companyCurrencyCd}</set>
<set var="applyAuthUserCode">${param.applyAuthUserCode}</set>

<set var="gadget" value="${appDto.gadgets[gadgetId]}" />
<set var="detail" value="${gadget.blocks[allowanceDetailTable]}" />

<!-- ttable -->
<set var="allowAddDetail" value="${'1' == detail.allowAddFlag}" />
<set var="allowDelDetail" value="${'1' == detail.allowDelFlag}" />

<!-- Message -->
<set var="msgRateReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0008) %></set>
<set var="msgAllowanceReget"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0009) %></set>
<set var="messageSearchWarning"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0010) %></set>
<set var="messageSearchWarningSub"><%=PropertyJspUtil.getMessage(ProductWorkflowProp.M.W0011) %></set>

<!-- Surface -->
<set var="surfaceChapterTitle">${gadget.surfaces["chapterTitle"]}</set>
<set var="surfaceAllowanceAmountReget">${gadget.surfaces["allowanceAmountReget"]}</set>
<set var="surfaceRateReget">${gadget.surfaces["rateReget"]}</set>
<set var="surfaceAllowance">${gadget.surfaces["allowance"]}</set>
<set var="surfaceTransAmount">${gadget.surfaces["amount"]}</set>
<set var="surfaceTransCurrency">${gadget.surfaces["amount_transCurrency"]}</set>
<set var="surfaceCompanyRate">${gadget.surfaces["amount_transRate"]}</set>
<set var="surfaceCompanyAmount">${gadget.surfaces["amount_companyAmount"]}</set>
<set var="surfaceQuantity">${gadget.surfaces["quantity"]}</set>
<set var="surfaceCompanyAmountSum">${gadget.surfaces["companyAmountSum"]}</set>
<set var="surfaceExpType">${gadget.surfaces["expType"]}</set>
<set var="surfaceAccount">${gadget.surfaces["account"]}</set>
<set var="surfaceTaxType">${gadget.surfaces["taxType"]}</set>
<set var="surfaceDepartment">${gadget.surfaces["department"]}</set>
<set var="surfaceProject">${gadget.surfaces["project"]}</set>

<!-- Mode -->
<set var="modeAllowanceAmountReget">${detail.fields["allowanceCd"].inputType}</set>
<set var="modeRateReget">${detail.fields["allowanceCd"].inputType}</set>
<set var="modeAllowance">${detail.fields["allowanceCd"].inputType}</set>
<!-- <set var="modeTransAmount">${detail.fields["transAmount"].inputType}</set> -->
<!-- <set var="modeTransCurrency">${detail.fields["transCurrencyCd"].inputType}</set> -->
<!-- <set var="modeCompanyRate">${detail.fields["companyRate"].inputType}</set> -->
<!-- <set var="modeCompanyAmount">${detail.fields["companyAmount"].inputType}</set> -->

```

```

<c:set var="modeTransAmount">${detail.fields["allowanceCd"].inputType}</c:set>
<c:set var="modeTransCurrency">${detail.fields["allowanceCd"].inputType}</c:set>
<c:set var="modeCompanyRate">${detail.fields["allowanceCd"].inputType}</c:set>
<c:set var="modeCompanyAmount">${detail.fields["allowanceCd"].inputType}</c:set>
<c:set var="modeQuantity">${detail.fields["quantity"].inputType}</c:set>
<%-- <c:set var="modeCompanyAmountSum">${detail.fields["companyAmountSum"].inputType}</c:set> --%>
<c:set var="modeCompanyAmountSum">${modeTransAmount == modeQuantity ? modeTransAmount : "0"}</c:set>
<c:set var="modeExpTypeCd">${detail.fields["expTypeCd"].inputType}</c:set>
<c:set var="modeAccountCd">${detail.fields["accountCd"].inputType}</c:set>
<c:set var="modeTaxTypeCd">${detail.fields["taxTypeCd"].inputType}</c:set>
<c:set var="modeDepartmentCd">${detail.fields["departmentCd"].inputType}</c:set>
<c:set var="modeProjectCd">${detail.fields["projectCd"].inputType}</c:set>

<%-- RequiredMark --%>
<c:set var="reqAllowance">${detail.fields["allowanceCd"].requiredFlag}</c:set>
<c:set var="reqTransAmount">${detail.fields["transAmount"].requiredFlag}</c:set>
<c:set var="reqTransCurrency">${detail.fields["transCurrencyCd"].requiredFlag}</c:set>
<c:set var="reqCompanyRate">${detail.fields["companyRate"].requiredFlag}</c:set>
<c:set var="reqCompanyAmount">${detail.fields["companyAmount"].requiredFlag}</c:set>
<c:set var="reqQuantity">${detail.fields["quantity"].requiredFlag}</c:set>
<c:set var="reqCompanyAmountSum">${detail.fields["companyAmountSum"].requiredFlag}</c:set>
<c:set var="reqExpTypeCd">${detail.fields["expTypeCd"].requiredFlag}</c:set>
<c:set var="reqAccountCd">${detail.fields["accountCd"].requiredFlag}</c:set>
<c:set var="reqTaxTypeCd">${detail.fields["taxTypeCd"].requiredFlag}</c:set>
<c:set var="reqDepartmentCd">${detail.fields["departmentCd"].requiredFlag}</c:set>
<c:set var="reqProjectCd">${detail.fields["projectCd"].requiredFlag}</c:set>

<%-- Validator --%>
<c:set var="validAllowanceCd_def">{k_allowanceExists:["allowanceCd", 'searchCriteriaCompany', 'searchCriteriaDate', 'false'], k_allowanceAmount:["allowanceCd", 'searchCriteriaComp
<c:set var="validAllowanceCd_apply">${("true" == reqAllowance ? "k_required:[""] : ""})</c:set>
<c:set var="validAllowanceCd_tempsave"></c:set>
<c:set var="validAllowanceCd_reapply">${("true" == reqAllowance ? "k_required:[""] : ""})</c:set>
<c:set var="validAllowanceCd_process">${("true" == reqAllowance ? "k_required:[""] : ""})</c:set>

<c:set var="validQuantity_def">{k_numeric:[""], k_numericRange:["1","999"]</c:set>
<c:set var="validQuantity_apply">${("true" == reqQuantity ? "k_required:[""] : ""})</c:set>
<c:set var="validQuantity_tempsave"></c:set>
<c:set var="validQuantity_reapply">${("true" == reqQuantity ? "k_required:[""] : ""})</c:set>
<c:set var="validQuantity_process">${("true" == reqQuantity ? "k_required:[""] : ""})</c:set>

<c:set var="validCompanyAmountSum_def">{k_currencyAmnt:["", 'searchCriteriaCompany'], k_numericMin:["0"]</c:set>
<c:set var="validCompanyAmountSum_apply">${("true" == reqCompanyAmountSum ? "k_required:[""] : ""})</c:set>
<c:set var="validCompanyAmountSum_tempsave"></c:set>
<c:set var="validCompanyAmountSum_reapply">${("true" == reqCompanyAmountSum ? "k_required:[""] : ""})</c:set>
<c:set var="validCompanyAmountSum_process">${("true" == reqCompanyAmountSum ? "k_required:[""] : ""})</c:set>

<c:set var="validExpTypeCd_def">{k_expTypeExists:["expTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']</c:set>
<c:set var="validExpTypeCd_apply">${("true" == reqExpTypeCd ? "k_required:[""] : ""})</c:set>
<c:set var="validExpTypeCd_tempsave"></c:set>
<c:set var="validExpTypeCd_reapply">${("true" == reqExpTypeCd ? "k_required:[""] : ""})</c:set>
<c:set var="validExpTypeCd_process">${("true" == reqExpTypeCd ? "k_required:[""] : ""})</c:set>

<c:set var="validAccountCd_def">{k_accountExists:["accountCd", 'searchCriteriaCompany', 'searchCriteriaDate']</c:set>
<c:set var="validAccountCd_apply">${("true" == reqAccountCd ? "k_required:[""] : ""})</c:set>
<c:set var="validAccountCd_tempsave"></c:set>
<c:set var="validAccountCd_reapply">${("true" == reqAccountCd ? "k_required:[""] : ""})</c:set>
<c:set var="validAccountCd_process">${("true" == reqAccountCd ? "k_required:[""] : ""})</c:set>

<c:set var="validTaxType_def">{k_taxTypeExists:["taxTypeCd", 'searchCriteriaCompany', 'searchCriteriaDate']</c:set>
<c:set var="validTaxType_apply">${("true" == reqTaxTypeCd ? "k_required:[""] : ""})</c:set>
<c:set var="validTaxType_tempsave"></c:set>
<c:set var="validTaxType_reapply">${("true" == reqTaxTypeCd ? "k_required:[""] : ""})</c:set>
<c:set var="validTaxType_process">${("true" == reqTaxTypeCd ? "k_required:[""] : ""})</c:set>

<c:set var="validDepartmentCd_def">{k_departmentExists:["searchCriteriaCompany", 'departmentSetCd', 'departmentCd', 'searchCriteriaDate']</c:set>
<c:set var="validDepartmentCd_apply">{k_requiredAny:["departmentCd", 'projectCd']</c:set>
<c:set var="validDepartmentCd_tempsave"></c:set>
<c:set var="validDepartmentCd_reapply">{k_requiredAny:["departmentCd", 'projectCd']</c:set>
<c:set var="validDepartmentCd_process">{k_requiredAny:["departmentCd", 'projectCd']</c:set>

<c:set var="validProjectCd_def">{k_projectExists:["projectCd", 'searchCriteriaCompany', 'searchCriteriaDate']</c:set>
<c:set var="validProjectCd_apply">{k_requiredAny:["departmentCd", 'projectCd']</c:set>
<c:set var="validProjectCd_tempsave"></c:set>
<c:set var="validProjectCd_reapply">{k_requiredAny:["departmentCd", 'projectCd']</c:set>
<c:set var="validProjectCd_process">{k_requiredAny:["departmentCd", 'projectCd']</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
  gadgetVariation="${gadgetVariation}"
  gadgetInstance="${gadgetInstance}"
  gadgetTitle="${surfaceChapterTitle}"
  gadgetId="${gadgetId}"
  hideGadgetTitle="true"
  style="width:850px;overflow:hidden;">
<k:blockContainer id="${gadgetBlockHeaderId}">
  <div class="imui-chapter-title">
    <h2 style="display:inline-block;width:50%">
      <k:surface

```

```

<n:surface
  name="chapterTitle"
  surfaceKey="{surfaceChapterTitle}"
/>
</h2>
<k:a
  class="kaiden-ui-icon-refresh-gray16_nosize"
  name="allowanceAmountReget"
  mode="{modeTransAmount}"
  text="{surfaceAllowanceAmountReget}"
  style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem;margin-left:1em;"
/>
<k:a
  class="kaiden-ui-icon-refresh-gray16_nosize"
  name="rateReget"
  mode="{modeRateReget}"
  text="{surfaceRateReget}"
  style="vertical-align:middle;float:right;padding-left:20px;font-size:1.2rem"
/>
</div>
</k:blockContainer>
<k:blockContainer id="caution" style="display:block;"></k:blockContainer>
<form id="gadget_form" onsubmit="return false;">
<k:tmtable id="{tmtableId}" class="imui-form kaiden-ui-tmt-form" style="table-layout:fixed" rowNumberTarget="rowNumber">
  <colgroup>
    <col style="width:40px;overflow:hidden">
    <col style="width:235px;overflow:hidden">
    <col style="width:185px;overflow:hidden">
    <col style="width:185px;overflow:hidden">
    <col style="width:60px;overflow:hidden">
    <col style="width:125px;overflow:hidden">
  </colgroup>
  <thead>
    <tr>
      <td rowspan="2" style="vertical-align:middle;">
        <k:tmtableOperationPanel position="header"
          allowAdd="{allowAddDetail}"
          allowDel="{allowDelDetail}"
        />
      </td>
      <th>
        <k:surface
          name="allowanceSurface"
          surfaceKey="{surfaceAllowance}"
          requiredMark="{reqAllowance}"
        />
      </th>
      <th colspan="2">
        <k:surface
          name="transAmountSurface"
          surfaceKey="{surfaceTransAmount}"
          requiredMark="{reqTransAmount}"
        /></th>
      <th>
        <k:surface
          name="quantitySurface"
          surfaceKey="{surfaceQuantity}"
          requiredMark="{reqQuantity}"
        />
      </th>
      <th>
        <k:surface
          name="companyAmountSumSurface"
          surfaceKey="{surfaceCompanyAmountSum}"
          requiredMark="{reqCompanyAmountSum}"
        />
      </th>
    </tr>
    <tr>
      <th colspan="2">
        <k:surface
          name="expTypeSurface"
          surfaceKey="{surfaceExpType}"
          requiredMark="{reqExpTypeCd}"
          style="display:inline;"
        />
        <k:surface
          name="accountSurface"
          surfaceKey="{surfaceAccount}"
          style="display:inline;"
        />
        <k:surface
          name="taxTypeSurface"
          surfaceKey="{surfaceTaxType}"
          style="display:inline;"
        />
      </th>

```

```

<th>
  <k:surface
    name="departmentSurface"
    surfaceKey="{surfaceDepartment}"
    requiredMark="{reqDepartmentCd}"
  />
</th>
<th colspan="2">
  <k:surface
    name="projectSurface"
    surfaceKey="{surfaceProject}"
    requiredMark="{reqProjectCd}"
  />
</th>
</tr>
</thead>
<tbody>
<tr>
  <td rowspan="2">
    <k:tmtableOperationPanel position="detail"
      allowAdd="{allowAddDetail}"
      allowDel="{allowDelDetail}"
      name="rowNumber"
    />
  </td>
  <td>
    <k:hidden
      name="allowanceCd"
      mode="0"
    />
    <kmsw:allowance
      name="allowanceName"
      mode="{modeAllowance}"
      imeMode="on"
      surfaceKey="{surfaceAllowance}"
      resultValue="allowanceName"
      resultChain="{transAmount:'amount','allowanceCd':allowanceCd,'transCurrencyCd':currencyCd}"
      primaryTarget="allowanceCd"
      style="width:150px;"
      allowanceCriteriaUserCd="{param.applyAuthUserCode}"
      searchCriteriaCompany="searchCriteriaCompany"
      searchCriteriaDate="searchCriteriaDate"
      searchCriteriaAllowanceCd="allowanceCd"
      validationDefault="{validAllowanceCd_def}"
      validationApply="{validAllowanceCd_apply}"
      validationTempsave="{validAllowanceCd_tempsave}"
      validationReapply="{validAllowanceCd_reapply}"
      validationProcess="{validAllowanceCd_process}"
    />
  </td>
<td colspan="2">
    <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/genericCurrencyInputParts.jsp">
      <jsp:param name="gadgetId" value="{gadgetId}" />
      <jsp:param name="tmtableId" value="{tmtableId}" />
      <jsp:param name="searchCriteriaCompany" value="{searchCriteriaCompany}" />
      <jsp:param name="companyCurrencyCd" value="{companyCurrencyCd}" />
      <%-- 金額 --%>
      <jsp:param name="modeTransAmnt" value="{modeTransAmount}" />
      <jsp:param name="surfaceTransAmnt" value="{surfaceTransAmount}" />
      <jsp:param name="requiredTransAmntApply" value="{reqTransAmount}" />
      <jsp:param name="requiredTransAmntReapply" value="{reqTransAmount}" />
      <jsp:param name="requiredTransAmntProcess" value="{reqTransAmount}" />
      <jsp:param name="readonlyTransAmount" value="true" />
      <%-- 通貨 --%>
      <jsp:param name="modeTransCryCd" value="{modeTransCurrency}" />
      <jsp:param name="surfaceTransCryCd" value="{surfaceTransCurrency}" />
      <jsp:param name="requiredTransCryCdApply" value="{reqTransCurrency}" />
      <jsp:param name="requiredTransCryCdReapply" value="{reqTransCurrency}" />
      <jsp:param name="requiredTransCryCdProcess" value="{reqTransCurrency}" />
      <jsp:param name="readonlyTransCryCd" value="true" />
      <%-- レート --%>
      <jsp:param name="modeCmpRate" value="{modeCompanyRate}" />
      <jsp:param name="surfaceCmpRate" value="{surfaceCompanyRate}" />
      <jsp:param name="requiredCmpRateApply" value="{reqTransCurrency}" />
      <jsp:param name="requiredCmpRateReapply" value="{reqTransCurrency}" />
      <jsp:param name="requiredCmpRateProcess" value="{reqTransCurrency}" />
      <%-- 換算額 --%>
      <jsp:param name="modeCmpAmnt" value="{modeCompanyAmount}" />
      <jsp:param name="surfaceCmpAmnt" value="{surfaceCompanyAmount}" />
      <jsp:param name="requiredCmpAmntApply" value="{reqCompanyAmount}" />
      <jsp:param name="requiredCmpAmntReapply" value="{reqCompanyAmount}" />
      <jsp:param name="requiredCmpAmntProcess" value="{reqCompanyAmount}" />
    </jsp:include>
  </td>
<td>
  <k:numericText

```

```

        name="quantity"
        mode="${modeQuantity}"
        surfaceKey="${surfaceQuantity}"
        style="width:40px;"
        maxLength="3"
        validationDefault="${validQuantity_def}"
        validationApply="${validQuantity_apply}"
        validationTempsave="${validQuantity_tempsave}"
        validationReapply="${validQuantity_reapply}"
        validationProcess="${validQuantity_process}"
    />
</td>
<td style="text-align:right">
    <k:numericText
        name="companyAmountSum"
        mode="${modeCompanyAmountSum}"
        formatted="view"
        surfaceKey="${surfaceCompanyAmouontSum}"
        class="kaiden-ui-cmpAmnt"
        style="width:100px;"
        tabIndex="-1"
        readonly="readonly"
        validationDefault="${validCompanyAmountSum_def}"
        validationApply="${validCompanyAmountSum_apply}"
        validationTempsave="${validCompanyAmountSum_tempsave}"
        validationReapply="${validCompanyAmountSum_reapply}"
        validationProcess="${validCompanyAmountSum_process}"
    />
</td>
</tr>
<tr>
<td colspan="2">
    <k:hidden
        name="expTypeCd"
        mode="0"
    />
    <kmsw:expType
        name="expTypeName"
        mode="${modeExpTypeCd}"
        imeMode="on"
        surfaceKey="${surfaceExpType}"
        resultValue="expTypeName"
        resultChain="{expTypeCd:'expTypeCd', accountCd:'accountCd', accountName:'accountName', taxTypeCd:'taxTypeCd', taxTypeName:'taxTypeName'}"
        primaryTarget="expTypeCd"
        searchCriteriaDate="searchCriteriaDate"
        searchCriteriaCompany="searchCriteriaCompany"
        searchCriteriaExpTypeCd="expTypeCd"
        style="width:150px;"
        validationDefault="${validExpTypeCd_def}"
        validationApply="${validExpTypeCd_apply}"
        validationTempsave="${validExpTypeCd_tempsave}"
        validationReapply="${validExpTypeCd_reapply}"
        validationProcess="${validExpTypeCd_process}"
    />
    <k:hidden
        name="accountCd"
        mode="${modeAccountCd}"
        surfaceKey="${surfaceAccount}"
        validationDefault="${validAccountCd_def}"
        validationApply="${validAccountCd_apply}"
        validationTempsave="${validAccountCd_tempsave}"
        validationReapply="${validAccountCd_reapply}"
        validationProcess="${validAccountCd_process}"
    />
    <k:text
        name="accountName"
        mode="1"
        class="kaiden-ui-mst-search-chain-item"
        style="width:115px;"
    />
    <k:hidden
        name="taxTypeCd"
        mode="${modeTaxTypeCd}"
        surfaceKey="${surfaceTaxType}"
        validationDefault="${validTaxType_def}"
        validationApply="${validTaxType_apply}"
        validationTempsave="${validTaxType_tempsave}"
        validationReapply="${validTaxType_reapply}"
        validationProcess="${validTaxType_process}"
    />
    <k:text
        name="taxTypeName"
        mode="1"
        class="kaiden-ui-mst-search-chain-item"
        style="width:115px;"
    />

```

```

</td>
<td>
  <k:hidden
    name="departmentSetCd"
    mode="{modeDepartmentCd}"
  />
  <k:hidden
    name="departmentCd"
    mode="{modeDepartmentCd}"
    surfaceKey="{surfaceDepartment}"
  />
  <kmsb:department
    name="departmentName"
    mode="{modeDepartmentCd}"
    imeMode="on"
    surfaceKey="{surfaceDepartment}"
    resultValue="departmentName"
    resultChain="{departmentSetCd:'departmentSetCd', departmentCd:'departmentCd'}"
    primaryTarget="departmentCd,departmentSetCd"
    searchCriteriaDate="searchCriteriaDate"
    searchCriteriaCompany="searchCriteriaCompany"
    searchCriteriaDepartmentCd="departmentCd"
    searchCriteriaDepartmentSetCd="departmentSetCd"
    style="width:150px;"
    validationDefault="{validDepartmentCd_def}"
    validationApply="{validDepartmentCd_apply}"
    validationTempsave="{validDepartmentCd_tempsave}"
    validationReapply="{validDepartmentCd_reapply}"
    validationProcess="{validDepartmentCd_process}"
  />
</td>
<td colspan="2">
  <k:hidden
    name="projectCd"
    mode="0"
    surfaceKey="{surfaceProject}"
  />
  <kmsw:project
    name="projectName"
    mode="{modeProjectCd}"
    imeMode="on"
    surfaceKey="{surfaceProject}"
    resultValue="projectName"
    resultChain="{projectCd:'projectCd'}"
    primaryTarget="projectCd"
    searchCriteriaDate="searchCriteriaDate"
    searchCriteriaCompany="searchCriteriaCompany"
    searchCriteriaProjectCd="projectCd"
    style="width:150px;"
    validationDefault="{validProjectCd_def}"
    validationApply="{validProjectCd_apply}"
    validationTempsave="{validProjectCd_tempsave}"
    validationReapply="{validProjectCd_reapply}"
    validationProcess="{validProjectCd_process}"
  />
</td>
</tr>
</tbody>
</k:tmttable>
</form>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($){
  var eventMan = KAIDEN.getGadget("{gadgetId}").eventManager
  , blockAccessor = KAIDEN.getGadget("{gadgetId}").blockAccessors["{gadgetBlockHeaderId}"]
  , tmt = KAIDEN.getGadget("{gadgetId}").tmtables["{tmttableId}"]
  , tma = tmt.tmttableAccessor;

  /*****
   * 内部イベント
   *****/

  //明細行:行挿入(データ設定前)
  tmt.addEventPostNewRowBeforeSetData(function(tupleId, tma, data) {

    //デフォルト部門を要求
    var defaultDept = KAIDEN.gadgetMan.execPull("claimDefaultDepartment", undefined, "{gadgetId}");
    defaultDept = defaultDept[0].result; //先頭行のみ利用
    tma("departmentSetCd", tupleId).setValue(defaultDept.departmentSetCd);
    tma("departmentCd", tupleId).setValue(defaultDept.departmentCd);
    tma("departmentName", tupleId).setValue(defaultDept.departmentName);

    //デフォルトプロジェクトを要求
    var defaultPrj = KAIDEN.gadgetMan.execPull("claimDefaultProject", undefined, "{gadgetId}");
    defaultPrj = defaultPrj[0].result; //先頭行のみ利用
    tma("projectCd", tupleId).setValue(defaultPrj.projectCd);

```

```

tma("projectName", tupleId).setValue(defaultPrj.projectName);

//取引通貨額、通貨を編集不可
tma("transAmount", tupleId).readonly().elem().attr("tabindex", "-1");
if (tma("transCurrencyCd", tupleId).elem(true).length > 0) {
    tma("transCurrencyCd", tupleId).disabled();
}

});

//明細行:行挿入(データ設定後)
tmt.addEventPostNewRow(function(tupleId, tma, data) {
    //マスタ検索部品の整合性確認。
    KAIDEN.MasterSearch.refresh(tma("allowanceName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("expTypeName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("departmentName", tupleId).elem().get(0));
    KAIDEN.MasterSearch.refresh(tma("projectName", tupleId).elem().get(0));

    //レートボックスのreadonly化
    if (tma("transCurrencyCd", tupleId).elem(true).length > 0) {
        var currencyCd = tma("transCurrencyCd", tupleId).getValue();
        if (currencyCd == "${companyCurrencyCd}") {
            tma("companyRate", tupleId).readonly();
        } else {
            tma("companyRate", tupleId).removeReadonly();
        }
    }

    //初期表示に先頭行を選択しているため、立替フラグ(hidden)の更新を実施 ※入力モードの場合のみ
    if (tma("payTypeCd", tupleId).elem(true).length > 0) {
        KAIDEN.base.foundation.SelectListData.setChainValue(tma("payTypeCd", tupleId).elem(true).get(0));
    }

});
//レート再取得ロジック
KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll = function() {
    tma("transCurrencyCd").elem(true).each(function() {
        tmt.variable.loadRate(this);
        tmt.variable.multiplyDays(this);
    });
}

//レート再取得:ボタン押下
blockAccessor("rateReget").elem(true).on("click", function(e){
    imuiConfirm("${msgRateReget}", "", function() {
        KAIDEN.getGadget("${gadgetId}").variable.reloadRateAll();
    }, undefined, false);
});

//日当金額再取得:ボタン押下
blockAccessor("allowanceAmountReget").elem(true).on("click", function(e){
    imuiConfirm("${msgAllowanceReget}", "", function() {

        var ajax = KAIDEN.Ajax()
        , criteria = {}
        , Accessor = KAIDEN.base.foundation.Accessor
        , ret;

        // 検索基準会社
        ret = KAIDEN.gadgetMan.execPull("claimSearchCriteriaCompany");
        criteria.searchCriteriaCompany = ret[0].result;
        // 検索基準日
        ret = KAIDEN.gadgetMan.execPull("claimSearchCriteriaDate");
        criteria.searchCriteriaDate = ret[0].result;
        // 申請権限者
        ret = KAIDEN.gadgetMan.execPull("claimApplyAuthUserCode");
        criteria.userCd = ret[0].result;

        tma("allowanceCd").elem(true).each(function() {
            var allowance = "";
            criteria.allowanceCd = ""
            criteria.allowanceCd = Accessor($(this)).getValue();

            if (criteria.allowanceCd) {
                allowance = ajax.postSync("kaiden/master/allowanceAmount", criteria);
            } else {
                return;
            }
        }

        var cur = KAIDEN.gadgetMan.getNeighboringAccessor(this, "transCurrencyCd")
        cur.setValue(allowance.currencyCd);
        tmt.variable.loadRate(cur.elem().get(0)); //レート取得

        KAIDEN.gadgetMan.getNeighboringAccessor(this, "transAmount").setValue(allowance.amount).elem().trigger("change");

    });
});

```

```

    }, undefined, false);
});

//日数*金額
tmt.variable.multiplyDays = function(target) {
    var companyAmount = KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmount").getValue();
    var quantity = KAIDEN.gadgetMan.getNeighboringAccessor(target, "quantity").getValue();

    if(($.isNumeric(companyAmount) && $.isNumeric(quantity))) {
        //金額項目に日数 * 換算額を設定する
        var total = new BigDecimal(companyAmount).multiply(new BigDecimal(quantity));
        KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue(total.toString());
    } else {
        KAIDEN.gadgetMan.getNeighboringAccessor(target, "companyAmountSum").setValue("");
    }
};

//日数入力後処理
tma("quantity").elem(true).live("change", function(e) {
    tmt.variable.multiplyDays(this);
});
//取引通貨額入力後処理(readonlyだがtrigger時にも利用できるよう付与)
tma("transAmount").elem(true).live("change", function(e) {
    tmt.variable.multiplyDays(this);
});
//レート入力後処理
tma("companyRate").elem(true).live("change", function(e) {
    tmt.variable.multiplyDays(this);
});

//マスタ検索確定時処理
tma("allowanceName").elem(true).live("choice", function(e, data) {
    //レートを再取得する
    var cur = KAIDEN.gadgetMan.getNeighboringAccessor(this, "transCurrencyCd")
    tmt.variable.loadRate(cur.elem().get(0));
    tmt.variable.multiplyDays(this);
});
//マスタ検索クリア時処理
tma("allowanceName").elem(true).live("clear", function(e) {
    //レート・換算額・金額(換算額*日数)をクリアする
    KAIDEN.gadgetMan.getNeighboringAccessor(this, "companyRate").setValue("");
    KAIDEN.gadgetMan.getNeighboringAccessor(this, "companyAmount").setValue("");
    KAIDEN.gadgetMan.getNeighboringAccessor(this, "companyAmountSum").setValue("");
});

/*****
* 受信イベント (push)
*****/

//loadData データ読み込み イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
    //明細行のクリア
    tmt.deleteAll();
    return KAIDEN.ImwModerator.genericLoadDataEvent("${gadgetId}", "${accessUrl}", arg);
});
//postLoadData データ読み込み後処理 イベントリスナの登録
eventMan.registPushListener("postLoadData", function(**Object*/arg) {
    //金額項目すべてに「マスタチェック:日当金額」を実行する
    var rv = true;
    tma("transAmount").elem(true).each(function() {
        KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
    });
});

//loadData4Copy 申請書コピー イベントリスナの登録
eventMan.registPushListener("loadData4Copy", function(**Object*/arg) {
    return KAIDEN.ImwModerator.genericLoadData4CopyEvent("${gadgetId}", "${accessUrl}", arg);
});

//postLoadData4Copy 申請書コピー後処理 イベントリスナの登録
eventMan.registPushListener("postLoadData4Copy", function(**Object*/arg) {
    var rv = true;
    tma("transAmount").elem(true).each(function() {
        KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
    });
});

//loadData4Settle 事前申請書取り込み イベントリスナの登録
eventMan.registPushListener("loadData4Settle", function(**Object*/arg) {
    return KAIDEN.ImwModerator.genericLoadData4SettleEvent("${gadgetId}", "${accessUrl}", arg);
});

//postLoadData4Settle 事前申請書取り込み後処理 イベントリスナの登録
eventMan.registPushListener("postLoadData4Settle", function(**Object*/arg) {
    var rv = true;
    tma("transAmount").elem(true).each(function() {

```



```

    KAIDEN.getGadget("${gadgetId").variable.validateTransAmount(this);
  });
});
KAIDEN.getGadget("${gadgetId").variable.validateTransAmount = function (/*HTMLElement*/target) {
  var rv = true;
  var val = KAIDEN.base.foundation.Accessor($(target)).getValue();
  var param = [ 'allowanceCd', 'searchCriteriaCompany', '${applyAuthUserCode}', 'transCurrencyCd', 'transAmount', 'searchCriteriaDate' ];
  rv = KAIDEN.validator.validateFunction("k_allowanceAmount")(val, target, param);
  if (!rv) {
    imuiShowWarningMessage("${messageSearchWarning}", "${messageSearchWarningSub}");
  }
};

//reloadAllCurrencyRate レート再取得 イベントリスナの登録
eventMan.registPushListener("reloadAllCurrencyRate", function(/**Object*/arg) {
  KAIDEN.getGadget("${gadgetId").variable.reloadRateAll();
});

/*****
 * 受信イベント (pull)
 *****/

//claimExpenseAmount 費用合計を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseAmount", function(/**Object*/arg) {
  var amounts = tmt.getColumnValues("companyAmountSum");
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
  return result.toString();
});

//claimTemporaryPayAmount 従業員立替金額を要求 イベントリスナの登録
eventMan.registPullListener("claimTemporaryPayAmount", function(/**Object*/arg) {
  var amounts = tmt.getColumnValues("companyAmountSum");
  var result = new BigDecimal().ZERO;
  $.each(amounts, function(index, value) {
    if ($.isNumeric(value)) {
      result = result.add(new BigDecimal(value));
    }
  });
  return result.toString();
});

//claimCompanyPayAmount 会社支払額を要求 イベントリスナの登録
eventMan.registPullListener("claimCompanyPayAmount", function(/**Object*/arg) {
  var result = new BigDecimal().ZERO;
  return result.toString();
});

//claimExpenseDetail 費用明細を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenseDetail", function(/**Object*/arg) {
  return tmt.getColumnValues("companyAmountSum");
});

//claimExpenCount 費用明細行数を要求 イベントリスナの登録
eventMan.registPullListener("claimExpenCount", function(/**Object*/arg) {
  return tmt.getCount();
});
});
</script>

```

SP

■ Path

src/main/webapp/WEB-INF/view/kaiden2/product_workflow/gadget/allowanceDetail/allowanceDetailV02_sp.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="imsp" uri="http://www.intra-mart.co.jp/taglib/imsp" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="kimw" uri="http://kaiden.slcs.co.jp/taglib/imw" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude_sp.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

```

```

<%-- allowanceDetail v02 --%>
<c:set var="gadgetClass">allowanceDetail</c:set>
<c:set var="gadgetVariation">v02</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${(gadgetClass)_${(gadgetVariation)_${(gadgetInstance)}}</c:set>
<c:set var="componentName">allowanceDetailV02Action</c:set>
<c:set var="imwSystemMatterId">${(param.imwSystemMatterId)}</c:set>
<c:set var="imwUserDataId">${(param.userDataId)}</c:set>
<c:set var="searchCriteriaCompany">${(param.searchCriteriaCompany)}</c:set>
<c:set var="searchCriteriaDate">${(param.searchCriteriaDate)}</c:set>
<c:set var="companyCurrencyCd">${(param.companyCurrencyCd)}</c:set>
<c:set var="tmtableId">allowanceDetailTable</c:set>

<c:set var="gadgetResponse" value="{kimw.load4View(componentName, imwSystemMatterId, imwUserDataId, gadgetId, searchCriteriaCompany))}"</c:set>
<c:set var="gadget" value="{appDto.gadgets[gadgetId]}" />

<%-- Surface --%>
<c:set var="surfaceChapterTitle">${(gadget.surfaces["chapterTitle"])}</c:set>
<c:set var="surfaceAllowance">${(gadget.surfaces["allowance"])}</c:set>
<c:set var="surfaceTransAmount">${(gadget.surfaces["amount"])}</c:set>
<c:set var="surfaceTransCurrency">${(gadget.surfaces["transCurrencyCd"])}</c:set>
<c:set var="surfaceCompanyRate">${(gadget.surfaces["companyRate"])}</c:set>
<c:set var="surfaceCompanyAmount">${(gadget.surfaces["companyAmount"])}</c:set>
<c:set var="surfaceQuantity">${(gadget.surfaces["quantity"])}</c:set>
<c:set var="surfaceCompanyAmouontSum">${(gadget.surfaces["companyAmountSum"])}</c:set>
<c:set var="surfaceExpType">${(gadget.surfaces["expType"])}</c:set>
<c:set var="surfaceAccount">${(gadget.surfaces["account"])}</c:set>
<c:set var="surfaceTaxType">${(gadget.surfaces["taxType"])}</c:set>
<c:set var="surfaceDepartment">${(gadget.surfaces["department"])}</c:set>
<c:set var="surfaceProject">${(gadget.surfaces["project"])}</c:set>
<c:set var="surfaceBack"><%= PropertyJspUtil.getSurface(BaseProp.S.BACK) %></c:set>

<div data-role="collapsible" data-collapsed="false" data-theme="b" data-content-theme="d" data-inset="false" data-collapsed-icon="arrow-d" data-expanded-icon="arrow-u" data-iconp
<h3>
  <span>${(surfaceChapterTitle)}</span>
  <span class="ui-btn-up-c ui-btn-corner-all" style="font-size:0.8em;padding:2px 5.5px"><c:out value="{fn:length(gadgetResponse.data[tmtableId])}"></c:out></span>
</h3>

<ul data-role="listview" data-divider-theme="b">
  <c:forEach items="{(gadgetResponse.data[tmtableId])}" var="detailData">
    <li>
      <span class="ui-li-heading"><c:out value="{(detailData.allowanceName)}"></c:out></span>
      <span class="ui-li-aside"><c:out value="{(detailData.companyAmountSum[1])}"></c:out></span>
      <div class="indent">
        <p class="kaiden-ui-overview-contents">
          <c:out value="{(surfaceTransAmount)}"></c:out>
          <c:out value="{(detailData.companyAmount[1])}"></c:out>
        </p>
        <p class="kaiden-ui-overview-contents">
          <c:out value="{(surfaceQuantity)}"></c:out>
          <c:out value="{(detailData.quantity[1])}"></c:out>
        </p>
      </div>
      <ul>
        <li data-icon="back" data-mini="true" data-theme="a"><a data-rel="back"><c:out value="{(surfaceBack)}"></c:out></a></li>
        <li>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceAllowance)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{(detailData.allowanceName)}"></c:out></label>
          </div>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceTransAmount)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents-sub"><c:out value="{(detailData.transAmount[1])}"></c:out></label>
            <label class="kaiden-ui-detail-item-contents-sub" style="margin-left:1em"><c:out value="{(detailData.transCurrencyName)}"></c:out></label>
          </div>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents-sub kaiden-ui-cmpRate"><c:out value="{(detailData.companyRate[1])}"></c:out></label>
          </div>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{(detailData.companyAmount[1])}"></c:out></label>
          </div>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceQuantity)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{(detailData.quantity[1])}"></c:out></label>
          </div>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceCompanyAmouontSum)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><strong><c:out value="{(detailData.companyAmountSum[1])}"></c:out></strong></label>
          </div>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceExpType)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{(detailData.expTypeName)}"></c:out></label>
          </div>
          <p class="kaiden-ui-detail-item-title"><c:out value="{(surfaceAccount)}"></c:out></p>
          <div class="indent">
            <label class="kaiden-ui-detail-item-contents"><c:out value="{(detailData.accountName)}"></c:out></label>
          </div>

```

```

</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceTaxType}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.taxTypeName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceDepartment}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.departmentName}"></c:out></label>
</div>
<p class="kaiden-ui-detail-item-title"><c:out value="\${surfaceProject}"></c:out></p>
<div class="indent">
  <label class="kaiden-ui-detail-item-contents"><c:out value="\${detailData.projectName}"></c:out></label>
</div>
</li>
</ul>
</li>
</c:forEach>
</ul>
</div>

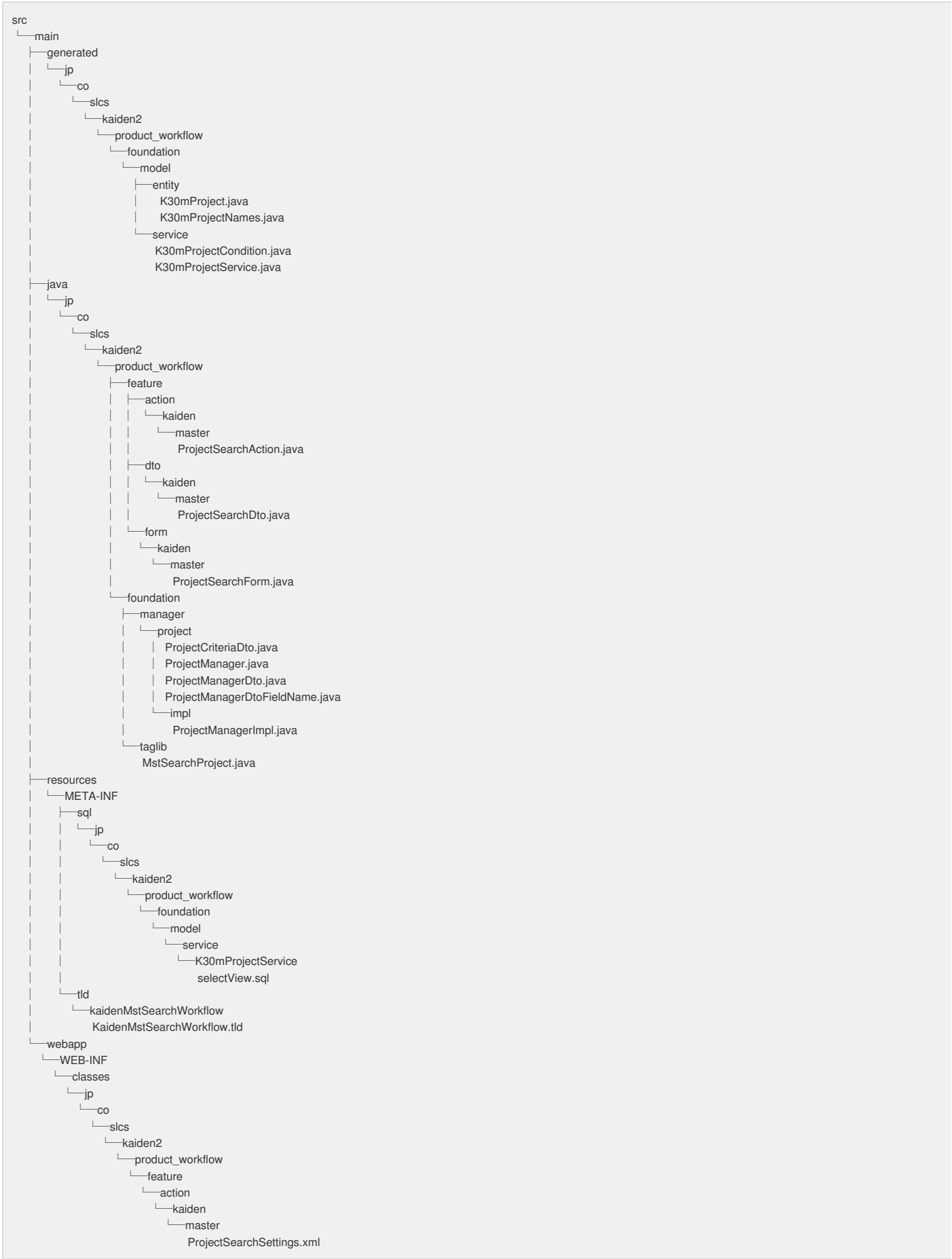
```


プロジェクトマスタ検索

プロジェクトマスタ検索のサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action](#)
 - [Dto](#)
 - [Form](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
 - [Taglib](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
- [Taglib](#)
- [XML](#)

[一覧](#)



Java

Action

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ProjectSearchAction.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master;

import net.arnx.jsonic.JSON;
import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.ContextHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.DateHelper;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchAction;
import jp.co.slcs.kaiden2.base.foundation.master.GenericSearchSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.HtmlEncodeUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master.ProjectSearchDto;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master.ProjectSearchForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;

/**
 * プロジェクトマスタ検索(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectSearchAction extends AbstractMasterSearchAction {

    /** ProjectSearchDto. */
    @Resource
    public ProjectSearchDto projectSearchDto;

    /** ProjectSearchForm. */
    @ActionForm
    @Resource
    public ProjectSearchForm projectSearchForm;

    /** ProjectManager. */
    @Resource
    public ProjectManager projectManager;

    /** ContextHelper. */
    @Resource
    public ContextHelper contextHelper;

    /** DateHelper. */
    @Resource
    public DateHelper dateHelper;

    /**
     * 検索.
     * <p>
     * 以下の設定ファイルに従って検索を行う.<br />
     * WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ProjectSearchSettings.xml
     * </p>
     * <p>
     * 検索条件は以下の通り
     * </p>
     * <table border="1">
     * <tr>
     * <th>フィールド</th>
     * <th>条件</th>
     * <th>値の取得元</th>
     * <th>省略</th>
     * </tr>
     * <tr>
     * <td>会社コード</td>
     * <td>完全一致</td>
     * <td>画面リクエスト</td>
     * <td>不可</td>
     * </tr>
     * <tr>
     * <td>プロジェクトコード</td>
     * <td>完全一致</td>
     * <td>画面リクエスト</td>
     * <td>可</td>
     * </tr>
     * <tr>
     * <td>ロケールID</td>
     * <td>完全一致</td>
     * <td>ログインユーザのロケールID</td>
     * <td>-</td>
     * </tr>
     */
}

```

```

* <tr>
* <td>検索基準日</td>
* <td>開始日-終了日の範囲</td>
* <td>画面リクエスト</td>
* <td>可(省略時はシステム日付を利用)</td>
* </tr>
* </table>
*
* @return 汎用検索JSP
* @throws KaidenSystemException XML読み込み例外
*/
@SuppressWarnings("unchecked")
@Execute(validator = false)
public String index() throws KaidenSystemException {
    searchDto = projectSearchDto;
    searchDto.searchKeywords = projectSearchForm.searchKeyword;
    // 設定ファイル読み込み
    searchDto.settings = new GenericSearchSettingsDto(readSetting(ProjectSearchAction.class));

    // キーワード条件
    KeywordsCriteriaDto keywordsCriteria = createKeywordCriteria(projectSearchForm.searchKeyword,
        projectSearchDto.settings);

    List<Map<String, Object>> searchResult = new ArrayList<Map<String, Object>>();
    // 検索条件の作成
    SearchCriteria criteria = new SearchCriteria();
    if (StringUtil.isNotEmpty(projectSearchForm.searchCriteriaCompany)) {
        criteria.companyCd = projectSearchForm.searchCriteriaCompany;

        if (StringUtil.isNotEmpty(projectSearchForm.searchCriteriaDate)) {
            criteria.searchCriteriaDate = dateHelper.toTimestamp(projectSearchForm.searchCriteriaDate);
        } else {
            criteria.searchCriteriaDate = dateHelper.getNowTimestamp();
        }
    }
    if (StringUtil.isNotEmpty(projectSearchForm.projectCd)) {
        criteria.projectCd = projectSearchForm.projectCd;
    }
    if ("true".equals(projectSearchForm.includeDisabled)) {
        criteria.deleteFlag = null;
    } else {
        criteria.deleteFlag = GenericFlag.OFF.getCode();
    }
    criteria.localeId = contextHelper.getLoginUserLocale().toString();

    Map<String, Object> ret = projectManager.selectListKeywordFilterIgnoreAuthCompany(criteria,
        getOrder(searchDto.settings.orderJson), keywordsCriteria);

    ConvertForJsonSource conv = new ConvertForJsonSource();
    for (Object dto : (List<Object>) ret.get("resultList")) {
        Map<String, Object> row = conv.dtoConvert((KaidenDto) dto);
        row.put("rowDataJson", HtmlEncodeUtil.escapeHtmlValue(JSON.encode(row)));
        searchResult.add(row);
    }
}

projectSearchDto.searchResult = searchResult;
return GENERIC_SEARCH_JSP;
}

/**
 * 検索条件.
 */
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public static class SearchCriteria extends KaidenDto {

    /** 会社コード. */
    public String companyCd;
    /** プロジェクトコード. */
    public String projectCd;
    /** 削除フラグ. */
    public String deleteFlag;
    /** ロケールID. */
    public String localeId;
    /** 検索基準日. */
    public Timestamp searchCriteriaDate;
}
}

```

Dto

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/dto/kaiden/master/ProjectSearchDto.java


```

package jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchDto;

/**
 * プロジェクトマスタ検索 (DTO).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectSearchDto extends AbstractMasterSearchDto {

}

```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/master/ProjectSearchForm.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchForm;

/**
 * プロジェクトマスタ検索 (From).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectSearchForm extends AbstractMasterSearchForm {

    /** プロジェクトコード. */
    public String projectCd;

}

```

Manager

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.ValidateMapHelper;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenAbstractValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidatorIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import org.seasar.framework.container.SingletonS2Container;

/**
 * プロジェクトマネージャのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ProjectManager extends TermMasterManager<ProjectManagerDto>,
    TermMasterManagerForJob<ProjectManagerDto> {

    /**
     * プロジェクトコードの最大長.
     */
    public static final int PROJECT_CD_LEN = 100;

    /**
     * プロジェクト名の最大長.
     */
    public static final int PROJECT_NAME_LEN = 250;

    /**
     * 新規登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();
}

```

```

/**
 * 更新登録用の検証定義リストを返却します。
 *
 * @return バリデート実行クラス用パラメタDTOリスト
 */
public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

/**
 * カスタムバリデート_自プロジェクト・上位プロジェクト同一チェック。
 * <p>
 * 自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。
 * </p>
 */
class CustomValidateNotEqualParent extends KaidenAbstractValidator implements KaidenValidatorIF {

    /** コンストラクタ。 */
    public CustomValidateNotEqualParent() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * チェックエラーとなる条件<br>
     * <ul>
     * <li>自プロジェクト・上位プロジェクト同一チェック。</li>
     * <li>自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。</li>
     * </ul>
     * 以下の場合、チェックを行いません。
     * <ul>
     * <li>自プロジェクトコード・自会社コードのいずれかが設定されていない場合</li>
     * <li>上位プロジェクトコード・上位会社コードの両方が未設定の場合</li>
     * </ul>
     * 以下の場合、パラメーターエラーとします。
     * <ul>
     * <li>上位プロジェクトコード・上位会社コードの片方が未設定の場合</li>
     * </ul>
     * </p>
     */
    @Override
    public KaidenValidateResult executeValidate(Map<String, Object> valueMap, String fieldKey, String[] args,
        Map<String, String> surfaces) {

        KaidenValidateResult result = super.makeNewValidateResult(fieldKey, surfaces);

        ValidateMapHelper validateMapHelper = SingletonS2Container.getComponent(ValidateMapHelper.class);

        String thisCompanyCd = validateMapHelper.getValueForString(valueMap, COMPANY_CD);
        String thisProjectCd = validateMapHelper.getValueForString(valueMap, PROJECT_CD);

        // 自プロジェクトコードまたは自会社コードが設定されていない場合は、エラーとしない
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            return result;
        }

        String parentCompanyCd = validateMapHelper.getValueForString(valueMap, PARENT_COMPANY_CD);
        String parentProjectCd = validateMapHelper.getValueForString(valueMap, PARENT_PROJECT_CD);

        // 上位プロジェクトコードと上位会社コードの両方が未設定の場合、エラーとしない
        if (StringUtil.isEmpty(parentCompanyCd, parentProjectCd)) {
            return result;
        }

        // 上位プロジェクトコード・上位会社コードの片方が未設定の場合、パラメーターエラー
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            throw KaidenIllegalParametersException.makeInstance();
        }

        // 自プロジェクトコードと上位プロジェクトコードが同一かつ、自会社コードと上位会社コードが同一の場合、チェックエラー
        if (thisCompanyCd.equals(parentCompanyCd) && thisProjectCd.equals(parentProjectCd)) {
            PropertyHelper propertyHelper = SingletonS2Container.getComponent(PropertyHelper.class);
            String fieldName = validateMapHelper.getSurfaces(valueMap, surfaces, PARENT_PROJECT_CD);
            result.setErrorStatus(true);
            result.setValidateClass(this.getClass());
            result.setFieldKey(PARENT_PROJECT_CD);
            result.setFieldNm(fieldName);
            result.setMainMessageKey(ProductWorkflowProp.M.Project.W001);
            result.setMainMessage(propertyHelper.getMessage(ProductWorkflowProp.M.Project.W001, fieldName));
        }

        return result;
    }
}

```

Manager (実装)

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/impl/ProjectManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mProjectService;
import org.seasar.framework.beans.util.Beans;

/**
 * プロジェクトマスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います、より業務的な機能を付加した操作を提供します.<br>
 * 使用するには、{@link ProjectManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ProjectManagerDto}クラスまたはそのリストが戻り値として返却されます.<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>

```

```

* <td>備考</td>
* </tr>
* <tr>
* <td>主キー検索</td>
* <td>select</td>
* <td>主キー検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>主キー検索 (認可設定無視)</td>
* <td>selectIgnoreAuthCompany</td>
* <td>主キー検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索</td>
* <td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索 (認可設定無視)</td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索 (認可設定無視)</td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索 (認可設定無視)</td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索 (認可設定無視)</td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化 (認可設定無視)</td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化 (認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>

```

```

* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録(認可設定無視)</td>
* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除(認可設定無視)</td>
* <td>removeIgnoreAuthCompany</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新</td>
* <td>renew</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新(認可設定無視)</td>
* <td>renewIgnoreAuthCompany</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能一覧(期間化)</b>
* <table border="1">
* <tr>
* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>期間検索</td>
* <td>selectTerms</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間検索(認可設定無視)</td>
* <td>selectTermsIgnoreAuthCompany</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索</td>
* <td>selectListIncluedsSearchDate</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索(認可設定無視)</td>
* <td>selectListIncluedsSearchDateIgnoreAuthCompany</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割</td>
* <td>divideTerm</td>
* <td>期間分割</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>期間分割(認可設定無視)</td>
* <td>divideTermIgnoreAuthCompany</td>
* <td>期間分割</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>期間変更</td>
* <td>moveTerm</td>

```

```
<td>期間変更</td>  
*<td>&nbsp;</td>  
*</tr>  
*<br>  
*<td>期間変更（認可設定無視）</td>  
*<td>moveTermIgnoreAuthCompany</td>  
*<td>期間変更</td>  
*<td>&nbsp;</td>  
*</tr>  
*</table>  
*<br>  
*<br>  
*機能に対するI/Oを以下に示します。（上記一覧のカテゴリ単位）<br>  
*<br>  
*[IN]  
*<table border="1">  
*<tr>  
*<td align="right">●</td>  
*<td>：必須</td>  
*</tr>  
*<tr>  
*<td align="right">△</td>  
*<td>：任意、検索時完全一致条件</td>  
*</tr>  
*<tr>  
*<td align="right">▲</td>  
*<td>：任意、検索時部分一致条件</td>  
*</tr>  
*<tr>  
*<td align="right">x</td>  
*<td>：不要（設定されていても無視します。）</td>  
*</tr>  
*<tr>  
*<td align="right">☆</td>  
*<td>：各APIの説明を参照</td>  
*</tr>  
*<tr>  
*<td align="right">-</td>  
*<td>：対象外</td>  
*</tr>  
*</table>  
*<br>  
*[OUT]  
*<table border="1">  
*<tr>  
*<td align="right">○</td>  
*<td>：あり</td>  
*</tr>  
*<tr>  
*<td align="right">x</td>  
*<td>：なし</td>  
*</tr>  
*<tr>  
*<td align="right">-</td>  
*<td>：対象外</td>  
*</tr>  
*</table>  
*<br>  
*<b>DTOクラスとメソッドのI/Oマトリクス(共通)</b>  
*<table border="1">  
*<tr>  
*<td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>  
*<td rowspan="2" align="center"><b>DTOフィールド名</b></td>  
*<td colspan="2" align="center"><b>主キー</b><br>  
検索</b></td>  
*<td colspan="2" align="center"><b>汎用</b><br>  
検索</b></td>  
*<td colspan="2" align="center"><b>ローケル</b><br>  
検索</b></td>  
*<td colspan="2" align="center"><b>参照</b><br>  
検索</b></td>  
*<td colspan="2" align="center"><b>キーワード</b><br>  
検索</b></td>  
*<td colspan="2" align="center"><b>削除フラグ</b><br>  
更新</b></td>  
*<td colspan="2" align="center"><b>登録</b></td>  
*<td colspan="2" align="center"><b>削除</b></td>  
*<td colspan="2" align="center"><b>更新</b></td>  
*</tr>  
*<tr>  
*<td align="center"><b>|</b></td>  
*<td align="center"><b>O</b></td>  
*<td align="center"><b>|</b></td>  
*<td align="center"><b>O</b></td>  
*<td align="center"><b>|</b></td>  
*<td align="center"><b>|</b></td>
```

[illegible]

[illegible]

<td align="center"></td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">△</td>
*<td align="center"></td>
*<td align="center">○</td>
*<td align="center">×</td>
*<td align="center">△</td>
*<td align="center"></td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">×</td>
*<td align="center"></td>
</tr>
<tr>
<td>会社コード(上位)</td>
<td-parentCompanyCd</td>
<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">△</td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">△</td>
*<td align="center"></td>
</tr>
<tr>
<td>プロジェクトコード(上位)</td>
<td-parentProjectCd</td>
<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">△</td>
*<td align="center"></td>
*<td align="center">×</td>
*<td align="center">△</td>
*<td align="center"></td>
</tr>
<tr>
*<td>予備項目A〜J</td>
<td-extensionA〜J</td>
<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">▲</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">△</td>
*<td align="center">x</td>
*<td align="center"></td>
*<td align="center">×</td>
*<td align="center">△</td>
*<td align="center"></td>
</tr>
<tr>
<td>登録日時</td>
<td-entryTs</td>
*<td align="center">x</td>

[illegible]

[illegible]

```

* </table>
* <br>
* <b>DTOクラスとメソッドのI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>
* <td rowspan="2" align="center"><b>DTOフィールド名</b></td>
* <td colspan="2" align="center"><b>期間検索</b></td>
* <td colspan="2" align="center"><b>基準日指定検索</b></td>
* <td colspan="2" align="center"><b>期間分割</b></td>
* <td colspan="2" align="center"><b>期間変更</b></td>
* </tr>
* <tr>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>プロジェクトコード(*)</b></td>
* <td><b>projectCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>startDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>endDate</td>
* <td align="center">x</td>

```

```

* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクト名称</td>
* <td>projectName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>会社コード(上位)</td>
* <td>parentCompanyCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクトコード(上位)</td>
* <td>parentProjectCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>

```

```

* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>
* <td>renewCnt</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新日時</td>
* <td>renewTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新者コード</td>
* <td>renewUserCode</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>組織名称</td>
* <td>companyName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクト名称(上位)</td>
* <td>parentProjectName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <br>
* テーブルのPK項目にはフィールド名に(*)を付与しています。<br>
* </p>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

```

```

public class ProjectManagerImpl extends AbstractTermMasterManager<ProjectManagerDto> implements ProjectManager {

    /** K30mProjectService. */
    @Resource
    protected K30mProjectService k30mProjectService;

    /**
     * {@inheritDoc}
     */
    @Override
    public ProjectManagerDto select(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
        return super.select(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public ProjectManagerDto selectIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.selectIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectList(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria,
        List<OrderDto> order) {
        return super.selectList(k30mProjectService, criteria, authCriteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListIgnoreAuthCompany(ProjectManagerDto criteria, List<OrderDto> order) {
        return super.selectListIgnoreAuthCompany(k30mProjectService, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
        AuthorityCriteriaDto authCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectBySql(k30mProjectService, "selectView.sql", criteria, order, authCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectBySqlIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectListByKeywords(k30mProjectService, "selectView.sql", criteria, order, keywordsCriteria,
            authCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectListByKeywordsIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order,
            keywordsCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */

```

```

    * @throws KaidenOptimisticLockException
    */
    @Override
    public void toEnabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        super.toEnabled(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     *
     * @throws KaidenOptimisticLockException
     */
    @Override
    public void toEnabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
        super.toEnabledIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toDisabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        super.toDisabled(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toDisabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
        super.toDisabledIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectLocales(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
        return super.selectLocales(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectLocalesIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.selectLocalesIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectTerms(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
        return super.selectTerms(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectTermsIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.selectTermsIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
        return super.entry(k30mProjectService, dataList, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entryIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenDuplicationEntryException {
        return super.entryIgnoreAuthCompany(k30mProjectService, dataList);
    }

```



```

}

/**
 * {@inheritDoc}
 */
@Override
public int remove(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException {
    return super.remove(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.removeIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int renew(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    return super.renew(k30mProjectService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int renewIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenOptimisticLockException {
    return super.renewIgnoreAuthCompany(k30mProjectService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
    super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
    String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
    KaidenValidateException {
    String termCd2nd = super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            toDisabled(criteria2nd, authCriteria);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            toEnabled(criteria2nd, authCriteria);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate, String deleteFlag2nd)
    throws KaidenOptimisticLockException, KaidenValidateException {
    String termCd2nd = super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {

```

```

        if (GenericFlag.isOff(deleteFlag2nd)) {
            toDisabledIgnoreAuthCompany(criteria2nd);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            toEnabledIgnoreAuthCompany(criteria2nd);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ProjectManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mProjectService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ProjectManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mProjectService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, PROJECT_CD});
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDate(ProjectManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mProjectService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ProjectManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mProjectService, criteria, order);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }

    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
 * </p>
 */

```

```

@Override
public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード</td>
 * <td>必須、皆伝コード、100桁</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>プロジェクト名</td>
 * <td>必須、250文字</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>会社コード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
 * </tr>
 * <tr>
 * <td>削除フラグ</td>
 * <td>フラグ妥当性</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
 * </tr>
 * </table>
 */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // プロジェクトコード
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
    // ロケールid
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日
    result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
    // プロジェクト名称
    result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
    // プロジェクトコード(上位)

```

```

result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
    "", "true"));
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));
// 削除フラグ
result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

return result;
}

```

```

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード</td>
 * <td>必須、皆伝コード、100桁</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、皆伝コード、50桁</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>プロジェクト名</td>
 * <td>必須、250文字</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>会社コード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
 * </tr>
 * </table>
 */

```

@Override

```

public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // プロジェクトコード
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
    // 期間コード
    result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
    result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
    // ロケールid
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日
    result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
    // プロジェクト名称

```

```

result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
// プロジェクトコード(上位)
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
    "", "true"));
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));

return result;
}

/** {@inheritDoc} */
@Override
public Class<ProjectManagerDto> getMainManagerDtoClass() {
    return ProjectManagerDto.class;
}

/** {@inheritDoc} */
@Override
public List<ProjectManagerDto> selectListIgnoreAuthCompanyWithLock(ProjectManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mProjectService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ProjectManagerDto.class,
        k30mProjectService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiledts() {
    return k30mProjectService.getPkFiledts();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ProjectManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicF<ProjectManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mProjectService, criteria, order, logic);
}
}

```

ManagerDto

- Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectCriteriaDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterMaintenanceCriteriaDto;

/**
 * プロジェクトマスタ 検索条件DTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectCriteriaDto extends TermMasterMaintenanceCriteriaDto {

    /** 会社コード. */
    // public String companyCd; //親クラスで定義済み

    /** プロジェクトコード. */
    public String projectCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールID. */
    // public String localeId; //親クラスで定義済み
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** プロジェクト名称. */
    public String projectName;
    /** 削除フラグ. */
    // public String deleteFlag; //親クラスで定義済み
    /** プロジェクトコード(上位). */
    public String parentProjectCd;
    /** 予備項目A. */
    public String extensionA;
    /** 予備項目B. */
    public String extensionB;
    /** 予備項目C. */
    public String extensionC;
    /** 予備項目D. */
    public String extensionD;
    /** 予備項目E. */
    public String extensionE;
    /** 予備項目F. */
    public String extensionF;
    /** 予備項目G. */
    public String extensionG;
    /** 予備項目H. */
    public String extensionH;
    /** 予備項目I. */
    public String extensionI;
    /** 予備項目J. */
    public String extensionJ;
    /** 会社名. */
    public String companyName;
    /** プロジェクト名称(上位). */
    public String parentProjectName;
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

/**
 * プロジェクトDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;

    /** プロジェクトコード. */
    public String projectCd;

    /** 期間コード. */
    public String termCd;

    /** ロケールID. */
    public String localeId;

    /** 開始日. */

```

```

public Timestamp startDate;

/** 終了日. */
public Timestamp endDate;

/** プロジェクト名称. */
public String projectName;

/** 削除フラグ. */
public String deleteFlag;

/** 会社コード(上位). */
public String parentCompanyCd;

/** プロジェクトコード(上位). */
public String parentProjectCd;

/** 予備項目A. */
public String extensionA;

/** 予備項目B. */
public String extensionB;

/** 予備項目C. */
public String extensionC;

/** 予備項目D. */
public String extensionD;

/** 予備項目E. */
public String extensionE;

/** 予備項目F. */
public String extensionF;

/** 予備項目G. */
public String extensionG;

/** 予備項目H. */
public String extensionH;

/** 予備項目I. */
public String extensionI;

/** 予備項目J. */
public String extensionJ;

/** 登録日時. */
public Timestamp entryTs;

/** 登録者コード. */
public String entryUserCd;

/** 更新カウンタ. */
public Long renewCnt;

/** 更新日時. */
public Timestamp renewTs;

/** 更新者コード. */
public String renewUserCd;

/** 組織名称. */
public String companyName;

/** プロジェクト名称(上位). */
public String parentProjectName;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityExtensionFieldNames;

/**
 * プロジェクトマスタDTO フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ProjectManagerDtoFieldName implements EntityExtensionFieldNames {

    /**
     * コンストラクタ.
     */
    private ProjectManagerDtoFieldName() {

    }

    /** 会社コード. */
    public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
    /** プロジェクトコード. */
    public static final String PROJECT_CD = "projectCd";
    /** 期間コード. */
    public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
    /** ロケールid. */
    public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
    /** プロジェクト名称. */
    public static final String PROJECT_NAME = "projectName";
    /** 開始日. */
    public static final String START_DATE = EntityCommonFieldNames.START_DATE;
    /** 終了日. */
    public static final String END_DATE = EntityCommonFieldNames.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;
    /** 会社コード(上位). */
    public static final String PARENT_COMPANY_CD = "parentCompanyCd";
    /** プロジェクトコード(上位). */
    public static final String PARENT_PROJECT_CD = "parentProjectCd";

    /** 会社名称(他マスタ名称). */
    public static final String COMPANY_NAME = "companyName";
    /** プロジェクト名称(上位). */
    public static final String PARENT_PROJECT_NAME = "parentProjectName";
}

```

Taglib

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/taglib/MstSearchProject.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.taglib;

import jp.co.slcs.kaiden2.base.foundation.taglib.AbstractMstSearch;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName;

/**
 * プロジェクトマスタ検索.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class MstSearchProject extends AbstractMstSearch {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /**
     * {@inheritDoc}
     */
    @Override
    protected String getSearchUrl() {
        return "projectSearch";
    }

    /**
     * 検索条件:プロジェクトコードを設定.
     *
     * @param searchCriteriaProjectCd 検索条件:プロジェクトコード
     */
    public void setSearchCriteriaProjectCd(String searchCriteriaProjectCd) {
        super.addSearchCriteria(ProjectManagerDtoFieldName.PROJECT_CD, searchCriteriaProjectCd);
    }
}

```


Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProject.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * プロジェクト.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30m_project")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mProject extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** プロジェクトコード. */
    @Id
    @Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String projectCd;

    /** 期間コード. */
    @Id
    @Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String termCd;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp endDate;

    /** プロジェクト名. */
    @Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
    public String projectName;

    /** 削除フラグ. */
    @Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
    public String deleteFlag;

    /** 会社コード(上位プロジェクト). */
    @Column(name = "parent_company_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String parentCompanyCd;

    /** プロジェクトコード(上位プロジェクト). */
    @Column(name = "parent_project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String parentProjectCd;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

    /** 予備項目b. */
    @Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionB;

    /** 予備項目c. */
    @Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionC;

```

```

public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProjectNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30mProject} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mProjectNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30mProjectNames() {
        super();
    }

    /**
     * companyCd のプロパティ名を返却.
     *
     * @return companyCd のプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * projectCd のプロパティ名を返却.
     *
     * @return projectCd のプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * termCd のプロパティ名を返却.
     *
     * @return termCd のプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }
}

```

```

    * localeIdのプロパティ名を返却.
    *
    * @return localeIdのプロパティ名
    */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * startDateのプロパティ名を返却.
     *
     * @return startDateのプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDateのプロパティ名を返却.
     *
     * @return endDateのプロパティ名
     */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }

    /**
     * projectNameのプロパティ名を返却.
     *
     * @return projectNameのプロパティ名
     */
    public static PropertyName<String> projectName() {
        return new PropertyName<String>("projectName");
    }

    /**
     * deleteFlagのプロパティ名を返却.
     *
     * @return deleteFlagのプロパティ名
     */
    public static PropertyName<String> deleteFlag() {
        return new PropertyName<String>("deleteFlag");
    }

    /**
     * parentCompanyCdのプロパティ名を返却.
     *
     * @return parentCompanyCdのプロパティ名
     */
    public static PropertyName<String> parentCompanyCd() {
        return new PropertyName<String>("parentCompanyCd");
    }

    /**
     * parentProjectCdのプロパティ名を返却.
     *
     * @return parentProjectCdのプロパティ名
     */
    public static PropertyName<String> parentProjectCd() {
        return new PropertyName<String>("parentProjectCd");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {

```

```

    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *

```

```

    * @return renewCntのプロパティ名
    */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30mProjectNames extends PropertyName<K30mProject> {

        /**
         * インスタンスを構築.
         */
        public _K30mProjectNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前
         */
        public _K30mProjectNames(final String name) {
            super(name);
        }

        /**
         * インスタンスを構築.
         *
         * @param parent 親
         * @param name 名前
         */
        public _K30mProjectNames(final PropertyName<?> parent, final String name) {
            super(parent, name);
        }

        /**
         * companyCdのプロパティ名を返却.
         *
         * @return companyCdのプロパティ名
         */
        public PropertyName<String> companyCd() {
            return new PropertyName<String>("companyCd");
        }

        /**
         * projectCdのプロパティ名を返却.
         *
         * @return projectCdのプロパティ名
         */
        public PropertyName<String> projectCd() {
            return new PropertyName<String>("projectCd");
        }

        /**
         * termCdのプロパティ名を返却.
         *
         * @return termCdのプロパティ名
         */
        public PropertyName<String> termCd() {
            return new PropertyName<String>("termCd");
        }

        /**
         * localeIdのプロパティ名を返却.
         *
         * @return localeIdのプロパティ名
         */
    }

```

```

/
public PropertyName<String> localeId() {
    return new PropertyName<String>((this, "localeId"));
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>((this, "startDate"));
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>((this, "endDate"));
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>((this, "projectName"));
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>((this, "deleteFlag"));
}

/**
 * parentCompanyCdのプロパティ名を返却.
 *
 * @return parentCompanyCdのプロパティ名
 */
public PropertyName<String> parentCompanyCd() {
    return new PropertyName<String>((this, "parentCompanyCd"));
}

/**
 * parentProjectCdのプロパティ名を返却.
 *
 * @return parentProjectCdのプロパティ名
 */
public PropertyName<String> parentProjectCd() {
    return new PropertyName<String>((this, "parentProjectCd"));
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>((this, "extensionA"));
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>((this, "extensionB"));
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>((this, "extensionC"));
}

/**

```

```

    * extensionDのプロパティ名を返却。
    *
    * @return extensionDのプロパティ名
    */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却。
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却。
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却。
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却。
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {

```

```

        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mProject;

/**
 * {@link K30mProject}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mProjectService extends GenerateService<K30mProject> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30mProject}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mProjectCondition extends
    AbstractEntityCondition<K30mProjectCondition> {

    /**
     * インスタンスを構築.
     */
    public K30mProjectCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30mProjectCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }
}

```



```

/** companyCdの条件. */
public NotNullableStringCondition<K30mProjectCondition> companyCd =
    new NotNullableStringCondition<K30mProjectCondition>("companyCd", this);

/** projectCdの条件. */
public NotNullableStringCondition<K30mProjectCondition> projectCd =
    new NotNullableStringCondition<K30mProjectCondition>("projectCd", this);

/** termCdの条件. */
public NotNullableStringCondition<K30mProjectCondition> termCd =
    new NotNullableStringCondition<K30mProjectCondition>("termCd", this);

/** localeIdの条件. */
public NotNullableStringCondition<K30mProjectCondition> localeId =
    new NotNullableStringCondition<K30mProjectCondition>("localeId", this);

/** startDateの条件. */
public NotNullableCondition<K30mProjectCondition, Timestamp> startDate =
    new NotNullableCondition<K30mProjectCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NotNullableCondition<K30mProjectCondition, Timestamp> endDate =
    new NotNullableCondition<K30mProjectCondition, Timestamp>("endDate", this);

/** projectNameの条件. */
public NotNullableStringCondition<K30mProjectCondition> projectName =
    new NotNullableStringCondition<K30mProjectCondition>("projectName", this);

/** deleteFlagの条件. */
public NotNullableStringCondition<K30mProjectCondition> deleteFlag =
    new NotNullableStringCondition<K30mProjectCondition>("deleteFlag", this);

/** parentCompanyCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentCompanyCd =
    new NullableStringCondition<K30mProjectCondition>("parentCompanyCd", this);

/** parentProjectCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentProjectCd =
    new NullableStringCondition<K30mProjectCondition>("parentProjectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30mProjectCondition> extensionA =
    new NullableStringCondition<K30mProjectCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30mProjectCondition> extensionB =
    new NullableStringCondition<K30mProjectCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30mProjectCondition> extensionC =
    new NullableStringCondition<K30mProjectCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30mProjectCondition> extensionD =
    new NullableStringCondition<K30mProjectCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30mProjectCondition> extensionE =
    new NullableStringCondition<K30mProjectCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mProjectCondition> extensionF =
    new NullableStringCondition<K30mProjectCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30mProjectCondition> extensionG =
    new NullableStringCondition<K30mProjectCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30mProjectCondition> extensionH =
    new NullableStringCondition<K30mProjectCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30mProjectCondition> extensionI =
    new NullableStringCondition<K30mProjectCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30mProjectCondition> extensionJ =
    new NullableStringCondition<K30mProjectCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullableCondition<K30mProjectCondition, Timestamp> entryTs =
    new NotNullableCondition<K30mProjectCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<K30mProjectCondition> entryUserCd =

```

```

public NotNullableStringCondition<K30mProjectCondition> entryUserCd =
    new NotNullableStringCondition<K30mProjectCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullableCondition<K30mProjectCondition, Long> renewCnt =
    new NotNullableCondition<K30mProjectCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullableCondition<K30mProjectCondition, Timestamp> renewTs =
    new NotNullableCondition<K30mProjectCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<K30mProjectCondition> renewUserCd =
    new NotNullableStringCondition<K30mProjectCondition>("renewUserCd", this);
}

```

SQL

検索

- **Path**
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService/selectView.sql

```

select
    prj.company_cd
    , prj.project_cd
    , prj.term_cd
    , prj.locale_id
    , prj.start_date
    , prj.end_date
    , prj.project_name
    , prj.delete_flag
    , prj.parent_company_cd
    , prj.parent_project_cd
    , prj.extension_a
    , prj.extension_b
    , prj.extension_c
    , prj.extension_d
    , prj.extension_e
    , prj.extension_f
    , prj.extension_g
    , prj.extension_h
    , prj.extension_i
    , prj.extension_j
    , prj.entry_ts
    , prj.entry_user_cd
    , prj.renew_cnt
    , prj.renew_ts
    , prj.renew_user_cd
    , dept.department_name as company_name
    , parent.project_name as parent_project_name
from k30m_project prj
    left join ( select imm_department.company_cd
        , imm_department.department_set_cd
        , imm_department.department_cd
        , imm_department.locale_id
        , imm_department.start_date
        , imm_department.end_date
        , imm_department.department_name
        from imm_department) dept
on prj.company_cd = dept.company_cd
and prj.company_cd = dept.department_set_cd
and prj.company_cd = dept.department_cd
and prj.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and dept.start_date <= prj.start_date
--ELSE and dept.end_date > prj.start_date
/*END*/
    left join k30m_project parent
on prj.parent_company_cd = parent.company_cd
and prj.parent_project_cd = parent.project_cd
and prj.locale_id = parent.locale_id
/*IF searchCriteriaDate != null*/
and parent.start_date <= /*searchCriteriaDate*/2000/01/01'
and parent.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and parent.start_date <= prj.start_date
--ELSE and parent.end_date > prj.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
    prj.company_cd in /*companyCd*/('comp')
/*END*/

```

```

/*IF projectCd != null*/
and prj.project_cd = /*projectCd*/projectCd'
/*END*/
/*IF termCd != null*/
and prj.term_cd = /*termCd*/termCd'
/*END*/
/*IF localeId != null*/
and prj.locale_id = /*localeId*/localeId'
/*END*/
/*IF searchCriteriaDate != null*/
and prj.start_date <= /*searchCriteriaDate*/1900/01/01'
and prj.end_date > /*searchCriteriaDate*/2999/12/31'
/*END*/
/*IF projectName != null*/
and prj.project_name like /*projectName*/%' ESCAPE '$'
/*END*/
/*IF deleteFlag != null*/
and prj.delete_flag = /*deleteFlag*/0'
/*END*/
/*IF parentCompanyCd != null*/
and prj.parent_company_cd = /*parentCompanyCd*/parentCompanyCd'
/*END*/
/*IF parentProjectCd != null*/
and prj.parent_project_cd = /*parentProjectCd*/parentProjectCd'
/*END*/
/*IF extensionA != null*/
and prj.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and prj.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and prj.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and prj.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and prj.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and prj.extension_f like /*extensionF*/extensionF' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
and prj.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
and prj.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
and prj.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
and prj.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF parentProjectName != null*/
and parent.project_name like /*parentProjectName*/parentProjectName' ESCAPE '$'
/*END*/
/*IF orderBy != null*/
ORDER BY /*orderBy*/company_cd
/*END*/

```

Taglib

- **Path**
src/main/resources/META-INF/tld/kaidenMstSearchWorkflow/KaidenMstSearchWorkflow.tld

```

<?xml version="1.0" encoding="UTF-8" ?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
  version="2.1">
  <display-name>Kaiden product_workflow Components Master Search Tag library</display-name>
  <tlib-version>1.0</tlib-version>
  <short-name>master search product_workflow</short-name>
  <uri>http://kaiden.slcs.co.jp/taglib/master_search/workflow</uri>
  <tag>
    <name>project</name>

```

```
<tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchCriteria</tag-class>
<body-content>empty</body-content>
<attribute>
  <name>searchCriteriaDate</name>
  <required>>false</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>searchCriteriaCompany</name>
  <required>>true</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>searchCriteriaProjectCd</name>
  <required>>true</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>resultChain</name>
  <required>>true</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>resultValue</name>
  <required>>true</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>primaryTarget</name>
  <required>>true</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<attribute>
  <name>includeDisabled</name>
  <required>>false</required>
  <rtextprvalue>>true</rtextprvalue>
</attribute>
<!-- 以下 kaidenui text と同様 -->
<attribute>
  <name>name</name>
  <required>true</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>surfaceKey</name>
  <required>>false</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>mode</name>
  <required>true</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>value</name>
  <required>>false</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>formatted</name>
  <required>>false</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>imeMode</name>
  <required>>false</required>
  <rtextprvalue>true</rtextprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>

<tag>
<name>expType</name>
<tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchExpType</tag-class>
<body-content>empty</body-content>
<attribute>
  <name>searchCriteriaDate</name>
  <required>>false</required>
  <rtextprvalue>true</rtextprvalue>
</attribute>
<attribute>
```

```

<name>searchCriteriaCompany</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>searchCriteriaExpTypeCd</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>resultChain</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>resultValue</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>primaryTarget</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>includeDisabled</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<!-- 以下 kaidenui text と同様 -->
<attribute>
<name>name</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<attribute>
<name>surfaceKey</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<attribute>
<name>mode</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<attribute>
<name>value</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<attribute>
<name>formatted</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<attribute>
<name>imeMode</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
<type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
<name>allowance</name>
<tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchAllowance</tag-class>
<body-content>empty</body-content>
<attribute>
<name>searchCriteriaDate</name>
<required>false</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>searchCriteriaCompany</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
<name>searchCriteriaAllowanceCd</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>

```

```

</attribute>
<attribute>
  <name>allowanceCriteriaUserCd</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>currencyCd</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>resultChain</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>resultValue</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>primaryTarget</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>includeDisabled</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<!-- 以下 kaidenui text と同様 -->
<attribute>
  <name>name</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>surfaceKey</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>mode</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>value</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>formatted</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>account</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchAccount</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaAccountCd</name>
    <required>true</required>

```

```

<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>resultChain</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>resultValue</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>primaryTarget</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>includeDisabled</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<!-- 以下 kaidenui text と同様 -->
<attribute>
  <name>name</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>surfaceKey</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>mode</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>value</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>formatted</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>vendor</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchVendor</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCustomerCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
</tag>
<attribute>

```

```

<name>resultValue</name>
<required>true</required>
<rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>primaryTarget</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<attribute>
  <name>includeDisabled</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
</attribute>
<!-- 以下 kaidenui text と同様 -->
<attribute>
  <name>name</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>surfaceKey</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>mode</name>
  <required>true</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>value</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>formatted</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
</taglib>

```

XML

- **Path**
src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ProjectSearchSettings.xml


```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericSearchSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericSearchSettings">

  <!-- 検索画面に表示する最大件数 -->
  <limit>20</limit>
  <!-- 検索画面の行番号ラベルID -->
  <rownumberSurfaceKey></rownumberSurfaceKey><!-- デフォルトを利用 -->
  <!-- 検索画面のカラム設定 -->
  <listCols>
    <listCol>
      <!-- 対象マスタのフィールドキー -->
      <fieldKey>projectCd</fieldKey>
      <!-- キーワード検索対象とするか否か -->
      <keywordSearchTarget>true</keywordSearchTarget>
      <!-- 表示幅 -->
      <width>100px</width>
      <!-- 表示位置 -->
      <align></align>
      <!-- ヘッダラベルID -->
      <headerSurfaceKey>PWKF.projectCd</headerSurfaceKey>
      <!-- ソート可否 -->
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>projectName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>150px</width>
      <align></align>
      <headerSurfaceKey>PWKF.projectName</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>parentProjectCd</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey>PWKF.master.project.parentProjectCd</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>parentProjectName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>150px</width>
      <align></align>
      <headerSurfaceKey>PWKF.master.project.parentProjectName</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
  </listCols>
  <!-- 検索画面のソート順 -->
  <orders>
    <order>
      <fieldKey>projectCd</fieldKey>
      <type>asc</type>
    </order>
  </orders>
</genericSearchSettings>

```

経費区分マスタ検索

経費区分マスタ検索のサンプルソースです。

本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action](#)
 - [Dto](#)
 - [Form](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
 - [Taglib](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
- [Taglib](#)
- [XML](#)

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   │   ├── K30mExpType.java
│   │   │   │   │   │   │   │   │   │   ├── K30mExpTypeNames.java
│   │   │   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │   │       ├── K30mExpTypeCondition.java
│   │   │   │   │   │   │   │   │       └── K30mExpTypeService.java
│   │   ├── java
│   │   │   ├── jp
│   │   │   │   ├── co
│   │   │   │   │   ├── slcs
│   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeSearchAction.java
│   │   │   │   │   │   │   │   │   │   ├── dto
│   │   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeSearchDto.java
│   │   │   │   │   │   │   │   │   │   ├── form
│   │   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeSearchForm.java
│   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │   │   │   │   │   ├── exptype
│   │   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeManager.java
│   │   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeManagerDto.java
│   │   │   │   │   │   │   │   │   │   │   │   ├── ExpTypeManagerDtoFieldName.java
│   │   │   │   │   │   │   │   │   │   │   │   └── impl
│   │   │   │   │   │   │   │   │   │   │   │       ├── ExpTypeManagerImpl.java
│   │   │   │   │   │   │   │   │   └── taglib
│   │   │   │   │   │   │   │   │       ├── MstSearchExpType.java
│   │   ├── resources
│   │   │   ├── META-INF
│   │   │   │   ├── sql
│   │   │   │   │   ├── jp
│   │   │   │   │   │   ├── co
│   │   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   │   │   │   │   ├── K30mExpTypeService
│   │   │   │   │   │   │   │   │   │   │   │       ├── selectView.sql
│   │   │   │   │   │   ├── tld
│   │   │   │   │   │   │   ├── kaidenMstSearchWorkflow
│   │   │   │   │   │   │   │   ├── KaidenMstSearchWorkflow.tld
│   │   ├── webapp
│   │   │   ├── WEB-INF
│   │   │   │   ├── classes
│   │   │   │   │   ├── jp
│   │   │   │   │   │   ├── co
│   │   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   │   │       ├── ExpTypeSearchSettings.xml

```

Java

Action

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ExpTypeSearchAction.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master;
```

```

import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.helper.ContextHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.DateHelper;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchAction;
import jp.co.slcs.kaiden2.base.foundation.master.GenericSearchSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.HtmlEncodeUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master.ExpTypeSearchDto;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master.ExpTypeSearchForm;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.master_filtering_filter.exp_type.MasterFilteringExpTypeFilter;
import net.arnx.jsonic.JSON;
import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;

/**
 * 経費区分マスタ検索(Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeSearchAction extends AbstractMasterSearchAction {

    /** ExpTypeSearchDto. */
    @Resource
    public ExpTypeSearchDto expTypeSearchDto;

    /** ExpTypeSearchForm. */
    @ActionForm
    @Resource
    public ExpTypeSearchForm expTypeSearchForm;

    /** ExpTypeManager. */
    @Resource
    public ExpTypeManager expTypeManager;

    /** ContextHelper. */
    @Resource
    public ContextHelper contextHelper;

    /** DateHelper. */
    @Resource
    public DateHelper dateHelper;

    /** MasterFilteringExpTypeFilter. */
    @Resource
    public MasterFilteringExpTypeFilter masterFilteringExpTypeFilter;

    /**
     * 検索.
     * <p>
     * 以下の設定ファイルに従って検索を行う.<br />
     * WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ExpTypeSearchSettings.xml
     * </p>
     * <p>
     * 検索条件は以下の通り
     * </p>
     * <table border="1">
     * <tr>
     * <th>フィールド</th>
     * <th>条件</th>
     * <th>値の取得元</th>
     * <th>省略</th>
     * </tr>
     * <tr>
     * <td>会社コード</td>
     * <td>完全一致</td>
     * <td>画面リクエスト</td>
     * <td>不可</td>
     * </tr>
     * <tr>
     * <td>経費区分トコード</td>
     * <td>完全一致</td>
     * <td>画面リクエスト</td>
     * <td>可</td>
     * </tr>
     */

```

```

* <td>ロケールID</td>
* <td>完全一致</td>
* <td>ログインユーザのロケールID</td>
* <td>~</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>開始日-終了日の範囲</td>
* <td>画面リクエスト</td>
* <td>可(省略時はシステム日付を利用)</td>
* </tr>
* </table>
*
* @return 汎用検索JSP
* @throws KaidenSystemException XML読み込み例外
*/
@SuppressWarnings("unchecked")
@Execute(validator = false)
public String index() throws KaidenSystemException {
    searchDto = expTypeSearchDto;
    searchDto.searchKeywords = expTypeSearchForm.searchKeyword;
    // 設定ファイル読み込み
    searchDto.settings = new GenericSearchSettingsDto(readSetting(ExpTypeSearchAction.class));

    // キーワード条件
    KeywordsCriteriaDto keywordsCriteria = createKeywordCriteria(expTypeSearchForm.searchKeyword,
        expTypeSearchDto.settings);

    List<Map<String, Object>> searchResult = new ArrayList<Map<String, Object>>();
    // 検索条件の作成
    SearchCriteria criteria = new SearchCriteria();
    if (StringUtil.isEmpty(expTypeSearchForm.searchCriteriaCompany)) {
        criteria.companyCd = expTypeSearchForm.searchCriteriaCompany;

        if (StringUtil.isEmpty(expTypeSearchForm.searchCriteriaDate)) {
            criteria.searchCriteriaDate = dateHelper.toTimestamp(expTypeSearchForm.searchCriteriaDate);
        } else {
            criteria.searchCriteriaDate = dateHelper.getNowTimestamp();
        }
        if (StringUtil.isEmpty(expTypeSearchForm.expTypeCd)) {
            criteria.expTypeCd = expTypeSearchForm.expTypeCd;
        }
        if ("true".equals(expTypeSearchForm.includeDisabled)) {
            criteria.deleteFlag = null;
        } else {
            criteria.deleteFlag = GenericFlag.OFF.getCode();
        }
        criteria.localeId = contextHelper.getLoginUserLocale().toString();

        SqlResultFilterIF filter = masterFilteringExpTypeFilter
            .getFilter(expTypeSearchForm.mstSearchConditionJsonDefault);

        Map<String, Object> ret = expTypeManager.selectList4MstSearch(criteria,
            getOrder(searchDto.settings.orderJson), keywordsCriteria, new SqlResultFilterIF[] {filter});
        ConvertForJsonSource conv = new ConvertForJsonSource();
        for (Object dto : (List<Object>) ret.get("resultList")) {
            Map<String, Object> row = conv.dtoConvert((KaidenDto) dto);
            row.put("rowDataJson", HtmlEncodeUtil.escapeHtmlValue(JSON.encode(row)));
            searchResult.add(row);
        }
    }
    expTypeSearchDto.searchResult = searchResult;
    return GENERIC_SEARCH_JSP;
}

/**
 * 検索条件.
 */
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public static class SearchCriteria extends KaidenDto {

    /** 会社コード. */
    public String companyCd;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 削除フラグ. */
    public String deleteFlag;
    /** ロケールID. */
    public String localeId;
    /** 検索基準日. */
    public Timestamp searchCriteriaDate;
}
}

```

Dto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/dto/kaiden/master/ExpTypeSearchDto.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchDto;

/**
 * 経費区分マスタ検索 (DTO).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeSearchDto extends AbstractMasterSearchDto {

}
```

Form

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/master/ExpTypeSearchForm.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterSearchForm;

/**
 * 経費区分マスタ検索 (Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeSearchForm extends AbstractMasterSearchForm {

    /** 経費区分コード. */
    public String expTypeCd;

}
```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;

/**
 * 経費区分マスタへの各種操作を提供するAPIのinterface.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ExpTypeManager extends TermMasterManagerForJob<ExpTypeManagerDto> {

    /** 経費区分コード. */
    public static final int EXP_TYPE_CD_LEN = 100;

    /** 経費区分名称. */
    public static final int EXP_TYPE_NAME_LEN = 250;

    /** 経費区分カテゴリ. */
    public static final int EXP_TYPE_CATEGORY_LEN = 250;

    /** 勘定科目コード. */
    public static final int ACCOUNT_CD_LEN = 100;

    /** 税区分コード. */
    public static final int TAX_TYPE_CD_LEN = 100;

    /** ソートキー. */
    public static final int SORT_KEY_LEN = 15;

    /**
     * 新規登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

    /**
     * 更新登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

    /**
     * マスタ検索部品向け 一覧取得.
     *
     * @param criteria 検索条件
     * @param order ソート順
     * @param filter フィルタークラス
     * @return 検索結果
     */
    public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter);
}

```

Manager(実装)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/impl/ExpTypeManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;

```

```

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_INTEGER;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.SORT_KEY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mExpTypeService;
import org.seasar.framework.beans.util.Beatns;

/**
 * 経費区分マスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 使用する際には、{@link ExpTypeManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ExpTypeManagerDto}クラスまたはそのリストが戻り値として返却されます。<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <th>機能概要</th>
 * <th>メソッド</th>
 * <th>カテゴリ</th>
 * <th>備考</th>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>select</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>主キー検索(認可設定無視)</td>
 * <td>selectIgnoreAuthCompany</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>汎用検索</td>
 * <td>selectAll</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>

```



```

<td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索(認可設定無視)</td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索(認可設定無視)</td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索(認可設定無視)</td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索(認可設定無視)</td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化(認可設定無視)</td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化(認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録(認可設定無視)</td>
* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>

```

```
* </td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>削除(認可設定無視)</td>* <td>removeIgnoreAuthCompany</td>* <td>削除</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>更新</td>* <td>renew</td>* <td>更新</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>更新(認可設定無視)</td>* <td>renewIgnoreAuthCompany</td>* <td>更新</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* </table>* <br>* <b>機能一覧(期間化)</b>* <table border="1">* <tr>* <th>機能概要</th>* <th>メソッド</th>* <th>カテゴリ</th>* <th>備考</th>* </tr>* <tr>* <td>期間検索</td>* <td>selectTerms</td>* <td>期間検索</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>期間検索(認可設定無視)</td>* <td>selectTermsIgnoreAuthCompany</td>* <td>期間検索</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>基準日指定検索</td>* <td>selectListIncludesSearchDate</td>* <td>基準日指定検索</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>基準日指定検索(認可設定無視)</td>* <td>selectListIncludesSearchDateIgnoreAuthCompany</td>* <td>基準日指定検索</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>期間分割</td>* <td>divideTerm</td>* <td>期間分割</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>期間分割(認可設定無視)</td>* <td>divideTermIgnoreAuthCompany</td>* <td>期間分割</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>期間変更</td>* <td>moveTerm</td>* <td>期間変更</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* <tr>* <td>期間変更(認可設定無視)</td>* <td>moveTermIgnoreAuthCompany</td>* <td>期間変更</td>* <td>&nbsp;&nbsp;&nbsp;</td>* </tr>* </table>* <br>* <br>* 機能に対するI/Oを以下に示します。(上記一覧のカテゴリ単位)<br>* <br>* [IN]* <table border="1">* <tr>* <td align="right">●</td>
```

```
* <td>:必須</td>
* </tr>
* <tr>
* <td align="right">△</td>
* <td>:任意、検索時完全一致条件</td>
* </tr>
* <tr>
* <td align="right">▲</td>
* <td>:任意、検索時部分一致条件</td>
* </tr>
* <tr>
* <td align="right">×</td>
* <td>:不要(設定されていても無視します。)</td>
* </tr>
* <tr>
* <td align="right">☆</td>
* <td>:各APIの説明を参照</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>:対象外</td>
* </tr>
* </table>
* <br>
* [OUT]
* <table border="1">
* <tr>
* <td align="right">○</td>
* <td>:あり</td>
* </tr>
* <tr>
* <td align="right">×</td>
* <td>:なし</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>:対象外</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(共通)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">主キー<br>
* 検索</th>
* <th colspan="2">汎用<br>
* 検索</th>
* <th colspan="2">ロケール<br>
* 検索</th>
* <th colspan="2">参照<br>
* 検索</th>
* <th colspan="2">キーワード<br>
* 検索</th>
* <th colspan="2">削除フラグ<br>
* 更新</th>
* <th colspan="2">登録</th>
* <th colspan="2">削除</th>
* <th colspan="2">更新</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* </tr>
* <tr>
* <td>税区分名称</td>
* <td>tax TypeName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center"></td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center"></td>
* <td align="center">☆</td>
* <td align="center"></td>
* <td align="center">☆</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* <td align="center">x</td>
* <td align="center"></td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">期間検索</th>
* <th colspan="2">基準日指定<br>
* 検索</th>
* <th colspan="2">期間分割</th>
* <th colspan="2">期間変更</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center"></td>
* <td align="center">●</td>
* <td align="center"></td>
* </tr>
* <tr>
* <td><b>経費区分コード(*)</b></td>
```

```

* <td><b>expTypeCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分名称</td>
* <td>expTypeName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分カテゴリ</td>
* <td>expTypeCategory</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>startDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>endDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>accountCd</td>
* <td align="center">x</td>
* <td align="center">○</td>

```

```

<td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>taxTypeCd</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>ソートキー</td>
* <td>sortKey</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>
* <td>renewCnt</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>

```

```

        <td align="center">☆</td>
        <td align="center"></td>
        <td align="center">☆</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>更新日時</td>
        <td>renewTs</td>
        <td align="center">x</td>
        <td align="center">○</td>
        <td align="center">△</td>
        <td align="center">○</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td>更新者コード</td>
        <td>renewUserCode</td>
        <td align="center">x</td>
        <td align="center">○</td>
        <td align="center">△</td>
        <td align="center">○</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>会社名称</td>
        <td>companyName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>勘定科目名称</td>
        <td>accountName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>税区分名称</td>
        <td>taxTypeName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>検索基準日</td>
        <td>searchCriteriaDate</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">●</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
    </tr>
    </table>
    <br>
    テーブルのPK項目にはフィールド名に(*)を付与しています。<br>
    </p>
    *
    * @author Sumitomo Life Information Systems Co.,Ltd.
    */
    public class ExpTypeManagerImpl extends AbstractTermMasterManager<ExpTypeManagerDto> implements ExpTypeManager {

        /** K30mExpTvdeService. */

```

```

@Resource
protected K30mExpTypeService k30mExpTypeService;

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto select(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.select(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto selectIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectList(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria,
    List<OrderDto> order) {
    return super.selectList(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompany(ExpTypeManagerDto criteria, List<OrderDto> order) {
    return super.selectListIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocales(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocalesIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
    AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySql(k30mExpTypeService, "selectView.sql", criteria, order, authCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySqlIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectListByKeywords(k30mExpTypeService, "selectView.sql", criteria, order, keywordsCriteria,
        authCriteria, likeTarget);
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectListByKeywordsIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order,
        keywordsCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public void toEnabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toEnabled(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toEnabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
    super.toEnabledIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toDisabled(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
    super.toDisabledIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
    return super.entry(k30mExpTypeService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entryIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenDuplicationEntryException {
    return super.entryIgnoreAuthCompany(k30mExpTypeService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException {
    return super.remove(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.removeIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}

```

```

*/
@Override
public int renew(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    return super.renew(k30mExpTypeService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int renewIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenOptimisticLockException {
    return super.renewIgnoreAuthCompany(k30mExpTypeService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTerms(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTermsIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDate(ExpTypeManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ExpTypeManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
    super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
    String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
    KaidenValidateException {
    String termCd2nd = super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabled(criteria2nd, authCriteria);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabled(criteria2nd, authCriteria);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate, String deleteFlag2nd)
    throws KaidenOptimisticLockException, KaidenValidateException {
    String termCd2nd = super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabledIgnoreAuthCompany(criteria2nd);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabledIgnoreAuthCompany(criteria2nd);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mExpTypeService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mExpTypeService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * 重複チェック確認フィールドを返却。
 * <p>
 * 新規登録時、登録済みデータと引数の登録データの重複チェックを行うフィールドを返します。
 * </p>
 *
 * @return 重複チェック確認フィールドリスト
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, EXP_TYPE_CD});
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override

```



```

public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecutor(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * <tr>
 * <td>削除フラグ</td>
 * <td>フラグ妥当性</td>
 * </tr>
 * </table>
 */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
    // ロケールID
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 経費区分名称
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
    // 経費区分カテゴリ
    result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日

```

```

result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));
// 削除フラグ
result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、50桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * </table>
 */

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_COMPANY_EXISTS, COMPANY_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));

```

```

result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
// 期間コード
result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
// ロケールID
result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
// 経費区分名称
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
// 開始日
result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
// 終了日
result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));

return result;
}

/** {@inheritDoc} */
@Override
public Class<ExpTypeManagerDto> getMainManagerDtoClass() {
    return ExpTypeManagerDto.class;
}

@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompanyWithLock(ExpTypeManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mExpTypeService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ExpTypeManagerDto.class,
        k30mExpTypeService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiled() {
    return k30mExpTypeService.getPkFiled();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ExpTypeManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicF<ExpTypeManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mExpTypeService, criteria, order, logic);
}

@Override
public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    Map<String, Object> resultMap = super.selectListByKeywordsFilter(k30mExpTypeService, "selectView.sql",
        criteria, order, keywordsCriteria, likeTarget, filter);
    @SuppressWarnings("unchecked")
    List<ExpTypeManagerDto> newList = (List<ExpTypeManagerDto>) resultMap.get("resultList");
    resultMap.put("resultList", newList);

    return resultMap;
}
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO.
 *
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールid. */
    public String localeId;
    /** 経費区分名称. */
    public String expTypeName;
    /** 経費区分カテゴリ. */
    public String expTypeCategory;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** ソートキー. */
    public Long sortKey;
    /** 削除フラグ. */
    public String deleteFlag;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 会社名称(他マスタ名称). */
    public String companyName;
    /** 勘定科目名称(他マスタ名称). */
    public String accountName;
    /** 税区分名称(他マスタ名称). */
    public String taxTypeName;
    /** 税率(他マスタ). */
    public BigDecimal taxRate;
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO フィールド名定義クラス。
 *
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 *
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpTypeManagerDtoFieldName extends EntityCommonFieldNames {

    /**
     * コンストラクタ。
     */
    private ExpTypeManagerDtoFieldName() {

        /** 会社コード。 */
        public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
        /** 経費区分コード。 */
        public static final String EXP_TYPE_CD = "expTypeCd";
        /** 期間コード。 */
        public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
        /** ロケールid。 */
        public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
        /** 経費区分名称。 */
        public static final String EXP_TYPE_NAME = "expTypeName";
        /** 経費区分カテゴリ。 */
        public static final String EXP_TYPE_CATEGORY = "expTypeCategory";
        /** 開始日。 */
        public static final String START_DATE = EntityCommonFieldNames.START_DATE;
        /** 終了日。 */
        public static final String END_DATE = EntityCommonFieldNames.END_DATE;
        /** 勘定科目コード。 */
        public static final String ACCOUNT_CD = "accountCd";
        /** 税区分コード。 */
        public static final String TAX_TYPE_CD = "taxTypeCd";
        /** ソートキー。 */
        public static final String SORT_KEY = EntityCommonFieldNames.SORT_KEY;
        /** 削除フラグ。 */
        public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;

        /** 会社名称(他マスタ名称)。 */
        public static final String COMPANY_NAME = "companyName";
        /** 勘定科目名称(他マスタ名称)。 */
        public static final String ACCOUNT_NAME = "accountName";
        /** 税区分名称(他マスタ名称)。 */
        public static final String TAX_TYPE_NAME = "taxTypeName";
        /** 税率(他マスタ)。 */
        public static final String TAX_RATE = "taxRate";
    }
}

```

Taglib

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/taglib/MstSearchExpType.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.taglib;

import jp.co.slcs.kaiden2.base.foundation.taglib.AbstractMstSearch;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName;

/**
 * 経費区分マスタ検索.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class MstSearchExpType extends AbstractMstSearch {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /**
     * {@inheritDoc}
     */
    @Override
    protected String getSearchUrl() {
        return "expTypeSearch";
    }

    /**
     * 検索条件:経費区分コードを設定.
     *
     * @param searchCriteriaExpTypeCd 検索条件:経費区分コード
     */
    public void setSearchCriteriaExpTypeCd(String searchCriteriaExpTypeCd) {
        super.addSearchCriteria(ExpTypeManagerDtoFieldName.EXP_TYPE_CD, searchCriteriaExpTypeCd);
    }
}

```

Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpType.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 経費区分マスタ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30m_exp_type")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mExpType extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** 経費区分コード. */
    @Id
    @Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String expTypeCd;

    /** 期間コード. */
    @Id
    @Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String termCd;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;
}

```

```

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
public String expTypeName;

/** 経費区分カテゴリ. */
@Column(name = "exp_type_category", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeCategory;

/** 開始日. */
@Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp endDate;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String taxTypeCd;

/** ソートキー. */
@Column(name = "sort_key", columnDefinition = "number(15,0)", nullable = false, unique = false)
public Long sortKey;

/** 削除フラグ. */
@Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String deleteFlag;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpTypeNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

```

```

/
* {@link K30mExpType}の名称クラス.
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mExpTypeNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30mExpTypeNames() {
        super();
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * expTypeCategoryのプロパティ名を返却.
     *
     * @return expTypeCategoryのプロパティ名
     */
    public static PropertyName<String> expTypeCategory() {
        return new PropertyName<String>("expTypeCategory");
    }

    /**
     * startDateのプロパティ名を返却.
     *
     * @return startDateのプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDateのプロパティ名を返却.
     *
     * @return endDateのプロパティ名
     */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }
}

```



```

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * sortKeyのプロパティ名を返却.
 *
 * @return sortKeyのプロパティ名
 */
public static PropertyName<Long> sortKey() {
    return new PropertyName<Long>("sortKey");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public static PropertyName<String> deleteFlag() {
    return new PropertyName<String>("deleteFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */

```

```

    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**

```

```

* @author S2JDBC-Gen
*/
public static class _K30mExpTypeNames extends PropertyName<K30mExpType> {

    /**
     * インスタンスを構築.
     */
    public _K30mExpTypeNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30mExpTypeNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30mExpTypeNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public PropertyName<String> companyCd() {
        return new PropertyName<String>(this, "companyCd");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public PropertyName<String> termCd() {
        return new PropertyName<String>(this, "termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>(this, "localeId");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public PropertyName<String> expTypeName() {
        return new PropertyName<String>(this, "expTypeName");
    }

    /**
     * expTypeCategoryのプロパティ名を返却.
     *
     * @return expTypeCategoryのプロパティ名
     */
    public PropertyName<String> expTypeCategory() {
        return new PropertyName<String>(this, "expTypeCategory");
    }

    /**
     * startDateのプロパティ名を返却.
     *

```

```

    * @return startDateのプロパティ名
    */
    public PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>(this, "startDate");
    }

    /**
     * endDateのプロパティ名を返却.
     *
     * @return endDateのプロパティ名
     */
    public PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>(this, "endDate");
    }

    /**
     * accountCdのプロパティ名を返却.
     *
     * @return accountCdのプロパティ名
     */
    public PropertyName<String> accountCd() {
        return new PropertyName<String>(this, "accountCd");
    }

    /**
     * taxTypeCdのプロパティ名を返却.
     *
     * @return taxTypeCdのプロパティ名
     */
    public PropertyName<String> taxTypeCd() {
        return new PropertyName<String>(this, "taxTypeCd");
    }

    /**
     * sortKeyのプロパティ名を返却.
     *
     * @return sortKeyのプロパティ名
     */
    public PropertyName<Long> sortKey() {
        return new PropertyName<Long>(this, "sortKey");
    }

    /**
     * deleteFlagのプロパティ名を返却.
     *
     * @return deleteFlagのプロパティ名
     */
    public PropertyName<String> deleteFlag() {
        return new PropertyName<String>(this, "deleteFlag");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public PropertyName<String> extensionA() {
        return new PropertyName<String>(this, "extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public PropertyName<String> extensionB() {
        return new PropertyName<String>(this, "extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public PropertyName<String> extensionC() {
        return new PropertyName<String>(this, "extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }
}

```

```

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *
 * @return entryTsのプロパティ名
 */
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
 * entryUserCdのプロパティ名を返却。
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却。
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却。
 *
 * @return renewTsのプロパティ名

```

```

    */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mExpType;

/**
 * {@link K30mExpType}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mExpTypeService extends GenerateService<K30mExpType> {
    //
}

```

Condition

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30mExpType}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mExpTypeCondition extends
    AbstractEntityCondition<K30mExpTypeCondition> {

    /**
     * インスタンスを構築.
     */
    public K30mExpTypeCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30mExpTypeCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** companyCdの条件. */
    public NotNulltableStringCondition<K30mExpTypeCondition> companyCd =
        new NotNulltableStringCondition<K30mExpTypeCondition>("companyCd", this);

    /** expTypeCdの条件. */
    public NotNulltableStringCondition<K30mExpTypeCondition> expTypeCd =
        new NotNulltableStringCondition<K30mExpTypeCondition>("expTypeCd", this);
}

```

```

/** termCdの条件. */
public NotNullStringCondition<K30mExpTypeCondition> termCd =
    new NotNullStringCondition<K30mExpTypeCondition>("termCd", this);

/** localeIdの条件. */
public NotNullStringCondition<K30mExpTypeCondition> localeId =
    new NotNullStringCondition<K30mExpTypeCondition>("localeId", this);

/** expTypeNameの条件. */
public NotNullStringCondition<K30mExpTypeCondition> expTypeName =
    new NotNullStringCondition<K30mExpTypeCondition>("expTypeName", this);

/** expTypeCategoryの条件. */
public NullableStringCondition<K30mExpTypeCondition> expTypeCategory =
    new NullableStringCondition<K30mExpTypeCondition>("expTypeCategory", this);

/** startDateの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> startDate =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> endDate =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("endDate", this);

/** accountCdの条件. */
public NotNullStringCondition<K30mExpTypeCondition> accountCd =
    new NotNullStringCondition<K30mExpTypeCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NotNullStringCondition<K30mExpTypeCondition> taxTypeCd =
    new NotNullStringCondition<K30mExpTypeCondition>("taxTypeCd", this);

/** sortKeyの条件. */
public NotNullCondition<K30mExpTypeCondition, Long> sortKey =
    new NotNullCondition<K30mExpTypeCondition, Long>("sortKey", this);

/** deleteFlagの条件. */
public NotNullStringCondition<K30mExpTypeCondition> deleteFlag =
    new NotNullStringCondition<K30mExpTypeCondition>("deleteFlag", this);

/** extensionAの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionA =
    new NullableStringCondition<K30mExpTypeCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionB =
    new NullableStringCondition<K30mExpTypeCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionC =
    new NullableStringCondition<K30mExpTypeCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionD =
    new NullableStringCondition<K30mExpTypeCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionE =
    new NullableStringCondition<K30mExpTypeCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionF =
    new NullableStringCondition<K30mExpTypeCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionG =
    new NullableStringCondition<K30mExpTypeCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionH =
    new NullableStringCondition<K30mExpTypeCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionI =
    new NullableStringCondition<K30mExpTypeCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionJ =
    new NullableStringCondition<K30mExpTypeCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> entryTs =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */

```

```

public NotNullStringCondition<K30mExpTypeCondition> entryUserCd =
    new NotNullStringCondition<K30mExpTypeCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30mExpTypeCondition, Long> renewCnt =
    new NotNullCondition<K30mExpTypeCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> renewTs =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30mExpTypeCondition> renewUserCd =
    new NotNullStringCondition<K30mExpTypeCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService/selectView.sql

```

select
    exp.company_cd
, exp.exp_type_cd
, exp.term_cd
, exp.locale_id
, exp.exp_type_name
, exp.exp_type_category
, exp.start_date
, exp.end_date
, exp.account_cd
, exp.tax_type_cd
, exp.sort_key
, exp.delete_flag
, exp.extension_a
, exp.extension_b
, exp.extension_c
, exp.extension_d
, exp.extension_e
, exp.extension_f
, exp.extension_g
, exp.extension_h
, exp.extension_i
, exp.extension_j
, exp.entry_ts
, exp.entry_user_cd
, exp.renew_cnt
, exp.renew_ts
, exp.renew_user_cd
, dept.department_name as company_name
, account.account_name as account_name
, tax.tax_type_name as tax_type_name
, tax.tax_rate as tax_rate
from k30m_exp_type exp
left join ( select imm_department.company_cd
, imm_department.department_set_cd
, imm_department.department_cd
, imm_department.locale_id
, imm_department.start_date
, imm_department.end_date
, imm_department.department_name
from imm_department) dept
on exp.company_cd = dept.company_cd
and exp.company_cd = dept.department_set_cd
and exp.company_cd = dept.department_cd
and exp.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and dept.start_date <= exp.start_date
--ELSE and dept.end_date > exp.start_date
/*END*/
left join k30m_account account
on exp.company_cd = account.company_cd
and exp.account_cd = account.account_cd
and exp.locale_id = account.locale_id
/*IF searchCriteriaDate != null*/
and account.start_date <= /*searchCriteriaDate*/2000/01/01'
and account.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and account.start_date <= exp.start_date
--ELSE and account.end_date > exp.start_date
/*END*/

```



```

left join k30m_tax_type tax
on exp.company_cd = tax.company_cd
and exp.tax_type_cd = tax.tax_type_cd
and exp.locale_id = tax.locale_id
/*IF searchCriteriaDate != null*/
and tax.start_date <= /*searchCriteriaDate*/'2000/01/01'
and tax.end_date > /*searchCriteriaDate*/'2000/01/01'
--ELSE and tax.start_date <= exp.start_date
--ELSE and tax.end_date > exp.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
exp.company_cd in /*companyCd*/('comp')
/*END*/
/*IF expTypeCd != null*/
and exp.exp_type_cd = /*expTypeCd*/expTypeCd'
/*END*/
/*IF termCd != null*/
and exp.term_cd = /*termCd*/termCd'
/*END*/
/*IF localeId != null*/
and exp.locale_id = /*localeId*/localeId'
/*END*/
/*IF expTypeName != null*/
and exp.exp_type_name like /*expTypeName*/'% ' ESCAPE '$'
/*END*/
/*IF expTypeCategory != null*/
and exp.exp_type_category like /*expTypeCategory*/expTypeCategory' ESCAPE '$'
/*END*/
/*IF searchCriteriaDate != null*/
and exp.start_date <= /*searchCriteriaDate*/'1900/01/01'
and exp.end_date > /*searchCriteriaDate*/'2999/12/31'
/*END*/
/*IF accountCd != null*/
and exp.account_cd = /*accountCd*/accountCd'
/*END*/
/*IF taxTypeCd != null*/
and exp.tax_type_cd = /*taxTypeCd*/taxTypeCd'
/*END*/
/*IF sortKey != null*/
and exp.sort_key = /*sortKey*/'0'
/*END*/
/*IF deleteFlag != null*/
and exp.delete_flag = /*deleteFlag*/'0'
/*END*/
/*IF extensionA != null*/
and exp.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and exp.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and exp.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and exp.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and exp.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and exp.extension_f like /*extensionF*/extensionF' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
and exp.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
and exp.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
and exp.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
and exp.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF accountName != null*/
and account.account_name like /*accountName*/accountName' ESCAPE '$'
/*END*/
/*IF taxTypeName != null*/
and tax.tax_type_name like /*taxTypeName*/taxTypeName' ESCAPE '$'
/*END*/
/*IF taxRate != null*/

```

```
and tax.tax_rate = /*taxRate*/0'
/*END*/
/*END*/
/*IF orderBy != null*/
ORDER BY /*$orderBy*/company_cd
/*END*/
```

Taglib

- **Path**

```
src/main/resources/META-INF/tld/kaidenMstSearchWorkflow/KaidenMstSearchWorkflow.tld
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
  version="2.1">

<display-name>KAIDEN product_workflow Components Master Search Tag library</display-name>
<lib-version>1.0</lib-version>
<short-name>master search product_workflow</short-name>
<uri>http://kaiden.slcs.co.jp/taglib/master_search/workflow</uri>
<tag>
  <name>project</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchProject</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaProjectCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultValue</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>primaryTarget</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>includeDisabled</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <!-- 以下 kaidenui text と同様 -->
  <attribute>
    <name>name</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>surfaceKey</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>mode</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>value</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
```

```

    <type>String</type>
  </attribute>
</attribute>
  <name>formatted</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
</attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>expType</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchExpType</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaExpTypeCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultValue</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>primaryTarget</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>includeDisabled</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <!-- 以下 kaidenui text と同様 -->
  <attribute>
    <name>name</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>surfaceKey</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>mode</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>value</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>formatted</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>

```

```

<attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>allowance</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchAllowance</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaAllowanceCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>allowanceCriteriaUserCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>currencyCd</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultValue</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>primaryTarget</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>includeDisabled</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <!-- 以下 kaidenui text と同様 -->
  <attribute>
    <name>name</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>surfaceKey</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>mode</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>value</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>formatted</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>

```

```

    <type>String</type>
  </attribute>
<attribute>
  <name>imeMode</name>
  <required>false</required>
  <rtexprvalue>true</rtexprvalue>
  <type>String</type>
</attribute>
<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>account</name>
  <tag-class>jp.co.sics.kaiden2.product_workflow.foundation.taglib.MstSearchAccount</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaAccountCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultValue</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>primaryTarget</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>includeDisabled</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <!-- 以下 kaidenui text と同様 -->
  <attribute>
    <name>name</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>surfaceKey</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>mode</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>value</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>formatted</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>imeMode</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>

```

```

<dynamic-attributes>true</dynamic-attributes>
</tag>
<tag>
  <name>vendor</name>
  <tag-class>jp.co.slcs.kaiden2.product_workflow.foundation.taglib.MstSearchVendor</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>searchCriteriaDate</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCompany</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>searchCriteriaCustomerCd</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultChain</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>resultValue</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>primaryTarget</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>includeDisabled</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <!-- 以下 kaidenui text と同様 -->
  <attribute>
    <name>name</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>surfaceKey</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>mode</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>value</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>formatted</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <attribute>
    <name>imeMode</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    <type>String</type>
  </attribute>
  <dynamic-attributes>true</dynamic-attributes>
</tag>
</taglib>

```

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ExpTypeSearchSettings.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericSearchSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericSearchSettings">

  <!-- 検索画面に表示する最大件数 -->
  <limit>20</limit>
  <!-- 検索画面の行番号ラベルID -->
  <rownumberSurfaceKey><!-- デフォルトを利用 -->
  <!-- 検索画面のカラム設定 -->
  <listCols>
    <listCol>
      <!-- 対象マスタのフィールドキー -->
      <fieldKey>expTypeCd</fieldKey>
      <!-- キーワード検索対象とするか否か -->
      <keywordSearchTarget>true</keywordSearchTarget>
      <!-- 表示幅 -->
      <width>100px</width>
      <!-- 表示位置 -->
      <align></align>
      <!-- ヘッダラベルID -->
      <headerSurfaceKey>PWKF.expTypeCd</headerSurfaceKey>
      <!-- ソート可否 -->
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>expTypeName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>150px</width>
      <align></align>
      <headerSurfaceKey>PWKF.expTypeName</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>accountCd</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey>PWKF.accountCd</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>accountName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>150px</width>
      <align></align>
      <headerSurfaceKey>PWKF.accountName</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>taxTypeCd</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey>PWKF.taxTypeCd</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
    <listCol>
      <fieldKey>taxTypeName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width>150px</width>
      <align></align>
      <headerSurfaceKey>PWKF.taxTypeName</headerSurfaceKey>
      <sortable>false</sortable>
    </listCol>
  </listCols>
  <!-- 検索画面のソート順 -->
  <orders>
    <order>
      <fieldKey>sortKey</fieldKey>
      <type>asc</type>
    </order>
    <order>
      <fieldKey>expTypeCd</fieldKey>
      <type>asc</type>
    </order>
  </orders>
</genericSearchSettings>

```


プロジェクトマスタメンテナンス

プロジェクトマスタメンテナンスのサンプルソースです。

本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action](#)
 - [Dto](#)
 - [Form](#)
 - [Logic](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
- [XML](#)
- [JSP](#)

[一覧](#)

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   └── co
│   │   │       └── slcs
│   │   │           └── kaiden2
│   │   │               └── product_workflow
│   │   │                   └── foundation
│   │   │                       └── model
│   │   │                           ├── entity
│   │   │                               ├── K30mProject.java
│   │   │                               ├── K30mProjectNames.java
│   │   │                               └── service
│   │   │                                   ├── K30mProjectCondition.java
│   │   │                                   └── K30mProjectService.java
│   ├── java
│   │   ├── jp
│   │   │   └── co
│   │   │       └── slcs
│   │   │           └── kaiden2
│   │   │               └── product_workflow
│   │   │                   ├── feature
│   │   │                       ├── action
│   │   │                           ├── kaiden
│   │   │                               └── master
│   │   │                                   ProjectAction.java
│   │   │                   ├── dto
│   │   │                       ├── kaiden
│   │   │                           └── master
│   │   │                               ProjectDto.java
│   │   │                   ├── form
│   │   │                       ├── kaiden
│   │   │                           └── master
│   │   │                               ProjectForm.java
│   │   │                   ├── logic
│   │   │                       ├── kaiden
│   │   │                           └── master
│   │   │                               ProjectLogic.java
│   │   │                   └── foundation
│   │   │                       ├── manager
│   │   │                           └── project
│   │   │                               ├── ProjectCriteriaDto.java
│   │   │                               ├── ProjectManager.java
│   │   │                               ├── ProjectManagerDto.java
│   │   │                               ├── ProjectManagerDtoFieldName.java
│   │   │                               └── impl
│   │   │                                   ProjectManagerImpl.java
│   ├── resources
│   │   ├── META-INF
│   │   ├── sql
│   │   │   └── jp
│   │   │       └── co
│   │   │           └── slcs
│   │   │               └── kaiden2
│   │   │                   └── product_workflow
│   │   │                       └── foundation
│   │   │                           └── model
│   │   │                               └── service
│   │   │                                   K30mProjectService
│   │   │                                       selectView.sql
│   ├── webapp
│   │   ├── WEB-INF
│   │   │   ├── classes
│   │   │   │   ├── jp
│   │   │   │   │   └── co
│   │   │   │       └── slcs
│   │   │   │           └── kaiden2
│   │   │   │               └── product_workflow
│   │   │   │                   └── feature
│   │   │   │                       ├── action
│   │   │   │                           ├── kaiden
│   │   │   │                               └── master
│   │   │   │                                   ProjectMasterMaintenanceSettings.xml
│   │   │   └── view
│   │   │       ├── kaiden2
│   │   │       │   └── product_workflow
│   │   │       │       └── master
│   │   │       │           └── project
│   │   │       │               projectMainte.jsp

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ProjectAction.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master;

import javax.annotation.Resource;
import jp.co.intra_mart.foundation.authz.annotation.Authz;
import jp.co.slcs.kaiden2.base.foundation.conf.ModuleEnum;
import jp.co.slcs.kaiden2.base.foundation.conf.ResultType;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceDto;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceAction;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.base.foundation.model.annotation.Module;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master.ProjectDto;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master.ProjectForm;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.master.ProjectLogic;
import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;

/**
 * プロジェクトマスタメンテナンス (Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectAction extends AbstractTermMasterMaintenanceAction {

    /** ProjectDto. */
    @Resource
    public ProjectDto projectDto;

    /** ProjectForm. */
    @ActionForm
    @Resource
    public ProjectForm projectForm;

    /** ProjectLogic. */
    @Resource
    public ProjectLogic projectLogic;

    /**
     * 入力ページのパスを返却.
     *
     * @return 入力ページのパス
     * @throws KaidenSystemException XML読み込み例外
     */
    @Override
    @Execute(validator = false)
    @Authz(uri = "service://kaiden/master/project", action = "execute")
    public String index() throws KaidenSystemException {

        // プロパティファイルから一覧画面の設定を読み込み
        MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
        projectDto.settings = new MasterMaintenanceSettingsDto(reader.read(ProjectAction.class));

        // 一覧画面の設定をレスポンスに設定
        return super.index();
    }

    /**
     * AJAXにて検索を実行.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    @Authz(uri = "service://kaiden/master/project", action = "execute")
    public String search() {
        return super.search();
    }

    /**
     * AJAXにてメンテナンス対象データを返却.
     *
     * @return メンテナンス対象データ
     */
    @Override

```

```

@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String select() {
    return super.select();
}

/**
 * AJAXにてデータを登録.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String entry() {
    return super.entry();
}

/**
 * AJAXにて無効化.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String toDisabled() {
    return super.toDisabled();
}

/**
 * AJAXにてデータ削除.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String remove() {
    return super.remove();
}

/**
 * AJAXにてデータ更新.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String renew() {
    return super.renew();
}

/**
 * AJAXにてメンテナンス対象データの期間データを返却.
 *
 * @return メンテナンス対象データ
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String selectTerm() {
    return super.selectTerm();
}

/**
 * AJAXにて有効化.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */

```

```

@Override
@Execute(validator = false)
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String toEnabled() {
    return super.toEnabled();
}

/**
 * AJAXにて期間分割.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String divideTerm() {
    return super.divideTerm();
}

/**
 * AJAXにて期間変更.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute(validator = false)
@Authz(uri = "service://kaiden/master/project", action = "execute")
public String moveTerm() {
    return super.moveTerm();
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceDto getDto() {
    return projectDto;
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceForm getForm() {
    return projectForm;
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceLogic getLogic() {
    return projectLogic;
}
}

```

Dto

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/dto/kaiden/master/ProjectDto.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceDto;

/**
 * プロジェクトマスタメンテナンス (DTO).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectDto extends AbstractTermMasterMaintenanceDto {
}

```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/master/ProjectForm.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm;

/**
 * プロジェクトマスタメンテナンス (Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectForm extends AbstractTermMasterMaintenanceForm {

}

```

Logic

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/master/ProjectLogic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.master;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.START_DATE;
import java.sql.Timestamp;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Stack;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.base.foundation.master.MasterManager;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master.ProjectAction;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;

/**
 * プロジェクトマスタメンテナンス (Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectLogic extends AbstractTermMasterMaintenanceLogic {

    /** ProjectManager. */
    @Resource
    public ProjectManager projectManager;

    /**
     * {@inheritDoc}
     */
    @Override
    public LogicResultDto selectTermAjax(AbstractTermMasterMaintenanceForm form) {
        Convert conv = new Convert();
        return super.selectTermAjax(form, conv);
    }

    /**
     * プロジェクトメンテナンス画面へデータを渡す際の書式変換.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    private static class Convert extends ConvertForJsonSource {

        /**
         * {@inheritDoc}
         *
         * <p>
         * 本サブクラスでは、以下の書式変換を行う。<br>
         * <table border="1">
         * <tr>
         * <td>項目名</td>
         * <td>書式</td>
         * </tr>
         * </table>
         */
    }
}

```

```

* </tr>
* <tr>
* <td>開始日</td>
* <td>システムの日付時刻形式</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>システムの日付時刻形式</td>
* </tr>
* </table>
* </p>
*/
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    // 標準書式変換
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 開始日の書式変換
    Object startDate = map.get(START_DATE);
    if (startDate != null) {
        result.put(START_DATE, dateHelper.formatDateSystemTz((Timestamp) startDate));
    }

    // 終了日の書式変換
    Object endDate = map.get(END_DATE);
    if (endDate != null) {
        result.put(END_DATE, dateHelper.formatDateSystemTz((Timestamp) endDate));
    }

    return result;
}

/**
 * {@inheritDoc}
 */
@Override
protected MasterManager<?> getManager() {
    return projectManager;
}

/**
 * {@inheritDoc}
 */
@Override
protected String[] getKeywordSearchTargetCol() {
    // プロパティファイルから一覧画面の設定を読み込み
    MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
    MasterMaintenanceSettingsDto settings;
    try {
        settings = new MasterMaintenanceSettingsDto(reader.read(ProjectAction.class));
    } catch (KaidenSystemException e) {
        throw KaidenRuntimeException.makeInstance(e);
    }

    return settings.searchTargetCol.toArray(new String[settings.searchTargetCol.size()]);
}

/**
 * {@inheritDoc}
 */
@Override
protected List<String> get18nTargetField() {
    return Arrays.asList("projectName");
}

/**
 * {@inheritDoc}
 */
@Override
protected Map<String, Object> getInputData(AbstractMasterMaintenanceForm form) {
    return JsonUtil.getMap(form.updateDataJson, Arrays.asList("inputArea"));
}

/**
 * {@inheritDoc}
 * <p>
 * ■設定値一覧<br>
 * <table border="1">
 * <tr>
 * <td>設定キー</td>
 * <td>設定値</td>
 * </tr>
 * <tr>
 * <td>companyCd</td>
 * <td>会社</td>
 * </tr>

```

```

</tr>
* <tr>
* <td>projectCd</td>
* <td>プロジェクトコード</td>
* </tr>
* <tr>
* <td>startDate</td>
* <td>開始日</td>
* </tr>
* <tr>
* <td>endDate</td>
* <td>終了日</td>
* </tr>
* <tr>
* <td>projectName</td>
* <td>プロジェクト名</td>
* </tr>
* <tr>
* <td>parentProjectCd</td>
* <td>上位プロジェクト</td>
* </tr>
* </table>
* </p>
*/
@Override
protected Map<String, String> getSurfaces() {

    Map<String, String> result = new HashMap<String, String>();

    result.put(COMPANY_CD, propertyHelper.getSurface(BaseProp.S.COMPANY));
    result.put(PROJECT_CD, propertyHelper.getSurface(ProductWorkflowProp.S.PROJECT_CD));
    result.put(START_DATE, propertyHelper.getSurface(BaseProp.S.START_DATE));
    result.put(END_DATE, propertyHelper.getSurface(BaseProp.S.END_DATE));
    result.put(PROJECT_NAME, propertyHelper.getSurface(ProductWorkflowProp.S.PROJECT_NAME));
    result.put(PARENT_PROJECT_CD, propertyHelper.getSurface(ProductWorkflowProp.S.Project.PARENT_PROJECT));

    return result;
}
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.ValidateMapHelper;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenAbstractValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidatorIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import org.seasar.framework.container.SingletonS2Container;

/**
 * プロジェクトマネージャのinterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ProjectManager extends TermMasterManager<ProjectManagerDto>,
    TermMasterManagerForJob<ProjectManagerDto> {

    /**
     * プロジェクトコードの最大長.
     */
    public static final int PROJECT_CD_LEN = 100;

    /**
     * プロジェクト名の最大長.
     */
    public static final int PROJECT_NAME_LEN = 250;

    /**
     * 新規登録用の検証定義リストを返却します.
     */
}

```



```

*
* @return バリデート実行クラス用パラメタDTOリスト
*/
public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

/**
 * 更新登録用の検証定義リストを返却します。
 *
 * @return バリデート実行クラス用パラメタDTOリスト
 */
public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

/**
 * カスタムバリデート_自プロジェクト・上位プロジェクト同一チェック。
 * <p>
 * 自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。
 * </p>
 */
class CustomValidateNotEqualParent extends KaidenAbstractValidator implements KaidenValidatorIF {

    /** コンストラクタ。 */
    public CustomValidateNotEqualParent() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * チェックエラーとなる条件<br>
     * <ul>
     * <li>自プロジェクト・上位プロジェクト同一チェック。</li>
     * <li>自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。</li>
     * </ul>
     * 以下の場合、チェックを行いません。
     * <ul>
     * <li>自プロジェクトコード・自会社コードのいずれかが設定されていない場合</li>
     * <li>上位プロジェクトコード・上位会社コードの両方が未設定の場合</li>
     * </ul>
     * 以下の場合、パラメーターエラーとします。
     * <ul>
     * <li>上位プロジェクトコード・上位会社コードの片方が未設定の場合</li>
     * </ul>
     * </p>
     */
    @Override
    public KaidenValidateResult executeValidate(Map<String, Object> valueMap, String fieldKey, String[] args,
        Map<String, String> surfaces) {

        KaidenValidateResult result = super.makeNewValidateResult(fieldKey, surfaces);

        ValidateMapHelper validateMapHelper = SingletonS2Container.getComponent(ValidateMapHelper.class);

        String thisCompanyCd = validateMapHelper.getValueForString(valueMap, COMPANY_CD);
        String thisProjectCd = validateMapHelper.getValueForString(valueMap, PROJECT_CD);

        // 自プロジェクトコードまたは自会社コードが設定されていない場合は、エラーとしない
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            return result;
        }

        String parentCompanyCd = validateMapHelper.getValueForString(valueMap, PARENT_COMPANY_CD);
        String parentProjectCd = validateMapHelper.getValueForString(valueMap, PARENT_PROJECT_CD);

        // 上位プロジェクトコードと上位会社コードの両方が未設定の場合、エラーとしない
        if (StringUtil.isEmpty(parentCompanyCd, parentProjectCd)) {
            return result;
        }

        // 上位プロジェクトコード・上位会社コードの片方が未設定の場合、パラメーターエラー
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            throw KaidenIllegalParametersException.makeInstance();
        }

        // 自プロジェクトコードと上位プロジェクトコードが同一かつ、自会社コードと上位会社コードが同一の場合、チェックエラー
        if (thisCompanyCd.equals(parentCompanyCd) && thisProjectCd.equals(parentProjectCd)) {
            PropertyHelper propertyHelper = SingletonS2Container.getComponent(PropertyHelper.class);
            String fieldName = validateMapHelper.getSurfaces(valueMap, surfaces, PARENT_PROJECT_CD);
            result.setErrorStatus(true);
            result.setValidateClass(this.getClass());
            result.setFieldKey(PARENT_PROJECT_CD);
            result.setFieldNm(fieldName);
            result.setMainMessageKey(ProductWorkflowProp.M.Project.W001);
            result.setMainMessage(propertyHelper.getMessage(ProductWorkflowProp.M.Project.W001, fieldName));
        }

        return result;
    }

```

```

    }
}
}

```

Manager (実装)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/impl/ProjectManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mProjectService;
import org.seasar.framework.beans.util.Beans;

/**
 * プロジェクトマスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います、より業務的な機能を付加した操作を提供します.<br>
 * 使用する際には、{@link ProjectManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ProjectManagerDto}クラスまたはそのリストが戻り値として返却されます.<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>

```

```

* <table border="1">
* <tr>
* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>主キー検索</td>
* <td>select</td>
* <td>主キー検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>主キー検索 (認可設定無視)</td>
* <td>selectIgnoreAuthCompany</td>
* <td>主キー検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索</td>
* <td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索 (認可設定無視)</td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索 (認可設定無視)</td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索 (認可設定無視)</td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索 (認可設定無視)</td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化 (認可設定無視)</td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化 (認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>

```

```

<td>無効化(認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録(認可設定無視)</td>
* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除(認可設定無視)</td>
* <td>removeIgnoreAuthCompany</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新</td>
* <td>renew</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新(認可設定無視)</td>
* <td>renewIgnoreAuthCompany</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能一覧(期間化)</b>
* <table border="1">
* <tr>
* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>期間検索</td>
* <td>selectTerms</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間検索(認可設定無視)</td>
* <td>selectTermsIgnoreAuthCompany</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索</td>
* <td>selectListIncluedsSearchDate</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索(認可設定無視)</td>
* <td>selectListIncluedsSearchDateIgnoreAuthCompany</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割</td>
* <td>divideTerm</td>
* <td>期間分割</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>期間分割(認可設定無視)</td>
* <td>divideTermIgnoreAuthCompany</td>
* <td>期間分割</td>
* <td>&nbsp;</td>

```

```
<table border="1">
<tr>
<td>期間変更</td>
<td>moveTerm</td>
<td>期間変更</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>期間変更(認可設定無視)</td>
<td>moveTermIgnoreAuthCompany</td>
<td>期間変更</td>
<td>&nbsp;</td>
</tr>
</table>
<br>
<br>
機能に対するI/Oを以下に示します。(上記一覧のカテゴリ単位)<br>
<br>
[IN]
<table border="1">
<tr>
<td align="right">●</td>
<td>: 必須</td>
</tr>
<tr>
<td align="right">△</td>
<td>: 任意、検索時完全一致条件</td>
</tr>
<tr>
<td align="right">▲</td>
<td>: 任意、検索時部分一致条件</td>
</tr>
<tr>
<td align="right">x</td>
<td>: 不要(設定されていても無視します。)</td>
</tr>
<tr>
<td align="right">☆</td>
<td>: 各APIの説明を参照</td>
</tr>
<tr>
<td align="right">—</td>
<td>: 対象外</td>
</tr>
</table>
<br>
[OUT]
<table border="1">
<tr>
<td align="right">○</td>
<td>: あり</td>
</tr>
<tr>
<td align="right">x</td>
<td>: なし</td>
</tr>
<tr>
<td align="right">—</td>
<td>: 対象外</td>
</tr>
</table>
<br>
<b>DTOクラスとメソッドのI/Oマトリクス(共通)</b>
<table border="1">
<tr>
<td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>
<td rowspan="2" align="center"><b>DTOフィールド名</b></td>
<td colspan="2" align="center"><b>主キー</b><br>
検索</td>
</tr>
<tr>
<td colspan="2" align="center"><b>汎用</b><br>
検索</td>
</tr>
<tr>
<td colspan="2" align="center"><b>ロケール</b><br>
検索</td>
<td colspan="2" align="center"><b>参照</b><br>
検索</td>
</tr>
<tr>
<td colspan="2" align="center"><b>キーワード</b><br>
検索</td>
<td colspan="2" align="center"><b>削除フラグ</b><br>
更新</td>
</tr>
<tr>
<td colspan="2" align="center"><b>登録</b></td>
<td colspan="2" align="center"><b>削除</b></td>
</tr>
<tr>
<td colspan="2" align="center"><b>更新</b></td>
<td colspan="2" align="center"><b>更新</b></td>
</tr>
</table>
<br>
<table border="1">
<tr>
<td align="center"><b>I</b></td>
<td align="center"><b>O</b></td>
</tr>
</table>
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスとメソッドのI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>
* <td rowspan="2" align="center"><b>DTOフィールド名</b></td>
* <td colspan="2" align="center"><b>期間検索</b></td>
* <td colspan="2" align="center"><b>基準日指定検索</b></td>
* <td colspan="2" align="center"><b>期間分割</b></td>
* <td colspan="2" align="center"><b>期間変更</b></td>
* </tr>
* <tr>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>プロジェクトコード(*)</b></td>
* <td><b>projectCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>startDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>

```

```

* </tr>
* <tr>
* <td>終了日</td>
* <td>endDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクト名称</td>
* <td>projectName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>会社コード(上位)</td>
* <td>parentCompanyCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクトコード(上位)</td>
* <td>parentProjectCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録者</td>

```

```

* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>
* <td>renewCnt</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">☆</td>
* <td align="center">×</td>
* <td align="center">☆</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>更新日時</td>
* <td>renewTs</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>更新者コード</td>
* <td>renewUserCd</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>組織名称</td>
* <td>companyName</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>プロジェクト名称(上位)</td>
* <td>parentProjectName</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">●</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* <td align="center">×</td>
* </tr>
* </table>
* <br>
* <br>
* テーブルのRZ項目にはフィールド名が格納されています。

```

```

* </p>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/

public class ProjectManagerImpl extends AbstractTermMasterManager<ProjectManagerDto> implements ProjectManager {

    /** K30mProjectService. */
    @Resource
    protected K30mProjectService k30mProjectService;

    /**
     * {@inheritDoc}
     */
    @Override
    public ProjectManagerDto select(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
        return super.select(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public ProjectManagerDto selectIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.selectIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectList(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria,
        List<OrderDto> order) {
        return super.selectList(k30mProjectService, criteria, authCriteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListIgnoreAuthCompany(ProjectManagerDto criteria, List<OrderDto> order) {
        return super.selectListIgnoreAuthCompany(k30mProjectService, criteria, order);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
        AuthorityCriteriaDto authCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectBySql(k30mProjectService, "selectView.sql", criteria, order, authCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public List<ProjectManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectBySqlIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectListByKeywords(k30mProjectService, "selectView.sql", criteria, order, keywordsCriteria,
            authCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria) {
        String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
            EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
        return super.selectListByKeywordsIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order,
            keywordsCriteria, likeTarget);
    }

```

```

    keyWordsCriteria, mtoTarget,
}

/**
 * {@inheritDoc}
 *
 * @throws KaidenOptimisticLockException
 */
@Override
public void toEnabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toEnabled(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 *
 * @throws KaidenOptimisticLockException
 */
@Override
public void toEnabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
    super.toEnabledIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toDisabled(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
    super.toDisabledIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectLocales(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectLocalesIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectTerms(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectTermsIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
    return super.entry(k30mProjectService, dataList, authCriteria);
}

/**
 * {@inheritDoc}

```

```

    /**
     * {@inheritDoc}
     */
    @Override
    public int entryIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenDuplicationEntryException {
        return super.entryIgnoreAuthCompany(k30mProjectService, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int remove(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException {
        return super.remove(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.removeIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        return super.renew(k30mProjectService, dataList, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renewIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenOptimisticLockException {
        return super.renewIgnoreAuthCompany(k30mProjectService, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
        super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
        String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
        KaidenValidateException {
        String termCd2nd = super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
        if (StringUtil.isEmpty(deleteFlag2nd)) {
            ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
            criteria2nd.termCd = termCd2nd;
            if (GenericFlag.isOn(deleteFlag2nd)) {
                toDisabled(criteria2nd, authCriteria);
            } else if (GenericFlag.isOff(deleteFlag2nd)) {
                toEnabled(criteria2nd, authCriteria);
            } else {
                throw KaidenIllegalParametersException.makeInstance();
            }
        }
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate)
        throws KaidenOptimisticLockException, KaidenValidateException {
        super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate, String deleteFlag2nd)
        throws KaidenOptimisticLockException, KaidenValidateException {

```



```

String termCd2nd = super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
if (StringUtil.isEmpty(deleteFlag2nd)) {
    ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
    criteria2nd.termCd = termCd2nd;
    if (GenericFlag.isOn(deleteFlag2nd)) {
        toDisabledIgnoreAuthCompany(criteria2nd);
    } else if (GenericFlag.isOff(deleteFlag2nd)) {
        toEnabledIgnoreAuthCompany(criteria2nd);
    } else {
        throw KaidenIllegalParametersException.makeInstance();
    }
}
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ProjectManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mProjectService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ProjectManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mProjectService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, PROJECT_CD});
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDate(ProjectManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mProjectService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ProjectManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mProjectService, criteria, order);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }

    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**

```

```

* {@inheritDoc}
* <p>
* {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
* </p>
*/

@Override
public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
* {@inheritDoc}
* <p>
* 以下のバリデート定義のリストを返却します。
* </p>
* <table border="1">
* <tr>
* <th>項目名</th>
* <th>検証内容</th>
* </tr>
* <tr>
* <td>会社コード</td>
* <td>必須、マスタ存在確認</td>
* </tr>
* <tr>
* <td>プロジェクトコード</td>
* <td>必須、昔伝コード、100桁</td>
* </tr>
* <tr>
* <td>ロケールID</td>
* <td>必須、ロケール妥当性</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>必須、日付(システム期間内)</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
* </tr>
* <tr>
* <td>プロジェクト名</td>
* <td>必須、250文字</td>
* </tr>
* <tr>
* <td>プロジェクトコード(上位プロジェクト)</td>
* <td>半角英数、マスタ存在確認</td>
* </tr>
* <tr>
* <td>会社コード(上位プロジェクト)</td>
* <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>フラグ妥当性</td>
* </tr>
* <tr>
* <td>プロジェクトコード(上位プロジェクト)</td>
* <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
* </tr>
* </table>
*/

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // プロジェクトコード
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
    // ロケールid
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日
    result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
}

```

```

result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// プロジェクト名称
result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
// プロジェクトコード(上位)
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
    "", "true"));
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));
// 削除フラグ
result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード</td>
 * <td>必須、皆伝コード、100桁</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、皆伝コード、50桁</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>プロジェクト名</td>
 * <td>必須、250文字</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>会社コード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
 * </tr>
 * </table>
 */

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // プロジェクトコード
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
    // 期間コード
    result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
    result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
    // ロケールid
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));

```

```

// 終了日
result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// プロジェクト名称
result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
// プロジェクトコード(上位)
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
    "", "true"));
result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));

return result;
}

/** {@inheritDoc} */
@Override
public Class<ProjectManagerDto> getMainManagerDtoClass() {
    return ProjectManagerDto.class;
}

/** {@inheritDoc} */
@Override
public List<ProjectManagerDto> selectListIgnoreAuthCompanyWithLock(ProjectManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mProjectService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ProjectManagerDto.class,
        k30mProjectService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiled() {
    return k30mProjectService.getPkFiled();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ProjectManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicIF<ProjectManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mProjectService, criteria, order, logic);
}
}

```

ManagerDto

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectCriteriaDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterMaintenanceCriteriaDto;

/**
 * プロジェクトマスタ 検索条件DTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectCriteriaDto extends TermMasterMaintenanceCriteriaDto {

    /** 会社コード. */
    // public String companyCd; //親クラスで定義済み

    /** プロジェクトコード. */
    public String projectCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールID. */
    // public String localeId; //親クラスで定義済み
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** プロジェクト名称. */
    public String projectName;
    /** 削除フラグ. */
    // public String deleteFlag; //親クラスで定義済み
    /** プロジェクトコード(上位). */
    public String parentProjectCd;
    /** 予備項目A. */
    public String extensionA;
    /** 予備項目B. */
    public String extensionB;
    /** 予備項目C. */
    public String extensionC;
    /** 予備項目D. */
    public String extensionD;
    /** 予備項目E. */
    public String extensionE;
    /** 予備項目F. */
    public String extensionF;
    /** 予備項目G. */
    public String extensionG;
    /** 予備項目H. */
    public String extensionH;
    /** 予備項目I. */
    public String extensionI;
    /** 予備項目J. */
    public String extensionJ;
    /** 会社名. */
    public String companyName;
    /** プロジェクト名称(上位). */
    public String parentProjectName;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

/**
 * プロジェクトDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;

    /** プロジェクトコード. */
    public String projectCd;

    /** 期間コード. */
    public String termCd;

    /** ロケールID. */
    public String localeId;

    /** 開始日. */

```

```

public Timestamp startDate;

    /** 終了日. */
    public Timestamp endDate;

    /** プロジェクト名称. */
    public String projectName;

    /** 削除フラグ. */
    public String deleteFlag;

    /** 会社コード(上位). */
    public String parentCompanyCd;

    /** プロジェクトコード(上位). */
    public String parentProjectCd;

    /** 予備項目A. */
    public String extensionA;

    /** 予備項目B. */
    public String extensionB;

    /** 予備項目C. */
    public String extensionC;

    /** 予備項目D. */
    public String extensionD;

    /** 予備項目E. */
    public String extensionE;

    /** 予備項目F. */
    public String extensionF;

    /** 予備項目G. */
    public String extensionG;

    /** 予備項目H. */
    public String extensionH;

    /** 予備項目I. */
    public String extensionI;

    /** 予備項目J. */
    public String extensionJ;

    /** 登録日時. */
    public Timestamp entryTs;

    /** 登録者コード. */
    public String entryUserCd;

    /** 更新カウンタ. */
    public Long renewCnt;

    /** 更新日時. */
    public Timestamp renewTs;

    /** 更新者コード. */
    public String renewUserCd;

    /** 組織名称. */
    public String companyName;

    /** プロジェクト名称(上位). */
    public String parentProjectName;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityExtensionFieldNames;

/**
 * プロジェクトマスタDTO フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ProjectManagerDtoFieldName implements EntityExtensionFieldNames {

    /**
     * コンストラクタ.
     */
    private ProjectManagerDtoFieldName() {

    }

    /** 会社コード. */
    public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
    /** プロジェクトコード. */
    public static final String PROJECT_CD = "projectCd";
    /** 期間コード. */
    public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
    /** ロケールid. */
    public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
    /** プロジェクト名称. */
    public static final String PROJECT_NAME = "projectName";
    /** 開始日. */
    public static final String START_DATE = EntityCommonFieldNames.START_DATE;
    /** 終了日. */
    public static final String END_DATE = EntityCommonFieldNames.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;
    /** 会社コード(上位). */
    public static final String PARENT_COMPANY_CD = "parentCompanyCd";
    /** プロジェクトコード(上位). */
    public static final String PARENT_PROJECT_CD = "parentProjectCd";

    /** 会社名称(他マスタ名称). */
    public static final String COMPANY_NAME = "companyName";
    /** プロジェクト名称(上位). */
    public static final String PARENT_PROJECT_NAME = "parentProjectName";
}

```

Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProject.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * プロジェクト.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30m_project")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mProject extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** プロジェクトコード. */
    @Id
    @Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)

```

```

@Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String projectCd;

/** 期間コード. */
@Id
@Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String termCd;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** 開始日. */
@Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp endDate;

/** プロジェクト名. */
@Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
public String projectName;

/** 削除フラグ. */
@Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String deleteFlag;

/** 会社コード(上位プロジェクト). */
@Column(name = "parent_company_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String parentCompanyCd;

/** プロジェクトコード(上位プロジェクト). */
@Column(name = "parent_project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
public String parentProjectCd;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProjectNames.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;
```

```
import java.sql.Timestamp;
```



```

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30mProject} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mProjectNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30mProjectNames() {
        super();
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * startDateのプロパティ名を返却.
     *
     * @return startDateのプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDateのプロパティ名を返却.
     *
     * @return endDateのプロパティ名
     */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }

    /**
     * projectNameのプロパティ名を返却.
     *
     * @return projectNameのプロパティ名
     */
    public static PropertyName<String> projectName() {
        return new PropertyName<String>("projectName");
    }

    /**
     * deleteFlagのプロパティ名を返却.
     *
     * @return deleteFlagのプロパティ名
     */

```

```

    public static PropertyName<String> deleteFlag() {
        return new PropertyName<String>("deleteFlag");
    }

    /**
     * parentCompanyCdのプロパティ名を返却.
     *
     * @return parentCompanyCdのプロパティ名
     */
    public static PropertyName<String> parentCompanyCd() {
        return new PropertyName<String>("parentCompanyCd");
    }

    /**
     * parentProjectCdのプロパティ名を返却.
     *
     * @return parentProjectCdのプロパティ名
     */
    public static PropertyName<String> parentProjectCd() {
        return new PropertyName<String>("parentProjectCd");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**

```

```

    * extensionHのプロパティ名を返却.
    *
    * @return extensionHのプロパティ名
    */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public static PropertyName<Long> renewCnt() {
        return new PropertyName<Long>("renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public static PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>("renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public static PropertyName<String> renewUserCd() {
        return new PropertyName<String>("renewUserCd");
    }

    /**
     * @author S2JDBC-Gen
     */
    public static class _K30mProjectNames extends PropertyName<K30mProject> {

        /**
         * インスタンスを構築.
         */
        public _K30mProjectNames() {
        }

        /**
         * インスタンスを構築.
         *
         * @param name 名前

```

```

*/
public _K30mProjectNames(final String name) {
    super(name);
}

/**
 * インスタンスを構築.
 *
 * @param parent 親
 * @param name 名前
 */
public K30mProjectNames(final PropertyName<?> parent, final String name) {
    super(parent, name);
}

/**
 * companyCdのプロパティ名を返却.
 *
 * @return companyCdのプロパティ名
 */
public PropertyName<String> companyCd() {
    return new PropertyName<String>(this, "companyCd");
}

/**
 * projectCdのプロパティ名を返却.
 *
 * @return projectCdのプロパティ名
 */
public PropertyName<String> projectCd() {
    return new PropertyName<String>(this, "projectCd");
}

/**
 * termCdのプロパティ名を返却.
 *
 * @return termCdのプロパティ名
 */
public PropertyName<String> termCd() {
    return new PropertyName<String>(this, "termCd");
}

/**
 * localeIdのプロパティ名を返却.
 *
 * @return localeIdのプロパティ名
 */
public PropertyName<String> localeId() {
    return new PropertyName<String>(this, "localeId");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>(this, "deleteFlag");
}

```

```

/**
 * parentCompanyCdのプロパティ名を返却.
 *
 * @return parentCompanyCdのプロパティ名
 */
public PropertyName<String> parentCompanyCd() {
    return new PropertyName<String>(this, "parentCompanyCd");
}

/**
 * parentProjectCdのプロパティ名を返却.
 *
 * @return parentProjectCdのプロパティ名
 */
public PropertyName<String> parentProjectCd() {
    return new PropertyName<String>(this, "parentProjectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名

```

```

    */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却。
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却。
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却。
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却。
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却。
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却。
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却。
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mProject;

/**
 * {@link K30mProject}のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mProjectService extends GenerateService<K30mProject> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NullableStringCondition;

/**
 * {@link K30mProject}の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mProjectCondition extends
    AbstractEntityCondition<K30mProjectCondition> {

    /**
     * インスタンスを構築。
     */
    public K30mProjectCondition() {
    }

    /**
     * インスタンスを構築。
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30mProjectCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** companyCdの条件。 */
    public NotNullableViewCondition<K30mProjectCondition> companyCd =
        new NotNullableViewCondition<K30mProjectCondition>("companyCd", this);

    /** projectCdの条件。 */
    public NotNullableViewCondition<K30mProjectCondition> projectCd =
        new NotNullableViewCondition<K30mProjectCondition>("projectCd", this);

    /** termCdの条件。 */
    public NotNullableViewCondition<K30mProjectCondition> termCd =
        new NotNullableViewCondition<K30mProjectCondition>("termCd", this);

    /** localeIdの条件。 */
    public NotNullableViewCondition<K30mProjectCondition> localeId =
        new NotNullableViewCondition<K30mProjectCondition>("localeId", this);

    /** startDateの条件。 */
    public NotNullableViewCondition<K30mProjectCondition, Timestamp> startDate =
        new NotNullableViewCondition<K30mProjectCondition, Timestamp>("startDate", this);

    /** endDateの条件。 */
    public NotNullableViewCondition<K30mProjectCondition, Timestamp> endDate =
        new NotNullableViewCondition<K30mProjectCondition, Timestamp>("endDate", this);

    /** projectNameの条件。 */
    public NotNullableViewCondition<K30mProjectCondition> projectName =
        new NotNullableViewCondition<K30mProjectCondition>("projectName", this);
}

```

```
/** deleteFlagの条件. */
public NullableStringCondition<K30mProjectCondition> deleteFlag =
    new NullableStringCondition<K30mProjectCondition>("deleteFlag", this);

/** parentCompanyCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentCompanyCd =
    new NullableStringCondition<K30mProjectCondition>("parentCompanyCd", this);

/** parentProjectCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentProjectCd =
    new NullableStringCondition<K30mProjectCondition>("parentProjectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30mProjectCondition> extensionA =
    new NullableStringCondition<K30mProjectCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30mProjectCondition> extensionB =
    new NullableStringCondition<K30mProjectCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30mProjectCondition> extensionC =
    new NullableStringCondition<K30mProjectCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30mProjectCondition> extensionD =
    new NullableStringCondition<K30mProjectCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30mProjectCondition> extensionE =
    new NullableStringCondition<K30mProjectCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mProjectCondition> extensionF =
    new NullableStringCondition<K30mProjectCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30mProjectCondition> extensionG =
    new NullableStringCondition<K30mProjectCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30mProjectCondition> extensionH =
    new NullableStringCondition<K30mProjectCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30mProjectCondition> extensionI =
    new NullableStringCondition<K30mProjectCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30mProjectCondition> extensionJ =
    new NullableStringCondition<K30mProjectCondition>("extensionJ", this);

/** entryTsの条件. */
public NullableCondition<K30mProjectCondition, Timestamp> entryTs =
    new NullableCondition<K30mProjectCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NullableStringCondition<K30mProjectCondition> entryUserCd =
    new NullableStringCondition<K30mProjectCondition>("entryUserCd", this);

/** renewCntの条件. */
public NullableCondition<K30mProjectCondition, Long> renewCnt =
    new NullableCondition<K30mProjectCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NullableCondition<K30mProjectCondition, Timestamp> renewTs =
    new NullableCondition<K30mProjectCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NullableStringCondition<K30mProjectCondition> renewUserCd =
    new NullableStringCondition<K30mProjectCondition>("renewUserCd", this);
}
```

SQL

検索

- Path
src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService/selectView.sql

```
select
  prj.company_cd
, prj.project_cd
, prj.term_cd
, prj.local_id
```



```

, prj.locale_id
, prj.start_date
, prj.end_date
, prj.project_name
, prj.delete_flag
, prj.parent_company_cd
, prj.parent_project_cd
, prj.extension_a
, prj.extension_b
, prj.extension_c
, prj.extension_d
, prj.extension_e
, prj.extension_f
, prj.extension_g
, prj.extension_h
, prj.extension_i
, prj.extension_j
, prj.entry_ts
, prj.entry_user_cd
, prj.renew_cnt
, prj.renew_ts
, prj.renew_user_cd
, dept.department_name as company_name
, parent.project_name as parent_project_name
from k30m_project prj
left join ( select imm_department.company_cd
, imm_department.department_set_cd
, imm_department.department_cd
, imm_department.locale_id
, imm_department.start_date
, imm_department.end_date
, imm_department.department_name
from imm_department) dept
on prj.company_cd = dept.company_cd
and prj.company_cd = dept.department_set_cd
and prj.company_cd = dept.department_cd
and prj.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and dept.start_date <= prj.start_date
--ELSE and dept.end_date > prj.start_date
/*END*/
left join k30m_project parent
on prj.parent_company_cd = parent.company_cd
and prj.parent_project_cd = parent.project_cd
and prj.locale_id = parent.locale_id
/*IF searchCriteriaDate != null*/
and parent.start_date <= /*searchCriteriaDate*/2000/01/01'
and parent.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and parent.start_date <= prj.start_date
--ELSE and parent.end_date > prj.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
prj.company_cd in /*companyCd*/('comp')
/*END*/
/*IF projectCd != null*/
and prj.project_cd = /*projectCd*/projectCd'
/*END*/
/*IF termCd != null*/
and prj.term_cd = /*termCd*/termCd'
/*END*/
/*IF localeId != null*/
and prj.locale_id = /*localeId*/localeId'
/*END*/
/*IF searchCriteriaDate != null*/
and prj.start_date <= /*searchCriteriaDate*/1900/01/01'
and prj.end_date > /*searchCriteriaDate*/2999/12/31'
/*END*/
/*IF projectName != null*/
and prj.project_name like /*projectName*/'% ' ESCAPE '$'
/*END*/
/*IF deleteFlag != null*/
and prj.delete_flag = /*deleteFlag*/0'
/*END*/
/*IF parentCompanyCd != null*/
and prj.parent_company_cd = /*parentCompanyCd*/parentCompanyCd'
/*END*/
/*IF parentProjectCd != null*/
and prj.parent_project_cd = /*parentProjectCd*/parentProjectCd'
/*END*/
/*IF extensionA != null*/
and prj.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/

```

```

/* IF extensionB != null */
and prj.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and prj.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and prj.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and prj.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and prj.extension_f like /*extensionF*/extensionF' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
and prj.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
and prj.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
and prj.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
and prj.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF parentProjectName != null*/
and parent.project_name like /*parentProjectName*/parentProjectName' ESCAPE '$'
/*END*/
/*END*/
/*IF orderBy != null*/
ORDER BY /*$orderBy*/company_cd
/*END*/

```

XML

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ProjectMasterMaintenanceSettings.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericMasterMaintenanceSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterMaintenanceSettings">
  <!-- 画面タイトルラベルID -->
  <titleSurfaceKey>PWKF.master.project.title</titleSurfaceKey>
  <!-- 検索画面JSPフルパス -->
  <searchGadgetJsp></searchGadgetJsp><!-- デフォルトを利用 -->
  <!-- メンテナンス画面JSPフルパス -->
  <maintenanceGadgetJsp>/WEB-INF/view/kaiden2/product_workflow/master/project/projectMainte.jsp</maintenanceGadgetJsp>
  <!-- AjaxAccessUrl -->
  <ajaxAccessUrl>kaiden/master/project</ajaxAccessUrl>
  <!-- 1ページあたりの件数(0以下はページングせずすべて表示) -->
  <limit>20</limit>
  <!-- 検索条件の検索基準日 使用/不使用 -->
  <useSearchCriteriaDate>true</useSearchCriteriaDate>
  <!-- 検索条件の検索基準日 必須/任意 -->
  <requiredSearchCriteriaDate>true</requiredSearchCriteriaDate>
  <!-- 検索条件の検索基準日 デフォルト値(システム日付) 設定/非設定(必須の場合にはこの設定にかかわらずデフォルト値が設定される) -->
  <defaultSearchCriteriaDate>true</defaultSearchCriteriaDate>
  <!-- 検索条件の会社 使用/不使用 -->
  <useSearchCompany>true</useSearchCompany>
  <!-- 検索条件のロケール 使用/不使用 -->
  <useSearchLocale>true</useSearchLocale>
  <!-- 検索条件のステータス無効データも検索。使用/不使用 -->
  <useSearchWithDisabled>true</useSearchWithDisabled>
  <!-- 汎用一覧・検索画面の行番号ラベルID -->
  <rownumberSurfaceKey></rownumberSurfaceKey><!-- デフォルトを利用 -->
  <!-- 汎用一覧・検索画面のカラム設定 -->
  <listCols>
    <listCol>
      <!-- 対象マスタのフィールドキー -->
      <fieldKey>projectCd</fieldKey>
      <!-- キーワード検索対象とするか否か -->
      <keywordSearchTarget>true</keywordSearchTarget>
      <!-- 表示幅 -->
      <width>150px</width>
      <!-- 表示位置 -->
      <align></align>
      <!-- ヘッダラベルID -->
      <headerSurfaceKey>PWKF.projectCd</headerSurfaceKey>
      <!-- ソート可能か否か -->
      <sortable>true</sortable>
    </listCol>
  </listCols>

```

```

<listCol>
  <fieldKey>projectName</fieldKey>
  <keywordSearchTarget>true</keywordSearchTarget>
  <width></width>
  <align></align>
  <headerSurfaceKey>PWKF.projectName</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>parentProjectName</fieldKey>
  <keywordSearchTarget>true</keywordSearchTarget>
  <width></width>
  <align></align>
  <headerSurfaceKey>PWKF.master.project.parentProjectName</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>deleteFlag</fieldKey>
  <keywordSearchTarget>false</keywordSearchTarget>
  <width>100px</width>
  <align>center</align>
  <headerSurfaceKey>BASE.status</headerSurfaceKey>
  <sortable>false</sortable>
</listCol>
</listCols>
<!-- 検索・一覧画面のソート順（汎用マスター一覧は、第二ソートキーまで有効）-->
<orders>
  <order>
    <fieldKey>projectCd</fieldKey>
    <type>asc</type>
  </order>
  <order>
    <fieldKey>projectName</fieldKey>
    <type>asc</type>
  </order>
</orders>
<!-- 対象マスタの主キーフィールド -->
<pkFields>
  <pkField>companyCd</pkField>
  <pkField>projectCd</pkField>
  <pkField>termCd</pkField>
  <pkField>localeId</pkField>
</pkFields>
</genericMasterMaintenanceSettings>

```

JSP

■ Path

```
src/main/webapp/WEB-INF/view/kaiden2/product_workflow/master/project/projectMainte.jsp
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_base" uri="http://kaiden.slcs.co.jp/taglib/listdata/base" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.util.PropertyJspUtil" %>

<!-- product_workflow 共通インクルード --%>
<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- プロジェクトマスタメンテナンス --%>
<!-- 設定 --%>
<c:set var="gadgetClass">projectMainte</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="gadgetTitle">${maintenanceDto.settings.title}</c:set>
<c:set var="accessUri">${maintenanceDto.settings.ajaxAccessUri}</c:set>

<c:set var="surfaceCompany"><%= BaseProp.S.COMPANY %></c:set>
<c:set var="surfaceProjectCd"><%= ProductWorkflowProp.S.PROJECT_CD %></c:set>
<c:set var="surfaceProjectName"><%= ProductWorkflowProp.S.PROJECT_NAME %></c:set>
<c:set var="surfaceStartDate"><%= BaseProp.S.START_DATE %></c:set>
<c:set var="surfaceConcatChar"><%= BaseProp.S.CONCAT_CHAR %></c:set>
<c:set var="surfaceEndDate"><%= BaseProp.S.END_DATE %></c:set>
<c:set var="surfaceParentProject"><%= ProductWorkflowProp.S.Project.PARENT_PROJECT %></c:set>

<!-- パリテーション --%>
<c:set var="validStartDate_def">{k_required:[], k_date:[]}</c:set>

```

```

<c:set var="validEndDate_def">{k_required:[''], k_date:[''], k_dateMin:['startDate']}

```

```

<td>
  <k:text-locales name="projectName"
    mode="0"
    style="width:300px;"
    maxlength="250"
    surfaceKey="${surfaceProjectName}"
    validationDefault="${validProjectName_def}"
    imeMode="on"
  />
</td>
</tr>
<tr>
<th>
  <k:surface surfaceKey="${surfaceParentProject}" />
</th>
<td>
  <kmsw:project
    name="parentProjectName"
    mode="0"
    resultChain="{parentCompanyCd:'companyCd', parentProjectCd:'projectCd}"
    searchCriteriaProjectCd="parentProjectCd"
    searchCriteriaCompany="companyCd"
    primaryTarget="parentCompanyCd,parentProjectCd"
    resultValue="projectName"
    includeDisabled="true"
    style="width:300px;"
    imeMode="on"
    surfaceKey="${surfaceParentProject}"
    validationDefault="${validParentProject_def}"
  />
  <k:hidden name="parentCompanyCd" mode="0"/>
  <k:hidden name="parentProjectCd" mode="0"/>
  <%--隠し項目 --%>
  <k:hidden name="renewCnt" mode="0"/>
  <k:hidden name="termCd" mode="0"/>
  <k:hidden name="deleteFlag" mode="0"/>
  <k:hidden name="pk_json" mode="0"/>

</td>
</tr>
</table>
</form>
</k:blockContainer>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {

var eventMan = KAIDEN.getGadget("${gadgetId}").eventManager
, inputAreaAccessor = KAIDEN.getGadget("${gadgetId}").blockAccessors["inputArea"];

//trunsNewEntry (新規入力画面表示) イベントリスナの登録
eventMan.registPushListener("trunsNewEntry", function(**Object*/arg) {
  //入力欄を一括クリア
  KAIDEN.getGadget("${gadgetId}").clearInput("inputArea");

  //開始日・終了日にシステム開始日・システム終了日を設定
  inputAreaAccessor("startDate").setValue("${k:getSystemStartDate()}");
  inputAreaAccessor("endDate").setValue("${k:getSystemEndDateDecrement()}");

  //開始日・終了日、会社、プロジェクトコードをdisabled
  inputAreaAccessor("startDate").removeDisabled();
  inputAreaAccessor("endDate").removeDisabled();
  inputAreaAccessor("companyCd").removeDisabled();
  inputAreaAccessor("projectCd").removeDisabled();

  if (inputAreaAccessor("companyCd").elem().children().length > 0) {
    inputAreaAccessor("companyCd").elem().get(0).selectedIndex = 0;
  }
  KAIDEN.MasterSearch.refreshAll(true);

  inputAreaAccessor("companyCd").elem().focus();

});

//loadData (表示対象データのロード) イベントリスナの登録
eventMan.registPushListener("loadData", function(**Object*/arg) {
  var ajax = KAIDEN.Ajax()
  , criteria = {}
  , stat = false;

  criteria.criteriaJson = arg;

  ajax.postSyncAuto("${accessUrl}/select", criteria
    , function(returnData) {
      //正常系
      //メンテナンス画面にデータセット

```

```

        for (var key in returnData.mainteTarget) {
            inputAreaAccessor(key).setValueIgnore(returnData.mainteTarget[key]);
        }
        inputAreaAccessor("pk_json").setValue(arg); //主キーを保存
        KAIDEN.MasterSearch.refreshAll(true);
        stat = true;
    }
    , function(returnData) {
        stat = false;
    }
    );
    return stat;
});

//trunsRenewEntry (更新画面表示) イベントリスナの登録
eventMan.registPushListener("trunsRenewEntry", function(**Object*/arg) {
    //開始日・終了日、会社、プロジェクトコードをdisabled
    inputAreaAccessor("startDate").disabled();
    inputAreaAccessor("endDate").disabled();
    inputAreaAccessor("companyCd").disabled();
    inputAreaAccessor("projectCd").disabled();

    $("div[data-kaiden-gadget-id=${gadgetId}] input[data-kaiden-org-name^=projectName]").eq(0).focus();

});

//currentPrimaryInfo (現在表示中データの主キーを要求) イベントリスナの登録
eventMan.registPullListener("claimCurrentDataPk", function(**Object*/arg) {
    return inputAreaAccessor("pk_json").getValue(); //主キーを返却
});

/**
 * 自身のプロジェクトと上位プロジェクトの同一チェック.
 * <p>自身のプロジェクトと上位プロジェクトの同一の場合エラー</p>
 * @param value 値
 * @param element 対象エレメント
 * @param param パラメータ (配列)
 */
KAIDEN.validator.addValidateMethod("k_notEqualParent", function(**String*/value, /**HtmlElement*/element, /**array*/param) {
    var na = KAIDEN.gadgetMan.getNeighboringAccessor
        , thisCompanyCd = na(element, "companyCd").getValue()
        , thisProjectCd = na(element, "projectCd").getValue()
        , parentCompanyCd = na(element, "parentCompanyCd").getValue()
        , parentProjectCd = na(element, "parentProjectCd").getValue();

    return !(thisCompanyCd === parentCompanyCd && thisProjectCd === parentProjectCd);

});

KAIDEN.validator.setInvalidMessage("k_notEqualParent", function(arg, element) {
    var msgParam = [];
    return KAIDEN.validator.formatMessage("${msgNotEqualParent}", element, msgParam);
});

inputAreaAccessor("companyCd").elem().on("change", function() {
    KAIDEN.MasterSearch.clear(inputAreaAccessor("parentProjectName").elem());
});

});
</script>
<!-- マスタメンテナンス汎用ボタンガジェット --%>
<div style="clear:both;">
<jsp:include page="/WEB-INF/view/kaiden2/base/foundation/gadget/genericMasterOperationButton.jsp" flush="true">
<jsp:param name="gadget_instance" value="0" />
<jsp:param name="maintenanceGadgetId" value="${gadgetId}" />
</jsp:include>
</div>
</div>

```

経費区分マスタメンテナンス

経費区分マスタメンテナンスのサンプルソースです。

本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Action](#)
 - [Dto](#)
 - [Form](#)
 - [Logic](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
- [XML](#)
- [JSP](#)

[一覧](#)

```

src
├── generated
│   ├── jp
│   │   ├── co
│   │   │   ├── slcs
│   │   │   │   ├── kaiden2
│   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   ├── K30mExpType.java
│   │   │   │   │   │   │   │   │   ├── K30mExpTypeNames.java
│   │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   │       ├── K30mExpTypeCondition.java
│   │   │   │   │   │   │   │   │       └── K30mExpTypeService.java
│   ├── java
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   └── ExpTypeAction.java
│   │   │   │   │   │   │   │   ├── dto
│   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   └── ExpTypeDto.java
│   │   │   │   │   │   │   │   ├── form
│   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   └── ExpTypeForm.java
│   │   │   │   │   │   │   │   ├── logic
│   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   └── ExpTypeLogic.java
│   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   ├── manager
│   │   │   │   │   │   │   │   │   ├── exptype
│   │   │   │   │   │   │   │   │   │   ├── ExpTypeManager.java
│   │   │   │   │   │   │   │   │   │   ├── ExpTypeManagerDto.java
│   │   │   │   │   │   │   │   │   │   ├── ExpTypeManagerDtoFieldName.java
│   │   │   │   │   │   │   │   │   │   └── impl
│   │   │   │   │   │   │   │   │   │   └── ExpTypeManagerImpl.java
│   ├── resources
│   │   ├── META-INF
│   │   │   ├── sql
│   │   │   │   ├── jp
│   │   │   │   │   ├── co
│   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   │   │   │   └── K30mExpTypeService
│   │   │   │   │   │   │   │   │   │   └── selectView.sql
│   ├── webapp
│   │   ├── WEB-INF
│   │   │   ├── classes
│   │   │   │   ├── jp
│   │   │   │   │   ├── co
│   │   │   │   │   │   ├── slcs
│   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   ├── feature
│   │   │   │   │   │   │   │   │   │   ├── action
│   │   │   │   │   │   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   └── ExpTypeMasterMaintenanceSettings.xml
│   │   │   │   │   │   │   │   │   │   └── view
│   │   │   │   │   │   │   │   │   │   │   ├── kaiden2
│   │   │   │   │   │   │   │   │   │   │   │   ├── product_workflow
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   │   │   │   │   ├── expType
│   │   │   │   │   │   │   │   │   │   │   │   │   └── expTypeMainte.jsp

```


■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ExpTypeAction.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master;

import javax.annotation.Resource;
import jp.co.intra_mart.foundation.authz.annotation.Authz;
import jp.co.slcs.kaiden2.base.foundation.conf.ModuleEnum;
import jp.co.slcs.kaiden2.base.foundation.conf.ResultType;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceDto;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceAction;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.base.foundation.model.annotation.Module;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master.ExpTypeDto;
import jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master.ExpTypeForm;
import jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.master.ExpTypeLogic;
import org.seasar.struts.annotation.ActionForm;
import org.seasar.struts.annotation.Execute;

/**
 * 経費区分マスタメンテナンス (Action).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeAction extends AbstractTermMasterMaintenanceAction {

    /** ExpTypeDto. */
    @Resource
    public ExpTypeDto expTypeDto;

    /** ExpTypeForm. */
    @ActionForm
    @Resource
    public ExpTypeForm expTypeForm;

    /** ExpTypeLogic. */
    @Resource
    public ExpTypeLogic expTypeLogic;

    /**
     * 入力ページのパスを返却.
     *
     * @return 入力ページのパス
     * @throws KaidenSystemException XML読み込み例外
     */
    @Override
    @Execute(validator = false)
    @Authz(uri = "service://kaiden/master/expType", action = "execute")
    public String index() throws KaidenSystemException {

        // プロパティファイルから一覧画面の設定を読み込み
        MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
        expTypeDto.settings = new MasterMaintenanceSettingsDto(reader.read(ExpTypeAction.class));

        // 一覧画面の設定をレスポンスに設定
        return super.index();
    }

    /**
     * AJAXにて検索を実行.
     * <p>
     * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
     * </p>
     *
     * @return null
     */
    @Override
    @Execute(validator = false)
    @Authz(uri = "service://kaiden/master/expType", action = "execute")
    public String search() {
        return super.search();
    }

    /**
     * AJAXにてメンテナンス対象データを返却.
     *
     * @return メンテナンス対象データ
     */
    @Override
    @Execute(validator = false)
    @Authz(uri = "service://kaiden/master/expType", action = "execute")

```

```

public String select() {
    return super.select();
}

/**
 * AJAXにてデータを登録.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String entry() {
    return super.entry();
}

/**
 * AJAXにて無効化.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String toDisabled() {
    return super.toDisabled();
}

/**
 * AJAXにてデータ削除.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String remove() {
    return super.remove();
}

/**
 * AJAXにてデータ更新.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String renew() {
    return super.renew();
}

/**
 * AJAXにてメンテナンス対象データの期間データを返却.
 *
 * @return メンテナンス対象データ
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String selectTerm() {
    return super.selectTerm();
}

/**
 * AJAXにて有効化.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false

```

```

@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String toEnabled() {
    return super.toEnabled();
}

/**
 * AJAXにて期間分割.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String divideTerm() {
    return super.divideTerm();
}

/**
 * AJAXにて期間変更.
 * <p>
 * レスポンスに、JSON化した、ロジッククラスの戻り値{@link LogicResultDto}を出力
 * </p>
 *
 * @return null
 */
@Override
@Execute validator = false
@Authz(uri = "service://kaiden/master/expType", action = "execute")
public String moveTerm() {
    return super.moveTerm();
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceDto getDto() {
    return expTypeDto;
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceForm getForm() {
    return expTypeForm;
}

/**
 * {@inheritDoc}
 */
@Override
protected AbstractMasterMaintenanceLogic getLogic() {
    return expTypeLogic;
}
}

```

Dto

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/dto/kaiden/master/ExpTypeDto.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.dto.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceDto;

/**
 * 経費区分マスタメンテナンス (DTO).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeDto extends AbstractTermMasterMaintenanceDto {
}

```

Form

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/form/kaiden/master/ExpTypeForm.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.form.kaiden.master;

import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm;

/**
 * 経費区分マスタメンテナンス (Form).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeForm extends AbstractTermMasterMaintenanceForm {

}

```

Logic

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/logic/kaiden/master/ExpTypeLogic.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.logic.kaiden.master;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.SORT_KEY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_CD;
import java.sql.Timestamp;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Stack;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceForm;
import jp.co.slcs.kaiden2.base.foundation.master.AbstractTermMasterMaintenanceLogic;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsDto;
import jp.co.slcs.kaiden2.base.foundation.master.MasterMaintenanceSettingsReader;
import jp.co.slcs.kaiden2.base.foundation.master.MasterManager;
import jp.co.slcs.kaiden2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden2.base.foundation.util.ConvertForJsonSource;
import jp.co.slcs.kaiden2.base.foundation.util.JsonUtil;
import jp.co.slcs.kaiden2.product_workflow.feature.action.kaiden.master.ExpTypeAction;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;

/**
 * 経費区分マスタメンテナンス (Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeLogic extends AbstractTermMasterMaintenanceLogic {

    /** ExpTypeManager. */
    @Resource
    public ExpTypeManager expTypeManager;

    /**
     * {@inheritDoc}
     */
    @Override
    public LogicResultDto selectTermAjax(AbstractTermMasterMaintenanceForm form) {
        Convert conv = new Convert();
        return super.selectTermAjax(form, conv);
    }

    /**
     * 経費区分メンテナンス画面へデータを渡す際の書式変換.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    private static class Convert extends ConvertForJsonSource {

        /**
         * {@inheritDoc}
         *
         * <p>
         * 本サブクラスでは、以下の書式変換を行う。<br>
         * ・・・
         */
    }

```

```

* <table border="1">
* <tr>
* <td>項目名</td>
* <td>書式</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>システムの日付時刻形式</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>システムの日付時刻形式</td>
* </tr>
* </table>
* </p>
*/
@Override
protected Map<String, String> formatFormattedValue(Map<String, Object> map, Stack<String> fldNameStack) {
    // 標準書式変換
    Map<String, String> result = super.formatFormattedValue(map, fldNameStack);

    // 開始日の書式変換
    Object startDate = map.get(START_DATE);
    if (startDate != null) {
        result.put(START_DATE, dateHelper.formatDateSystemTz((Timestamp) startDate));
    }

    // 終了日の書式変換
    Object endDate = map.get(END_DATE);
    if (endDate != null) {
        result.put(END_DATE, dateHelper.formatDateSystemTz((Timestamp) endDate));
    }

    return result;
}

}

/**
 * {@inheritDoc}
 */
@Override
protected MasterManager<?> getManager() {
    return expTypeManager;
}

/**
 * {@inheritDoc}
 */
@Override
protected String[] getKeywordSearchTargetCol() {
    // プロパティファイルから一覧画面の設定を読み込み
    MasterMaintenanceSettingsReader reader = new MasterMaintenanceSettingsReader();
    MasterMaintenanceSettingsDto settings;
    try {
        settings = new MasterMaintenanceSettingsDto(reader.read(ExpTypeAction.class));
    } catch (KaidenSystemException e) {
        throw KaidenRuntimeException.makeInstance(e);
    }

    return settings.searchTargetCol.toArray(new String[settings.searchTargetCol.size()]);
}

/**
 * {@inheritDoc}
 */
@Override
protected List<String> getI18nTargetField() {
    return Arrays.asList(EXP_TYPE_NAME);
}

/**
 * {@inheritDoc}
 */
@Override
protected Map<String, Object> getInputData(AbstractMasterMaintenanceForm form) {
    Map<String, Object> map = JsonUtil.getMap(form.updateDataJson(), Arrays.asList("inputArea"));

    return map;
}

/**
 * {@inheritDoc}
 * <p>
 * ■ 入力項目名... 書式...

```

```

    ■設定値一覧<DT>
    * <table border="1">
    * <tr>
    * <td>設定キー</td>
    * <td>設定値</td>
    * </tr>
    * <tr>
    * <td>companyCd</td>
    * <td>会社</td>
    * </tr>
    * <tr>
    * <td>expTypeCd</td>
    * <td>経費区分コード</td>
    * </tr>
    * <tr>
    * <td>startDate</td>
    * <td>開始日</td>
    * </tr>
    * <tr>
    * <td>endDate</td>
    * <td>終了日</td>
    * </tr>
    * <tr>
    * <td>expTypeName</td>
    * <td>経費区分名</td>
    * </tr>
    * <tr>
    * <td>expTypeCategory</td>
    * <td>経費区分カテゴリ</td>
    * </tr>
    * <tr>
    * <td>accountCd</td>
    * <td>勘定科目</td>
    * </tr>
    * <tr>
    * <td>taxTypeCd</td>
    * <td>税区分</td>
    * </tr>
    * <tr>
    * <td>sortKey</td>
    * <td>ソートキー</td>
    * </tr>
    * <tr>
    * <td>deleteFlag</td>
    * <td>ステータス</td>
    * </tr>
    * </table>
    * </p>
    */
    @Override
    protected Map<String, String> getSurfaces() {

        Map<String, String> result = new HashMap<String, String>();

        result.put(COMPANY_CD, propertyHelper.getSurface(BaseProp.S.COMPANY));
        result.put(EXP_TYPE_CD, propertyHelper.getSurface(ProductWorkflowProp.S.EXP_TYPE_CD));
        result.put(START_DATE, propertyHelper.getSurface(BaseProp.S.START_DATE));
        result.put(END_DATE, propertyHelper.getSurface(BaseProp.S.END_DATE));
        result.put(EXP_TYPE_NAME, propertyHelper.getSurface(ProductWorkflowProp.S.EXP_TYPE_NAME));
        result.put(EXP_TYPE_CATEGORY, propertyHelper.getSurface(ProductWorkflowProp.S.ExpType.EXP_TYPE_CATEGORY));
        result.put(ACCOUNT_CD, propertyHelper.getSurface(ProductWorkflowProp.S.ACCOUNT));
        result.put(TAX_TYPE_CD, propertyHelper.getSurface(ProductWorkflowProp.S.TAX_TYPE));
        result.put(SORT_KEY, propertyHelper.getSurface(BaseProp.S.SORT_KEY));
        result.put(DELETE_FLAG, propertyHelper.getSurface(BaseProp.S.STATUS));

        return result;
    }
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;

/**
 * 経費区分マスタへの各種操作を提供するAPIのinterface.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ExpTypeManager extends TermMasterManagerForJob<ExpTypeManagerDto> {

    /** 経費区分コード. */
    public static final int EXP_TYPE_CD_LEN = 100;

    /** 経費区分名称. */
    public static final int EXP_TYPE_NAME_LEN = 250;

    /** 経費区分カテゴリ. */
    public static final int EXP_TYPE_CATEGORY_LEN = 250;

    /** 勘定科目コード. */
    public static final int ACCOUNT_CD_LEN = 100;

    /** 税区分コード. */
    public static final int TAX_TYPE_CD_LEN = 100;

    /** ソートキー. */
    public static final int SORT_KEY_LEN = 15;

    /**
     * 新規登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

    /**
     * 更新登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

    /**
     * マスタ検索部品向け 一覧取得.
     *
     * @param criteria 検索条件
     * @param order ソート順
     * @param filter フィルタークラス
     * @return 検索結果
     */
    public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter);
}

```

Manager(実装)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/impl/ExpTypeManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;

```

```

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_INTEGER;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.SORT_KEY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mExpTypeService;
import org.seasar.framework.beans.util.Beatns;

/**
 * 経費区分マスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 使用する際には、{@link ExpTypeManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ExpTypeManagerDto}クラスまたはそのリストが戻り値として返却されます。<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <th>機能概要</th>
 * <th>メソッド</th>
 * <th>カテゴリ</th>
 * <th>備考</th>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>select</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>主キー検索(認可設定無視)</td>
 * <td>selectIgnoreAuthCompany</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>汎用検索</td>
 * <td>selectAll</td>
 * <td>汎用検索</td>
 * <td>&nbsp;</td>
 * </tr>

```



```

<td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索(認可設定無視)</td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索(認可設定無視)</td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索(認可設定無視)</td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索(認可設定無視)</td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化(認可設定無視)</td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化(認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録(認可設定無視)</td>
* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>

```

```

</td><td></td>
* </tr>
* <tr>
* <td>削除 (認可設定無視)</td>
* <td>removeIgnoreAuthCompany</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新</td>
* <td>renew</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新 (認可設定無視)</td>
* <td>renewIgnoreAuthCompany</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <b>機能一覧(期間化)</b>
* <table border="1">
* <tr>
* <th>機能概要</th>
* <th>メソッド</th>
* <th>カテゴリ</th>
* <th>備考</th>
* </tr>
* <tr>
* <td>期間検索</td>
* <td>selectTerms</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間検索 (認可設定無視)</td>
* <td>selectTermsIgnoreAuthCompany</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索</td>
* <td>selectListIncludesSearchDate</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索 (認可設定無視)</td>
* <td>selectListIncludesSearchDateIgnoreAuthCompany</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割</td>
* <td>divideTerm</td>
* <td>期間分割</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割 (認可設定無視)</td>
* <td>divideTermIgnoreAuthCompany</td>
* <td>期間分割</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間変更</td>
* <td>moveTerm</td>
* <td>期間変更</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間変更 (認可設定無視)</td>
* <td>moveTermIgnoreAuthCompany</td>
* <td>期間変更</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* 機能に対するI/Oを以下に示します。(上記一覧のカテゴリ単位)<br>
* <br>
* [IN]
* <table border="1">
* <tr>
* <td align="right"></td>

```

```
* <td align="right">△</td>
* </tr>
* <tr>
* <td align="right">△</td>
* <td>: 任意、検索時完全一致条件</td>
* </tr>
* <tr>
* <td align="right">▲</td>
* <td>: 任意、検索時部分一致条件</td>
* </tr>
* <tr>
* <td align="right">×</td>
* <td>: 不要(設定されていても無視します。)</td>
* </tr>
* <tr>
* <td align="right">☆</td>
* <td>: 各APIの説明を参照</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* [OUT]
* <table border="1">
* <tr>
* <td align="right">○</td>
* <td>: あり</td>
* </tr>
* <tr>
* <td align="right">×</td>
* <td>: なし</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(共通)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">主キー<br>
* 検索</th>
* <th colspan="2">汎用<br>
* 検索</th>
* <th colspan="2">ロケール<br>
* 検索</th>
* <th colspan="2">参照<br>
* 検索</th>
* <th colspan="2">キーワード<br>
* 検索</th>
* <th colspan="2">削除フラグ<br>
* 更新</th>
* <th colspan="2">登録</th>
* <th colspan="2">削除</th>
* <th colspan="2">更新</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>税区分名称</td>
* <td>tax TypeName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">期間検索</th>
* <th colspan="2">基準日指定<br>
* 検索</th>
* <th colspan="2">期間分割</th>
* <th colspan="2">期間変更</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>経費区分コード(*)</b></td>
```

```

* <td><b>expTypeCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分名称</td>
* <td>expTypeName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分カテゴリ</td>
* <td>expTypeCategory</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>startDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>endDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>accountCd</td>
* <td align="center">x</td>
* <td align="center">○</td>

```

```

<td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>taxTypeCd</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>ソートキー</td>
* <td>sortKey</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">〇</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>
* <td>renewCnt</td>
* <td align="center">×</td>
* <td align="center">〇</td>
* <td align="center">△</td>
* <td align="center">〇</td>

```

```

* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新日時</td>
* <td>renewTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <tr>
* <td>更新者コード</td>
* <td>renewUserCode</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>会社名称</td>
* <td>companyName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>勘定科目名称</td>
* <td>accountName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>税区分名称</td>
* <td>taxTypeName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* テーブルのPK項目にはフィールド名に(*)を付与しています。<br>
* </p>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public class ExpTypeManagerImpl extends AbstractTermMasterManager<ExpTypeManagerDto> implements ExpTypeManager {

/** K30mExpTvdeService. */

```

```

@Resource
protected K30mExpTypeService k30mExpTypeService;

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto select(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.select(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto selectIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectList(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria,
    List<OrderDto> order) {
    return super.selectList(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompany(ExpTypeManagerDto criteria, List<OrderDto> order) {
    return super.selectListIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocales(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocalesIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
    AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySql(k30mExpTypeService, "selectView.sql", criteria, order, authCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySqlIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectListByKeywords(k30mExpTypeService, "selectView.sql", criteria, order, keywordsCriteria,
        authCriteria, likeTarget);
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectListByKeywordsIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order,
        keywordsCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public void toEnabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toEnabled(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toEnabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
    super.toEnabledIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toDisabled(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
    super.toDisabledIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
    return super.entry(k30mExpTypeService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entryIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenDuplicationEntryException {
    return super.entryIgnoreAuthCompany(k30mExpTypeService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException {
    return super.remove(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int removeIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.removeIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}

```

```

*/
@Override
public int renew(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    return super.renew(k30mExpTypeService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int renewIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenOptimisticLockException {
    return super.renewIgnoreAuthCompany(k30mExpTypeService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTerms(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTermsIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDate(ExpTypeManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ExpTypeManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
    super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
    String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
    KaidenValidateException {
    String termCd2nd = super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabled(criteria2nd, authCriteria);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabled(criteria2nd, authCriteria);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate, String deleteFlag2nd)
    throws KaidenOptimisticLockException, KaidenValidateException {
    String termCd2nd = super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabledIgnoreAuthCompany(criteria2nd);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabledIgnoreAuthCompany(criteria2nd);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mExpTypeService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mExpTypeService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * 重複チェック確認フィールドを返却。
 * <p>
 * 新規登録時、登録済みデータと引数の登録データの重複チェックを行うフィールドを返します。
 * </p>
 *
 * @return 重複チェック確認フィールドリスト
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, EXP_TYPE_CD});
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override

```



```

public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecutor(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * <tr>
 * <td>削除フラグ</td>
 * <td>フラグ妥当性</td>
 * </tr>
 * </table>
 */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
    // ロケールID
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 経費区分名称
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
    // 経費区分カテゴリ
    result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日

```

```

result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));
// 削除フラグ
result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、50桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * </table>
 */

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_COMPANY_EXISTS, COMPANY_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));

```

```

result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
// 期間コード
result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
// ロケールID
result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
// 経費区分名称
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
// 開始日
result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
// 終了日
result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));

return result;
}

/** {@inheritDoc} */
@Override
public Class<ExpTypeManagerDto> getMainManagerDtoClass() {
    return ExpTypeManagerDto.class;
}

@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompanyWithLock(ExpTypeManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mExpTypeService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ExpTypeManagerDto.class,
        k30mExpTypeService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiled() {
    return k30mExpTypeService.getPkFiled();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ExpTypeManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicF<ExpTypeManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mExpTypeService, criteria, order, logic);
}

@Override
public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    Map<String, Object> resultMap = super.selectListByKeywordsFilter(k30mExpTypeService, "selectView.sql",
        criteria, order, keywordsCriteria, likeTarget, filter);
    @SuppressWarnings("unchecked")
    List<ExpTypeManagerDto> newList = (List<ExpTypeManagerDto>) resultMap.get("resultList");
    resultMap.put("resultList", newList);

    return resultMap;
}
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO.
 *
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールid. */
    public String localeId;
    /** 経費区分名称. */
    public String expTypeName;
    /** 経費区分カテゴリ. */
    public String expTypeCategory;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** ソートキー. */
    public Long sortKey;
    /** 削除フラグ. */
    public String deleteFlag;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 会社名称(他マスタ名称). */
    public String companyName;
    /** 勘定科目名称(他マスタ名称). */
    public String accountName;
    /** 税区分名称(他マスタ名称). */
    public String taxTypeName;
    /** 税率(他マスタ). */
    public BigDecimal taxRate;
}

```

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO フィールド名定義クラス。
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpTypeManagerDtoFieldName extends EntityCommonFieldNames {

    /**
     * コンストラクタ。
     */
    private ExpTypeManagerDtoFieldName() {

    }

    /** 会社コード。 */
    public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
    /** 経費区分コード。 */
    public static final String EXP_TYPE_CD = "expTypeCd";
    /** 期間コード。 */
    public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
    /** ロケールid。 */
    public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
    /** 経費区分名称。 */
    public static final String EXP_TYPE_NAME = "expTypeName";
    /** 経費区分カテゴリ。 */
    public static final String EXP_TYPE_CATEGORY = "expTypeCategory";
    /** 開始日。 */
    public static final String START_DATE = EntityCommonFieldNames.START_DATE;
    /** 終了日。 */
    public static final String END_DATE = EntityCommonFieldNames.END_DATE;
    /** 勘定科目コード。 */
    public static final String ACCOUNT_CD = "accountCd";
    /** 税区分コード。 */
    public static final String TAX_TYPE_CD = "taxTypeCd";
    /** ソートキー。 */
    public static final String SORT_KEY = EntityCommonFieldNames.SORT_KEY;
    /** 削除フラグ。 */
    public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;

    /** 会社名称(他マスタ名称)。 */
    public static final String COMPANY_NAME = "companyName";
    /** 勘定科目名称(他マスタ名称)。 */
    public static final String ACCOUNT_NAME = "accountName";
    /** 税区分名称(他マスタ名称)。 */
    public static final String TAX_TYPE_NAME = "taxTypeName";
    /** 税率(他マスタ)。 */
    public static final String TAX_RATE = "taxRate";
}

```

Java (S2JDBC)

Entity

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpType.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 経費区分マスタ。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Entity
@Table(name = "k30m_exp_type")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mExpType extends GenerateEntity implements Serializable {

```

```

/** シリアルバージョンUID. */
private static final long serialVersionUID = 1L;

/** 会社コード. */
@Id
@Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String companyCd;

/** 経費区分コード. */
@Id
@Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String expTypeCd;

/** 期間コード. */
@Id
@Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String termCd;

/** ロケールid. */
@Id
@Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
public String localeId;

/** 経費区分名称. */
@Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
public String expTypeName;

/** 経費区分カテゴリ. */
@Column(name = "exp_type_category", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String expTypeCategory;

/** 開始日. */
@Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
public Timestamp endDate;

/** 勘定科目コード. */
@Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String accountCd;

/** 税区分コード. */
@Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
public String taxTypeCd;

/** ソートキー. */
@Column(name = "sort_key", columnDefinition = "number(15,0)", nullable = false, unique = false)
public Long sortKey;

/** 削除フラグ. */
@Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
public String deleteFlag;

/** 予備項目a. */
@Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionA;

/** 予備項目b. */
@Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionB;

/** 予備項目c. */
@Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionC;

/** 予備項目d. */
@Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionD;

/** 予備項目e. */
@Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionE;

/** 予備項目f. */
@Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionF;

/** 予備項目g. */
@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

```

```

// 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensioni;

// 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpTypeNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30mExpType}の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mExpTypeNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30mExpTypeNames() {
        super();
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * expTypeNameのプロパティ名を返却.
     *
     * @return expTypeNameのプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }

    /**
     * expTypeCategoryのプロパティ名を返却.
     *
     * @return expTypeCategoryのプロパティ名
     */
}

```

```

    @return expTypeCategoryのプロパティ名
*/
public static PropertyName<String> expTypeCategory() {
    return new PropertyName<String>("expTypeCategory");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public static PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>("startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public static PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>("endDate");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * sortKeyのプロパティ名を返却.
 *
 * @return sortKeyのプロパティ名
 */
public static PropertyName<Long> sortKey() {
    return new PropertyName<Long>("sortKey");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public static PropertyName<String> deleteFlag() {
    return new PropertyName<String>("deleteFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public static PropertyName<String> extensionC() {
    return new PropertyName<String>("extensionC");
}

```



```

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public static PropertyName<String> extensionD() {
    return new PropertyName<String>("extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public static PropertyName<String> extensionE() {
    return new PropertyName<String>("extensionE");
}

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */

```

```

public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30mExpTypeNames extends PropertyName<K30mExpType> {

    /**
     * インスタンスを構築.
     */
    public _K30mExpTypeNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30mExpTypeNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30mExpTypeNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public PropertyName<String> companyCd() {
        return new PropertyName<String>(this, "companyCd");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public PropertyName<String> termCd() {
        return new PropertyName<String>(this, "termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>(this, "localeId");
    }
}

```

```

}

/**
 * expTypeNameのプロパティ名を返却.
 *
 * @return expTypeNameのプロパティ名
 */
public PropertyName<String> expTypeName() {
    return new PropertyName<String>(this, "expTypeName");
}

/**
 * expTypeCategoryのプロパティ名を返却.
 *
 * @return expTypeCategoryのプロパティ名
 */
public PropertyName<String> expTypeCategory() {
    return new PropertyName<String>(this, "expTypeCategory");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>(this, "startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public PropertyName<String> accountCd() {
    return new PropertyName<String>(this, "accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>(this, "taxTypeCd");
}

/**
 * sortKeyのプロパティ名を返却.
 *
 * @return sortKeyのプロパティ名
 */
public PropertyName<Long> sortKey() {
    return new PropertyName<Long>(this, "sortKey");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>(this, "deleteFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *

```

```

    * @return extensionBのプロパティ名
    */
    public PropertyName<String> extensionB() {
        return new PropertyName<String>(this, "extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public PropertyName<String> extensionC() {
        return new PropertyName<String>(this, "extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public PropertyName<String> extensionD() {
        return new PropertyName<String>(this, "extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public PropertyName<String> extensionE() {
        return new PropertyName<String>(this, "extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public PropertyName<String> extensionF() {
        return new PropertyName<String>(this, "extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public PropertyName<String> extensionG() {
        return new PropertyName<String>(this, "extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public PropertyName<String> extensionH() {
        return new PropertyName<String>(this, "extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public PropertyName<String> extensionI() {
        return new PropertyName<String>(this, "extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public PropertyName<String> extensionJ() {
        return new PropertyName<String>(this, "extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }
}

```

```

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}
}
}

```

Service

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mExpType;

/**
 * {@link K30mExpType}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mExpTypeService extends GenerateService<K30mExpType> {
    //
}

```

Condition

Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30mExpType}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mExpTypeCondition extends
    AbstractEntityCondition<K30mExpTypeCondition> {
}

```

```

/**
 * インスタンスを構築
 */
public K30mExpTypeCondition() {
}

/**
 * インスタンスを構築
 *
 * @param prefix プレフィックス
 * @param where 検索条件
 */
public K30mExpTypeCondition(String prefix, ComplexWhere where) {
    super(prefix, where);
}

/** companyCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> companyCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("companyCd", this);

/** expTypeCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> expTypeCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("expTypeCd", this);

/** termCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> termCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("termCd", this);

/** localeIdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> localeId =
    new NotNullableStringCondition<K30mExpTypeCondition>("localeId", this);

/** expTypeNameの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> expTypeName =
    new NotNullableStringCondition<K30mExpTypeCondition>("expTypeName", this);

/** expTypeCategoryの条件. */
public NullableStringCondition<K30mExpTypeCondition> expTypeCategory =
    new NullableStringCondition<K30mExpTypeCondition>("expTypeCategory", this);

/** startDateの条件. */
public NotNullableCondition<K30mExpTypeCondition, Timestamp> startDate =
    new NotNullableCondition<K30mExpTypeCondition, Timestamp>("startDate", this);

/** endDateの条件. */
public NotNullableCondition<K30mExpTypeCondition, Timestamp> endDate =
    new NotNullableCondition<K30mExpTypeCondition, Timestamp>("endDate", this);

/** accountCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> accountCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("accountCd", this);

/** taxTypeCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> taxTypeCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("taxTypeCd", this);

/** sortKeyの条件. */
public NotNullableCondition<K30mExpTypeCondition, Long> sortKey =
    new NotNullableCondition<K30mExpTypeCondition, Long>("sortKey", this);

/** deleteFlagの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> deleteFlag =
    new NotNullableStringCondition<K30mExpTypeCondition>("deleteFlag", this);

/** extensionAの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionA =
    new NullableStringCondition<K30mExpTypeCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionB =
    new NullableStringCondition<K30mExpTypeCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionC =
    new NullableStringCondition<K30mExpTypeCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionD =
    new NullableStringCondition<K30mExpTypeCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionE =
    new NullableStringCondition<K30mExpTypeCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionF =

```

```

    new NullableStringCondition<K30mExpTypeCondition>("extensionF", this);

    /** extensionGの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionG =
        new NullableStringCondition<K30mExpTypeCondition>("extensionG", this);

    /** extensionHの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionH =
        new NullableStringCondition<K30mExpTypeCondition>("extensionH", this);

    /** extensionIの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionI =
        new NullableStringCondition<K30mExpTypeCondition>("extensionI", this);

    /** extensionJの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionJ =
        new NullableStringCondition<K30mExpTypeCondition>("extensionJ", this);

    /** entryTsの条件. */
    public NotNullCondition<K30mExpTypeCondition, Timestamp> entryTs =
        new NotNullCondition<K30mExpTypeCondition, Timestamp>("entryTs", this);

    /** entryUserCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> entryUserCd =
        new NotNullStringCondition<K30mExpTypeCondition>("entryUserCd", this);

    /** renewCntの条件. */
    public NotNullCondition<K30mExpTypeCondition, Long> renewCnt =
        new NotNullCondition<K30mExpTypeCondition, Long>("renewCnt", this);

    /** renewTsの条件. */
    public NotNullCondition<K30mExpTypeCondition, Timestamp> renewTs =
        new NotNullCondition<K30mExpTypeCondition, Timestamp>("renewTs", this);

    /** renewUserCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> renewUserCd =
        new NotNullStringCondition<K30mExpTypeCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService/selectView.sql

```

select
    exp.company_cd
    , exp.exp_type_cd
    , exp.term_cd
    , exp.locale_id
    , exp.exp_type_name
    , exp.exp_type_category
    , exp.start_date
    , exp.end_date
    , exp.account_cd
    , exp.tax_type_cd
    , exp.sort_key
    , exp.delete_flag
    , exp.extension_a
    , exp.extension_b
    , exp.extension_c
    , exp.extension_d
    , exp.extension_e
    , exp.extension_f
    , exp.extension_g
    , exp.extension_h
    , exp.extension_i
    , exp.extension_j
    , exp.entry_ts
    , exp.entry_user_cd
    , exp.renew_cnt
    , exp.renew_ts
    , exp.renew_user_cd
    , dept.department_name as company_name
    , account.account_name as account_name
    , tax.tax_type_name as tax_type_name
    , tax.tax_rate as tax_rate
from k30m_exp_type exp
left join ( select imm_department.company_cd
    , imm_department.department_set_cd
    , imm_department.department_cd
    , imm_department.locale_id
    , imm_department.start_date

```

```

        , imm_department.end_date
        , imm_department.department_name
    from imm_department) dept
on exp.company_cd = dept.company_cd
and exp.company_cd = dept.department_set_cd
and exp.company_cd = dept.department_cd
and exp.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and dept.start_date <= exp.start_date
--ELSE and dept.end_date > exp.start_date
/*END*/
left join k30m_account account
on exp.company_cd = account.company_cd
and exp.account_cd = account.account_cd
and exp.locale_id = account.locale_id
/*IF searchCriteriaDate != null*/
and account.start_date <= /*searchCriteriaDate*/2000/01/01'
and account.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and account.start_date <= exp.start_date
--ELSE and account.end_date > exp.start_date
/*END*/
left join k30m_tax_type tax
on exp.company_cd = tax.company_cd
and exp.tax_type_cd = tax.tax_type_cd
and exp.locale_id = tax.locale_id
/*IF searchCriteriaDate != null*/
and tax.start_date <= /*searchCriteriaDate*/2000/01/01'
and tax.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and tax.start_date <= exp.start_date
--ELSE and tax.end_date > exp.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
exp.company_cd in /*companyCd*/('comp')
/*END*/
/*IF expTypeCd != null*/
and exp.exp_type_cd = /*expTypeCd*/expTypeCd'
/*END*/
/*IF termCd != null*/
and exp.term_cd = /*termCd*/termCd'
/*END*/
/*IF localeId != null*/
and exp.locale_id = /*localeId*/localeId'
/*END*/
/*IF expTypeName != null*/
and exp.exp_type_name like /*expTypeName*/%' ESCAPE '$'
/*END*/
/*IF expTypeCategory != null*/
and exp.exp_type_category like /*expTypeCategory*/expTypeCategory' ESCAPE '$'
/*END*/
/*IF searchCriteriaDate != null*/
and exp.start_date <= /*searchCriteriaDate*/1900/01/01'
and exp.end_date > /*searchCriteriaDate*/2999/12/31'
/*END*/
/*IF accountCd != null*/
and exp.account_cd = /*accountCd*/accountCd'
/*END*/
/*IF taxTypeCd != null*/
and exp.tax_type_cd = /*taxTypeCd*/taxTypeCd'
/*END*/
/*IF sortKey != null*/
and exp.sort_key = /*sortKey*/0'
/*END*/
/*IF deleteFlag != null*/
and exp.delete_flag = /*deleteFlag*/0'
/*END*/
/*IF extensionA != null*/
and exp.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and exp.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and exp.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and exp.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and exp.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and exp.extension_f like /*extensionF*/extensionF' ESCAPE '$'

```



```

/*END*/
/*IF extensionG != null*/
  and exp.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
  and exp.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
  and exp.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
  and exp.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
  and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF accountName != null*/
  and account.account_name like /*accountName*/accountName' ESCAPE '$'
/*END*/
/*IF taxTypeName != null*/
  and tax.tax_type_name like /*taxTypeName*/taxTypeName' ESCAPE '$'
/*END*/
/*IF taxRate != null*/
  and tax.tax_rate = /*taxRate*/0'
/*END*/
/*END*/
/*IF orderBy != null*/
  ORDER BY /*$orderBy*/company_cd
/*END*/

```

XML

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/feature/action/kaiden/master/ExpTypeMasterMaintenanceSettings.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericMasterMaintenanceSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterMaintenanceSettings">
  <!-- 画面タイトルラベルID -->
  <titleSurfaceKey>PWKF.master.expType.title</titleSurfaceKey>
  <!-- 検索画面JSPフルパス -->
  <searchGadgetJsp></searchGadgetJsp><!-- デフォルトを利用 -->
  <!-- メンテナンス画面JSPフルパス -->
  <maintenanceGadgetJsp>/WEB-INF/view/kaiden2/product_workflow/master/expType/expTypeMainte.jsp</maintenanceGadgetJsp>
  <!-- AjaxAccessUrl -->
  <ajaxAccessUrl>kaiden/master/expType</ajaxAccessUrl>
  <!-- 1ページあたりの件数 (0以下はページングせずすべて表示) -->
  <limit>20</limit>
  <!-- 検索条件の検索基準日 使用/不使用 -->
  <useSearchCriteriaDate>true</useSearchCriteriaDate>
  <!-- 検索条件の検索基準日 必須/任意 -->
  <requiredSearchCriteriaDate>true</requiredSearchCriteriaDate>
  <!-- 検索条件の検索基準日 デフォルト値(システム日付) 設定/非設定 (必須の場合にはこの設定にかかわらずデフォルト値が設定される) -->
  <defaultSearchCriteriaDate>true</defaultSearchCriteriaDate>
  <!-- 検索条件の会社 使用/不使用 -->
  <useSearchCompany>true</useSearchCompany>
  <!-- 検索条件のロケール 使用/不使用 -->
  <useSearchLocale>true</useSearchLocale>
  <!-- 検索条件のステータス無効データも検索。使用/不使用 -->
  <useSearchWithDisabled>true</useSearchWithDisabled>
  <!-- 汎用一覧・検索画面の行番号ラベルID -->
  <rownumberSurfaceKey></rownumberSurfaceKey><!-- デフォルトを利用 -->
  <!-- 汎用一覧・検索画面のカラム設定 -->
  <listCols>
    <listCol>
      <!-- 対象マスタのフィールドキー -->
      <fieldKey>expTypeCd</fieldKey>
      <!-- キーワード検索対象とするか否か -->
      <keywordSearchTarget>true</keywordSearchTarget>
      <!-- 表示幅 -->
      <width>120px</width>
      <!-- 表示位置 -->
      <align></align>
      <!-- ヘッダラベルID -->
      <headerSurfaceKey>PWKF.expTypeCd</headerSurfaceKey>
      <!-- ソート可能か否か -->
      <sortable>true</sortable>
    </listCol>
    <listCol>
      <fieldKey>expTypeName</fieldKey>
      <keywordSearchTarget>true</keywordSearchTarget>
      <width></width>
      <align></align>
      <headerSurfaceKey>PWKF.expTypeName</headerSurfaceKey>
      <sortable>true</sortable>
    </listCol>
  </listCols>

```

```

<listCol>
  <fieldKey>accountCd</fieldKey>
  <keywordSearchTarget>true</keywordSearchTarget>
  <width>120px</width>
  <align></align>
  <headerSurfaceKey>PWKF.accountCd</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>accountName</fieldKey>
  <keywordSearchTarget>true</keywordSearchTarget>
  <width></width>
  <align></align>
  <headerSurfaceKey>PWKF.accountName</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>taxTypeName</fieldKey>
  <keywordSearchTarget>true</keywordSearchTarget>
  <width></width>
  <align></align>
  <headerSurfaceKey>PWKF.taxTypeName</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>sortKey</fieldKey>
  <keywordSearchTarget>false</keywordSearchTarget>
  <width>90px</width>
  <align>right</align>
  <headerSurfaceKey>BASE.sortKey</headerSurfaceKey>
  <sortable>true</sortable>
</listCol>
<listCol>
  <fieldKey>deleteFlag</fieldKey>
  <keywordSearchTarget>false</keywordSearchTarget>
  <width>100px</width>
  <align>center</align>
  <headerSurfaceKey>BASE.status</headerSurfaceKey>
  <sortable>false</sortable>
</listCol>
</listCols>
<!-- 検索・一覧画面のソート順（汎用マスター一覧は、第二ソートキーまで有効）-->
<orders>
  <order>
    <fieldKey>sortKey</fieldKey>
    <type>asc</type>
  </order>
  <order>
    <fieldKey>expTypeCd</fieldKey>
    <type>asc</type>
  </order>
</orders>
<!-- 対象マスタの主キーフィールド -->
<pkFields>
  <pkField>companyCd</pkField>
  <pkField>expTypeCd</pkField>
  <pkField>termCd</pkField>
  <pkField>localeId</pkField>
</pkFields>
</genericMasterMaintenanceSettings>

```

JSP

■ Path

```
src/main/webapp/WEB-INF/view/kaiden2/product_workflow/master/expType/expTypeMainte.jsp
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="im" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_base" uri="http://kaiden.slcs.co.jp/taglib/listdata/base" %>
<%@ taglib prefix="klist_workflow" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
<%@ page import="jp.co.slcs.kaiden2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp" %>
<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>

<!-- product_workflow 共通インクルード --%>
<c:if test="{empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>

<!-- 汎用マスター一覧・検索 --%>

```

```

<?-- 設定 --?>
<c:set var="gadgetClass">expTypeMainte</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>
<c:set var="gadgetTitle">${maintenanceDto.settings.title}</c:set>
<c:set var="accessUrl">${maintenanceDto.settings.ajaxAccessUrl}</c:set>

<c:set var="surfaceStartDate"><%= BaseProp.S.START_DATE %></c:set>
<c:set var="surfaceConcatChar"><%= BaseProp.S.CONCAT_CHAR %></c:set>
<c:set var="surfaceEndDate"><%= BaseProp.S.END_DATE %></c:set>
<c:set var="surfaceCompany"><%= BaseProp.S.COMPANY %></c:set>
<c:set var="surfaceExpTypeCd"><%= ProductWorkflowProp.S.EXP_TYPE_CD %></c:set>
<c:set var="surfaceExpTypeName"><%= ProductWorkflowProp.S.EXP_TYPE_NAME %></c:set>
<c:set var="surfaceExpTypeCategory"><%= ProductWorkflowProp.S.ExpType.EXP_TYPE_CATEGORY %></c:set>
<c:set var="surfaceAccount"><%= ProductWorkflowProp.S.ACCOUNT %></c:set>
<c:set var="surfaceTaxType"><%= ProductWorkflowProp.S.TAX_TYPE %></c:set>
<c:set var="surfaceSortKey"><%= BaseProp.S.SORT_KEY %></c:set>

<?-- バリデーション --?>
<c:set var="validStartDate_def">{k_required:[], k_date:[]}</c:set>
<c:set var="validEndDate_def">{k_required:[], k_date:[], k_dateMin:['startDate']}</c:set>
<c:set var="validCompanyCd_def">{k_required:[], k_companyExists:['companyCd', '', 'true']}</c:set>
<c:set var="validExpTypeCd_def">{k_required:[], k_kaidenId:[], k_maxLength:['100']}</c:set>
<c:set var="validExpTypeName_def">{k_required:[], k_maxLength:['250']}</c:set>
<c:set var="validExpTypeCategory_def">{k_maxLength:['250']}</c:set>
<c:set var="validAccount_def">{k_required:[], k_accountExists:['accountCd', 'companyCd', '', 'true']}</c:set>
<c:set var="validTaxType_def">{k_required:[], k_taxTypeExists:['taxTypeCd', 'companyCd', '', 'true']}</c:set>
<c:set var="validSortKey_def">{k_required:[], k_numeric:[], k_maxLength:['15']}</c:set>

<?-- 期間一覧ガジェット --?>
<div class="kaiden-ui-master-term-list">
  <jsp:include page="/WEB-INF/view/kaiden2/base/foundation/gadget/genericMasterTermList.jsp" flush="true">
    <jsp:param name="gadget_instance" value="0" />
  </jsp:include>
</div>
<div class="kaiden-ui-master-maintenance-area">
  <k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="${gadgetTitle}"
    gadgetId="${gadgetId}"
    hideGadgetTitle="true"
    class="kaiden-ui-master-maintenance-input-area">

    <k:blockContainer id="inputArea">
      <form id="inputForm" onsubmit="return false;">
        <table class="imui-form" style="table-layout:fixed">
          <colgroup>
            <col style="width:150px;" />
            <col />
          </colgroup>
          <tr style="height:30px;">
            <th>
              <k:surface surfaceKey="${surfaceStartDate}" style="display:inline-block" requiredMark="true"/>
              <k:surface surfaceKey="${surfaceConcatChar}" style="display:inline-block" />
              <k:surface surfaceKey="${surfaceEndDate}" style="display:inline-block" requiredMark="true"/>
            </th>
            <td>
              <k:date name="startDate"
                mode="0"
                surfaceKey="${surfaceStartDate}"
                validationDefault="${validStartDate_def}"
              />
              <k:surface surfaceKey="${surfaceConcatChar}" style="display:inline-block;margin-left:1em;margin-right:1em;" />
              <k:date name="endDate"
                mode="0"
                surfaceKey="${surfaceEndDate}"
                validationDefault="${validEndDate_def}"
              />
            </td>
          </tr>
          <tr>
            <th>
              <k:surface surfaceKey="${surfaceCompany}" requiredMark="true"/>
            </th>
            <td>
              <klist_base:listDataCompany name="readParmitCompany" type="readParmitCompany" blank="false"/>
              <k:select name="companyCd"
                mode="0"
                validationDefault="${validCompanyCd_def}"
                surfaceKey="${surfaceCompany}"
                dataName="readParmitCompany"
                style="width:300px;"
                includeDisabled="true"/>
            </td>
          </tr>
        </table>
      </form>
    </k:gadgetContainer>
  </div>

```

```

</td>
<tr>
  <th>
    <k:surface surfaceKey="{surfaceExpTypeCd}" requiredMark="true"/>
  </th>
  <td>
    <k:codeText name="expTypeCd"
      mode="0"
      style="width:300px;"
      surfaceKey="{surfaceExpTypeCd}"
      validationDefault="{validExpTypeCd_def}"
      maxlength="100"
    />
  </td>
</tr>
<tr>
  <th>
    <k:surface surfaceKey="{surfaceExpTypeName}" requiredMark="true"/>
  </th>
  <td>
    <k:text-locales name="expTypeName"
      mode="0"
      style="width:300px;"
      surfaceKey="{surfaceExpTypeName}"
      validationDefault="{validExpTypeName_def}"
      imeMode="on"
      maxlength="250"
    />
  </td>
</tr>
<tr>
  <th>
    <k:surface surfaceKey="{surfaceExpTypeCategory}" />
  </th>
  <td>
    <k:text name="expTypeCategory"
      mode="0"
      style="width:300px;"
      surfaceKey="{surfaceExpTypeCategory}"
      validationDefault="{validExpTypeCategory_def}"
      imeMode="on"
      maxlength="250"
    />
  </td>
</tr>
<tr>
  <th>
    <k:surface surfaceKey="{surfaceAccount}" requiredMark="true"/>
  </th>
  <td>
    <kmsw:account name="accountName"
      mode="0"
      resultChain="{accountCd:accountCd}"
      searchCriteriaAccountCd="accountCd"
      searchCriteriaCompany="companyCd"
      resultValue="accountName"
      primaryTarget="accountCd"
      includeDisabled="true"
      style="width:280px"
      imeMode="on"
      surfaceKey="{surfaceAccount}"
      validationDefault="{validAccount_def}"
    />
    <k:hidden name="accountCd" mode="0" />
  </td>
</tr>
<tr>
  <th>
    <k:surface mode="0" surfaceKey="{surfaceTaxType}" requiredMark="true"/>
  </th>
  <td>
    <klist_workflow:listDataTaxType name="taxTypeName" blank="false" companyCd="" />
    <k:select name="taxTypeCd"
      mode="0"
      validationDefault="{validTaxType_def}"
      surfaceKey="{surfaceTaxType}"
      dataName="taxTypeName"
      style="width:300px;"
      includeDisabled="true"
    />
  </td>
</tr>
<tr>
  <th>
    <k:surface mode="0" surfaceKey="{surfaceSortKey}" requiredMark="true"/>
  </th>
  <td>

```

```

        <k:numericText name="sortKey"
            mode="0"
            style="width:50px;"
            surfaceKey="{surfaceSortKey}"
            validationDefault="{validSortKey_def}"
            maxlength="15"
        />
    </td>
</tr>
</table>
</form>
</k:blockContainer>
</k:gadgetContainer>
<script type="text/javascript">
jQuery(function($) {

    var eventMan = KAIDEN.getGadget("${gadgetId").eventManager
        , inputAreaAccessor = KAIDEN.getGadget("${gadgetId").blockAccessors["inputArea"];

    //trunsNewEntry (新規入力画面表示) イベントリスナの登録
    eventMan.registPushListener("trunsNewEntry", function(**Object*/arg) {
        //入力欄を一括クリア
        KAIDEN.getGadget("${gadgetId").clearInput("inputArea");

        //開始日・終了日にシステム開始日・システム終了日を設定
        inputAreaAccessor("startDate").setValue("${k:getSystemStartDate()}");
        inputAreaAccessor("endDate").setValue("${k:getSystemEndDateDecrement()}");

        //開始日・終了日、会社、経費区分コードをdisabled
        inputAreaAccessor("startDate").removeDisabled();
        inputAreaAccessor("endDate").removeDisabled();
        inputAreaAccessor("companyCd").removeDisabled();
        inputAreaAccessor("expTypeCd").removeDisabled();

        if (inputAreaAccessor("companyCd").elem().children().length > 0) {
            inputAreaAccessor("companyCd").elem().get(0).selectedIndex = 0;
            inputAreaAccessor("companyCd").elem().trigger("change");
        }
        KAIDEN.MasterSearch.refreshAll(true);

        //会社をフォーカス
        inputAreaAccessor("companyCd").elem().focus();

    });

    //loadData (表示対象データのロード) イベントリスナの登録
    eventMan.registPushListener("loadData", function(**Object*/arg) {
        var ajax = KAIDEN.Ajax()
            , criteria = {}
            , stat = false;

        criteria.criteriaJson = arg;

        ajax.postSyncAuto("${accessUrl}/select", criteria
            , function(returnData) {
                //正常系
                //メンテナンス画面にデータセット
                for (var key in returnData.mainteTarget) {
                    inputAreaAccessor(key).setValueIgnore(returnData.mainteTarget[key]);
                }
                inputAreaAccessor("pk_json").setValue(arg); //主キーを保存
                inputAreaAccessor("companyCd").elem().trigger("change4Load");
                inputAreaAccessor("taxTypeCd").setValue(returnData.mainteTarget["taxTypeCd"])
                KAIDEN.MasterSearch.refreshAll(true);
                stat = true;
            }
            , function(returnData) {
                stat = false;
            }
        );
        return stat;
    });

    //trunsRenewEntry (更新画面表示) イベントリスナの登録
    eventMan.registPushListener("trunsRenewEntry", function(**Object*/arg) {
        //開始日・終了日、会社、経費区分コードをdisabled
        inputAreaAccessor("startDate").disabled();
        inputAreaAccessor("endDate").disabled();
        inputAreaAccessor("companyCd").disabled();
        inputAreaAccessor("expTypeCd").disabled();
    });

```

```

    $("div[data-kaiden-gadget-id='${gadgetId}'] input[data-kaiden-org-name^='expTypeName']").eq(0).focus();

});

//currentPrimaryInfo (現在表示中データの主キーを要求) イベントリスナの登録
eventMan.registPullListener("claimCurrentDataPk", function(**Object*/arg) {
    return inputAreaAccessor("pk_json").getValue(); //主キーを返却
});

inputAreaAccessor("companyCd").elem().on("change", function() {
    var criteria = {};
    criteria.companyCd = inputAreaAccessor("companyCd").getValue();
    KAIDEN.SelectListData.refreshList("taxTypeName", criteria);
    KAIDEN.MasterSearch.clear(inputAreaAccessor("accountName").elem());
});

inputAreaAccessor("companyCd").elem().on("change4Load", function() {
    var criteria = {};
    criteria.companyCd = inputAreaAccessor("companyCd").getValue();
    KAIDEN.SelectListData.refreshList("taxTypeName", criteria);
});

});
</script>
<!-- マスタメンテナンス汎用ボタンガジェット --%>
<div style="clear:both;">
<jsp:include page="/WEB-INF/view/kaiden2/base/foundation/gadget/genericMasterOperationButton.jsp" flush="true">
    <jsp:param name="gadget_instance" value="0" />
    <jsp:param name="maintenanceGadgetId" value="${gadgetId}" />
</jsp:include>
</div>
</div>

```

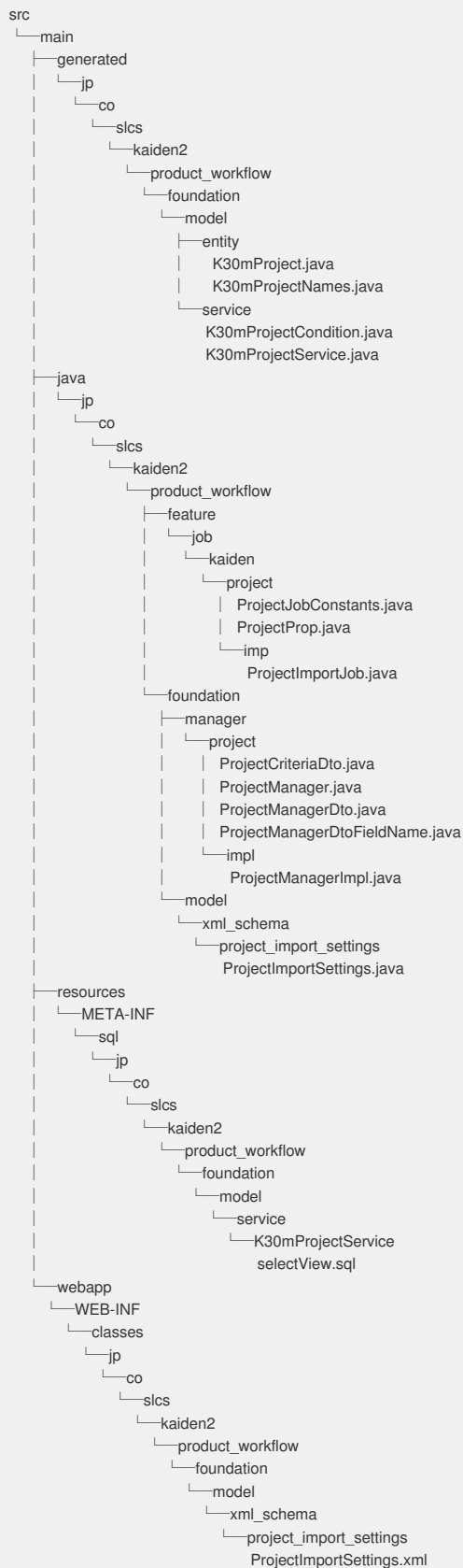

プロジェクトマスタインポート

プロジェクトマスタインポートのサンプルソースです。

本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

- [一覧](#)
- [Java](#)
 - [Job](#)
 - [Manager](#)
 - [Manager \(実装\)](#)
 - [ManagerDto](#)
 - [Settings](#)
- [Java \(S2JDBC\)](#)
 - [Entity](#)
 - [Names](#)
 - [Service](#)
 - [Condition](#)
- [SQL](#)
 - [検索](#)
- [XML](#)

[一覧](#)



Java

Job

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/project/imp/ProjectImportJob.java

```
package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.project.imp;
```

```
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
```

```
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
```

```
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
```

```
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
```

```
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
```

```

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerUtil.FUNCTIONAL_INTERFACE_ROLE_ID_CD;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.job.AbstractTermMasterImportJob;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobProcessType;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.xml_schema.generic_jop_process_result.GenericJobProcessResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.project.ProjectJobConstants;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.xml_schema.project_import_settings.ProjectImportSettings;
import jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow;
import org.seasar.framework.container.SingletonS2Container;

/**
 * プロジェクトマスタImporter.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectImportJob extends AbstractTermMasterImportJob<ProjectImportSettings, GenericJobProcessResult> {

    /** ProjectManager. */
    private final ProjectManager projectManager;

    /**
     * コンストラクタ.
     */
    public ProjectImportJob() {
        super(KaidenJobProcessType.IMPORT, ProjectJobConstants.KAIDEN_JOB_ID, ErrorMode.ERROR_CONTINUE);
        projectManager = SingletonS2Container.getComponent("projectManager");
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected ProjectImportSettings readSettingsForImport() {
        return super.read(ProjectImportSettings.class);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected GenericJobProcessResult getGenericJobProcessResult() {
        return new GenericJobProcessResult();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected LinkedHashMap<String, Map<String, String>> getSurfacesMap() {
        LinkedHashMap<String, Map<String, String>> surfacesMap = new LinkedHashMap<String, Map<String, String>>();
        surfacesMap.put(createUniqueKey(TARGET_MASTER_KEY), ProjectJobConstants.PROJECT_MST_SURFACES);
        return surfacesMap;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected List<ValidateDefsDto> addValidateRuleSub() {

        // 新規登録用の検証定義リスト取得
        List<ValidateDefsDto> vDefs = projectManager.getValidateDefsDtoList4Entry();

        int trglDx = -1;

        for (int i = 0; i < vDefs.size(); i++) {
            if (ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS.equals(vDefs.get(i).singleValidationId)) {
                trglDx = i;
                break;
            }
        }

        // プロジェクトマスタ存在チェックを削除 (これから登録するので存在チェックは行わない)
        if (trglDx >= 0) {
            vDefs.remove(trglDx);
        }

        // インポートバッチ独自実装分追加
        // 会社コード(上位)
        vDefs.add(new ValidateDefsDto(PARENT_COMPANY_CD, VALID_COMPANY_EXISTS, PARENT_COMPANY_CD, "", "true"));
    }

```

```

        validateDefsDto(PARENT_PROJECT_CD, VALID_MAX_LENGTH, String
        // プロジェクトコード(上位)
        vDefs.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_MAX_LENGTH, String
        .valueOf(ProjectManager.PROJECT_CD_LEN)));
        vDefs.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_KAIDEN_ID));

        return vDefs;
    }

    /** {@inheritDoc} */
    @SuppressWarnings("unchecked")
    @Override
    protected TermMasterManagerForJob<? extends KaidenDto> getDefaultManager() {
        return (TermMasterManagerForJob<? extends KaidenDto>) projectManager;
    }
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/project/ProjectJobConstants.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.project;

import java.util.LinkedHashMap;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobIdIF;

/**
 * プロジェクトマスタImporter/expoter定数.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ProjectJobConstants {

    /**
     * コンストラクタ.
     */
    private ProjectJobConstants() {
    }

    /** 処理対象機能皆伝ジョブID:プロジェクトマスタ. */
    public static final KaidenJobIdIF KAIDEN_JOB_ID = ProjectProp.KaidenJobID.KAIDEN_JOB_ID_IMPORT;
    /** プロジェクトマスタレベル定義. */
    public static final LinkedHashMap<String, String> PROJECT_MST_SURFACES = ProjectProp.Surfaces.getSurfaces();
    /** プロジェクトマスタファイルIndex. */
    public static final int DATA_FILE_INDEX = 0;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/project/ProjectProp.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.project;

import java.util.LinkedHashMap;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobIdIF;
import jp.co.slcs.kaiden2.base.foundation.job.SurfacesIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ExtensionsProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName;
import org.seasar.framework.container.SingletonS2Container;

/**
 * プロジェクトマスタDBカラムプロパティ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ProjectProp {

    /**
     * プロジェクトマスタDBカラム.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    public static final class C {

        /** 皆伝ジョブID. */
        public static final String KAIDEN_JOB_ID_IMPORT = "kaiden-job-master-import-project";
        /** 会社コード. */
        public static final String COMPANY_CD = ProjectManagerDtoFieldName.COMPANY_CD;
        /** プロジェクトコード. */
        public static final String PROJECT_CD = ProjectManagerDtoFieldName.PROJECT_CD;
        /** 期間コード. */
        public static final String TERM_CD = ProjectManagerDtoFieldName.TERM_CD;
        /** ロケールID. */

```

```

    public static final String LOCALE_ID = ProjectManagerDtoFieldName.LOCALE_ID;
    /** プロジェクト名称. */
    public static final String PROJECT_NAME = ProjectManagerDtoFieldName.PROJECT_NAME;
    /** 開始日. */
    public static final String START_DATE = ProjectManagerDtoFieldName.START_DATE;
    /** 終了日. */
    public static final String END_DATE = ProjectManagerDtoFieldName.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = ProjectManagerDtoFieldName.DELETE_FLAG;
    /** 会社コード(上位プロジェクト). */
    public static final String PARENT_COMPANY_CD = ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
    /** プロジェクトコード(上位プロジェクト). */
    public static final String PARENT_PROJECT_CD = ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
}

/**
 * プロジェクトマスタDBカラム名称.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public static final class S {

    /** ジョブ名称. */
    public static final String JOB_NAME_IMPORT = ProductWorkflowProp.S.ProjectJob.JOB_NAME_IMPORT;
    /** 会社コード. */
    public static final String COMPANY_CD = BaseProp.S.COMPANY_CD;
    /** プロジェクトコード. */
    public static final String PROJECT_CD = ProductWorkflowProp.S.PROJECT_CD;
    /** 期間コード. */
    public static final String TERM_CD = BaseProp.S.TERM_CD;
    /** ロケールId. */
    public static final String LOCALE_ID = BaseProp.S.LOCALE_ID;
    /** プロジェクト名称. */
    public static final String PROJECT_NAME = ProductWorkflowProp.S.PROJECT_NAME;
    /** 開始日. */
    public static final String START_DATE = BaseProp.S.START_DATE;
    /** 終了日. */
    public static final String END_DATE = BaseProp.S.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = BaseProp.S.DELETE_FLAG;
    /** 会社コード(上位プロジェクト). */
    public static final String PARENT_COMPANY_CD = ProductWorkflowProp.S.Project.PARENT_COMPANY_CD;
    /** プロジェクトコード(上位プロジェクト). */
    public static final String PARENT_PROJECT_CD = ProductWorkflowProp.S.Project.PARENT_PROJECT_CD;
}

/**
 * プロジェクトマスタJob IDの列挙.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public static enum KaidenJobID implements KaidenJobIdIF {

    /** 皆伝ジョブID(インポート). */
    KAIDEN_JOB_ID_IMPORT(ProjectProp.C.KAIDEN_JOB_ID_IMPORT, ProjectProp.S.JOB_NAME_IMPORT);

    /** 皆伝ジョブID. */
    private final String kaidenJobId;
    /** ジョブ名称用サーフェスキー. */
    private final String surfaceKey;

    /**
     * コンストラクタ.
     *
     * @param kaidenJobId ジョブID
     * @param surfaceKey サーフェスキー
     */
    private KaidenJobID(String kaidenJobId, String surfaceKey) {
        this.kaidenJobId = kaidenJobId;
        this.surfaceKey = surfaceKey;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getKaidenJobId() {
        return kaidenJobId;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getKaidenJobName() {
        return SingletonS2Container.getComponent(PropertyHelper.class).getSurface(surfaceKey);
    }
}

```

```

        return SingletonS2Container.getComponent(PropertyHelper.class).getSurface(surfaceKey);
    }
}

/**
 * プロジェクトマスタラベル定義の列挙.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public enum Surfaces implements SurfacesIF {

    /** 会社コード. */
    COMPANY_CD(ProjectProp.C.COMPANY_CD, ProjectProp.S.COMPANY_CD),
    /** プロジェクトコード. */
    PROJECT_CD(ProjectProp.C.PROJECT_CD, ProjectProp.S.PROJECT_CD),
    /** 期間コード. */
    TERM_CD(ProjectProp.C.TERM_CD, ProjectProp.S.TERM_CD),
    /** ロケールId. */
    LOCALE_ID(ProjectProp.C.LOCALE_ID, ProjectProp.S.LOCALE_ID),
    /** プロジェクト名称. */
    PROJECT_NAME(ProjectProp.C.PROJECT_NAME, ProjectProp.S.PROJECT_NAME),
    /** 開始日. */
    START_DATE(ProjectProp.C.START_DATE, ProjectProp.S.START_DATE),
    /** 終了日. */
    END_DATE(ProjectProp.C.END_DATE, ProjectProp.S.END_DATE),
    /** 削除フラグ. */
    DELETE_FLAG(ProjectProp.C.DELETE_FLAG, ProjectProp.S.DELETE_FLAG),
    /** 会社コード(上位プロジェクト). */
    PARENT_COMPANY_CD(ProjectProp.C.PARENT_COMPANY_CD, ProjectProp.S.PARENT_COMPANY_CD),
    /** プロジェクトコード(上位プロジェクト). */
    PARENT_PROJECT_CD(ProjectProp.C.PARENT_PROJECT_CD, ProjectProp.S.PARENT_PROJECT_CD);

    /** ラベルキー. */
    private final String key;
    /** ラベル名称. */
    private final String name;

    /**
     * コンストラクタ.
     *
     * @param columnName ラベルキー
     * @param surfaceKey サーフィスキー
     */
    private Surfaces(String columnName, String surfaceKey) {
        key = columnName;
        name = SingletonS2Container.getComponent(PropertyHelper.class).getSurface(surfaceKey);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getKey() {
        return key;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getValue() {
        return name;
    }

    /**
     * ラベルキーから自身を返却.
     *
     * @param key ラベルキー値
     * @return 自身
     */
    public static Surfaces codeOf(String key) {
        for (Surfaces that : Surfaces.values()) {
            if (that.getKey().equals(key)) {
                return that;
            }
        }
        throw KaidenIllegalParametersException.makeInstance();
    }

    /**
     * ラベル名称から自身を返却.
     *
     * @param name ラベル名称
     * @return 自身
     */
    public static Surfaces nameOf(String name) {
        for (Surfaces that : Surfaces.values()) {

```

```

        if (that.getValue().equals(name)) {
            return that;
        }
    }
    throw KaidenIllegalParametersException.makeInstance();
}

/**
 * ラベル定義を取得します。
 *
 * @return ラベル定義
 */
public static LinkedHashMap<String, String> getSurfaces() {

    LinkedHashMap<String, String> surfaces = new LinkedHashMap<String, String>();

    for (Surfaces projectSurfaces : Surfaces.values()) {
        surfaces.put(projectSurfaces.getKey(), projectSurfaces.getValue());
    }

    surfaces.putAll(ExtensionsProp.Surfaces.getSurfaces());

    return surfaces;
}
}

```

Manager

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden2.base.foundation.helper.ValidateMapHelper;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenAbstractValidator;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidatorIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import org.seasar.framework.container.SingletonS2Container;

/**
 * プロジェクトマネージャのInterface.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ProjectManager extends TermMasterManager<ProjectManagerDto>,
    TermMasterManagerForJob<ProjectManagerDto> {

    /**
     * プロジェクトコードの最大長.
     */
    public static final int PROJECT_CD_LEN = 100;

    /**
     * プロジェクト名の最大長.
     */
    public static final int PROJECT_NAME_LEN = 250;

    /**
     * 新規登録用の検証定義リストを返却します。
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

    /**
     * 更新登録用の検証定義リストを返却します。
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

    /**

```

```

* カスタムバリデート_自プロジェクト・上位プロジェクト同一チェック。
* <p>
* 自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。
* </p>
*/

class CustomValidateNotEqualParent extends KaidenAbstractValidator implements KaidenValidatorIF {

    /** コンストラクタ。 */
    public CustomValidateNotEqualParent() {
        super();
    }

    /**
     * {@inheritDoc}
     * <p>
     * チェックエラーとなる条件<br>
     * <ul>
     * <li>自プロジェクト・上位プロジェクト同一チェック。</li>
     * <li>自プロジェクトと上位プロジェクトが同一コードの場合はエラーとする。</li>
     * </ul>
     * 以下の場合、チェックを行いません。
     * <ul>
     * <li>自プロジェクトコード・自会社コードのいずれかが設定されていない場合</li>
     * <li>上位プロジェクトコード・上位会社コードの両方が未設定の場合</li>
     * </ul>
     * 以下の場合、パラメーターエラーとします。
     * <ul>
     * <li>上位プロジェクトコード・上位会社コードの片方が未設定の場合</li>
     * </ul>
     * </p>
     */
    @Override
    public KaidenValidateResult executeValidate(Map<String, Object> valueMap, String fieldKey, String[] args,
        Map<String, String> surfaces) {

        KaidenValidateResult result = super.makeNewValidateResult(fieldKey, surfaces);

        ValidateMapHelper validateMapHelper = SingletonS2Container.getComponent(ValidateMapHelper.class);

        String thisCompanyCd = validateMapHelper.getValueForString(valueMap, COMPANY_CD);
        String thisProjectCd = validateMapHelper.getValueForString(valueMap, PROJECT_CD);

        // 自プロジェクトコードまたは自会社コードが設定されていない場合は、エラーとしない
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            return result;
        }

        String parentCompanyCd = validateMapHelper.getValueForString(valueMap, PARENT_COMPANY_CD);
        String parentProjectCd = validateMapHelper.getValueForString(valueMap, PARENT_PROJECT_CD);

        // 上位プロジェクトコードと上位会社コードの両方が未設定の場合、エラーとしない
        if (StringUtil.isEmpty(parentCompanyCd, parentProjectCd)) {
            return result;
        }

        // 上位プロジェクトコード・上位会社コードの片方が未設定の場合、パラメーターエラー
        if (StringUtil.isEmptyAnyOne(thisCompanyCd, thisProjectCd)) {
            throw KaidenIllegalParametersException.makeInstance();
        }

        // 自プロジェクトコードと上位プロジェクトコードが同一かつ、自会社コードと上位会社コードが同一の場合、チェックエラー
        if (thisCompanyCd.equals(parentCompanyCd) && thisProjectCd.equals(parentProjectCd)) {
            PropertyHelper propertyHelper = SingletonS2Container.getComponent(PropertyHelper.class);
            String fieldName = validateMapHelper.getSurfaces(valueMap, surfaces, PARENT_PROJECT_CD);
            result.setErrorStatus(true);
            result.setValidateClass(this.getClass());
            result.setFieldKey(PARENT_PROJECT_CD);
            result.setFieldNm(fieldName);
            result.setMainMessageKey(ProductWorkflowProp.M.Project.W001);
            result.setMainMessage(propertyHelper.getMessage(ProductWorkflowProp.M.Project.W001, fieldName));
        }

        return result;
    }
}

```

Manager (実装)

- Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/impl/ProjectManagerImpl.java

```
package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.impl;
```

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_PROJECT_EXISTS;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mProjectService;
import org.seasar.framework.beans.util.Beans;

/**
 * プロジェクトマスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います、より業務的な機能を付加した操作を提供します。<br>
 * 使用するには、{@link ProjectManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ProjectManagerDto}クラスまたはそのリストが戻り値として返却されます。<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <td>機能概要</td>
 * <td>メソッド</td>
 * <td>カテゴリ</td>
 * <td>備考</td>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>select</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>

```



```

*
* <tr>
* <td>主キー検索 (認可設定無視) </td>
* <td>selectIgnoreAuthCompany</td>
* <td>主キー検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>汎用検索</td>
* <td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>汎用検索 (認可設定無視) </td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>ロケール検索 (認可設定無視) </td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>参照検索 (認可設定無視) </td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>キーワード検索 (認可設定無視) </td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>有効化 (認可設定無視) </td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>無効化 (認可設定無視) </td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
*
* <tr>
* <td>登録 (認可設定無視) </td>

```

```

* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除(認可設定無視)</td>
* <td>removeIgnoreAuthCompany</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新</td>
* <td>renew</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新(認可設定無視)</td>
* <td>renewIgnoreAuthCompany</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* <b>機能一覧(期間化)</b>
* <table border="1">
* <tr>
* <td>機能概要</td>
* <td>メソッド</td>
* <td>カテゴリ</td>
* <td>備考</td>
* </tr>
* <tr>
* <td>期間検索</td>
* <td>selectTerms</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間検索(認可設定無視)</td>
* <td>selectTermsIgnoreAuthCompany</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索</td>
* <td>selectListIncludesSearchDate</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索(認可設定無視)</td>
* <td>selectListIncludesSearchDateIgnoreAuthCompany</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割</td>
* <td>divideTerm</td>
* <td>期間分割</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>期間分割(認可設定無視)</td>
* <td>divideTermIgnoreAuthCompany</td>
* <td>期間分割</td>
* <td><b>&nbsp;</b></td>
* </tr>
* <tr>
* <td>期間変更</td>
* <td>moveTerm</td>
* <td>期間変更</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間変更(認可設定無視)</td>
* <td>moveTermIgnoreAuthCompany</td>
* <td>期間変更</td>
* <td>&nbsp;</td>

```

```
* </tr>
* </table>
* <br>
* <br>
* 機能に対するI/Oを以下に示します。(上記一覧のカテゴリ単位)<br>
* <br>
* [IN]
* <table border="1">
* <tr>
* <td align="right">●</td>
* <td>: 必須</td>
* </tr>
* <tr>
* <td align="right">△</td>
* <td>: 任意、検索時完全一致条件</td>
* </tr>
* <tr>
* <td align="right">▲</td>
* <td>: 任意、検索時部分一致条件</td>
* </tr>
* <tr>
* <td align="right">x</td>
* <td>: 不要(設定されていても無視します。)</td>
* </tr>
* <tr>
* <td align="right">☆</td>
* <td>: 各APIの説明を参照</td>
* </tr>
* <tr>
* <td align="right">‐</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* [OUT]
* <table border="1">
* <tr>
* <td align="right">○</td>
* <td>: あり</td>
* </tr>
* <tr>
* <td align="right">x</td>
* <td>: なし</td>
* </tr>
* <tr>
* <td align="right">‐</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスとメソッドのI/Oマトリクス(共通)</b>
* <table border="1">
* <tr>
* <td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>
* <td rowspan="2" align="center"><b>DTOフィールド名</b></td>
* <td colspan="2" align="center"><b>主キー</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>汎用</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>ロケール</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>参照</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>キーワード</b><br>
* 検索</td>
* <td colspan="2" align="center"><b>削除フラグ</b><br>
* 更新</td>
* <td colspan="2" align="center"><b>登録</b></td>
* <td colspan="2" align="center"><b>削除</b></td>
* <td colspan="2" align="center"><b>更新</b></td>
* </tr>
* <tr>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>I</b></td>
* <td align="center"><b>O</b></td>
```

[illegible]

[illegible]

<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
<td align="center">x</td>
<td align="center"></td>
<td align="center">x</td>
<td align="center"></td>
</tr>
<tr>
<td>会社コード(上位)</td>
<td>parentCompanyCd</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"> Δ </td>
<td align="center"></td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"> $\blacktriangle$ </td>
<td align="center"></td>
<td align="center"></td>
<td align="center"> $\blacktriangle$ </td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"></td>
<td align="center"></td>
</tr>
<tr>
<td>プロジェクトコード(上位)</td>
<td>parentProjectCd</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"></td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"></td>
<td align="center">O</td>
<td align="center"> $\blacktriangle$ </td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
</tr>
<tr>
<td>予備項目A～J</td>
<td>extensionA～J</td>
<td align="center"></td>
<td align="center">O</td>
<td align="center"> Δ </td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"> $\blacktriangle$ </td>
<td align="center">O</td>
<td align="center"> $\blacktriangle$ </td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
<td align="center">x</td>
<td align="center"></td>
<td align="center"> Δ </td>
<td align="center"></td>
</tr>
<tr>
<td>登録日時</td>
<td>entryTs</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"> Δ </td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center">x</td>
<td align="center">O</td>
<td align="center"></td>
</tr>

[illegible]

```
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">△</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>組織名称</td>
* <td>companyName</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクト名称(上位)</td>
* <td>parentProjectName</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスとメソッドのI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <td rowspan="2" align="center"><b>DTOフィールド論理名</b></td>
* <td rowspan="2" align="center"><b>DTOフィールド名</b></td>
* <td colspan="2" align="center"><b>期間検索</b></td>
```



```

<td colspan=2 align= center ><U>期間検索</U></td>
* <td colspan="2" align="center"><b>基準日指定検索</b></td>
* <td colspan="2" align="center"><b>期間分割</b></td>
* <td colspan="2" align="center"><b>期間変更</b></td>
* </tr>
* <tr>
* <td align="center"><b>|</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>|</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>|</b></td>
* <td align="center"><b>O</b></td>
* <td align="center"><b>|</b></td>
* <td align="center"><b>O</b></td>
* </tr>
* <tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">O</td>
* <td align="center">△</td>
* <td align="center">O</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>プロジェクトコード(*)</b></td>
* <td><b>projectCd</b></td>
* <td align="center">●</td>
* <td align="center">O</td>
* <td align="center">△</td>
* <td align="center">O</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">O</td>
* <td align="center">△</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>開始日</b></td>
* <td><b>startDate</b></td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>終了日</b></td>
* <td><b>endDate</b></td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">O</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>

```

```

* <tr>
* <td>プロジェクト名称</td>
* <td>projectName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>会社コード(上位)</td>
* <td>parentCompanyCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクトコード(上位)</td>
* <td>parentProjectCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>

```

```

* <td>renewCnt</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新日時</td>
* <td>renewTs</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新者コード</td>
* <td>renewUserCd</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>組織名称</td>
* <td>companyName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>プロジェクト名称(上位)</td>
* <td>parentProjectName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <br>
* テーブルのPK項目にはフィールド名に(*)を付与しています。<br>
* </p>
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
public class ProjectManagerImpl extends AbstractTermMasterManager<ProjectManagerDto> implements ProjectManager {

    /** K30mProjectService. */
    @Resource
    protected K30mProjectService k30mProjectService;

    /**
     * {@inheritDoc}

```

```

*/

@Override
public ProjectManagerDto select(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.select(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */

@Override
public ProjectManagerDto selectIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.selectIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */

@Override
public List<ProjectManagerDto> selectList(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria,
    List<OrderDto> order) {
    return super.selectList(k30mProjectService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */

@Override
public List<ProjectManagerDto> selectListIgnoreAuthCompany(ProjectManagerDto criteria, List<OrderDto> order) {
    return super.selectListIgnoreAuthCompany(k30mProjectService, criteria, order);
}

/**
 * {@inheritDoc}
 */

@Override
public List<ProjectManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
    AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
        EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
    return super.selectBySql(k30mProjectService, "selectView.sql", criteria, order, authCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */

@Override
public List<ProjectManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
    String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
        EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
    return super.selectBySqlIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order, likeTarget);
}

/**
 * {@inheritDoc}
 */

@Override
public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
        EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
    return super.selectListByKeywords(k30mProjectService, "selectView.sql", criteria, order, keywordsCriteria,
        authCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */

@Override
public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria) {
    String[] likeTarget = {PROJECT_NAME, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D, EXTENSION_E,
        EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME, PARENT_PROJECT_NAME};
    return super.selectListByKeywordsIgnoreAuthCompany(k30mProjectService, "selectView.sql", criteria, order,
        keywordsCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 *
 * @throws KaidenOptimisticLockException
 */

@Override
public void toEnabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toEnabled(k30mProjectService, criteria, authCriteria);
}

```

```

/**
 * {@inheritDoc}
 */
* @throws KaidenOptimisticLockException
*/
@Override
public void toEnabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
    super.toEnabledIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabled(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    super.toDisabled(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void toDisabledIgnoreAuthCompany(ProjectManagerDto criteria) throws KaidenOptimisticLockException {
    super.toDisabledIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectLocales(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectLocalesIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectTerms(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k30mProjectService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectTermsIgnoreAuthCompany(ProjectManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k30mProjectService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entry(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
    return super.entry(k30mProjectService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int entryIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenDuplicationEntryException {
    return super.entryIgnoreAuthCompany(k30mProjectService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public int remove(ProjectManagerDto criteria, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException {

```

```

        return super.remove(k30mProjectService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeIgnoreAuthCompany(ProjectManagerDto criteria) {
        return super.removeIgnoreAuthCompany(k30mProjectService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renew(List<ProjectManagerDto> dataList, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        return super.renew(k30mProjectService, dataList, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int renewIgnoreAuthCompany(List<ProjectManagerDto> dataList) throws KaidenOptimisticLockException {
        return super.renewIgnoreAuthCompany(k30mProjectService, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
        super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTerm(ProjectManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
        String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
        KaidenValidateException {
        String termCd2nd = super.divideTerm(k30mProjectService, criteria, divideDate, authCriteria);
        if (StringUtil.isEmpty(deleteFlag2nd)) {
            ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
            criteria2nd.termCd = termCd2nd;
            if (GenericFlag.isOn(deleteFlag2nd)) {
                toDisabled(criteria2nd, authCriteria);
            } else if (GenericFlag.isOff(deleteFlag2nd)) {
                toEnabled(criteria2nd, authCriteria);
            } else {
                throw KaidenIllegalParametersException.makeInstance();
            }
        }
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate)
        throws KaidenOptimisticLockException, KaidenValidateException {
        super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void divideTermIgnoreAuthCompany(ProjectManagerDto criteria, Date divideDate, String deleteFlag2nd)
        throws KaidenOptimisticLockException, KaidenValidateException {
        String termCd2nd = super.divideTermIgnoreAuthCompany(k30mProjectService, criteria, divideDate);
        if (StringUtil.isEmpty(deleteFlag2nd)) {
            ProjectManagerDto criteria2nd = Beans.createAndCopy(ProjectManagerDto.class, criteria).execute();
            criteria2nd.termCd = termCd2nd;
            if (GenericFlag.isOn(deleteFlag2nd)) {
                toDisabledIgnoreAuthCompany(criteria2nd);
            } else if (GenericFlag.isOff(deleteFlag2nd)) {
                toEnabledIgnoreAuthCompany(criteria2nd);
            } else {
                throw KaidenIllegalParametersException.makeInstance();
            }
        }
    }
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ProjectManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mProjectService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ProjectManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mProjectService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * {@inheritDoc}
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, PROJECT_CD});
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDate(ProjectManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mProjectService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ProjectManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ProjectManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mProjectService, criteria, order);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }

    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {

    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }

    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(

```

```

        ErrorMode.ERROR_CONTINUE);
    }

    /**
     * {@inheritDoc}
     * <p>
     * 以下のバリデート定義のリストを返却します。
     * </p>
     * <table border="1">
     * <tr>
     * <th>項目名</th>
     * <th>検証内容</th>
     * </tr>
     * <tr>
     * <td>会社コード</td>
     * <td>必須、マスタ存在確認</td>
     * </tr>
     * <tr>
     * <td>プロジェクトコード</td>
     * <td>必須、皆伝コード、100桁</td>
     * </tr>
     * <tr>
     * <td>ロケールID</td>
     * <td>必須、ロケール妥当性</td>
     * </tr>
     * <tr>
     * <td>開始日</td>
     * <td>必須、日付(システム期間内)</td>
     * </tr>
     * <tr>
     * <td>終了日</td>
     * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
     * </tr>
     * <tr>
     * <td>プロジェクト名</td>
     * <td>必須、250文字</td>
     * </tr>
     * <tr>
     * <td>プロジェクトコード(上位プロジェクト)</td>
     * <td>半角英数、マスタ存在確認</td>
     * </tr>
     * <tr>
     * <td>会社コード(上位プロジェクト)</td>
     * <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
     * </tr>
     * <tr>
     * <td>削除フラグ</td>
     * <td>フラグ妥当性</td>
     * </tr>
     * <tr>
     * <td>プロジェクトコード(上位プロジェクト)</td>
     * <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
     * </tr>
     * </table>
     */
    @Override
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
        List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
        // 会社コード
        result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
        result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
        // プロジェクトコード
        result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
        result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
        result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
        // ロケールid
        result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
        result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
        // 開始日
        result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
        result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
        // 終了日
        result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
        result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
        result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
        // プロジェクト名称
        result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
        result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
        // プロジェクトコード(上位)
        result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
            "", "true"));
        result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));
        // 削除フラグ
        result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

        return result;
    }

```



```

}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード</td>
 * <td>必須、皆伝コード、100桁</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、皆伝コード、50桁</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>プロジェクト名</td>
 * <td>必須、250文字</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>会社コード(上位プロジェクト)</td>
 * <td>半角英数、マスタ存在確認※プロジェクトコード(上位プロジェクト)のマスタ存在確認に利用</td>
 * </tr>
 * <tr>
 * <td>プロジェクトコード(上位プロジェクト)</td>
 * <td>自身のプロジェクトと上位プロジェクトの同一チェック</td>
 * </tr>
 * </table>
 */

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMpany_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMpany_CD, VALID_COMPANY_EXISTS, COMpany_CD, "", "true"));
    // プロジェクトコード
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_MAX_LENGTH, String.valueOf(PROJECT_CD_LEN)));
    result.add(new ValidateDefsDto(PROJECT_CD, VALID_KAIDEN_ID));
    // 期間コード
    result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
    result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
    // ロケールid
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日
    result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
    result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
    // プロジェクト名称
    result.add(new ValidateDefsDto(PROJECT_NAME, VALID_REQUIRED));
    result.add(new ValidateDefsDto(PROJECT_NAME, VALID_MAX_LENGTH, String.valueOf(PROJECT_NAME_LEN)));
    // プロジェクトコード(上位)
    result.add(new ValidateDefsDto(PARENT_PROJECT_CD, VALID_PROJECT_EXISTS, PARENT_PROJECT_CD, PARENT_COMPANY_CD,
        "", "true"));
    result.add(new ValidateDefsDto(PARENT_PROJECT_CD, new CustomValidateNotEqualParent()));
}

```

```

    return result;
}

/** {@inheritDoc} */
@Override
public Class<ProjectManagerDto> getMainManagerDtoClass() {
    return ProjectManagerDto.class;
}

/** {@inheritDoc} */
@Override
public List<ProjectManagerDto> selectListIgnoreAuthCompanyWithLock(ProjectManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mProjectService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ProjectManagerDto.class,
        k30mProjectService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiled() {
    return k30mProjectService.getPkFiled();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ProjectManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicF<ProjectManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mProjectService, criteria, order, logic);
}
}

```

ManagerDto

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectCriteriaDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterMaintenanceCriteriaDto;

/**
 * プロジェクトマスタ 検索条件DTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectCriteriaDto extends TermMasterMaintenanceCriteriaDto {

    /** 会社コード. */
    // public String companyCd; //親クラスで定義済み

    /** プロジェクトコード. */
    public String projectCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールID. */
    // public String localeId; //親クラスで定義済み
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** プロジェクト名称. */
    public String projectName;
    /** 削除フラグ. */
    // public String deleteFlag; //親クラスで定義済み
    /** プロジェクトコード(上位). */
    public String parentProjectCd;
    /** 予備項目A. */
    public String extensionA;
    /** 予備項目B. */
    public String extensionB;
    /** 予備項目C. */
    public String extensionC;
    /** 予備項目D. */
    public String extensionD;
    /** 予備項目E. */
    public String extensionE;
    /** 予備項目F. */
    public String extensionF;
    /** 予備項目G. */
    public String extensionG;
    /** 予備項目H. */
    public String extensionH;
    /** 予備項目I. */
    public String extensionI;
    /** 予備項目J. */
    public String extensionJ;
    /** 会社名. */
    public String companyName;
    /** プロジェクト名称(上位). */
    public String parentProjectName;
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;

/**
 * プロジェクトDTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ProjectManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;

    /** プロジェクトコード. */
    public String projectCd;

    /** 期間コード. */
    public String termCd;

    /** ロケールID. */
    public String localeId;

    /** 開始日. */

```

```

public Timestamp startDate;

/** 終了日. */
public Timestamp endDate;

/** プロジェクト名称. */
public String projectName;

/** 削除フラグ. */
public String deleteFlag;

/** 会社コード(上位). */
public String parentCompanyCd;

/** プロジェクトコード(上位). */
public String parentProjectCd;

/** 予備項目A. */
public String extensionA;

/** 予備項目B. */
public String extensionB;

/** 予備項目C. */
public String extensionC;

/** 予備項目D. */
public String extensionD;

/** 予備項目E. */
public String extensionE;

/** 予備項目F. */
public String extensionF;

/** 予備項目G. */
public String extensionG;

/** 予備項目H. */
public String extensionH;

/** 予備項目I. */
public String extensionI;

/** 予備項目J. */
public String extensionJ;

/** 登録日時. */
public Timestamp entryTs;

/** 登録者コード. */
public String entryUserCd;

/** 更新カウンタ. */
public Long renewCnt;

/** 更新日時. */
public Timestamp renewTs;

/** 更新者コード. */
public String renewUserCd;

/** 組織名称. */
public String companyName;

/** プロジェクト名称(上位). */
public String parentProjectName;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/project/ProjectManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.project;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityExtensionFieldNames;

/**
 * プロジェクトマスタDTO フィールド名定義クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ProjectManagerDtoFieldName implements EntityExtensionFieldNames {

    /**
     * コンストラクタ.
     */
    private ProjectManagerDtoFieldName() {

    }

    /** 会社コード. */
    public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
    /** プロジェクトコード. */
    public static final String PROJECT_CD = "projectCd";
    /** 期間コード. */
    public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
    /** ロケールid. */
    public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
    /** プロジェクト名称. */
    public static final String PROJECT_NAME = "projectName";
    /** 開始日. */
    public static final String START_DATE = EntityCommonFieldNames.START_DATE;
    /** 終了日. */
    public static final String END_DATE = EntityCommonFieldNames.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;
    /** 会社コード(上位). */
    public static final String PARENT_COMPANY_CD = "parentCompanyCd";
    /** プロジェクトコード(上位). */
    public static final String PARENT_PROJECT_CD = "parentProjectCd";

    /** 会社名称(他マスタ名称). */
    public static final String COMPANY_NAME = "companyName";
    /** プロジェクト名称(上位). */
    public static final String PARENT_PROJECT_NAME = "parentProjectName";
}

```

Settings

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/model/xml_schema/project_import_settings/ProjectImportSettings.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.xml_schema.project_import_settings;

import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PARENT_PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.PROJECT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.project.ProjectManagerDtoFieldName.START_DATE;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;
import jp.co.slcs.kaiden2.base.foundation.model.xml_schema.generic_import_settings.GenericImportSettings;

/**
 * プロジェクトマスタ読込設定DTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@XmlRootElement
@XmlType(propOrder = {})
public class ProjectImportSettings extends GenericImportSettings {

    /** serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /**
     * フィールド定義配列.
     */
    protected String[] filedNames = new String[] {COMPANY_CD, PROJECT_CD, LOCALE_ID, START_DATE, END_DATE,
        PROJECT_NAME, DELETE_FLAG, PARENT_COMPANY_CD, PARENT_PROJECT_CD};

    /** {@inheritDoc} */
    @Override
    protected void initOrders() {
        super.getOrders().add(new Order(COMPANY_CD, Order.SORT_ASC));
        super.getOrders().add(new Order(PROJECT_CD, Order.SORT_ASC));
        super.getOrders().add(new Order(LOCALE_ID, Order.SORT_ASC));
    }

    /** {@inheritDoc} */
    @Override
    protected void initPkFields() {
        super.getPkFields().add(COMPANY_CD);
        super.getPkFields().add(PROJECT_CD);
        super.getPkFields().add(LOCALE_ID);
    }

    /** {@inheritDoc} */
    @Override
    protected void init18NFields() {
        super.get18NFields().add(PROJECT_NAME);
    }

    /** {@inheritDoc} */
    @Override
    protected void initFieldsOrder() {
        for (String j : filedNames) {
            super.getFieldsOrder().add(j);
        }
    }
}

```

Java (S2JDBC)

Entity

- Path
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProject.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**

```

```

* プロジェクト.
*
* @author Sumitomo Life Information Systems Co.,Ltd.
*/
@Entity
@Table(name = "k30m_project")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mProject extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** プロジェクトコード. */
    @Id
    @Column(name = "project_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String projectCd;

    /** 期間コード. */
    @Id
    @Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String termCd;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp endDate;

    /** プロジェクト名. */
    @Column(name = "project_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
    public String projectName;

    /** 削除フラグ. */
    @Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
    public String deleteFlag;

    /** 会社コード(上位プロジェクト). */
    @Column(name = "parent_company_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String parentCompanyCd;

    /** プロジェクトコード(上位プロジェクト). */
    @Column(name = "parent_project_cd", columnDefinition = "varchar2(100)", nullable = true, unique = false)
    public String parentProjectCd;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

    /** 予備項目b. */
    @Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionB;

    /** 予備項目c. */
    @Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionC;

    /** 予備項目d. */
    @Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionD;

    /** 予備項目e. */
    @Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionE;

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionG;

    /** 予備項目h. */
    @Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

    @Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false);
    public String extensionH;

    /** 予備項目i. */
    @Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionI;

    /** 予備項目j. */
    @Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mProjectNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30mProject} の名称クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mProjectNames {

    /**
     * デフォルトコンストラクタ。
     */
    private K30mProjectNames() {
        super();
    }

    /**
     * companyCd のプロパティ名を返却。
     *
     * @return companyCd のプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * projectCd のプロパティ名を返却。
     *
     * @return projectCd のプロパティ名
     */
    public static PropertyName<String> projectCd() {
        return new PropertyName<String>("projectCd");
    }

    /**
     * termCd のプロパティ名を返却。
     *
     * @return termCd のプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * localeId のプロパティ名を返却。
     *
     * @return localeId のプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * startDate のプロパティ名を返却。
     *
     * @return startDate のプロパティ名
     */
    public static PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>("startDate");
    }

    /**
     * endDate のプロパティ名を返却。
     *

```



```

    * @return endDateのプロパティ名
    */
    public static PropertyName<Timestamp> endDate() {
        return new PropertyName<Timestamp>("endDate");
    }

    /**
     * projectNameのプロパティ名を返却.
     *
     * @return projectNameのプロパティ名
     */
    public static PropertyName<String> projectName() {
        return new PropertyName<String>("projectName");
    }

    /**
     * deleteFlagのプロパティ名を返却.
     *
     * @return deleteFlagのプロパティ名
     */
    public static PropertyName<String> deleteFlag() {
        return new PropertyName<String>("deleteFlag");
    }

    /**
     * parentCompanyCdのプロパティ名を返却.
     *
     * @return parentCompanyCdのプロパティ名
     */
    public static PropertyName<String> parentCompanyCd() {
        return new PropertyName<String>("parentCompanyCd");
    }

    /**
     * parentProjectCdのプロパティ名を返却.
     *
     * @return parentProjectCdのプロパティ名
     */
    public static PropertyName<String> parentProjectCd() {
        return new PropertyName<String>("parentProjectCd");
    }

    /**
     * extensionAのプロパティ名を返却.
     *
     * @return extensionAのプロパティ名
     */
    public static PropertyName<String> extensionA() {
        return new PropertyName<String>("extensionA");
    }

    /**
     * extensionBのプロパティ名を返却.
     *
     * @return extensionBのプロパティ名
     */
    public static PropertyName<String> extensionB() {
        return new PropertyName<String>("extensionB");
    }

    /**
     * extensionCのプロパティ名を返却.
     *
     * @return extensionCのプロパティ名
     */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }
}

```

```

/**
 * extensionFのプロパティ名を返却.
 *
 * @return extensionFのプロパティ名
 */
public static PropertyName<String> extensionF() {
    return new PropertyName<String>("extensionF");
}

/**
 * extensionGのプロパティ名を返却.
 *
 * @return extensionGのプロパティ名
 */
public static PropertyName<String> extensionG() {
    return new PropertyName<String>("extensionG");
}

/**
 * extensionHのプロパティ名を返却.
 *
 * @return extensionHのプロパティ名
 */
public static PropertyName<String> extensionH() {
    return new PropertyName<String>("extensionH");
}

/**
 * extensionIのプロパティ名を返却.
 *
 * @return extensionIのプロパティ名
 */
public static PropertyName<String> extensionI() {
    return new PropertyName<String>("extensionI");
}

/**
 * extensionJのプロパティ名を返却.
 *
 * @return extensionJのプロパティ名
 */
public static PropertyName<String> extensionJ() {
    return new PropertyName<String>("extensionJ");
}

/**
 * entryTsのプロパティ名を返却.
 *
 * @return entryTsのプロパティ名
 */
public static PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>("entryTs");
}

/**
 * entryUserCdのプロパティ名を返却.
 *
 * @return entryUserCdのプロパティ名
 */
public static PropertyName<String> entryUserCd() {
    return new PropertyName<String>("entryUserCd");
}

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名

```

```

*/
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30mProjectNames extends PropertyName<K30mProject> {

    /**
     * インスタンスを構築.
     */
    public _K30mProjectNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30mProjectNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30mProjectNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public PropertyName<String> companyCd() {
        return new PropertyName<String>(this, "companyCd");
    }

    /**
     * projectCdのプロパティ名を返却.
     *
     * @return projectCdのプロパティ名
     */
    public PropertyName<String> projectCd() {
        return new PropertyName<String>(this, "projectCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public PropertyName<String> termCd() {
        return new PropertyName<String>(this, "termCd");
    }

    /**
     * localeIdのプロパティ名を返却.
     *
     * @return localeIdのプロパティ名
     */
    public PropertyName<String> localeId() {
        return new PropertyName<String>(this, "localeId");
    }

    /**
     * startDateのプロパティ名を返却.
     *
     * @return startDateのプロパティ名
     */
    public PropertyName<Timestamp> startDate() {
        return new PropertyName<Timestamp>(this, "startDate");
    }

    /**
     * endDateのプロパティ名を返却.
     *
     * @return endDateのプロパティ名
     */
    public PropertyName<Timestamp> endDate() {

```

```

    return new PropertyName<Timestamp>(this, "endDate");
}

/**
 * projectNameのプロパティ名を返却.
 *
 * @return projectNameのプロパティ名
 */
public PropertyName<String> projectName() {
    return new PropertyName<String>(this, "projectName");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>(this, "deleteFlag");
}

/**
 * parentCompanyCdのプロパティ名を返却.
 *
 * @return parentCompanyCdのプロパティ名
 */
public PropertyName<String> parentCompanyCd() {
    return new PropertyName<String>(this, "parentCompanyCd");
}

/**
 * parentProjectCdのプロパティ名を返却.
 *
 * @return parentProjectCdのプロパティ名
 */
public PropertyName<String> parentProjectCd() {
    return new PropertyName<String>(this, "parentProjectCd");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public PropertyName<String> extensionA() {
    return new PropertyName<String>(this, "extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却.
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却.
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却.

```

```

*
* @return extensionFのプロパティ名
*/
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
* extensionGのプロパティ名を返却.
*
* @return extensionGのプロパティ名
*/
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
* extensionHのプロパティ名を返却.
*
* @return extensionHのプロパティ名
*/
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
* extensionIのプロパティ名を返却.
*
* @return extensionIのプロパティ名
*/
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
* extensionJのプロパティ名を返却.
*
* @return extensionJのプロパティ名
*/
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
* entryTsのプロパティ名を返却.
*
* @return entryTsのプロパティ名
*/
public PropertyName<Timestamp> entryTs() {
    return new PropertyName<Timestamp>(this, "entryTs");
}

/**
* entryUserCdのプロパティ名を返却.
*
* @return entryUserCdのプロパティ名
*/
public PropertyName<String> entryUserCd() {
    return new PropertyName<String>(this, "entryUserCd");
}

/**
* renewCntのプロパティ名を返却.
*
* @return renewCntのプロパティ名
*/
public PropertyName<Long> renewCnt() {
    return new PropertyName<Long>(this, "renewCnt");
}

/**
* renewTsのプロパティ名を返却.
*
* @return renewTsのプロパティ名
*/
public PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>(this, "renewTs");
}

/**
* renewUserCdのプロパティ名を返却.
*
* @return renewUserCdのプロパティ名
*/
public PropertyName<String> renewUserCd() {
    return new PropertyName<String>(this, "renewUserCd");
}

```

```

    }
  }
}

```

Service

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mProject;

/**
 * {@link K30mProject} のサービスクラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mProjectService extends GenerateService<K30mProject> {
    //
}

```

Condition

- **Path**
src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30mProject} の条件クラス。
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mProjectCondition extends
    AbstractEntityCondition<K30mProjectCondition> {

    /**
     * インスタンスを構築。
     */
    public K30mProjectCondition() {
    }

    /**
     * インスタンスを構築。
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30mProjectCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** companyCdの条件。 */
    public NotNulltableStringCondition<K30mProjectCondition> companyCd =
        new NotNulltableStringCondition<K30mProjectCondition>("companyCd", this);

    /** projectCdの条件。 */
    public NotNulltableStringCondition<K30mProjectCondition> projectCd =
        new NotNulltableStringCondition<K30mProjectCondition>("projectCd", this);

    /** termCdの条件。 */
    public NotNulltableStringCondition<K30mProjectCondition> termCd =
        new NotNulltableStringCondition<K30mProjectCondition>("termCd", this);

    /** localeIdの条件。 */
    public NotNulltableStringCondition<K30mProjectCondition> localeId =
        new NotNulltableStringCondition<K30mProjectCondition>("localeId", this);

    /** startDateの条件。 */
    public NotNulltableCondition<K30mProjectCondition, Timestamp> startDate =
        new NotNulltableCondition<K30mProjectCondition, Timestamp>("startDate", this);
}

```

```

/** endDateの条件. */
public NotNullCondition<K30mProjectCondition, Timestamp> endDate =
    new NotNullCondition<K30mProjectCondition, Timestamp>("endDate", this);

/** projectNameの条件. */
public NotNullStringCondition<K30mProjectCondition> projectName =
    new NotNullStringCondition<K30mProjectCondition>("projectName", this);

/** deleteFlagの条件. */
public NotNullStringCondition<K30mProjectCondition> deleteFlag =
    new NotNullStringCondition<K30mProjectCondition>("deleteFlag", this);

/** parentCompanyCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentCompanyCd =
    new NullableStringCondition<K30mProjectCondition>("parentCompanyCd", this);

/** parentProjectCdの条件. */
public NullableStringCondition<K30mProjectCondition> parentProjectCd =
    new NullableStringCondition<K30mProjectCondition>("parentProjectCd", this);

/** extensionAの条件. */
public NullableStringCondition<K30mProjectCondition> extensionA =
    new NullableStringCondition<K30mProjectCondition>("extensionA", this);

/** extensionBの条件. */
public NullableStringCondition<K30mProjectCondition> extensionB =
    new NullableStringCondition<K30mProjectCondition>("extensionB", this);

/** extensionCの条件. */
public NullableStringCondition<K30mProjectCondition> extensionC =
    new NullableStringCondition<K30mProjectCondition>("extensionC", this);

/** extensionDの条件. */
public NullableStringCondition<K30mProjectCondition> extensionD =
    new NullableStringCondition<K30mProjectCondition>("extensionD", this);

/** extensionEの条件. */
public NullableStringCondition<K30mProjectCondition> extensionE =
    new NullableStringCondition<K30mProjectCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mProjectCondition> extensionF =
    new NullableStringCondition<K30mProjectCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30mProjectCondition> extensionG =
    new NullableStringCondition<K30mProjectCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30mProjectCondition> extensionH =
    new NullableStringCondition<K30mProjectCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30mProjectCondition> extensionI =
    new NullableStringCondition<K30mProjectCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30mProjectCondition> extensionJ =
    new NullableStringCondition<K30mProjectCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30mProjectCondition, Timestamp> entryTs =
    new NotNullCondition<K30mProjectCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullStringCondition<K30mProjectCondition> entryUserCd =
    new NotNullStringCondition<K30mProjectCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30mProjectCondition, Long> renewCnt =
    new NotNullCondition<K30mProjectCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30mProjectCondition, Timestamp> renewTs =
    new NotNullCondition<K30mProjectCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullStringCondition<K30mProjectCondition> renewUserCd =
    new NotNullStringCondition<K30mProjectCondition>("renewUserCd", this);
}

```

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mProjectService/selectView.sql

```

select
  prj.company_cd
, prj.project_cd
, prj.term_cd
, prj.locale_id
, prj.start_date
, prj.end_date
, prj.project_name
, prj.delete_flag
, prj.parent_company_cd
, prj.parent_project_cd
, prj.extension_a
, prj.extension_b
, prj.extension_c
, prj.extension_d
, prj.extension_e
, prj.extension_f
, prj.extension_g
, prj.extension_h
, prj.extension_i
, prj.extension_j
, prj.entry_ts
, prj.entry_user_cd
, prj.renew_cnt
, prj.renew_ts
, prj.renew_user_cd
, dept.department_name as company_name
, parent.project_name as parent_project_name
from k30m_project prj
  left join ( select imm_department.company_cd
                    , imm_department.department_set_cd
                    , imm_department.department_cd
                    , imm_department.locale_id
                    , imm_department.start_date
                    , imm_department.end_date
                    , imm_department.department_name
                  from imm_department) dept
on prj.company_cd = dept.company_cd
and prj.company_cd = dept.department_set_cd
and prj.company_cd = dept.department_cd
and prj.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/'2000/01/01'
and dept.end_date > /*searchCriteriaDate*/'2000/01/01'
--ELSE and dept.start_date <= prj.start_date
--ELSE and dept.end_date > prj.start_date
/*END*/
  left join k30m_project parent
on prj.parent_company_cd = parent.company_cd
and prj.parent_project_cd = parent.project_cd
and prj.locale_id = parent.locale_id
/*IF searchCriteriaDate != null*/
and parent.start_date <= /*searchCriteriaDate*/'2000/01/01'
and parent.end_date > /*searchCriteriaDate*/'2000/01/01'
--ELSE and parent.start_date <= prj.start_date
--ELSE and parent.end_date > prj.start_date
/*END*/
/*BEGIN*/
where
/*IF companyCd != null*/
  prj.company_cd in /*companyCd*/('comp')
/*END*/
/*IF projectCd != null*/
and prj.project_cd = /*projectCd*/'projectCd'
/*END*/
/*IF termCd != null*/
and prj.term_cd = /*termCd*/'termCd'
/*END*/
/*IF localeId != null*/
and prj.locale_id = /*localeId*/'localeId'
/*END*/
/*IF searchCriteriaDate != null*/
and prj.start_date <= /*searchCriteriaDate*/'1900/01/01'
and prj.end_date > /*searchCriteriaDate*/'2999/12/31'
/*END*/
/*IF projectName != null*/
and prj.project_name like /*projectName*/'% ' ESCAPE '$'
/*END*/
/*IF deleteFlag != null*/
and prj.delete_flag = /*deleteFlag*/'0'
/*END*/

```



```

/*IF parentCompanyCd != null*/
and prj.parent_company_cd = /*parentCompanyCd*/parentCompanyCd'
/*END*/
/*IF parentProjectCd != null*/
and prj.parent_project_cd = /*parentProjectCd*/parentProjectCd'
/*END*/
/*IF extensionA != null*/
and prj.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and prj.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and prj.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and prj.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/
/*IF extensionE != null*/
and prj.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
and prj.extension_f like /*extensionF*/extensionF' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
and prj.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
and prj.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
and prj.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
and prj.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF parentProjectName != null*/
and parent.project_name like /*parentProjectName*/parentProjectName' ESCAPE '$'
/*END*/
/*END*/
/*IF orderBy != null*/
ORDER BY /*$orderBy*/company_cd
/*END*/

```

XML

- **Path**
src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/foundation/model/xml_schema/project_import_settings/ProjectImportSettings.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericImportSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericImportSettings" xmlns:tns="http://kaiden.slcs.co.jp/xmlschema/genericImportSettings">

  <!-- デフォルト設定値 -->
  <defaultSettings>
    <import_mode>1</import_mode>
    <start_date></start_date>
    <end_date></end_date>
    <shift_date></shift_date>
    <company_cd></company_cd>
    <character>UTF-8</character>
    <delimiter></delimiter>
    <enclosing>&quot;</enclosing>
    <normal_log>true</normal_log>
    <target_files>
      <target_file>kaiden/import/project/project_import.csv</target_file>
    </target_files>
    <start_rows>0</start_rows>
    <archive_files>
      <archive_file>kaiden/archive/import/project/project_import_{YMDHMSN}.csv</archive_file>
    </archive_files>
    <delete_flag>false</delete_flag>
  </defaultSettings>

  <!-- 会社ごとの設定(サンプル) -->
  <!-- companySettings -->
  <companySetting>
    <settings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="settings">
      <import_mode>1</import_mode>
      <start_date></start_date>
      <end_date></end_date>
      <shift_date>1</shift_date>
      <company_cd>sample_company_01</company_cd>
      <character>UTF-8</character>
      <delimiter></delimiter>
      <enclosing>&quot;</enclosing>
      <normal_log>true</normal_log>
      <target_files>
        <target_file>sample_company_01_project.csv</target_file>
      </target_files>
      <start_rows>1</start_rows>
      <archive_files>
        <archive_file>archive/sample_company_01_project.csv</archive_file>
      </archive_files>
      <delete_flag>true</delete_flag>
    </settings>
  </companySetting>
</companySettings-->
  <!-- Import時ソート順リスト -->
</genericImportSettings>

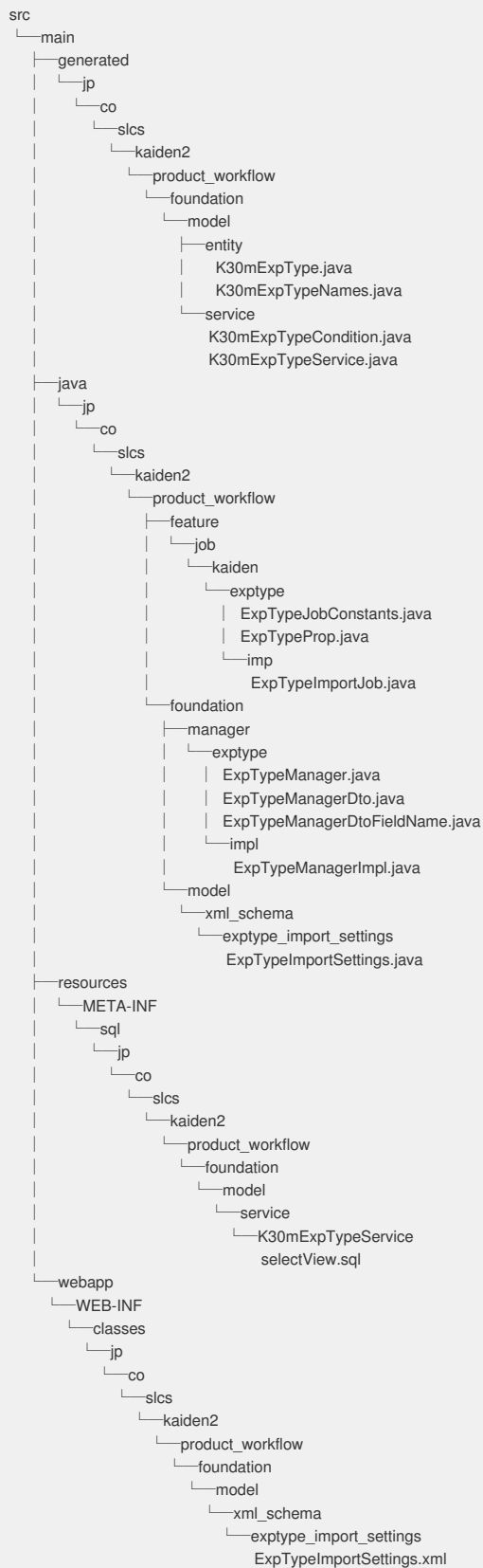
```

経費区分マスタインポート

経費区分マスタインポートのサンプルソースです。
本サンプルソースは「intra-mart Accel Kaiden! ワークフローモジュール 2.0.2」から抜粋したソースです。

■	一覧
■	Java
■	Job
■	Manager
■	Manager (実装)
■	ManagerDto
■	Settings
■	Java (S2JDBC)
■	Entity
■	Names
■	Service
■	Condition
■	SQL
■	検索
■	XML

[一覧](#)



Java

Job

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/exptype/imp/ExpTypeImportJob.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.exptype.imp;

import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.job.AbstractTermMasterImportJob;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobProcessType;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.xml_schema.generic_jop_process_result.GenericJobProcessResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.exptype.ExpTypeJobConstants;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.xml_schema.exptype_import_settings.ExpTypeImportSettings;
import org.seasar.framework.container.SingletonS2Container;

/**
 * 経費区分マスタImporter.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeImportJob extends AbstractTermMasterImportJob<ExpTypeImportSettings, GenericJobProcessResult> {

    /** ExpTypeManager. */
    private final ExpTypeManager expTypeManager;

    /**
     * コンストラクタ.
     */
    public ExpTypeImportJob() {
        super(KaidenJobProcessType.IMPORT, ExpTypeJobConstants.KAIDEN_JOB_ID_IMPORT, ErrorMode.ERROR_CONTINUE);
        expTypeManager = SingletonS2Container.getComponent("expTypeManager");
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected ExpTypeImportSettings readSettingsForImport() {
        return super.read(ExpTypeImportSettings.class);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected GenericJobProcessResult getGenericJobProcessResult() {
        return new GenericJobProcessResult();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected LinkedHashMap<String, Map<String, String>> getSurfacesMap() {
        LinkedHashMap<String, Map<String, String>> surfacesMap = new LinkedHashMap<String, Map<String, String>>();
        surfacesMap.put(createUniqueKey(TARGET_MASTER_KEY), ExpTypeJobConstants.EXPTYPE_MST_SURFACES);
        return surfacesMap;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    protected List<ValidateDefsDto> addValidateRuleSub() {
        return expTypeManager.getValidateDefsDtoList4Entry();
    }

    /** {@inheritDoc} */
    @Override
    protected <DTO> TermMasterManagerForJob<? extends KaidenDto> getDefaultManager() {
        return expTypeManager;
    }
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/exptype/ExpTypeJobConstants.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.exptype;

import java.util.LinkedHashMap;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobIdIF;

/**
 * 経費区分マスタImporter/expoter定数.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpTypeJobConstants {

    /**
     * コンストラクタ.
     */
    private ExpTypeJobConstants() {

    }

    /** 処理対象機能皆伝ジョブID:経費区分マスタ. */
    public static final KaidenJobIdIF KAIDEN_JOB_ID_IMPORT = ExpTypeProp.KaidenJobID.KAIDEN_JOB_ID_IMPORT;
    /** 経費区分マスタラベル定義. */
    public static final LinkedHashMap<String, String> EXPTYPE_MST_SURFACES = ExpTypeProp.Surfaces.getSurfaces();
    /** 経費区分マスタファイルIndex. */
    public static final int DATA_FILE_INDEX = 0;
}

```

- Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/feature/job/kaiden/exptype/ExpTypeProp.java

```

package jp.co.slcs.kaiden2.product_workflow.feature.job.kaiden.exptype;

import java.util.LinkedHashMap;
import jp.co.slcs.kaiden2.base.foundation.conf.BaseProp;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden2.base.foundation.job.KaidenJobIdIF;
import jp.co.slcs.kaiden2.base.foundation.job.SurfacesIF;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ExtensionsProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.conf.ProductWorkflowProp;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName;
import org.seasar.framework.container.SingletonS2Container;

/**
 * 経費区分マスタDBカラムプロパティ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpTypeProp {

    /**
     * 経費区分マスタDBカラム.
     *
     * @author Sumitomo Life Information Systems Co.,Ltd.
     */
    public static final class C {

        /** 皆伝ジョブID. */
        public static final String KAIDEN_JOB_ID_IMPORT = "kaiden-job-master-import-exptype";
        /** 会社コード. */
        public static final String COMPANY_CD = ExpTypeManagerDtoFieldName.COMPANY_CD;
        /** 経費区分コード. */
        public static final String EXPTYPE_CD = ExpTypeManagerDtoFieldName.EXP_TYPE_CD;
        /** 期間コード. */
        public static final String TERM_CD = ExpTypeManagerDtoFieldName.TERM_CD;
        /** ロケールId. */
        public static final String LOCALE_ID = ExpTypeManagerDtoFieldName.LOCALE_ID;
        /** 経費区分名称. */
        public static final String EXPTYPE_NAME = ExpTypeManagerDtoFieldName.EXP_TYPE_NAME;
        /** 開始日. */
        public static final String START_DATE = ExpTypeManagerDtoFieldName.START_DATE;
        /** 終了日. */
        public static final String END_DATE = ExpTypeManagerDtoFieldName.END_DATE;
        /** 経費区分カテゴリ. */
        public static final String EXP_TYPE_CATEGORY = ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY;
        /** 勘定科目コード. */
        public static final String ACCOUNT_CD = ExpTypeManagerDtoFieldName.ACCOUNT_CD;
        /** 税区分コード. */
        public static final String TAX_TYPE_CD = ExpTypeManagerDtoFieldName.TAX_TYPE_CD;
        /** ソートキー. */
        public static final String SORT_KEY = ExpTypeManagerDtoFieldName.SORT_KEY;
        /** 削除フラグ. */
        public static final String DELETE_FLAG = ExpTypeManagerDtoFieldName.DELETE_FLAG;
    }
}

```

```

* 経費区分マスタDBカラム名称.
*

* @author Sumitomo Life Information Systems Co.,Ltd.
*/

public static final class S {

    /** ジョブ名称. */
    public static final String JOB_NAME_IMPORT = ProductWorkflowProp.S.ExpTypeJob.JOB_NAME_IMPORT;
    /** 会社コード. */
    public static final String COMPANY_CD = BaseProp.S.COMPANY_CD;
    /** 経費区分コード. */
    public static final String EXPTYPE_CD = ProductWorkflowProp.S.EXP_TYPE_CD;
    /** 期間コード. */
    public static final String TERM_CD = BaseProp.S.TERM_CD;
    /** ロケールid. */
    public static final String LOCALE_ID = BaseProp.S.LOCALE_ID;
    /** 経費区分名称. */
    public static final String EXPTYPE_NAME = ProductWorkflowProp.S.EXP_TYPE_NAME;
    /** 開始日. */
    public static final String START_DATE = BaseProp.S.START_DATE;
    /** 終了日. */
    public static final String END_DATE = BaseProp.S.END_DATE;
    /** 削除フラグ. */
    public static final String DELETE_FLAG = BaseProp.S.DELETE_FLAG;
    /** 経費区分カテゴリ. */
    public static final String EXP_TYPE_CATEGORY = ProductWorkflowProp.S.ExpType.EXP_TYPE_CATEGORY;
    /** 勘定科目コード. */
    public static final String ACCOUNT_CD = ProductWorkflowProp.S.ACCOUNT_CD;
    /** 税区分コード. */
    public static final String TAX_TYPE_CD = ProductWorkflowProp.S.TAX_TYPE_CD;
    /** ソートキー. */
    public static final String SORT_KEY = BaseProp.S.SORT_KEY;
}

/**
 * 経費区分マスタJob IDの列挙.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public static enum KaidenJobID implements KaidenJobIdIF {

    /** 皆伝ジョブID. */
    KAIDEN_JOB_ID_IMPORT(ExpTypeProp.C.KAIDEN_JOB_ID_IMPORT, ExpTypeProp.S.JOB_NAME_IMPORT);

    /** 皆伝ジョブID. */
    private final String kaidenJobId;
    /** ジョブ名称用サーフェスキー. */
    private final String surfaceKey;

    /**
     * コンストラクタ.
     *
     * @param kaidenJobId ジョブID
     * @param surfaceKey サーフェスキー
     */
    private KaidenJobID(String kaidenJobId, String surfaceKey) {
        this.kaidenJobId = kaidenJobId;
        this.surfaceKey = surfaceKey;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getKaidenJobId() {
        return kaidenJobId;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public String getKaidenJobName() {
        return SingletonS2Container.getComponent(PropertyHelper.class).getSurface(surfaceKey);
    }
}

/**
 * 経費区分マスタラベル定義の列挙.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public enum Surfaces implements SurfacesIF {

    /** 会社コード. */
    COMPANY_CD(ExpTypeProp.C.COMPANY_CD, ExpTypeProp.S.COMPANY_CD),

```

```

/** 経費区分コード. */
EXPTYPE_CD(ExpTypeProp.C.EXPTYPE_CD, ExpTypeProp.S.EXPTYPE_CD),
/** 期間コード. */
TERM_CD(ExpTypeProp.C.TERM_CD, ExpTypeProp.S.TERM_CD),
/** ロケールId. */
LOCALE_ID(ExpTypeProp.C.LOCALE_ID, ExpTypeProp.S.LOCALE_ID),
/** 経費区分名称. */
EXPTYPE_NAME(ExpTypeProp.C.EXPTYPE_NAME, ExpTypeProp.S.EXPTYPE_NAME),
/** 開始日. */
START_DATE(ExpTypeProp.C.START_DATE, ExpTypeProp.S.START_DATE),
/** 終了日. */
END_DATE(ExpTypeProp.C.END_DATE, ExpTypeProp.S.END_DATE),
/** 経費区分カテゴリ. */
EXP_TYPE_CATEGORY(ExpTypeProp.C.EXP_TYPE_CATEGORY, ExpTypeProp.S.EXP_TYPE_CATEGORY),
/** 勘定科目コード. */
ACCOUNT_CD(ExpTypeProp.C.ACCOUNT_CD, ExpTypeProp.S.ACCOUNT_CD),
/** 税区分コード. */
TAX_TYPE_CD(ExpTypeProp.C.TAX_TYPE_CD, ExpTypeProp.S.TAX_TYPE_CD),
/** ソートキー. */
SORT_KEY(ExpTypeProp.C.SORT_KEY, ExpTypeProp.S.SORT_KEY),
/** 削除フラグ. */
DELETE_FLAG(ExpTypeProp.C.DELETE_FLAG, ExpTypeProp.S.DELETE_FLAG);

/** ラベルキー. */
private final String key;
/** ラベル名称. */
private final String name;

/**
 * コンストラクタ.
 *
 * @param columnName ラベルキー
 * @param surfaceKey サーフスキー
 */
private Surfaces(String columnName, String surfaceKey) {
    key = columnName;
    name = SingletonS2Container.getComponent(PropertyHelper.class).getSurface(surfaceKey);
}

/**
 * {@inheritDoc}
 */
@Override
public String getKey() {
    return key;
}

/**
 * {@inheritDoc}
 */
@Override
public String getValue() {
    return name;
}

/**
 * ラベルキーから自身を返却.
 *
 * @param key ラベルキー値
 * @return 自身
 */
public static Surfaces codeOf(String key) {
    for (Surfaces that : Surfaces.values()) {
        if (that.getKey().equals(key)) {
            return that;
        }
    }
    throw KaidenIllegalParametersException.makeInstance();
}

/**
 * ラベル名称から自身を返却.
 *
 * @param name ラベル名称
 * @return 自身
 */
public static Surfaces nameOf(String name) {
    for (Surfaces that : Surfaces.values()) {
        if (that.getValue().equals(name)) {
            return that;
        }
    }
    throw KaidenIllegalParametersException.makeInstance();
}

```



```

* ラベル定義を取得します。
*
* @return ラベル定義
*/
public static LinkedHashMap<String, String> getSurfaces() {

    LinkedHashMap<String, String> surfaces = new LinkedHashMap<String, String>();

    for (Surfaces exptypeSurfaces : Surfaces.values()) {
        surfaces.put(exptypeSurfaces.getKey(), exptypeSurfaces.getValue());
    }

    surfaces.putAll(ExtensionsProp.Surfaces.getSurfaces());

    return surfaces;
}
}
}

```

Manager

- **Path**
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManager.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.util.List;
import java.util.Map;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.master.TermMasterManagerForJob;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;

/**
 * 経費区分マスタへの各種操作を提供するAPIのinterface.<br>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public interface ExpTypeManager extends TermMasterManagerForJob<ExpTypeManagerDto> {

    /** 経費区分コード. */
    public static final int EXP_TYPE_CD_LEN = 100;

    /** 経費区分名称. */
    public static final int EXP_TYPE_NAME_LEN = 250;

    /** 経費区分カテゴリ. */
    public static final int EXP_TYPE_CATEGORY_LEN = 250;

    /** 勘定科目コード. */
    public static final int ACCOUNT_CD_LEN = 100;

    /** 税区分コード. */
    public static final int TAX_TYPE_CD_LEN = 100;

    /** ソートキー. */
    public static final int SORT_KEY_LEN = 15;

    /**
     * 新規登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Entry();

    /**
     * 更新登録用の検証定義リストを返却します.
     *
     * @return バリデート実行クラス用パラメタDTOリスト
     */
    public List<ValidateDefsDto> getValidateDefsDtoList4Renew();

    /**
     * マスタ検索部品向け 一覧取得.
     *
     * @param criteria 検索条件
     * @param order ソート順
     * @param filter フィルタークラス
     * @return 検索結果
     */
    public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter);
}

```

Manager(実装)

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/impl/ExpTypeManagerImpl.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl;

import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldLength.TERM_CD_LEN;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_A;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_B;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_C;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_D;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_E;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_F;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_G;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_H;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_I;
import static jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames.EXTENSION_J;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_COMPANY_EXISTS;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_DATE_MIN;

```

```

import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_FLAG;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_INTEGER;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_KAIDEN_ID;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_LOCALE;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_MAX_LENGTH;
import static jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ConstValidationIdKaidenBase.VALID_REQUIRED;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.ACCOUNT_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.COMPANY_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.DELETE_FLAG;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.END_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.EXP_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.LOCALE_ID;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.SORT_KEY;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.START_DATE;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TAX_TYPE_NAME;
import static jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDtoFieldName.TERM_CD;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_ACCOUNT_EXISTS;
import static jp.co.slcs.kaiden2.product_workflow.foundation.util.validate.conf.ConstValidationIdKaidenProductWorkflow.VALID_TAX_TYPE_EXISTS;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import javax.annotation.Resource;
import jp.co.slcs.kaiden2.base.foundation.conf.GenericFlag;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenApplicationException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenDuplicationEntryException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenHaventAuthorityToCompanyException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenOptimisticLockException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden2.base.foundation.exception.KaidenValidateException;
import jp.co.slcs.kaiden2.base.foundation.manager.AbstractTermMasterManager;
import jp.co.slcs.kaiden2.base.foundation.manager.AuthorityCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.KeywordsCriteriaDto;
import jp.co.slcs.kaiden2.base.foundation.manager.common.IterateBusinessLogicIF;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenDto;
import jp.co.slcs.kaiden2.base.foundation.model.dto.OrderDto;
import jp.co.slcs.kaiden2.base.foundation.model.service.SqlResultFilterIF;
import jp.co.slcs.kaiden2.base.foundation.util.CollectionUtil;
import jp.co.slcs.kaiden2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden2.base.foundation.util.validate.KaidenValidateResult;
import jp.co.slcs.kaiden2.base.foundation.util.validate.conf.ErrorMode;
import jp.co.slcs.kaiden2.base.foundation.util.validate.model.dto.ValidateDefsDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManager;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.ExpTypeManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.service.K30mExpTypeService;
import org.seasar.framework.beans.util.Beat;

/**
 * 経費区分マスタへの各種操作を提供するAPI.<br>
 * <br>
 * <p>
 * 本APIは、主にデータベース操作を行います。より業務的な機能を付加した操作を提供します。<br>
 * 使用する際には、{@link ExpTypeManagerDto}クラスまたはそのリストをパラメータとして渡すと、<br>
 * {@link ExpTypeManagerDto}クラスまたはそのリストが戻り値として返却されます。<br>
 * (一部例外あり。各APIを参照。)<br>
 * <br>
 * <b>機能一覧(共通)</b>
 * <table border="1">
 * <tr>
 * <th>機能概要</th>
 * <th>メソッド</th>
 * <th>カテゴリ</th>
 * <th>備考</th>
 * </tr>
 * <tr>
 * <td>主キー検索</td>
 * <td>select</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>主キー検索(認可設定無視)</td>
 * <td>selectIgnoreAuthCompany</td>
 * <td>主キー検索</td>
 * <td>&nbsp;</td>
 * </tr>
 * <tr>
 * <td>汎用検索</td>

```

```

<td>selectList</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>汎用検索(認可設定無視)</td>
* <td>selectListIgnoreAuthCompany</td>
* <td>汎用検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索</td>
* <td>selectLocales</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>ロケール検索(認可設定無視)</td>
* <td>selectLocalesIgnoreAuthCompany</td>
* <td>ロケール検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索</td>
* <td>selectListWithReference</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>参照検索(認可設定無視)</td>
* <td>selectListWithReferenceIgnoreAuthCompany</td>
* <td>参照検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索</td>
* <td>selectListKeywordFilter</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>キーワード検索(認可設定無視)</td>
* <td>selectListKeywordFilterIgnoreAuthCompany</td>
* <td>キーワード検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化</td>
* <td>toEnabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>有効化(認可設定無視)</td>
* <td>toEnabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化</td>
* <td>toDisabled</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>無効化(認可設定無視)</td>
* <td>toDisabledIgnoreAuthCompany</td>
* <td>削除フラグ更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録</td>
* <td>entry</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>登録(認可設定無視)</td>
* <td>entryIgnoreAuthCompany</td>
* <td>登録</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>削除</td>
* <td>remove</td>
* <td>削除</td>
* <td>&nbsp;</td>

```

```

</td><td></td>
* </tr>
* <tr>
* <td>削除 (認可設定無視)</td>
* <td>removeIgnoreAuthCompany</td>
* <td>削除</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新</td>
* <td>renew</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>更新 (認可設定無視)</td>
* <td>renewIgnoreAuthCompany</td>
* <td>更新</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <b>機能一覧(期間化)</b>
* <table border="1">
* <tr>
* <th>機能概要</th>
* <th>メソッド</th>
* <th>カテゴリ</th>
* <th>備考</th>
* </tr>
* <tr>
* <td>期間検索</td>
* <td>selectTerms</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間検索 (認可設定無視)</td>
* <td>selectTermsIgnoreAuthCompany</td>
* <td>期間検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索</td>
* <td>selectListIncludesSearchDate</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>基準日指定検索 (認可設定無視)</td>
* <td>selectListIncludesSearchDateIgnoreAuthCompany</td>
* <td>基準日指定検索</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割</td>
* <td>divideTerm</td>
* <td>期間分割</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間分割 (認可設定無視)</td>
* <td>divideTermIgnoreAuthCompany</td>
* <td>期間分割</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間変更</td>
* <td>moveTerm</td>
* <td>期間変更</td>
* <td>&nbsp;</td>
* </tr>
* <tr>
* <td>期間変更 (認可設定無視)</td>
* <td>moveTermIgnoreAuthCompany</td>
* <td>期間変更</td>
* <td>&nbsp;</td>
* </tr>
* </table>
* <br>
* <br>
* 機能に対するI/Oを以下に示します。(上記一覧のカテゴリ単位)<br>
* <br>
* [IN]
* <table border="1">
* <tr>
* <td align="right">●</td>

```

```
* <td align="right">△</td>
* </tr>
* <tr>
* <td align="right">△</td>
* <td>: 任意、検索時完全一致条件</td>
* </tr>
* <tr>
* <td align="right">▲</td>
* <td>: 任意、検索時部分一致条件</td>
* </tr>
* <tr>
* <td align="right">x</td>
* <td>: 不要(設定されていても無視します。)</td>
* </tr>
* <tr>
* <td align="right">☆</td>
* <td>: 各APIの説明を参照</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* [OUT]
* <table border="1">
* <tr>
* <td align="right">○</td>
* <td>: あり</td>
* </tr>
* <tr>
* <td align="right">x</td>
* <td>: なし</td>
* </tr>
* <tr>
* <td align="right">—</td>
* <td>: 対象外</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(共通)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">主キー<br>
* 検索</th>
* <th colspan="2">汎用<br>
* 検索</th>
* <th colspan="2">ロケール<br>
* 検索</th>
* <th colspan="2">参照<br>
* 検索</th>
* <th colspan="2">キーワード<br>
* 検索</th>
* <th colspan="2">削除フラグ<br>
* 更新</th>
* <th colspan="2">登録</th>
* <th colspan="2">削除</th>
* <th colspan="2">更新</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <td><b>会社コード(*)</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
```

<td align="center"></td>
*<td align="center">○</td>
*<td align="center">●</td>
*<td align="center">"center">○</td>
*<td align="center">"center">△</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">"center">●</td>
*<td align="center">-</td>
*<td align="center">●</td>
*<td align="center">"center">○</td>
*<td align="center">●</td>
*<td align="center">-</td>
*<td align="center">●</td>
*<td align="center">-</td>
*
</tr>
<tr>
<td>経費区分コード()</td>
*<td>expTypeCd</td>
*<td align="center">●</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">●</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">●</td>
*<td align="center">-</td>
*<td align="center">●</td>
*<td align="center">-</td>
*<td align="center">○</td>
*<td align="center">-</td>
*
</tr>
<tr>
<td>期間コード()</td>
*<td>termCd</td>
*<td align="center">●</td>
*<td align="center">○</td>
*<td align="center">○</td>
*<td align="center">●</td>
*<td align="center">○</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">●</td>
*<td align="center">-x</td>
*<td align="center">-</td>
*<td align="center">"center">x</td>
*<td align="center">-</td>
*<td align="center">●</td>
*<td align="center">-</td>
*
</tr>
<tr>
<td>ロケールID()</td>
*<td>localeId</td>
*<td align="center">○</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">△</td>
*<td align="center">○</td>
*<td align="center">x</td>
*<td align="center">-</td>
*<td align="center">●</td>
*<td align="center">-</td>
*
</tr>
<tr>
*<td>経費区分名称</td>
*<td>expTypeName</td>

[illegible]

[illegible]

[illegible]

[illegible]

```

<td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </tr>
* <td>税区分名称</td>
* <td>taxTypeName</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">▲</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* </tr>
* <td>検索基準日</td>
* <td>searchCriteriaDate</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">☆</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">-</td>
* </tr>
* </table>
* <br>
* <b>DTOクラスと機能のI/Oマトリクス(期間化)</b>
* <table border="1">
* <tr>
* <th rowspan="2">DTOフィールド論理名</th>
* <th rowspan="2">DTOフィールド名</th>
* <th colspan="2">期間検索</th>
* <th colspan="2">基準日指定</th>
* <th colspan="2">検索</th>
* <th colspan="2">期間分割</th>
* <th colspan="2">期間変更</th>
* </tr>
* <tr>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* <th>I</th>
* <th>O</th>
* </tr>
* <tr>
* <td><b>会社コード(")</b></td>
* <td><b>companyCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* </tr>
* <td><b>経費区コード(")</b></td>

```

```

<td><b>expTypeCd</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>期間コード(*)</b></td>
* <td><b>termCd</b></td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">●</td>
* <td align="center">-</td>
* <td align="center">●</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td><b>ロケールID(*)</b></td>
* <td><b>localeId</b></td>
* <td align="center">●</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分名称</td>
* <td>expTypeName</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>経費区分カテゴリ</td>
* <td>expTypeCategory</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>開始日</td>
* <td>startDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>終了日</td>
* <td>endDate</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">○</td>
* <td align="center">x</td>
* <td align="center">-</td>
* <td align="center">x</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>勘定科目コード</td>
* <td>accountCd</td>
* <td align="center">x</td>
* <td align="center">○</td>

```

```

<td align="center"></td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>税区分コード</td>
* <td>taxTypeCd</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>ソートキー</td>
* <td>sortKey</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>削除フラグ</td>
* <td>deleteFlag</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>予備項目A～J</td>
* <td>extensionA～J</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録日時</td>
* <td>entryTs</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>登録者コード</td>
* <td>entryUserCd</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>
* <td align="center">×</td>
* <td align="center">-</td>
* <td align="center">×</td>
* <td align="center">-</td>
* </tr>
* <tr>
* <td>更新カウンタ</td>
* <td>renewCnt</td>
* <td align="center">×</td>
* <td align="center">○</td>
* <td align="center">△</td>
* <td align="center">○</td>

```

```

        <td align="center">☆</td>
        <td align="center"></td>
        <td align="center">☆</td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>更新日時</td>
        <td>renewTs</td>
        <td align="center">x</td>
        <td align="center">○</td>
        <td align="center">△</td>
        <td align="center">○</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>
    <td>更新者コード</td>
    <td>renewUserCode</td>
    <td align="center">x</td>
    <td align="center">○</td>
    <td align="center">△</td>
    <td align="center">○</td>
    <td align="center">x</td>
    <td align="center"></td>
    <td align="center">x</td>
    <td align="center"></td>
    <td align="center"></td>
    </tr>
    <tr>
        <td>会社名称</td>
        <td>companyName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>勘定科目名称</td>
        <td>accountName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>税区分名称</td>
        <td>taxTypeName</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>検索基準日</td>
        <td>searchCriteriaDate</td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">●</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center">x</td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>
    <tr>
        <td>
            テーブルのPK項目にはフィールド名に(*)を付与しています。<br>
        </td>
    </tr>
    </table>
    <br>
    @author Sumitomo Life Information Systems Co.,Ltd.
    */
    public class ExpTypeManagerImpl extends AbstractTermMasterManager<ExpTypeManagerDto> implements ExpTypeManager {

        /** K30mExpTvdeService. */

```

```

@Resource
protected K30mExpTypeService k30mExpTypeService;

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto select(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.select(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public ExpTypeManagerDto selectIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectList(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria,
    List<OrderDto> order) {
    return super.selectList(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompany(ExpTypeManagerDto criteria, List<OrderDto> order) {
    return super.selectListIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocales(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectLocales(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectLocalesIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectLocalesIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReference(KaidenDto criteria, List<OrderDto> order,
    AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySql(k30mExpTypeService, "selectView.sql", criteria, order, authCriteria, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListWithReferenceIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectBySqlIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order, likeTarget);
}

/**
 * {@inheritDoc}
 */
@Override
public Map<String, Object> selectListKeywordFilter(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, AuthorityCriteriaDto authCriteria) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    return super.selectListByKeywords(k30mExpTypeService, "selectView.sql", criteria, order, keywordsCriteria,
        authCriteria, likeTarget);
}

```



```

    }

    /**
     * {@inheritDoc}
     */
    @Override
    public Map<String, Object> selectListKeywordFilterIgnoreAuthCompany(KaidenDto criteria, List<OrderDto> order,
        KeywordsCriteriaDto keywordsCriteria) {
        String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
            EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
            ACCOUNT_NAME, TAX_TYPE_NAME};
        return super.selectListByKeywordsIgnoreAuthority(k30mExpTypeService, "selectView.sql", criteria, order,
            keywordsCriteria, likeTarget);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toEnabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        super.toEnabled(k30mExpTypeService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toEnabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
        super.toEnabledIgnoreAuthCompany(k30mExpTypeService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toDisabled(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
        super.toDisabled(k30mExpTypeService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void toDisabledIgnoreAuthCompany(ExpTypeManagerDto criteria) throws KaidenOptimisticLockException {
        super.toDisabledIgnoreAuthCompany(k30mExpTypeService, criteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entry(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException, KaidenDuplicationEntryException {
        return super.entry(k30mExpTypeService, dataList, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int entryIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenDuplicationEntryException {
        return super.entryIgnoreAuthCompany(k30mExpTypeService, dataList);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int remove(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria)
        throws KaidenHaventAuthorityToCompanyException {
        return super.remove(k30mExpTypeService, criteria, authCriteria);
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public int removeIgnoreAuthCompany(ExpTypeManagerDto criteria) {
        return super.removeIgnoreAuthCompany(k30mExpTypeService, criteria);
    }

    /**
     * {@inheritDoc}

```

```

*/
@Override
public int renew(List<ExpTypeManagerDto> dataList, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException {
    return super.renew(k30mExpTypeService, dataList, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public int renewIgnoreAuthCompany(List<ExpTypeManagerDto> dataList) throws KaidenOptimisticLockException {
    return super.renewIgnoreAuthCompany(k30mExpTypeService, dataList);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTerms(ExpTypeManagerDto criteria, AuthorityCriteriaDto authCriteria) {
    return super.selectTerms(k30mExpTypeService, criteria, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectTermsIgnoreAuthCompany(ExpTypeManagerDto criteria) {
    return super.selectTermsIgnoreAuthCompany(k30mExpTypeService, criteria);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDate(ExpTypeManagerDto criteria,
    AuthorityCriteriaDto authCriteria, List<OrderDto> order) {
    return super.selectListIncludesSearchDate(k30mExpTypeService, criteria, authCriteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public List<ExpTypeManagerDto> selectListIncludesSearchDateIgnoreAuthCompany(ExpTypeManagerDto criteria,
    List<OrderDto> order) {
    return super.selectListIncludesSearchDateIgnoreAuthCompany(k30mExpTypeService, criteria, order);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria)
    throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException, KaidenValidateException {
    super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTerm(ExpTypeManagerDto criteria, Date divideDate, AuthorityCriteriaDto authCriteria,
    String deleteFlag2nd) throws KaidenHaventAuthorityToCompanyException, KaidenOptimisticLockException,
    KaidenValidateException {
    String termCd2nd = super.divideTerm(k30mExpTypeService, criteria, divideDate, authCriteria);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabled(criteria2nd, authCriteria);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabled(criteria2nd, authCriteria);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
}

```

```

}

/**
 * {@inheritDoc}
 */
@Override
public void divideTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date divideDate, String deleteFlag2nd)
    throws KaidenOptimisticLockException, KaidenValidateException {
    String termCd2nd = super.divideTermIgnoreAuthCompany(k30mExpTypeService, criteria, divideDate);
    if (StringUtil.isEmpty(deleteFlag2nd)) {
        ExpTypeManagerDto criteria2nd = Beans.createAndCopy(ExpTypeManagerDto.class, criteria).execute();
        criteria2nd.termCd = termCd2nd;
        if (GenericFlag.isOn(deleteFlag2nd)) {
            this.toDisabledIgnoreAuthCompany(criteria2nd);
        } else if (GenericFlag.isOff(deleteFlag2nd)) {
            this.toEnabledIgnoreAuthCompany(criteria2nd);
        } else {
            throw KaidenIllegalParametersException.makeInstance();
        }
    }
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTerm(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate,
    AuthorityCriteriaDto authCriteria) throws KaidenHaventAuthorityToCompanyException,
    KaidenOptimisticLockException, KaidenValidateException {
    super.moveTerm(k30mExpTypeService, criteria, newStartDate, newEndDate, authCriteria);
}

/**
 * {@inheritDoc}
 */
@Override
public void moveTermIgnoreAuthCompany(ExpTypeManagerDto criteria, Date newStartDate, Date newEndDate)
    throws KaidenOptimisticLockException, KaidenValidateException {
    super.moveTermIgnoreAuthCompany(k30mExpTypeService, criteria, newStartDate, newEndDate);
}

/**
 * {@inheritDoc}
 */
@Override
public Class<? extends KaidenDto> getDtoClass() {
    return super.dtoClass;
}

/**
 * 重複チェック確認フィールドを返却。
 * <p>
 * 新規登録時、登録済みデータと引数の登録データの重複チェックを行うフィールドを返します。
 * </p>
 *
 * @return 重複チェック確認フィールドリスト
 */
@Override
protected List<String> getDuplicationCheckField() {
    return Arrays.asList(new String[] {COMPANY_CD, EXP_TYPE_CD});
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Entry()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override
public List<KaidenValidateResult> validate4Entry(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecuter(dtoList, surfaces, getValidateDefsDtoList4Entry()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * {@link #getValidateDefsDtoList4Renew()} で取得した定義を基にバリデートを実行します。
 * </p>
 */
@Override

```

```

public List<KaidenValidateResult> validate4Renew(List<?> dtoList, Map<String, String> surfaces) {
    if (dtoList == null) {
        throw KaidenIllegalParametersException.makeInstance();
    }
    return makeValidateExecutor(dtoList, surfaces, getValidateDefsDtoList4Renew()).executeAll(
        ErrorMode.ERROR_CONTINUE);
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * <tr>
 * <td>削除フラグ</td>
 * <td>フラグ妥当性</td>
 * </tr>
 * </table>
 */
@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Entry() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_COMPANY_EXISTS, COMPANY_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
    // ロケールID
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
    result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
    // 経費区分名称
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
    // 経費区分カテゴリ
    result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
    // 開始日
    result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
    result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
    // 終了日

```

```

result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));
// 削除フラグ
result.add(new ValidateDefsDto(DELETE_FLAG, VALID_FLAG));

return result;
}

/**
 * {@inheritDoc}
 * <p>
 * 以下のバリデート定義のリストを返却します。
 * </p>
 * <table border="1">
 * <tr>
 * <th>項目名</th>
 * <th>検証内容</th>
 * </tr>
 * <tr>
 * <td>会社コード</td>
 * <td>必須、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>経費区分コード</td>
 * <td>必須、100桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>期間コード</td>
 * <td>必須、50桁、皆伝コード</td>
 * </tr>
 * <tr>
 * <td>ロケールID</td>
 * <td>必須、ロケール妥当性</td>
 * </tr>
 * <tr>
 * <td>経費区分名称</td>
 * <td>必須、250桁</td>
 * </tr>
 * <tr>
 * <td>経費区分カテゴリ</td>
 * <td>250桁</td>
 * </tr>
 * <tr>
 * <td>開始日</td>
 * <td>必須、日付(システム期間内)</td>
 * </tr>
 * <tr>
 * <td>終了日</td>
 * <td>必須、日付(システム期間内)、開始日・終了日大小チェック</td>
 * </tr>
 * <tr>
 * <td>勘定科目コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>税区分コード</td>
 * <td>必須、100桁、皆伝コード、マスタ存在確認</td>
 * </tr>
 * <tr>
 * <td>ソートキー</td>
 * <td>必須、数値</td>
 * </tr>
 * </table>
 */

@Override
public List<ValidateDefsDto> getValidateDefsDtoList4Renew() {
    List<ValidateDefsDto> result = new ArrayList<ValidateDefsDto>();
    // 会社コード
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(COMPANY_CD, VALID_COMPANY_EXISTS, COMPANY_CD, "", "true"));
    // 経費区分コード
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_REQUIRED));
    result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));

```

```

result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CD, VALID_KAIDEN_ID));
// 期間コード
result.add(new ValidateDefsDto(TERM_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TERM_CD, VALID_MAX_LENGTH, String.valueOf(TERM_CD_LEN)));
result.add(new ValidateDefsDto(TERM_CD, VALID_KAIDEN_ID));
// ロケールID
result.add(new ValidateDefsDto(LOCALE_ID, VALID_REQUIRED));
result.add(new ValidateDefsDto(LOCALE_ID, VALID_LOCALE));
// 経費区分名称
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_REQUIRED));
result.add(new ValidateDefsDto(EXP_TYPE_NAME, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_NAME_LEN)));
result.add(new ValidateDefsDto(EXP_TYPE_CATEGORY, VALID_MAX_LENGTH, String.valueOf(EXP_TYPE_CATEGORY_LEN)));
// 開始日
result.add(new ValidateDefsDto(START_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(START_DATE, VALID_DATE));
// 終了日
result.add(new ValidateDefsDto(END_DATE, VALID_REQUIRED));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE));
result.add(new ValidateDefsDto(END_DATE, VALID_DATE_MIN, START_DATE));
// 勘定科目コード
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_MAX_LENGTH, String.valueOf(ACCOUNT_CD_LEN)));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(ACCOUNT_CD, VALID_ACCOUNT_EXISTS, ACCOUNT_CD, COMPANY_CD, "", "true"));
// 税区分コード
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_REQUIRED));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_MAX_LENGTH, String.valueOf(TAX_TYPE_CD_LEN)));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_KAIDEN_ID));
result.add(new ValidateDefsDto(TAX_TYPE_CD, VALID_TAX_TYPE_EXISTS, TAX_TYPE_CD, COMPANY_CD, "", "true"));
// ソートキー
result.add(new ValidateDefsDto(SORT_KEY, VALID_REQUIRED));
result.add(new ValidateDefsDto(SORT_KEY, VALID_INTEGER, String.valueOf(SORT_KEY_LEN)));

return result;
}

/** {@inheritDoc} */
@Override
public Class<ExpTypeManagerDto> getMainManagerDtoClass() {
    return ExpTypeManagerDto.class;
}

@Override
public List<ExpTypeManagerDto> selectListIgnoreAuthCompanyWithLock(ExpTypeManagerDto criteria, List<OrderDto> order) {
    Map<String, Object> criteriaMap = k30mExpTypeService.getCriteria(criteria);
    return complementEndDateForDisp(CollectionUtil.createListAndCopy(ExpTypeManagerDto.class,
        k30mExpTypeService.genericSelectList4Update(criteriaMap, order)));
}

/** {@inheritDoc} */
@Override
public List<String> getPkFiled() {
    return k30mExpTypeService.getPkFiled();
}

/**
 * {@inheritDoc}
 */
@Override
public void executeIterateLogic(ExpTypeManagerDto criteria, List<OrderDto> order,
    IterateBusinessLogicIF<ExpTypeManagerDto> logic) throws KaidenSystemException, KaidenApplicationException {
    super.iterateLogic(k30mExpTypeService, criteria, order, logic);
}

@Override
public Map<String, Object> selectList4MstSearch(KaidenDto criteria, List<OrderDto> order,
    KeywordsCriteriaDto keywordsCriteria, SqlResultFilterIF[] filter) {
    String[] likeTarget = {EXP_TYPE_NAME, EXP_TYPE_CATEGORY, EXTENSION_A, EXTENSION_B, EXTENSION_C, EXTENSION_D,
        EXTENSION_E, EXTENSION_F, EXTENSION_G, EXTENSION_H, EXTENSION_I, EXTENSION_J, COMPANY_NAME,
        ACCOUNT_NAME, TAX_TYPE_NAME};
    Map<String, Object> resultMap = super.selectListByKeywordsFilter(k30mExpTypeService, "selectView.sql",
        criteria, order, keywordsCriteria, likeTarget, filter);
    @SuppressWarnings("unchecked")
    List<ExpTypeManagerDto> newList = (List<ExpTypeManagerDto>) resultMap.get("resultList");
    resultMap.put("resultList", newList);

    return resultMap;
}
}

```

ManagerDto

■ Path

src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDto.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import java.math.BigDecimal;
import java.sql.Timestamp;
import jp.co.slcs.kaiden2.base.foundation.model.dto.KaidenMasterManagerDto;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO.
 *
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public class ExpTypeManagerDto extends KaidenMasterManagerDto {

    /** 会社コード. */
    public String companyCd;
    /** 経費区分コード. */
    public String expTypeCd;
    /** 期間コード. */
    public String termCd;
    /** ロケールid. */
    public String localeId;
    /** 経費区分名称. */
    public String expTypeName;
    /** 経費区分カテゴリ. */
    public String expTypeCategory;
    /** 開始日. */
    public Timestamp startDate;
    /** 終了日. */
    public Timestamp endDate;
    /** 勘定科目コード. */
    public String accountCd;
    /** 税区分コード. */
    public String taxTypeCd;
    /** ソートキー. */
    public Long sortKey;
    /** 削除フラグ. */
    public String deleteFlag;
    /** 予備項目a. */
    public String extensionA;
    /** 予備項目b. */
    public String extensionB;
    /** 予備項目c. */
    public String extensionC;
    /** 予備項目d. */
    public String extensionD;
    /** 予備項目e. */
    public String extensionE;
    /** 予備項目f. */
    public String extensionF;
    /** 予備項目g. */
    public String extensionG;
    /** 予備項目h. */
    public String extensionH;
    /** 予備項目i. */
    public String extensionI;
    /** 予備項目j. */
    public String extensionJ;
    /** 登録日時. */
    public Timestamp entryTs;
    /** 登録者コード. */
    public String entryUserCd;
    /** 更新カウンタ. */
    public Long renewCnt;
    /** 更新日時. */
    public Timestamp renewTs;
    /** 更新者コード. */
    public String renewUserCd;

    /** 会社名称(他マスタ名称). */
    public String companyName;
    /** 勘定科目名称(他マスタ名称). */
    public String accountName;
    /** 税区分名称(他マスタ名称). */
    public String taxTypeName;
    /** 税率(他マスタ). */
    public BigDecimal taxRate;
}

```

- Path
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/manager/exptype/ExpTypeManagerDtoFieldName.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype;

import jp.co.slcs.kaiden2.base.foundation.model.conf.EntityCommonFieldNames;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exptype.impl.ExpTypeManagerImpl;

/**
 * 経費区分マスタDTO フィールド名定義クラス。
 *
 * <p>
 * マネージャクラス(@link ExpTypeManagerImpl)へ渡すパラメータ、及びマネージャからの結果返却で当クラスを使用します。
 * </p>
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public final class ExpTypeManagerDtoFieldName extends EntityCommonFieldNames {

    /**
     * コンストラクタ。
     */
    private ExpTypeManagerDtoFieldName() {

        /** 会社コード。 */
        public static final String COMPANY_CD = EntityCommonFieldNames.COMPANY_CD;
        /** 経費区分コード。 */
        public static final String EXP_TYPE_CD = "expTypeCd";
        /** 期間コード。 */
        public static final String TERM_CD = EntityCommonFieldNames.TERM_CD;
        /** ロケールid。 */
        public static final String LOCALE_ID = EntityCommonFieldNames.LOCALE_ID;
        /** 経費区分名称。 */
        public static final String EXP_TYPE_NAME = "expTypeName";
        /** 経費区分カテゴリ。 */
        public static final String EXP_TYPE_CATEGORY = "expTypeCategory";
        /** 開始日。 */
        public static final String START_DATE = EntityCommonFieldNames.START_DATE;
        /** 終了日。 */
        public static final String END_DATE = EntityCommonFieldNames.END_DATE;
        /** 勘定科目コード。 */
        public static final String ACCOUNT_CD = "accountCd";
        /** 税区分コード。 */
        public static final String TAX_TYPE_CD = "taxTypeCd";
        /** ソートキー。 */
        public static final String SORT_KEY = EntityCommonFieldNames.SORT_KEY;
        /** 削除フラグ。 */
        public static final String DELETE_FLAG = EntityCommonFieldNames.DELETE_FLAG;

        /** 会社名称(他マスタ名称)。 */
        public static final String COMPANY_NAME = "companyName";
        /** 勘定科目名称(他マスタ名称)。 */
        public static final String ACCOUNT_NAME = "accountName";
        /** 税区分名称(他マスタ名称)。 */
        public static final String TAX_TYPE_NAME = "taxTypeName";
        /** 税率(他マスタ)。 */
        public static final String TAX_RATE = "taxRate";
    }
}

```

Settings

■ Path

```
src/main/java/jp/co/slcs/kaiden2/product_workflow/foundation/model/xml_schema/exptype_import_settings/ExpTypeImportSettings.java
```



```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.xml_schema.exctype_import_settings;

import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;
import jp.co.slcs.kaiden2.base.foundation.model.xml_schema.generic_import_settings.GenericImportSettings;
import jp.co.slcs.kaiden2.product_workflow.foundation.manager.exctype.ExpTypeManagerDtoFieldName;

/**
 * 経費区分マスタ読み設定DTO.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@XmlRootElement
@XmlType(propOrder = {})
public class ExpTypeImportSettings extends GenericImportSettings {

    /** serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /**
     * フィールド定義配列.
     */
    protected String[] filedNames = new String[] {ExpTypeManagerDtoFieldName.COMPANY_CD,
        ExpTypeManagerDtoFieldName.EXP_TYPE_CD, ExpTypeManagerDtoFieldName.LOCALE_ID,
        ExpTypeManagerDtoFieldName.START_DATE, ExpTypeManagerDtoFieldName.END_DATE,
        ExpTypeManagerDtoFieldName.EXP_TYPE_NAME, ExpTypeManagerDtoFieldName.EXP_TYPE_CATEGORY,
        ExpTypeManagerDtoFieldName.ACCOUNT_CD, ExpTypeManagerDtoFieldName.TAX_TYPE_CD,
        ExpTypeManagerDtoFieldName.SORT_KEY, ExpTypeManagerDtoFieldName.DELETE_FLAG};

    /** {@inheritDoc} */
    @Override
    protected void initOrders() {
        super.getOrders().add(new Order(ExpTypeManagerDtoFieldName.COMPANY_CD, Order.SORT_ASC));
        super.getOrders().add(new Order(ExpTypeManagerDtoFieldName.EXP_TYPE_CD, Order.SORT_ASC));
        super.getOrders().add(new Order(ExpTypeManagerDtoFieldName.LOCALE_ID, Order.SORT_ASC));
    }

    /** {@inheritDoc} */
    @Override
    protected void initPkFields() {
        super.getPkFields().add(ExpTypeManagerDtoFieldName.COMPANY_CD);
        super.getPkFields().add(ExpTypeManagerDtoFieldName.EXP_TYPE_CD);
        super.getPkFields().add(ExpTypeManagerDtoFieldName.LOCALE_ID);
    }

    /** {@inheritDoc} */
    @Override
    protected void initI18NFields() {
        super.getI18NFields().add(ExpTypeManagerDtoFieldName.EXP_TYPE_NAME);
    }

    /** {@inheritDoc} */
    @Override
    protected void initFieldsOrder() {
        for (String j : filedNames) {
            super.getFieldsOrder().add(j);
        }
    }
}

```

Java (S2JDBC)

Entity

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpType.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import javax.annotation.Generated;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import jp.co.slcs.kaiden2.base.foundation.model.entity.GenerateEntity;

/**
 * 経費区分マスタ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */

```

```

@Entity
@Table(name = "k30m_exp_type")
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl" })
public final class K30mExpType extends GenerateEntity implements Serializable {

    /** シリアルバージョンUID. */
    private static final long serialVersionUID = 1L;

    /** 会社コード. */
    @Id
    @Column(name = "company_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String companyCd;

    /** 経費区分コード. */
    @Id
    @Column(name = "exp_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String expTypeCd;

    /** 期間コード. */
    @Id
    @Column(name = "term_cd", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String termCd;

    /** ロケールid. */
    @Id
    @Column(name = "locale_id", columnDefinition = "varchar2(50)", nullable = false, unique = false)
    public String localeId;

    /** 経費区分名称. */
    @Column(name = "exp_type_name", columnDefinition = "varchar2(1000)", nullable = false, unique = false)
    public String expTypeName;

    /** 経費区分カテゴリ. */
    @Column(name = "exp_type_category", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String expTypeCategory;

    /** 開始日. */
    @Column(name = "start_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp startDate;

    /** 終了日. */
    @Column(name = "end_date", columnDefinition = "timestamp(6)", nullable = false, unique = false)
    public Timestamp endDate;

    /** 勘定科目コード. */
    @Column(name = "account_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String accountCd;

    /** 税区分コード. */
    @Column(name = "tax_type_cd", columnDefinition = "varchar2(100)", nullable = false, unique = false)
    public String taxTypeCd;

    /** ソートキー. */
    @Column(name = "sort_key", columnDefinition = "number(15,0)", nullable = false, unique = false)
    public Long sortKey;

    /** 削除フラグ. */
    @Column(name = "delete_flag", columnDefinition = "varchar2(1)", nullable = false, unique = false)
    public String deleteFlag;

    /** 予備項目a. */
    @Column(name = "extension_a", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionA;

    /** 予備項目b. */
    @Column(name = "extension_b", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionB;

    /** 予備項目c. */
    @Column(name = "extension_c", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionC;

    /** 予備項目d. */
    @Column(name = "extension_d", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionD;

    /** 予備項目e. */
    @Column(name = "extension_e", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionE;

    /** 予備項目f. */
    @Column(name = "extension_f", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
    public String extensionF;

    /** 予備項目g. */
    @Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)

```

```

@Column(name = "extension_g", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionG;

/** 予備項目h. */
@Column(name = "extension_h", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionH;

/** 予備項目i. */
@Column(name = "extension_i", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionI;

/** 予備項目j. */
@Column(name = "extension_j", columnDefinition = "varchar2(1000)", nullable = true, unique = false)
public String extensionJ;
}

```

Names

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/entity/K30mExpTypeNames.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.entity;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.name.PropertyName;

/**
 * {@link K30mExpType} の名称クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.NamesModelFactoryImpl" })
public final class K30mExpTypeNames {

    /**
     * デフォルトコンストラクタ.
     */
    private K30mExpTypeNames() {
        super();
    }

    /**
     * companyCd のプロパティ名を返却.
     *
     * @return companyCd のプロパティ名
     */
    public static PropertyName<String> companyCd() {
        return new PropertyName<String>("companyCd");
    }

    /**
     * expTypeCd のプロパティ名を返却.
     *
     * @return expTypeCd のプロパティ名
     */
    public static PropertyName<String> expTypeCd() {
        return new PropertyName<String>("expTypeCd");
    }

    /**
     * termCd のプロパティ名を返却.
     *
     * @return termCd のプロパティ名
     */
    public static PropertyName<String> termCd() {
        return new PropertyName<String>("termCd");
    }

    /**
     * localeId のプロパティ名を返却.
     *
     * @return localeId のプロパティ名
     */
    public static PropertyName<String> localeId() {
        return new PropertyName<String>("localeId");
    }

    /**
     * expTypeName のプロパティ名を返却.
     *
     * @return expTypeName のプロパティ名
     */
    public static PropertyName<String> expTypeName() {
        return new PropertyName<String>("expTypeName");
    }
}

```

```

/**
 * expTypeCategoryのプロパティ名を返却.
 *
 * @return expTypeCategoryのプロパティ名
 */
public static PropertyName<String> expTypeCategory() {
    return new PropertyName<String>("expTypeCategory");
}

/**
 * startDateのプロパティ名を返却.
 *
 * @return startDateのプロパティ名
 */
public static PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>("startDate");
}

/**
 * endDateのプロパティ名を返却.
 *
 * @return endDateのプロパティ名
 */
public static PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>("endDate");
}

/**
 * accountCdのプロパティ名を返却.
 *
 * @return accountCdのプロパティ名
 */
public static PropertyName<String> accountCd() {
    return new PropertyName<String>("accountCd");
}

/**
 * taxTypeCdのプロパティ名を返却.
 *
 * @return taxTypeCdのプロパティ名
 */
public static PropertyName<String> taxTypeCd() {
    return new PropertyName<String>("taxTypeCd");
}

/**
 * sortKeyのプロパティ名を返却.
 *
 * @return sortKeyのプロパティ名
 */
public static PropertyName<Long> sortKey() {
    return new PropertyName<Long>("sortKey");
}

/**
 * deleteFlagのプロパティ名を返却.
 *
 * @return deleteFlagのプロパティ名
 */
public static PropertyName<String> deleteFlag() {
    return new PropertyName<String>("deleteFlag");
}

/**
 * extensionAのプロパティ名を返却.
 *
 * @return extensionAのプロパティ名
 */
public static PropertyName<String> extensionA() {
    return new PropertyName<String>("extensionA");
}

/**
 * extensionBのプロパティ名を返却.
 *
 * @return extensionBのプロパティ名
 */
public static PropertyName<String> extensionB() {
    return new PropertyName<String>("extensionB");
}

/**
 * extensionCのプロパティ名を返却.
 *
 * @return extensionCのプロパティ名

```

```

    */
    public static PropertyName<String> extensionC() {
        return new PropertyName<String>("extensionC");
    }

    /**
     * extensionDのプロパティ名を返却.
     *
     * @return extensionDのプロパティ名
     */
    public static PropertyName<String> extensionD() {
        return new PropertyName<String>("extensionD");
    }

    /**
     * extensionEのプロパティ名を返却.
     *
     * @return extensionEのプロパティ名
     */
    public static PropertyName<String> extensionE() {
        return new PropertyName<String>("extensionE");
    }

    /**
     * extensionFのプロパティ名を返却.
     *
     * @return extensionFのプロパティ名
     */
    public static PropertyName<String> extensionF() {
        return new PropertyName<String>("extensionF");
    }

    /**
     * extensionGのプロパティ名を返却.
     *
     * @return extensionGのプロパティ名
     */
    public static PropertyName<String> extensionG() {
        return new PropertyName<String>("extensionG");
    }

    /**
     * extensionHのプロパティ名を返却.
     *
     * @return extensionHのプロパティ名
     */
    public static PropertyName<String> extensionH() {
        return new PropertyName<String>("extensionH");
    }

    /**
     * extensionIのプロパティ名を返却.
     *
     * @return extensionIのプロパティ名
     */
    public static PropertyName<String> extensionI() {
        return new PropertyName<String>("extensionI");
    }

    /**
     * extensionJのプロパティ名を返却.
     *
     * @return extensionJのプロパティ名
     */
    public static PropertyName<String> extensionJ() {
        return new PropertyName<String>("extensionJ");
    }

    /**
     * entryTsのプロパティ名を返却.
     *
     * @return entryTsのプロパティ名
     */
    public static PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>("entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public static PropertyName<String> entryUserCd() {
        return new PropertyName<String>("entryUserCd");
    }
}

```

```

/**
 * renewCntのプロパティ名を返却.
 *
 * @return renewCntのプロパティ名
 */
public static PropertyName<Long> renewCnt() {
    return new PropertyName<Long>("renewCnt");
}

/**
 * renewTsのプロパティ名を返却.
 *
 * @return renewTsのプロパティ名
 */
public static PropertyName<Timestamp> renewTs() {
    return new PropertyName<Timestamp>("renewTs");
}

/**
 * renewUserCdのプロパティ名を返却.
 *
 * @return renewUserCdのプロパティ名
 */
public static PropertyName<String> renewUserCd() {
    return new PropertyName<String>("renewUserCd");
}

/**
 * @author S2JDBC-Gen
 */
public static class _K30mExpTypeNames extends PropertyName<K30mExpType> {

    /**
     * インスタンスを構築.
     */
    public _K30mExpTypeNames() {
    }

    /**
     * インスタンスを構築.
     *
     * @param name 名前
     */
    public _K30mExpTypeNames(final String name) {
        super(name);
    }

    /**
     * インスタンスを構築.
     *
     * @param parent 親
     * @param name 名前
     */
    public _K30mExpTypeNames(final PropertyName<?> parent, final String name) {
        super(parent, name);
    }

    /**
     * companyCdのプロパティ名を返却.
     *
     * @return companyCdのプロパティ名
     */
    public PropertyName<String> companyCd() {
        return new PropertyName<String>(this, "companyCd");
    }

    /**
     * expTypeCdのプロパティ名を返却.
     *
     * @return expTypeCdのプロパティ名
     */
    public PropertyName<String> expTypeCd() {
        return new PropertyName<String>(this, "expTypeCd");
    }

    /**
     * termCdのプロパティ名を返却.
     *
     * @return termCdのプロパティ名
     */
    public PropertyName<String> termCd() {
        return new PropertyName<String>(this, "termCd");
    }

    /**
     * localeIdのプロパティ名を返却.

```

```

*
* @return localeIdのプロパティ名
*/
public PropertyName<String> localeId() {
    return new PropertyName<String>((this, "localeId"));
}

/**
* expTypeNameのプロパティ名を返却.
*
* @return expTypeNameのプロパティ名
*/
public PropertyName<String> expTypeName() {
    return new PropertyName<String>((this, "expTypeName"));
}

/**
* expTypeCategoryのプロパティ名を返却.
*
* @return expTypeCategoryのプロパティ名
*/
public PropertyName<String> expTypeCategory() {
    return new PropertyName<String>((this, "expTypeCategory"));
}

/**
* startDateのプロパティ名を返却.
*
* @return startDateのプロパティ名
*/
public PropertyName<Timestamp> startDate() {
    return new PropertyName<Timestamp>((this, "startDate"));
}

/**
* endDateのプロパティ名を返却.
*
* @return endDateのプロパティ名
*/
public PropertyName<Timestamp> endDate() {
    return new PropertyName<Timestamp>((this, "endDate"));
}

/**
* accountCdのプロパティ名を返却.
*
* @return accountCdのプロパティ名
*/
public PropertyName<String> accountCd() {
    return new PropertyName<String>((this, "accountCd"));
}

/**
* taxTypeCdのプロパティ名を返却.
*
* @return taxTypeCdのプロパティ名
*/
public PropertyName<String> taxTypeCd() {
    return new PropertyName<String>((this, "taxTypeCd"));
}

/**
* sortKeyのプロパティ名を返却.
*
* @return sortKeyのプロパティ名
*/
public PropertyName<Long> sortKey() {
    return new PropertyName<Long>((this, "sortKey"));
}

/**
* deleteFlagのプロパティ名を返却.
*
* @return deleteFlagのプロパティ名
*/
public PropertyName<String> deleteFlag() {
    return new PropertyName<String>((this, "deleteFlag"));
}

/**
* extensionAのプロパティ名を返却.
*
* @return extensionAのプロパティ名
*/
public PropertyName<String> extensionA() {
    return new PropertyName<String>((this, "extensionA"));
}

```

```

}

/**
 * extensionBのプロパティ名を返却。
 *
 * @return extensionBのプロパティ名
 */
public PropertyName<String> extensionB() {
    return new PropertyName<String>(this, "extensionB");
}

/**
 * extensionCのプロパティ名を返却。
 *
 * @return extensionCのプロパティ名
 */
public PropertyName<String> extensionC() {
    return new PropertyName<String>(this, "extensionC");
}

/**
 * extensionDのプロパティ名を返却。
 *
 * @return extensionDのプロパティ名
 */
public PropertyName<String> extensionD() {
    return new PropertyName<String>(this, "extensionD");
}

/**
 * extensionEのプロパティ名を返却。
 *
 * @return extensionEのプロパティ名
 */
public PropertyName<String> extensionE() {
    return new PropertyName<String>(this, "extensionE");
}

/**
 * extensionFのプロパティ名を返却。
 *
 * @return extensionFのプロパティ名
 */
public PropertyName<String> extensionF() {
    return new PropertyName<String>(this, "extensionF");
}

/**
 * extensionGのプロパティ名を返却。
 *
 * @return extensionGのプロパティ名
 */
public PropertyName<String> extensionG() {
    return new PropertyName<String>(this, "extensionG");
}

/**
 * extensionHのプロパティ名を返却。
 *
 * @return extensionHのプロパティ名
 */
public PropertyName<String> extensionH() {
    return new PropertyName<String>(this, "extensionH");
}

/**
 * extensionIのプロパティ名を返却。
 *
 * @return extensionIのプロパティ名
 */
public PropertyName<String> extensionI() {
    return new PropertyName<String>(this, "extensionI");
}

/**
 * extensionJのプロパティ名を返却。
 *
 * @return extensionJのプロパティ名
 */
public PropertyName<String> extensionJ() {
    return new PropertyName<String>(this, "extensionJ");
}

/**
 * entryTsのプロパティ名を返却。
 *

```



```

    * @return entryTsのプロパティ名
    */
    public PropertyName<Timestamp> entryTs() {
        return new PropertyName<Timestamp>(this, "entryTs");
    }

    /**
     * entryUserCdのプロパティ名を返却.
     *
     * @return entryUserCdのプロパティ名
     */
    public PropertyName<String> entryUserCd() {
        return new PropertyName<String>(this, "entryUserCd");
    }

    /**
     * renewCntのプロパティ名を返却.
     *
     * @return renewCntのプロパティ名
     */
    public PropertyName<Long> renewCnt() {
        return new PropertyName<Long>(this, "renewCnt");
    }

    /**
     * renewTsのプロパティ名を返却.
     *
     * @return renewTsのプロパティ名
     */
    public PropertyName<Timestamp> renewTs() {
        return new PropertyName<Timestamp>(this, "renewTs");
    }

    /**
     * renewUserCdのプロパティ名を返却.
     *
     * @return renewUserCdのプロパティ名
     */
    public PropertyName<String> renewUserCd() {
        return new PropertyName<String>(this, "renewUserCd");
    }
}

```

Service

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import javax.annotation.Generated;
import jp.co.slcs.kaiden2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden2.product_workflow.foundation.model.entity.K30mExpType;

/**
 * {@link K30mExpType}のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl" })
public final class K30mExpTypeService extends GenerateService<K30mExpType> {
    //
}

```

Condition

■ Path

src/main/generated/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeCondition.java

```

package jp.co.slcs.kaiden2.product_workflow.foundation.model.service;

import java.sql.Timestamp;
import javax.annotation.Generated;
import org.seasar.extension.jdbc.where.ComplexWhere;
import org.seasar.extension.jdbc.where.condition.AbstractEntityCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableCondition;
import org.seasar.extension.jdbc.where.condition.NotNullableStringCondition;
import org.seasar.extension.jdbc.where.condition.NulltableStringCondition;

/**
 * {@link K30mExpType}の条件クラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.

```

```

*/
@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.ConditionModelFactoryImpl" })
public final class K30mExpTypeCondition extends
    AbstractEntityCondition<K30mExpTypeCondition> {

    /**
     * インスタンスを構築.
     */
    public K30mExpTypeCondition() {
    }

    /**
     * インスタンスを構築.
     *
     * @param prefix プレフィックス
     * @param where 検索条件
     */
    public K30mExpTypeCondition(String prefix, ComplexWhere where) {
        super(prefix, where);
    }

    /** companyCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> companyCd =
        new NotNullStringCondition<K30mExpTypeCondition>("companyCd", this);

    /** expTypeCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> expTypeCd =
        new NotNullStringCondition<K30mExpTypeCondition>("expTypeCd", this);

    /** termCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> termCd =
        new NotNullStringCondition<K30mExpTypeCondition>("termCd", this);

    /** localeIdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> localeId =
        new NotNullStringCondition<K30mExpTypeCondition>("localeId", this);

    /** expTypeNameの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> expTypeName =
        new NotNullStringCondition<K30mExpTypeCondition>("expTypeName", this);

    /** expTypeCategoryの条件. */
    public NullableStringCondition<K30mExpTypeCondition> expTypeCategory =
        new NullableStringCondition<K30mExpTypeCondition>("expTypeCategory", this);

    /** startDateの条件. */
    public NotNullCondition<K30mExpTypeCondition, Timestamp> startDate =
        new NotNullCondition<K30mExpTypeCondition, Timestamp>("startDate", this);

    /** endDateの条件. */
    public NotNullCondition<K30mExpTypeCondition, Timestamp> endDate =
        new NotNullCondition<K30mExpTypeCondition, Timestamp>("endDate", this);

    /** accountCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> accountCd =
        new NotNullStringCondition<K30mExpTypeCondition>("accountCd", this);

    /** taxTypeCdの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> taxTypeCd =
        new NotNullStringCondition<K30mExpTypeCondition>("taxTypeCd", this);

    /** sortKeyの条件. */
    public NotNullCondition<K30mExpTypeCondition, Long> sortKey =
        new NotNullCondition<K30mExpTypeCondition, Long>("sortKey", this);

    /** deleteFlagの条件. */
    public NotNullStringCondition<K30mExpTypeCondition> deleteFlag =
        new NotNullStringCondition<K30mExpTypeCondition>("deleteFlag", this);

    /** extensionAの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionA =
        new NullableStringCondition<K30mExpTypeCondition>("extensionA", this);

    /** extensionBの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionB =
        new NullableStringCondition<K30mExpTypeCondition>("extensionB", this);

    /** extensionCの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionC =
        new NullableStringCondition<K30mExpTypeCondition>("extensionC", this);

    /** extensionDの条件. */
    public NullableStringCondition<K30mExpTypeCondition> extensionD =
        new NullableStringCondition<K30mExpTypeCondition>("extensionD", this);

    /** extensionEの条件. */

```

```

public NullableStringCondition<K30mExpTypeCondition> extensionE =
    new NullableStringCondition<K30mExpTypeCondition>("extensionE", this);

/** extensionFの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionF =
    new NullableStringCondition<K30mExpTypeCondition>("extensionF", this);

/** extensionGの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionG =
    new NullableStringCondition<K30mExpTypeCondition>("extensionG", this);

/** extensionHの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionH =
    new NullableStringCondition<K30mExpTypeCondition>("extensionH", this);

/** extensionIの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionI =
    new NullableStringCondition<K30mExpTypeCondition>("extensionI", this);

/** extensionJの条件. */
public NullableStringCondition<K30mExpTypeCondition> extensionJ =
    new NullableStringCondition<K30mExpTypeCondition>("extensionJ", this);

/** entryTsの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> entryTs =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("entryTs", this);

/** entryUserCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> entryUserCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("entryUserCd", this);

/** renewCntの条件. */
public NotNullCondition<K30mExpTypeCondition, Long> renewCnt =
    new NotNullCondition<K30mExpTypeCondition, Long>("renewCnt", this);

/** renewTsの条件. */
public NotNullCondition<K30mExpTypeCondition, Timestamp> renewTs =
    new NotNullCondition<K30mExpTypeCondition, Timestamp>("renewTs", this);

/** renewUserCdの条件. */
public NotNullableStringCondition<K30mExpTypeCondition> renewUserCd =
    new NotNullableStringCondition<K30mExpTypeCondition>("renewUserCd", this);
}

```

SQL

検索

■ Path

src/main/resources/META-INF/sql/jp/co/slcs/kaiden2/product_workflow/foundation/model/service/K30mExpTypeService/selectView.sql

```

select
    exp.company_cd
, exp.exp_type_cd
, exp.term_cd
, exp.locale_id
, exp.exp_type_name
, exp.exp_type_category
, exp.start_date
, exp.end_date
, exp.account_cd
, exp.tax_type_cd
, exp.sort_key
, exp.delete_flag
, exp.extension_a
, exp.extension_b
, exp.extension_c
, exp.extension_d
, exp.extension_e
, exp.extension_f
, exp.extension_g
, exp.extension_h
, exp.extension_i
, exp.extension_j
, exp.entry_ts
, exp.entry_user_cd
, exp.renew_cnt
, exp.renew_ts
, exp.renew_user_cd
, dept.department_name as company_name
, account.account_name as account_name
, tax.tax_type_name as tax_type_name
, tax.tax_rate as tax_rate
from k30m_exp_type exp

```

```

left join ( select imm_department.company_cd
, imm_department.department_set_cd
, imm_department.department_cd
, imm_department.locale_id
, imm_department.start_date
, imm_department.end_date
, imm_department.department_name
from imm_department) dept
on exp.company_cd = dept.company_cd
and exp.company_cd = dept.department_set_cd
and exp.company_cd = dept.department_cd
and exp.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and dept.start_date <= exp.start_date
--ELSE and dept.end_date > exp.start_date
/*END*/

left join k30m_account account
on exp.company_cd = account.company_cd
and exp.account_cd = account.account_cd
and exp.locale_id = account.locale_id
/*IF searchCriteriaDate != null*/
and account.start_date <= /*searchCriteriaDate*/2000/01/01'
and account.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and account.start_date <= exp.start_date
--ELSE and account.end_date > exp.start_date
/*END*/

left join k30m_tax_type tax
on exp.company_cd = tax.company_cd
and exp.tax_type_cd = tax.tax_type_cd
and exp.locale_id = tax.locale_id
/*IF searchCriteriaDate != null*/
and tax.start_date <= /*searchCriteriaDate*/2000/01/01'
and tax.end_date > /*searchCriteriaDate*/2000/01/01'
--ELSE and tax.start_date <= exp.start_date
--ELSE and tax.end_date > exp.start_date
/*END*/

/*BEGIN*/
where
/*IF companyCd != null*/
exp.company_cd in /*companyCd*/('comp')
/*END*/
/*IF expTypeCd != null*/
and exp.exp_type_cd = /*expTypeCd*/expTypeCd'
/*END*/
/*IF termCd != null*/
and exp.term_cd = /*termCd*/termCd'
/*END*/
/*IF localeId != null*/
and exp.locale_id = /*localeId*/localeId'
/*END*/
/*IF expTypeName != null*/
and exp.exp_type_name like /*expTypeName*/'% ' ESCAPE '$'
/*END*/
/*IF expTypeCategory != null*/
and exp.exp_type_category like /*expTypeCategory*/expTypeCategory' ESCAPE '$'
/*END*/
/*IF searchCriteriaDate != null*/
and exp.start_date <= /*searchCriteriaDate*/1900/01/01'
and exp.end_date > /*searchCriteriaDate*/2999/12/31'
/*END*/
/*IF accountCd != null*/
and exp.account_cd = /*accountCd*/accountCd'
/*END*/
/*IF taxTypeCd != null*/
and exp.tax_type_cd = /*taxTypeCd*/taxTypeCd'
/*END*/
/*IF sortKey != null*/
and exp.sort_key = /*sortKey*/0'
/*END*/
/*IF deleteFlag != null*/
and exp.delete_flag = /*deleteFlag*/0'
/*END*/
/*IF extensionA != null*/
and exp.extension_a like /*extensionA*/extensionA' ESCAPE '$'
/*END*/
/*IF extensionB != null*/
and exp.extension_b like /*extensionB*/extensionB' ESCAPE '$'
/*END*/
/*IF extensionC != null*/
and exp.extension_c like /*extensionC*/extensionC' ESCAPE '$'
/*END*/
/*IF extensionD != null*/
and exp.extension_d like /*extensionD*/extensionD' ESCAPE '$'
/*END*/

```

```

/*IF extensionE != null*/
  and exp.extension_e like /*extensionE*/extensionE' ESCAPE '$'
/*END*/
/*IF extensionF != null*/
  and exp.extension_f like /*extensionF*/extensionF' ESCAPE '$'
/*END*/
/*IF extensionG != null*/
  and exp.extension_g like /*extensionG*/extensionG' ESCAPE '$'
/*END*/
/*IF extensionH != null*/
  and exp.extension_h like /*extensionH*/extensionH' ESCAPE '$'
/*END*/
/*IF extensionI != null*/
  and exp.extension_i like /*extensionI*/extensionI' ESCAPE '$'
/*END*/
/*IF extensionJ != null*/
  and exp.extension_j like /*extensionJ*/extensionJ' ESCAPE '$'
/*END*/
/*IF companyName != null*/
  and dept.department_name like /*companyName*/companyName' ESCAPE '$'
/*END*/
/*IF accountName != null*/
  and account.account_name like /*accountName*/accountName' ESCAPE '$'
/*END*/
/*IF taxTypeName != null*/
  and tax.tax_type_name like /*taxTypeName*/taxTypeName' ESCAPE '$'
/*END*/
/*IF taxRate != null*/
  and tax.tax_rate = /*taxRate*/0
/*END*/
/*END*/
/*IF orderBy != null*/
  ORDER BY /*$orderBy*/company_cd
/*END*/

```

XML

- **Path**
src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/product_workflow/foundation/model/xml_schema/exptype_import_settings/ExpTypeImportSettings.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<genericImportSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericImportSettings" xmlns:tns="http://kaiden.slcs.co.jp/xmlschema/genericImportSettings">

  <!-- デフォルト設定値 -->
  <defaultSettings>
    <import_mode>1</import_mode>
    <start_date></start_date>
    <end_date></end_date>
    <shift_date></shift_date>
    <company_cd></company_cd>
    <character>UTF-8</character>
    <delimiter></delimiter>
    <enclosing>&quot;</enclosing>
    <normal_log>true</normal_log>
    <target_files>
      <target_file>kaiden/import/exptype/exptype_import.csv</target_file>
    </target_files>
    <start_rows>1</start_rows>
    <archive_files>
      <archive_file>kaiden/archive/import/exptype/exptype_import_{YMDHMSN}.csv</archive_file>
    </archive_files>
    <delete_flag>false</delete_flag>
  </defaultSettings>

  <!-- 会社ごとの設定(サンプル) -->
  <!-- companySettings -->
  <companySetting>
    <settings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="settings">
      <import_mode>1</import_mode>
      <start_date></start_date>
      <end_date></end_date>
      <shift_date>1</shift_date>
      <company_cd>sample_company_01</company_cd>
      <character>UTF-8</character>
      <delimiter></delimiter>
      <enclosing>&quot;</enclosing>
      <normal_log>true</normal_log>
      <target_files>
        <target_file>sample_company_01_exptype.csv</target_file>
      </target_files>
      <start_rows>1</start_rows>
      <archive_files>
        <archive_file>archive/sample_company_01_exptype.csv</archive_file>
      </archive_files>
      <delete_flag>true</delete_flag>
    </settings>
  </companySetting>
</companySettings-->
</genericImportSettings>

```


■ はじめに ■ 本バリデーションを使用する際の注意点 ■ intra-mart Accel Kaiden! 基盤モジュール ■ 必須 ■ いずれか必須 ■ いずれか必須 (checkbox) ■ コード (intra-mart Accel Kaiden!) ■ コード (intra-mart Accel Platform) ■ ロケール ■ フラグ ■ 最小文字数 ■ 最大文字数 ■ 固定文字数 ■ 半角英字 ■ 半角英数字 ■ 半角数字 ■ 半角英数記号 ■ 整数 ■ 小数 ■ 数値範囲 ■ 数値範囲 (不等号選択) ■ 最小数値 ■ 最小数値 (不等号選択) ■ 最大数値 ■ 最大数値 (不等号選択) ■ 日付形式 ■ 日付範囲 ■ 日付範囲 (不等号選択) ■ 最小日付 ■ 最小日付 (不等号選択) ■ 最大日付 ■ 最大日付 (不等号選択) ■ 時刻 ■ 時刻範囲 ■ 時刻範囲 (不等号選択) ■ 最小時刻 ■ 最小時刻 (不等号選択) ■ 最大時刻 ■ 最大時刻 (不等号選択) ■ 時刻単位 ■ 正規表現 ■ 不一致 ■ 通貨額 ■ FB文字 ■ intra-mart Accel Kaiden! IM-Workflow連携モジュール ■ ワークフロー日付

はじめに

本バリデーションを使用する際の注意点

本バリデーションはバリデーション対象の項目に設定する必要があります。

```
<k:text
  name="sample"
  mode="0"
  validationDefault="{\"k_required:[\"]}"
/>
```

intra-mart Accel Kaiden! 基盤モジュール

必須

■ 概要 本バリデーションが設定された項目に、値が入力されているか否かを判定します。 チェックOK :対象項目に値が入力されている。 チェックNG :対象項目に値が入力されていない。	
■ 詳細 バリデーションクラス RequiredValidator	

ID (JavaScript)	k_required
ID (Java)	required
出力メッセージ	必須です。

- 引数
本バリデーションに引数はありません。

いずれか必須

- 概要
引数で指定されたすべての項目に、値が入力されているか否かを判定します。

チェックOK : 引数で指定されたいずれかの項目に値が入力されている。
 チェックNG : 引数で指定されたすべての項目に値が入力されていない。

- 詳細

バリデーションクラス	RequiredAnyValidator
ID (JavaScript)	k_requiredAny
ID (Java)	requiredAny
出力メッセージ	いずれか必須です。{0}
	{0}: 引数で指定されたフィールドの名称を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	チェック対象フィールド(「name」属性)の配列を設定します。	○

いずれか必須 (checkbox)

- 概要
引数で指定されたすべての項目 (checkbox) が、選択されているか否かを判定します。

チェックOK : 引数で指定されたいずれかの項目が選択されている。
 チェックNG : 引数で指定されたすべての項目が選択されていない。

- 詳細

バリデーションクラス	AnyOnForCheckValidator
ID (JavaScript)	k_anyOnForCheck
ID (Java)	anyOnForCheck
出力メッセージ	いずれか必須です。{0}
	{0}: 引数で指定されたフィールドの名称を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	チェック対象フィールド(「name」属性)の配列を設定します。	○

コード (intra-mart Accel Kaiden!)

- 概要
本バリデーションが設定された項目の値が、intra-mart Accel Kaiden!のコード体系で入力されているか否かを判定します。

チェックOK : 対象項目の値がintra-mart Accel Kaiden!のコード体系のみで入力されている。
 チェックNG : 対象項目の値にintra-mart Accel Kaiden!のコード体系以外の文字が含まれている。



コラム

intra-mart Accel Kaiden!のコード体系に含まれる文字は次の通りです。

- 半角英数字: A-Za-z0-9
- 記号: _-

- 詳細

バリデーションクラス	KaidenIdValidator
ID (JavaScript)	k_kaidenId
ID (Java)	kaidenId
出力メッセージ	半角英数字、「_」、「-」で入力してください。

- 引数
本バリデーションに引数はありません。

コード (intra-mart Accel Platform)

- 概要
本バリデーションが設定された項目の値が、intra-mart Accel Platformのコード体系で入力されているか否かを判定します。

チェックOK : 対象項目の値がintra-mart Accel Platformのコード体系のみで入力されている。

チェックNG : 対象項目の値にintra-mart Accel Platformのコード体系以外の文字が含まれている。



コラム

intra-mart Accel Platformのコード体系に含まれる文字は次の通りです。

- 半角英数字: A-Za-z0-9
- 記号: _-@. +!

- 詳細

バリデーションクラス	ImIdValidator
ID (JavaScript)	k_imId
ID (Java)	imId
出力メッセージ	半角英数字、「_」、「-」、「@」、「.」、「+」、「!」で入力してください。

- 引数
本バリデーションに引数はありません。

ロケール

- 概要
本バリデーションが設定された項目の値が、intra-mart Accel Platformのシステムロケールに存在するロケールか否かを判定します。

チェックOK : 対象項目の値が、intra-mart Accel Platformのシステムロケールに存在する。

チェックNG : 対象項目の値が、intra-mart Accel Platformのシステムロケールに存在しない。

- 詳細

バリデーションクラス	LocaleValidator
ID (JavaScript)	(JavaScriptでのバリデーションなし)
ID (Java)	locale
出力メッセージ	指定されたロケールは存在しません。

- 引数
本バリデーションに引数はありません。

フラグ

- 概要
本バリデーションが設定された項目の値が、「0」または「1」で入力されているか否かを判定します。

チェックOK : 対象項目の値が「0」または「1」で入力されている。

チェックNG : 対象項目の値が「0」または「1」で入力されていない。

- 詳細

バリデーションクラス	FlagValidator
ID (JavaScript)	k_flag
ID (Java)	flag
出力メッセージ	フラグとして認識できない文字です。

- 引数
本バリデーションに引数はありません。

最小文字数

■ 概要

本バリデーションが設定された項目の値が、「最小文字数 <= 入力文字数」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小文字数 <= 入力文字数」の範囲内で入力されている。

チェックNG :対象項目の値が「最小文字数 <= 入力文字数」の範囲を超えて入力されている。

■ 詳細

バリデーションクラス	MinLengthValidator
ID (JavaScript)	k_minLength
ID (Java)	minLength
出力メッセージ	{0}文字以上で入力してください。
	{0}: 引数で指定された最小文字数を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小文字数を設定します。	○

最大文字数

■ 概要

本バリデーションが設定された項目の値が、「入力文字数 <= 最大文字数」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「入力文字数 <= 最大文字数」の範囲内で入力されている。

チェックNG :対象項目の値が「入力文字数 <= 最大文字数」の範囲を超えて入力されている。

■ 詳細

バリデーションクラス	MaxLengthValidator
ID (JavaScript)	k_maxLength
ID (Java)	maxLength
出力メッセージ	{0}文字以下で入力してください。
	{0}: 引数で指定された最大文字数を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大文字数を設定します。	○

固定文字数

■ 概要

本バリデーションが設定された項目の値が、固定文字数で入力されているか否かを判定します。

チェックOK :対象項目の値が固定文字数で入力されている。

チェックNG :対象項目の値が固定文字数以外の文字数で入力されている。

■ 詳細

バリデーションクラス	FixedLengthValidator
ID (JavaScript)	k_fixedLength
ID (Java)	fixedLength
出力メッセージ	{0}文字で入力してください。
	{0}: 引数で指定された固定文字数を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定 順	説明	必須
1	固定文字数を設定します。	○

半角英字

■ 概要

本バリデーションが設定された項目の値が、半角英字 (A-Za-z) で入力されているか否かを判定します。

チェックOK : 対象項目の値が半角英字のみで入力されている。

チェックNG : 対象項目の値に半角英字以外の文字が含まれている。

■ 詳細

バリデーションクラス	AlphaValidator
ID (JavaScript)	k_alpha
ID (Java)	alpha
出力メッセージ	半角英字で入力してください。

■ 引数

本バリデーションに引数はありません。

半角英数字

■ 概要

本バリデーションが設定された項目の値が、半角英数字 (A-Za-z0-9) で入力されているか否かを判定します。

チェックOK : 対象項目の値が半角英数字のみで入力されている。

チェックNG : 対象項目の値に半角英数字以外の文字が含まれている。

■ 詳細

バリデーションクラス	AlphaNumericValidator
ID (JavaScript)	k_alphaNumeric
ID (Java)	alphaNumeric
出力メッセージ	半角英数字で入力してください。

■ 引数

本バリデーションに引数はありません。

半角数字

■ 概要

本バリデーションが設定された項目の値が、半角数字 (0-9) で入力されているか否かを判定します。

チェックOK : 対象項目の値が半角数字のみで入力されている。

チェックNG : 対象項目の値に半角数字以外の文字が含まれている。

■ 詳細

バリデーションクラス	NumericValidator
ID (JavaScript)	k_numeric
ID (Java)	numeric
出力メッセージ	半角数字で入力してください。

■ 引数

本バリデーションに引数はありません。

半角英数記号

■ 概要

本バリデーションが設定された項目の値が、半角英数 (A-Za-z0-9) と半角記号で入力されているか否かを判定します。

チェックOK : 対象項目の値が半角英数記号のみで入力されている。

チェックNG : 対象項目の値に半角英数記号以外の文字が含まれている。

■ 詳細

バリデーションクラス	AlphaNumericSymbolValidator
ID (JavaScript)	k_alphaNumericSymbol
ID (Java)	alphaNumericSymbol

出力メッセージ	半角英数字記号で入力してください。

- 引数
本バリデーションに引数はありません。

整数

- 概要
本バリデーションが設定された項目の値が、半角数字、マイナス(0-9-)で入力されているか否かを判定します。

チェックOK :対象項目の値が半角数字、マイナスのみで入力されており、整数部桁数以内の文字数で入力されている。
チェックNG :対象項目の値に半角数字、マイナス以外の文字が含まれている、または、整数部桁数を超える文字数で入力されている。

- 詳細

バリデーションクラス	IntegerValidator
ID (JavaScript)	k_integer
ID (Java)	integer
出力メッセージ	{0}桁までの数値を入力してください。 {0}:引数で指定された整数部の桁数を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	マイナス記号の桁数を除いた整数部の桁数を設定します。	☐

小数

- 概要
本バリデーションが設定された項目の値が、半角数字、ドット、マイナス(0-9.-)で入力されているか否かを判定します。

チェックOK :対象項目の値が半角数字、ドット、マイナスのみで入力されており、整数部、小数部が指定桁数以内の文字数で入力されている。
チェックNG :対象項目の値に半角数字、ドット、マイナス以外の文字が含まれている、または、整数部、小数部のいずれかが指定桁数を超える文字数で入力されている。

 コラム

整数部桁数に「0」を指定した場合でも、整数部の値が「0」であればチェックOKとします。

- 詳細

バリデーションクラス	DigitsValidator
ID (JavaScript)	k_digits
ID (Java)	digits
出力メッセージ	整数部{0}桁、小数部{1}桁を入力してください。 {0}:引数で指定された整数部の桁数を出力します。 {1}:引数で指定された小数部の桁数を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	整数部の桁数を設定します。	☐
2	小数部の桁数を設定します。	☐

数値範囲

- 概要
本バリデーションが設定された項目の値が、「最小数値 <= 入力値 <= 最大数値」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小数値 <= 入力値 <= 最大数値」の範囲内で入力されている。
チェックNG :対象項目の値が「最小数値 <= 入力値 <= 最大数値」の範囲を超えて入力されている。

 注意

最小数値、最大数値、入力値のすべてが数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック(数字・整数・小数)と組み合わせて使用してください。

詳細

バリデーションクラス	NumericRangeValidator
ID (JavaScript)	k_numericRange
ID (Java)	numericRange
出力メッセージ	{0} - {1}の範囲で入力してください。 {0}: 引数で指定された最小数値を出力します。 {1}: 引数で指定された最大数値を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小数値を設定します。 最小数値を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大数値を設定します。 最大数値を保持するフィールドの「name」属性を設定することも可能です。	○

数値範囲 (不等号選択)

概要

本バリデーションが設定された項目の値が、「最小数値 <= (または <) 入力値 <= (または <) 最大数値」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「最小数値 <= (または <) 入力値 <= (または <) 最大数値」の範囲内で入力されている。
チェックNG : 対象項目の値が「最小数値 <= (または <) 入力値 <= (または <) 最大数値」の範囲を超えて入力されている。

**注意**
最小数値、最大数値、入力値のすべてが数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック (数字・整数・小数) と組み合わせて使用してください。

詳細

バリデーションクラス	NumericRange2Validator
ID (JavaScript)	k_numericRange2
ID (Java)	numericRange2
出力メッセージ	引数の条件で次のメッセージを表示します。 {0}以上、{1}以下の範囲で入力してください。 {0}以上、{1}未満の範囲で入力してください。 {0}超、{1}以下の範囲で入力してください。 {0}超、{1}未満の範囲で入力してください。 {0}: 引数で指定された最小数値を出力します。 {1}: 引数で指定された最大数値を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小数値を設定します。 最小数値を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大数値を設定します。 最大数値を保持するフィールドの「name」属性を設定することも可能です。	○
3	入力値が最小数値と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(最小数値 <= 入力値) "false": チェックNGとします。(最小数値 < 入力値) ※省略時は"false"が設定されたものとします。	
4	入力値が最大数値と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(入力値 <= 最大数値) "false": チェックNGとします。(入力値 < 最大数値) ※省略時は"false"が設定されたものとします。	

最小数値

概要

本バリデーションが設定された項目の値が、「最小数値 <= 入力値」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「最小数値 <= 入力値」の範囲内で入力されている。
チェックNG : 対象項目の値が「最小数値 <= 入力値」の範囲を超えて入力されている。

**注意**

最小数値、入力値の両方が数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック(数字・整数・小数)と組み合わせて使用してください。

詳細

バリデーションクラス	NumericMinValidator
ID (JavaScript)	k_numericMin
ID (Java)	numericMin
出力メッセージ	{0}以上で入力してください。 {0}: 引数で指定された最小数値を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小数値を設定します。 最小数値を保持するフィールドの「name」属性を設定することも可能です。	○

最小数値(不等号選択)

概要

本バリデーションが設定された項目の値が、「最小数値 <= (または <) 入力値」の範囲で入力されているか否かを判定します。

チェックOK: 対象項目の値が「最小数値 <= (または <) 入力値」の範囲内で入力されている。

チェックNG: 対象項目の値が「最小数値 <= (または <) 入力値」の範囲を超えて入力されている。



注意

最小数値、入力値の両方が数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック(数字・整数・小数)と組み合わせて使用してください。

詳細

バリデーションクラス	NumericMin2Validator
ID (JavaScript)	k_numericMin2
ID (Java)	numericMin2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以上で入力してください。 {0}より大きい値で入力してください。 {0}: 引数で指定された最小数値を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小数値を設定します。 最小数値を保持するフィールドの「name」属性を設定することも可能です。	○
2	入力値が最小数値と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(最小数値 <= 入力値) "false": チェックNGとします。(最小数値 < 入力値) ※省略時は"false"が設定されたものとします。	

最大数値

概要

本バリデーションが設定された項目の値が、「入力値 <= 最大数値」の範囲で入力されているか否かを判定します。

チェックOK: 対象項目の値が「入力値 <= 最大数値」の範囲内で入力されている。

チェックNG: 対象項目の値が「入力値 <= 最大数値」の範囲を超えて入力されている。



注意

最大数値、入力値の両方が数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック(数字・整数・小数)と組み合わせて使用してください。

詳細

バリデーションクラス	NumericMaxValidator
ID (JavaScript)	k_numericMax
ID (Java)	numericMax

出力メッセージ	{0}以下で入力してください。
	{0}: 引数で指定された最大数値を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大数値を設定します。 最大数値を保持するフィールドの「name」属性を設定することも可能です。	○

最大数値（不等号選択）

- 概要
本バリデーションが設定された項目の値が、「入力値 <=（または <）最大数値」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「入力値 <=（または <）最大数値」の範囲内で入力されている。
チェックNG : 対象項目の値が「入力値 <=（または <）最大数値」の範囲を超えて入力されている。

 **注意**
最大数値、入力値の両方が数値変換できない場合はチェックを実施しません。
数値型のチェックは含まれないため、他のチェック（数字・整数・小数）と組み合わせて使用してください。

- 詳細

バリデーションクラス	NumericMax2Validator
ID (JavaScript)	k_numericMax2
ID (Java)	numericMax2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以下で入力してください。 {0}未満で入力してください。 {0}: 引数で指定された最大数値を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大数値を設定します。 最大数値を保持するフィールドの「name」属性を設定することも可能です。	○
2	入力値が最大数値と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(入力値 <= 最大数値) "false": チェックNGとします。(入力値 < 最大数値) ※省略時は"false"が設定されたものとします。	

日付形式

- 概要
本バリデーションが設定された項目の値が、「日付形式 (YYYY/MM/DD) かつ、システムの有効期間内」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が日付形式で入力されており、システムの有効期間の範囲内で入力されている。
チェックNG : 対象項目の値が日付形式以外の形式で入力されている、またはシステムの有効期間の範囲を超えて入力されている。

 **コラム**
システムの有効期間（初期値）は次の通りです。

- システムの有効期間開始日 : 1900/01/01
- システムの有効期間終了日 : 2999/12/31

- 詳細

バリデーションクラス	DateValidator
ID (JavaScript)	k_date
ID (Java)	date
出力メッセージ	{0} - {1}の範囲で入力してください。 {0}: システム有効期間開始日 {1}: システム有効期間終了日

- 引数
本バリデーションに引数はありません。

日付範囲

- 概要
本バリデーションが設定された項目の値が、「最小日付 <= 入力値 <= 最大日付」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小日付 <= 入力値 <= 最大日付」の範囲内で入力されている。
チェックNG :対象項目の値が「最小日付 <= 入力値 <= 最大日付」の範囲を超えて入力されている。



注意

最小日付、最大日付、入力値のすべてが日付変換できない場合はチェックを実施しません。
 日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

- 詳細

バリデーションクラス	DateRangeValidator
ID (JavaScript)	k_dateRange
ID (Java)	dateRange
出力メッセージ	{0} - {1}の範囲で入力してください。 {0}: 引数で指定された最小日付を出力します。 {1}: 引数で指定された最大日付を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小日付 (最過去日付) を設定します。 最小日付 (最過去日付) を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大日付 (最未来日付) を設定します。 最大日付 (最未来日付) を保持するフィールドの「name」属性を設定することも可能です。	○

日付範囲(不等号選択)

- 概要
本バリデーションが設定された項目の値が、「最小日付 <= (または <) 入力値 <= (または <) 最大日付」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小日付 <= (または <) 入力値 <= (または <) 最大日付」の範囲内で入力されている。
チェックNG :対象項目の値が「最小日付 <= (または <) 入力値 <= (または <) 最大日付」の範囲を超えて入力されている。



注意

最小日付、最大日付、入力値のすべてが日付変換できない場合はチェックを実施しません。
 日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

- 詳細

バリデーションクラス	DateRange2Validator
ID (JavaScript)	k_dateRange2
ID (Java)	dateRange2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以上、{1}以下の範囲で入力してください。 {0}以上、{1}未満の範囲で入力してください。 {0}超、{1}以下の範囲で入力してください。 {0}超、{1}未満の範囲で入力してください。 {0}: 引数で指定された最小日付を出力します。 {1}: 引数で指定された最大日付を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小日付 (最過去日付) を設定します。 最小日付 (最過去日付) を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大日付 (最未来日付) を設定します。 最大日付 (最未来日付) を保持するフィールドの「name」属性を設定することも可能です。	○

設定順	説明	必須
3	入力値が最小日付（最過去日付）と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。（最小日付 <= 入力値） "false": チェックNGとします。（最小日付 < 入力値） ※省略時は"false"が設定されたものとします。	
4	入力値が最大日付（最未来日付）と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。（入力値 <= 最大日付） "false": チェックNGとします。（入力値 < 最大日付） ※省略時は"false"が設定されたものとします。	

最小日付

■ 概要

本バリデーションが設定された項目の値が、「最小日付 <= 入力値」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「最小日付 <= 入力値」の範囲内で入力されている。

チェックNG : 対象項目の値が「最小日付 <= 入力値」の範囲を超えて入力されている。



注意

最小日付、入力値の両方が日付変換できない場合はチェックを実施しません。
日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

■ 詳細

バリデーションクラス DateMinValidator

ID (JavaScript)	k_dateMin

ID (Java)	dateMin

出力メッセージ	{0}以上で入力してください。

	{0}: 引数で指定された最小日付を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小日付（最過去日付）を設定します。 最小日付（最過去日付）を保持するフィールドの「name」属性を設定することも可能です。	○

最小日付（不等号選択）

■ 概要

本バリデーションが設定された項目の値が、「最小日付 <=（または <）入力値」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「最小日付 <=（または <）入力値」の範囲内で入力されている。

チェックNG : 対象項目の値が「最小日付 <=（または <）入力値」の範囲を超えて入力されている。



注意

最小日付、入力値の両方が日付変換できない場合はチェックを実施しません。
日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

■ 詳細

バリデーションクラス DateMin2Validator

ID (JavaScript)	k_dateMin2

ID (Java)	dateMin2

出力メッセージ	引数の条件で以下のメッセージを表示します。

	{0}以上で入力してください。

	{0}より大きい値で入力してください。

	{0}: 引数で指定された最小日付を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小日付（最過去日付）を設定します。 最小日付（最過去日付）を保持するフィールドの「name」属性を設定することも可能です。	○

設定順	説明	必須
2	入力値が最小日付（最過去日付）と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。（最小日付 <= 入力値） "false": チェックNGとします。（最小日付 < 入力値） ※省略時は"false"が設定されたものとします。	

最大日付

- 概要
本バリデーションが設定された項目の値が、「入力値 <= 最大日付」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「入力値 <= 最大日付」の範囲内で入力されている。
チェックNG : 対象項目の値が「入力値 <= 最大日付」の範囲を超えて入力されている。

 注意

最小日付、入力値の両方が日付変換できない場合はチェックを実施しません。
日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

- 詳細

バリデーションクラス	DateMaxValidator
ID (JavaScript)	k_dateMax
ID (Java)	dateMax
出力メッセージ	{0}以下で入力してください。 {0}: 引数で指定された最大日付を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大日付（最未来日付）を設定します。 最大日付（最未来日付）を保持するフィールドの「name」属性を設定することも可能です。	○

最大日付（不等号選択）

- 概要
本バリデーションが設定された項目の値が、「入力値 <=（または <）最大日付」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「入力値 <=（または <）最大日付」の範囲内で入力されている。
チェックNG : 対象項目の値が「入力値 <=（または <）最大日付」の範囲を超えて入力されている。

 注意

最小日付、入力値の両方が日付変換できない場合はチェックを実施しません。
日付型のチェックは含まれないため、他のチェック（日付）と組み合わせて使用してください。

- 詳細

バリデーションクラス	DateMax2Validator
ID (JavaScript)	k_dateMax2
ID (Java)	dateMax2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以下で入力してください。 {0}未満で入力してください。 {0}: 引数で指定された最大日付を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大日付（最未来日付）を設定します。 最大日付（最未来日付）を保持するフィールドの「name」属性を設定することも可能です。	○
2	入力値が最大日付（最未来日付）と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。（入力値 <= 最大日付） "false": チェックNGとします。（入力値 < 最大日付） ※省略時は"false"が設定されたものとします。	

概要

本バリデーションが設定された項目の値が、時刻形式(H:mm)で入力されているか否かを判定します。

チェックOK :対象項目の値が時刻形式で入力されている。

チェックNG :対象項目の値が時刻形式で入力されていない。

コラム

24時以降(99:59)と入力された場合でもチェックNGにはなりません。

詳細

バリデーションクラス	TimeValidator
ID (JavaScript)	k_time
ID (Java)	time
出力メッセージ	時刻の形式で入力してください。

引数

本バリデーションに引数はありません。

時刻範囲

概要

本バリデーションが設定された項目の値が、「最小時刻 <= 入力値 <= 最大時刻」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小時刻 <= 入力値 <= 最大時刻」の範囲内で入力されている。

チェックNG :対象項目の値が「最小時刻 <= 入力値 <= 最大時刻」の範囲を超えて入力されている。

注意

最小時刻、最大時刻、入力値のすべてが時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

詳細

バリデーションクラス	TimeRangeValidator
ID (JavaScript)	k_timeRange
ID (Java)	timeRange
出力メッセージ	{0}- {1}の範囲で入力してください。 {0}: 引数で指定された最小時刻を出力します。 {1}: 引数で指定された最大時刻を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小時刻を設定します。 最小時刻を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大時刻を設定します。 最大時刻を保持するフィールドの「name」属性を設定することも可能です。	○

時刻範囲(不等号選択)

概要

本バリデーションが設定された項目の値が、「最小時刻 <= (または <) 入力値 <= (または <) 最大時刻」の範囲で入力されているか否かを判定します。

チェックOK :対象項目の値が「最小時刻 <= (または <) 入力値 <= (または <) 最大時刻」の範囲内で入力されている。

チェックNG :対象項目の値が「最小時刻 <= (または <) 入力値 <= (または <) 最大時刻」の範囲を超えて入力されている。

注意

最小時刻、最大時刻、入力値のすべてが時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

詳細

バリデーションクラス	TimeRange2Validator
ID (JavaScript)	k_timeRange2
ID (Java)	timeRange2

出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以上、{1}以下の範囲で入力してください。 {0}以上、{1}未満の範囲で入力してください。 {0}超、{1}以下の範囲で入力してください。 {0}超、{1}未満の範囲で入力してください。
	{0}: 引数で指定された最小時刻を出力します。 {1}: 引数で指定された最大時刻を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小時刻を設定します。 最小時刻を保持するフィールドの「name」属性を設定することも可能です。	○
2	最大時刻を設定します。 最大時刻を保持するフィールドの「name」属性を設定することも可能です。	○
3	入力値が最小時刻と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(最小時刻 <= 入力値) "false": チェックNGとします。(最小時刻 < 入力値) ※省略時は"false"が設定されたものとします。	
4	入力値が最大時刻と同じ値の場合にチェックOKとするか否かを設定します。 "true": チェックOKとします。(入力値 <= 最大時刻) "false": チェックNGとします。(入力値 < 最大時刻) ※省略時は"false"が設定されたものとします。	

最小時刻

■ 概要

本バリデーションが設定された項目の値が、「最小時刻 <= 入力値」の範囲で入力されているか否かを判定します。

チェックOK: 対象項目の値が「最小時刻 <= 入力値」の範囲内で入力されている。

チェックNG: 対象項目の値が「最小時刻 <= 入力値」の範囲を超えて入力されている。



注意

最小時刻、入力値の両方が時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

■ 詳細

バリデーションクラス	TimeMinValidator
ID (JavaScript)	k_timeMin
ID (Java)	timeMin
出力メッセージ	{0}以上で入力してください。 {0}: 引数で指定された最小時刻を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小時刻を設定します。 最小時刻を保持するフィールドの「name」属性を設定することも可能です。	○

最小時刻(不等号選択)

■ 概要

本バリデーションが設定された項目の値が、「最小時刻 <= (または <) 入力値」の範囲で入力されているか否かを判定します。

チェックOK: 対象項目の値が「最小時刻 <= (または <) 入力値」の範囲内で入力されている。

チェックNG: 対象項目の値が「最小時刻 <= (または <) 入力値」の範囲を超えて入力されている。



注意

最小時刻、入力値の両方が時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

■ 詳細

バリデーションクラス	TimeMin2Validator
ID (JavaScript)	k_timeMin2

ID (Java)	timeMin2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以上で入力してください。 {0}より大きい値で入力してください。
	{0}: 引数で指定された最小時刻を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最小時刻を設定します。 最小時刻を保持するフィールドの「name」属性を設定することも可能です。	○
2	入力値が最小時刻と同じ値の場合にチェックOKとするか否かを設定します。 “true”: チェックOKとします。(最小時刻 <= 入力値) “false”: チェックNGとします。(最小時刻 < 入力値) ※省略時は“false”が設定されたものとします。	

最大時刻

■ 概要

本バリデーションが設定された項目の値が、「入力値 <= 最大時刻」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「入力値 <= 最大時刻」の範囲内で入力されている。

チェックNG : 対象項目の値が「入力値 <= 最大時刻」の範囲を超えて入力されている。



注意

最大時刻、入力値の両方が時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

■ 詳細

バリデーションクラス	TimeMaxValidator
ID (JavaScript)	k_timeMax
ID (Java)	timeMax
出力メッセージ	{0}以下で入力してください。 {0}: 引数で指定された最大時刻を出力します。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大時刻を設定します。 最大時刻を保持するフィールドの「name」属性を設定することも可能です。	○

最大時刻(不等号選択)

■ 概要

本バリデーションが設定された項目の値が、「入力値 <= (または <) 最大時刻」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が「入力値 <= (または <) 最大時刻」の範囲内で入力されている。

チェックNG : 対象項目の値が「入力値 <= (または <) 最大時刻」の範囲を超えて入力されている。



注意

最大時刻、入力値の両方が時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

■ 詳細

バリデーションクラス	TimeMax2Validator
ID (JavaScript)	k_timeMax2
ID (Java)	timeMax2
出力メッセージ	引数の条件で以下のメッセージを表示します。 {0}以下で入力してください。 {0}未満で入力してください。 {0}: 引数で指定された最大時刻を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	最大時刻を設定します。 最大時刻を保持するフィールドの「name」属性を設定することも可能です。	○
2	入力値が最大時刻と同じ値の場合にチェックOKとするか否かを設定します。 “true”:チェックOKとします。(入力値 <= 最大時刻) “false”:チェックNGとします。(入力値 < 最大時刻) ※省略時は“false”が設定されたものとします。	

時刻単位

- 概要
本バリデーションが設定された項目の値が、引数で指定された時刻単位で入力されているか否かを判定します。

チェックOK :対象項目の値が引数で指定された時刻単位で入力されている。
チェックNG :対象項目の値が引数で指定された時刻単位で入力されていない。

 コラム

分 (mm) に相当する値を数値に変換し、時刻単位で除算した結果、余りがない場合にチェックOKとします。

 注意

入力値が時刻変換できない場合はチェックを実施しません。
時刻型のチェックは含まれないため、他のチェック(時刻)と組み合わせて使用してください。

- 詳細

バリデーションクラス TimeUnitValidator	
ID (JavaScript)	k_timeUnit
ID (Java)	timeUnit
出力メッセージ	時刻は{0}単位で入力してください。 {0}: 引数で指定された時刻単位を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	時刻単位(分)を設定します。	○

正規表現

- 概要
本バリデーションが設定された項目の値が、指定された正規表現で入力されているか否かを判定します。

チェックOK :対象項目の値が指定された正規表現に一致する。
チェックNG :対象項目の値が指定された正規表現に一致しない。

- 詳細

バリデーションクラス RegularExpressionValidator	
ID (JavaScript)	k_regularExpression
ID (Java)	regularExpression
出力メッセージ	引数のメッセージIDに指定されたメッセージを表示します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	正規表現を設定します。	○
2	メッセージIDを設定します。	○

不一致

- 概要
引数で指定された比較対象1と比較対象2の値が異なる値か否かを判定します。

チェックOK : 引数で指定された比較対象1と比較対象2の値が異なる。
チェックNG : 引数で指定された比較対象1と比較対象2の値が同じ。

詳細

バリデーションクラス NotEqualValidator	
ID (JavaScript)	k_notEqual
ID (Java)	notEqual
出力メッセージ	{0}と{1}は同一の設定はできません。 {0}: 引数で指定された比較対象1を出力します。 {1}: 引数で指定された比較対象2を出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	比較対象1を設定します。	○
2	比較対象2を設定します。	○

通貨額

概要

本バリデーションが設定された項目の値が、「対象通貨の精度・スケール」で入力されているか否かを判定します。

チェックOK : 対象項目の値が「対象通貨の精度・スケール」で入力されている。
チェックNG : 対象項目の値に「対象通貨の精度・スケール」を超えて入力されている。

 コラム

通貨コードの精度・スケールは通貨精度区分が「KAIDEN」のデータのみをチェック対象とします。
通貨精度が見つからなかった場合は「整数部9、小数部4 (constansts.xmlの定義値)」を使用します。

詳細

バリデーションクラス CurrencyAmntValidator	
ID (JavaScript)	k_currencyAmnt
ID (Java)	currencyAmnt
出力メッセージ	整数部{0}桁、小数部{1}桁を入力してください。 {0}: 引数で指定された通貨の精度を出力します。 {1}: 引数で指定された通貨のスケールを出力します。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	通貨コードを保持するフィールドの「name」属性を設定します。	
2	会社コードを保持するフィールドの「name」属性を設定します。	○

FB文字

概要

本バリデーションが設定された項目の値が、「振込データで使用可能な文字」で入力されているか否かを判定します。

チェックOK : 対象項目の値が「振込データで使用可能な文字」のみで入力されている。
チェックNG : 対象項目の値に「振込データで使用可能な文字」以外の文字が含まれている。

 コラム

振込データで使用可能な文字は次の通りです。

- カナ：アからン(半角)※小文字含まない
- 英字：AからZ(半角)
- 数字：0から9(半角)
- 濁点・半濁点：・゜(半角)
- 記号：() 「 」 - / . スペース(半角)

詳細

バリデーションクラス	FbCharValidator
ID (JavaScript)	k_fbChar
ID (Java)	fbChar
出力メッセージ	入力できない文字が含まれています。

- 引数
本バリデーションに引数はありません。

intra-mart Accel Kaiden! IM-Workflow連携モジュール

ワークフロー日付

- 概要
本バリデーションが設定された項目の値が、「日付形式 (YYYY/MM/DD) かつ、ワークフローパラメータの有効期間内」の範囲で入力されているか否かを判定します。

チェックOK : 対象項目の値が日付形式で入力されており、ワークフローパラメータの有効期間の範囲内で入力されている。

チェックNG : 対象項目の値が日付形式以外の形式で入力されている、またはワークフローパラメータの有効期間の範囲を超えて入力されている。

コラム

ワークフローパラメータの有効期間 (初期値) は次の通りです。

- ワークフローパラメータの有効期間開始日: 1900/01/01
- ワークフローパラメータの有効期間終了日: 2999/12/31

- 詳細

バリデーションクラス	ImwDateValidator
ID (JavaScript)	k_imwDate
ID (Java)	imwDate
出力メッセージ	{0} - {1}の範囲で入力してください。 {0}:ワークフローパラメータの有効期間開始日 {1}:ワークフローパラメータの有効期間終了日

- 引数
本バリデーションに引数はありません。

- はじめに
 - intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用する際の設定について
 - ガジェットでの引数「会社コード」について
 - ガジェットでの引数「検索基準日」について
- intra-mart Accel Kaiden! 基盤モジュール
 - 会社
 - 組織
 - ユーザ
 - ロール
 - 通貨
 - 項目区分
 - 項目区分 追加可否
 - 項目
 - 金融機関
 - 会社口座
- intra-mart Accel Kaiden! ワークフローモジュール
 - プロジェクト
 - 経費区分
 - 勘定科目
 - 税区分
 - 日当
 - 日当金額
 - 役職(日当関連)
 - 仕入先
 - 支払方法
 - 支払分類
 - 会計期間
- intra-mart Accel Kaiden! 勤務管理モジュール
 - 勤務体系カテゴリ
 - 勤務体系
 - 勤務状況カテゴリ
 - 勤務状況
 - 勤務日付区分
 - 休憩・控除・時間年中分類
 - 休憩・控除・時間年休
 - 勤務カレンダー
 - 権限セット

はじめに

intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用する際の設定について

intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用するJSPのページで、次の様に共通JSPをインクルードする必要があります。

```
<c:if test="${empty pwkfCommonIncluded}" scope="request">
  <jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>
```



コラム

次の様な設定でも問題ありませんが、同じ画面に複数のガジェットが表示されることを考慮し、画面全体で一度だけ読み込まれるようにしています。

```
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
```

ガジェットでの引数「会社コード」について

ガジェット内では、バリデーションへの引数「会社コード」に「searchCriteriaCompany」を設定することも可能です。
「searchCriteriaCompany」を設定することで、申請画面や処理画面自体で保持する検索基準会社のコード値を取得可能です。

ガジェットでの引数「検索基準日」について

ガジェット内では、バリデーションへの引数「検索基準日」に「searchCriteriaDate」を設定することも可能です。
「searchCriteriaDate」を設定することで、申請画面や処理画面自体で保持する検索基準日を取得可能です。

intra-mart Accel Kaiden! 基盤モジュール

会社

- 概要
引数に指定した値が「IM-共通マスタの会社(組織)マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「IM-共通マスタの会社（組織）マスタ」に存在する。
チェックNG :対象項目の値が、「IM-共通マスタの会社（組織）マスタ」に存在しない。

■ 詳細

バリデーションクラス	CompanyExistsValidator
ID (JavaScript)	k_companyExists
ID (Java)	companyExists
出力メッセージ	指定された会社は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	会社コードを保持するフィールドの「name」属性を設定します。	○
2	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
3	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

組織

■ 概要

引数に指定した値が「IM-共通マスタの組織マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「IM-共通マスタの組織マスタ」に存在する。
チェックNG :対象項目の値が、「IM-共通マスタの組織マスタ」に存在しない。

■ 詳細

バリデーションクラス	DepartmentExistsValidator
ID (JavaScript)	k_departmentExists
ID (Java)	departmentExists
出力メッセージ	指定された組織は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	会社コードを保持するフィールドの「name」属性を設定します。	○
2	組織セットコードを保持するフィールドの「name」属性を設定します。	○
3	組織コードを保持するフィールドの「name」属性を設定します。	○
4	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
5	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

ユーザ

■ 概要

引数に指定した値が「IM-共通マスタのユーザマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「IM-共通マスタのユーザマスタ」に存在する。
チェックNG :対象項目の値が、「IM-共通マスタのユーザマスタ」に存在しない。

■ 詳細

バリデーションクラス	UserExistsValidator
ID (JavaScript)	k_userExists
ID (Java)	userExists
出力メッセージ	指定されたユーザは存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	ユーザコードを保持するフィールドの「name」属性を設定します。	○
2	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
3	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

ロール

- 概要
引数に指定した値が「IM-共通マスタのロールマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「IM-共通マスタのロールマスタ」に存在する。
チェックNG :対象項目の値が、「IM-共通マスタのロールマスタ」に存在しない。

 コラム

ロールのカテゴリが「KAIDEN」のデータのみをチェック対象とします。

- 詳細

バリデーションクラス	RoleExistsValidator
ID (JavaScript)	k_roleExists
ID (Java)	roleExists
出力メッセージ	指定されたロールは存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	ロールIDを保持するフィールドの「name」属性を設定します。	○

通貨

- 概要
引数に指定した値が「IM-共通マスタの通貨マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「IM-共通マスタのユーザマスタ」に存在する。
チェックNG :対象項目の値が、「IM-共通マスタのユーザマスタ」に存在しない。

- 詳細

バリデーションクラス	ImmCurrencyExistsValidator
ID (JavaScript)	k_immCurrencyExists
ID (Java)	immCurrencyExists
出力メッセージ	指定された通貨は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	通貨コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

項目区分

- 概要
引数に指定した値が「項目区分マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「項目区分マスタ」に存在する。
チェックNG :対象項目の値が、「項目区分マスタ」に存在しない。

詳細

バリデーションクラス	ItemTypeExistsValidator
ID (JavaScript)	k_itemTypeExists
ID (Java)	itemTypeExists
出力メッセージ	指定された項目区分は存在しません。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	項目区分コードを保持するフィールドの「name」属性を設定します。	○

項目区分 追加可否

概要

引数に指定した項目区分が項目の追加を許可してるか否かを判定します。

チェックOK : 指定した項目区分が項目の追加を許可している。

チェックNG : 指定した項目区分が項目の追加を許可していない。

詳細

バリデーションクラス	ItemTypeAllowAddItemValidator
ID (JavaScript)	k_itemTypeAllowAddItem
ID (Java)	itemTypeExists
出力メッセージ	指定された項目区分には項目を追加できません。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	項目区分コードを保持するフィールドの「name」属性を設定します。	○

項目

概要

引数に指定した値が「項目マスタ」に存在するか否かを判定します。

チェックOK : 対象項目の値が、「項目マスタ」に存在する。

チェックNG : 対象項目の値が、「項目マスタ」に存在しない。

 **コラム**

会社コードフィールドに設定された会社が存在せず、対象項目区分が追加・削除とも不可の場合は、ビルトインデータのみをチェック対象とします。
会社コードフィールドに設定された会社が存在せず、対象項目区分が上記以外の場合はチェックNGとします。

詳細

バリデーションクラス	ItemExistsValidator
ID (JavaScript)	k_itemExists
ID (Java)	itemExists
出力メッセージ	指定された項目は存在しません。

引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	項目IDを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	ステータスを設定します。 “true”: 論理削除されている場合でもチェックOKとします。 “false”: 論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

金融機関

概要

引数に指定した値が「金融機関マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「金融機関マスタ」に存在する。
チェックNG :対象項目の値が、「金融機関マスタ」に存在しない。

■ 詳細

バリデーションクラス BankFacilExistsValidator	
ID (JavaScript)	k_bankFacilExists
ID (Java)	bankFacilExists
出力メッセージ	指定された金融機関は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	金融機関コードを保持するフィールドの「name」属性を設定します。	○
2	金融機関支店番号を保持するフィールドの「name」属性を設定します。	○
3	金融機関並びコードを保持するフィールドの「name」属性を設定します。 ※省略時は検索条件に含みません。	

会社口座

■ 概要

引数に指定した値が「会社口座マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「会社口座マスタ」に存在する。
チェックNG :対象項目の値が、「会社口座マスタ」に存在しない。

■ 詳細

バリデーションクラス CompanyBankExistsValidator	
ID (JavaScript)	k_companyBankExists
ID (Java)	companyBankExists
出力メッセージ	指定された会社口座は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	会社口座コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとしします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとしします。	

intra-mart Accel Kaiden! ワークフローモジュール

プロジェクト

■ 概要

引数に指定した値が「プロジェクトマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「プロジェクトマスタ」に存在する。
チェックNG :対象項目の値が、「プロジェクトマスタ」に存在しない。

■ 詳細

バリデーションクラス ProjectExistsValidator	
ID (JavaScript)	k_projectExists
ID (Java)	projectExists
出力メッセージ	指定されたプロジェクトは存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	プロジェクトコードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

経費区分

- 概要
引数に指定した値が「経費区分マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「経費区分マスタ」に存在する。
チェックNG :対象項目の値が、「経費区分マスタ」に存在しない。

- 詳細
バリデーションクラス ExpTypeExistsValidator

ID (JavaScript)	k_expTypeExists
ID (Java)	expTypeExists
出力メッセージ	指定された経費区分は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	経費区分コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

勘定科目

- 概要
引数に指定した値が「勘定科目マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勘定科目マスタ」に存在する。
チェックNG :対象項目の値が、「勘定科目マスタ」に存在しない。

- 詳細
バリデーションクラス AccountExistsValidator

ID (JavaScript)	k_accountExists
ID (Java)	accountExists
出力メッセージ	指定された勘定科目は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勘定科目コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

税区分

- 概要
引数に指定した値が「税区分マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「税区分マスタ」に存在する。
チェックNG :対象項目の値が、「税区分マスタ」に存在しない。

- 詳細

バリデーションクラス TaxTypeExistsValidator	
ID (JavaScript)	k_taxTypeExists
ID (Java)	taxTypeExists
出力メッセージ	指定された税区分は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	税区分コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

日当

- 概要
引数に指定した値が「日当マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「日当マスタ」に存在する。
チェックNG :対象項目の値が、「日当マスタ」に存在しない。

- 詳細

バリデーションクラス AllowanceExistsValidator	
ID (JavaScript)	k_allowanceExists
ID (Java)	allowanceExists
出力メッセージ	指定された日当は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	日当コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

日当金額

- 概要
引数に指定した日当が使用できるか否かを判定します。

チェックOK :引数に指定した日当が使用できる。
チェックNG :引数に指定した日当が使用できない。

 コラム

引数で指定したユーザの資格種別（役職/ロール）を基に、引数で指定した日当が使用可能かをチェックします。

- 詳細

バリデーションクラス AllowanceAmountValidator	
ID (JavaScript)	k_allowanceAmount

ID (Java)	allowanceAmount
出力メッセージ	日当額が正しくありません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	日当コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	ユーザコードを保持するフィールドの「name」属性を設定します。	○
4	通貨コードを保持するフィールドの「name」属性を設定します。	○
5	日当金額を保持するフィールドの「name」属性を設定します。	○
6	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
7	ステータスを設定します。 “true”: 論理削除されている場合でもチェックOKとします。 “false”: 論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

役職 (日当関連)

■ 概要

引数に指定した値が「IM-共通マスタの役職マスタ」に存在するか否かを判定します。

チェックOK : 対象項目の値が、「IM-共通マスタの役職マスタ」に存在する。

チェックNG : 対象項目の値が、「IM-共通マスタの役職マスタ」に存在しない。



コラム

「ProductWorkflowSettings.xml」のdecisionAllowanceDepartmentSetCdiに設定した組織セットコードに存在するかをチェックします。

■ 詳細

バリデーションクラス AllowancePostExistsValidator

ID (JavaScript)	k_allowancePostExists
ID (Java)	allowancePostExists
出力メッセージ	指定された役職は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	役職コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”: 論理削除されている場合でもチェックOKとします。 “false”: 論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

仕入先

■ 概要

引数に指定した値が「仕入先マスタ」に存在するか否かを判定します。

チェックOK : 対象項目の値が、「仕入先マスタ」に存在する。

チェックNG : 対象項目の値が、「仕入先マスタ」に存在しない。

■ 詳細

バリデーションクラス VendorExistsValidator

ID (JavaScript)	k_vendorExists
ID (Java)	vendorExists
出力メッセージ	指定された仕入先は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	仕入先コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

支払方法

- 概要
引数に指定した値が「支払方法マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「支払方法マスタ」に存在する。
チェックNG :対象項目の値が、「支払方法マスタ」に存在しない。

- 詳細

バリデーションクラス PayMethodExistsValidator	
ID (JavaScript)	k_payMethodExists
ID (Java)	payMethodExists
出力メッセージ	指定された支払方法は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	支払方法コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

支払分類

- 概要
引数に指定した値が「支払分類マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「支払分類マスタ」に存在する。
チェックNG :対象項目の値が、「支払分類マスタ」に存在しない。

- 詳細

バリデーションクラス PayTypeExistsValidator	
ID (JavaScript)	k_payTypeExists
ID (Java)	payTypeExists
出力メッセージ	指定された支払分類は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	支払分類コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”:論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

会計期間

- 概要**
引数に指定した値が「会計期間マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「会計期間マスタ」に存在し、次の条件(※)に一致する。
チェックNG :対象項目の値が、「会計期間マスタ」に存在しない、または次の条件(※)に一致しない。

※条件
処理種別=0(申請)の場合:会計期間オープン日 <= システム日付、会計期間申請締日 > システム日付
処理種別=1(承認)の場合:会計期間オープン日 <= システム日付、会計期間承認締日 > システム日付

- 詳細**

バリデーションクラス	AccountingPeriodExistsValidator
ID (JavaScript)	k_accountingPeriodExists
ID (Java)	accountingPeriodExists
出力メッセージ	指定された会計期間は存在しません。

- 引数**
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	会社コードを保持するフィールドの「name」属性を設定します。	○
2	処理種別を設定します。0:申請1:承認	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

intra-mart Accel Kaiden! 勤務管理モジュール

勤務体系カテゴリ

- 概要**
引数に指定した値が「勤務体系カテゴリマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務体系カテゴリマスタ」に存在する。
チェックNG :対象項目の値が、「勤務体系カテゴリマスタ」に存在しない。

- 詳細**

バリデーションクラス	WsCategoryExistsValidator
ID (JavaScript)	k_wsCategoryExists
ID (Java)	wsCategoryExists
出力メッセージ	指定された勤務体系カテゴリは存在しません。

- 引数**
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務体系カテゴリを保持するフィールドの「name」属性を設定します。	○

勤務体系

- 概要**
引数に指定した値が「勤務体系マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務体系マスタ」に存在する。
チェックNG :対象項目の値が、「勤務体系マスタ」に存在しない。

- 詳細**

バリデーションクラス	WsAgreementExistsValidator
ID (JavaScript)	k_wsAgreementExists
ID (Java)	wsAgreementExists
出力メッセージ	指定された勤務体系は存在しません。

- 引数**
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務体系コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○

勤務状況カテゴリ

■ 概要

引数に指定した値が「勤務状況カテゴリマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務状況カテゴリマスタ」に存在する。

チェックNG :対象項目の値が、「勤務状況カテゴリマスタ」に存在しない。

■ 詳細

バリデーションクラス	WorkStatusCatValidator
ID (JavaScript)	k_workStatusCatExists
ID (Java)	workStatusCatExists
出力メッセージ	指定された勤務状況カテゴリは存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務状況カテゴリを保持するフィールドの「name」属性を設定します。	○

勤務状況

■ 概要

引数に指定した値が「勤務状況マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務状況マスタ」に存在する。

チェックNG :対象項目の値が、「勤務状況マスタ」に存在しない。

■ 詳細

バリデーションクラス	WorkStatusExistsValidator
ID (JavaScript)	k_workStatusExists
ID (Java)	workStatusExists
出力メッセージ	指定された勤務状況は存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務状況コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
4	ステータスを設定します。 “true”:論理削除されている場合でもチェックOKとします。 “false”: 論理削除されている場合はチェックNGとします。 ※省略時は“false”が設定されたものとします。	

勤務日付区分

■ 概要

引数に指定した値が「勤務日付区分マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務日付区分マスタ」に存在する。

チェックNG :対象項目の値が、「勤務日付区分マスタ」に存在しない。

■ 詳細

バリデーションクラス	LbrDayTypeExistsValidator
ID (JavaScript)	k_lbrDayTypeExists
ID (Java)	lbrDayTypeExists
出力メッセージ	指定された勤務日付区分は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務日付区分コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	勤務日付区分カテゴリを保持するフィールドの「name」属性を設定します。 ※省略時は検索条件に含みません。	

休憩・控除・時間年休中分類

- 概要
引数に指定した値が「休憩・控除・時間年休中分類マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「休憩・控除・時間年休中分類マスタ」に存在する。
チェックNG :対象項目の値が、「休憩・控除・時間年休中分類マスタ」に存在しない。

- 詳細

バリデーションクラス	DeduMcateExistsValidator
ID (JavaScript)	k_deduMcateExists
ID (Java)	deduMcateExists
出力メッセージ	指定された休憩・控除・時間年休中分類は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	休憩・控除・時間年休中分類コードを保持するフィールドの「name」属性を設定します。	○

休憩・控除・時間年休

- 概要
引数に指定した値が「休憩・控除・時間年休マスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「休憩・控除・時間年休マスタ」に存在する。
チェックNG :対象項目の値が、「休憩・控除・時間年休マスタ」に存在しない。

- 詳細

バリデーションクラス	DeductionExistsValidator
ID (JavaScript)	k_deductionExists
ID (Java)	deductionExists
出力メッセージ	指定された休憩・控除・時間年休は存在しません。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	休憩・控除・時間年休コードを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	検索基準日、または検索基準日を保持するフィールドの「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

勤務カレンダー

- 概要
引数に指定した値が「勤務カレンダーマスタ」に存在するか否かを判定します。

チェックOK :対象項目の値が、「勤務カレンダーマスタ」に存在する。
チェックNG :対象項目の値が、「勤務カレンダーマスタ」に存在しない。

- 詳細

バリデーションクラス	LbrCalendarExistsValidator
ID (JavaScript)	k_lbrCalendarExists
ID (Java)	lbrCalendarExists
出力メッセージ	指定された勤務カレンダーは存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	勤務カレンダーIDを保持するフィールドの「name」属性を設定します。	○
2	会社コードを保持するフィールドの「name」属性を設定します。	○
3	ステータスを設定します。 "true": 論理削除されている場合でもチェックOKとします。 "false": 論理削除されている場合はチェックNGとします。 ※省略時は"false"が設定されたものとします。	

権限セット

■ 概要

引数に指定した値が「権限セットマスタ」に存在するか否かを判定します。

チェックOK : 対象項目の値が、「権限セットマスタ」に存在する。

チェックNG : 対象項目の値が、「権限セットマスタ」に存在しない。

■ 詳細

バリデーションクラス	AuthoritySetExistsValidator
ID (JavaScript)	k_authoritySetExists
ID (Java)	authoritySetExists
出力メッセージ	指定された権限セットは存在しません。

■ 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	権限セットIDを保持するフィールドの「name」属性を設定します。	○

- はじめに
 - intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用する際の設定について
- intra-mart Accel Kaiden! 基盤モジュール
 - 列挙型存在チェック
- intra-mart Accel Kaiden! ワークフローモジュール
 - 精算ステータス
 - 精算金額

はじめに

intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用する際の設定について

intra-mart Accel Kaiden! ワークフローモジュールのバリデーションを使用するJSPのページで、次の様に共通JSPをインクルードする必要があります。

```
<c:if test="${empty pwkfCommonIncluded}" scope="request">
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
<c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>
```



コラム

次の様な設定でも問題ありませんが、同じ画面に複数のガジェットが表示されることを考慮し、画面全体で一度だけ読み込まれるようにしています。

```
<jsp:include page="/WEB-INF/view/kaiden2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
```

intra-mart Accel Kaiden! 基盤モジュール

列挙型存在チェック

- 概要
引数で指定されたコードが引数の列挙型クラスに存在するか否かを判定します。

チェックOK : 引数で指定されたコードが引数の列挙型クラスに存在する。
チェックNG : 引数で指定されたコードが引数の列挙型クラスに存在しない。

- 詳細

バリデーションクラス	EnumExistsValidator
ID (JavaScript)	(JavaScriptでのバリデーションなし)
ID (Java)	enumExists
出力メッセージ	指定された{0}は存在しません。 {0}: 引数で指定された列挙型のコード値を出力します。

- 引数
本バリデーションの引数は次の通りです。

設定順	説明	必須
1	列挙型のコード値を設定します。	○
2	列挙型のクラスを設定します。	○

intra-mart Accel Kaiden! ワークフローモジュール

精算ステータス

- 概要
引数に指定した事前申請書の精算可否を判定します。

チェックOK : 事前申請書の精算ステータスが「0」(未精算)の場合、または、事前申請書の精算ステータスが「1」(精算中)かつ、引数に指定した精算申請諸が紐づく場合。
チェックNG : 上記以外の場合。

- 詳細

バリデーションクラス	AdvanceAdjustmentStatusValidator
ID (JavaScript)	k_advanceAdjustmentStatus
ID (Java)	advanceAdjustmentStatus
出力メッセージ	精算中もしくは精算が完了している申請書のため、処理できません。

- 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	精算申請書のシステム案件IDを保持するフィールドの「name」属性を設定します。	○
2	精算申請書のユーザデータIDを保持するフィールドの「name」属性を設定します。	○
3	精算申請書のガジェットクラスを保持するフィールドの「name」属性を設定します。	○
4	精算申請書のガジェットインスタンスを保持するフィールドの「name」属性を設定します。	○
5	事前申請書のシステム案件IDを保持するフィールドの「name」属性を設定します。	○
6	事前申請書のユーザデータIDを保持するフィールドの「name」属性を設定します。	○
7	事前申請書のガジェットクラスを保持するフィールドの「name」属性を設定します。	○
8	事前申請書のガジェットインスタンスを保持するフィールドの「name」属性を設定します。	○

精算金額

- 概要

引数に指定した精算予定額が引数に指定した事前申請書の仮払残高を超えるか否かを判定します。

チェックOK : 引数に指定した精算予定額が引数に指定した事前申請書の仮払残高を超えない。

チェックNG : 引数に指定した精算予定額が引数に指定した事前申請書の仮払残高を超える。

- 詳細

バリデーションクラス	AdvanceBalanceValidator
ID (JavaScript)	(JavaScriptでのバリデーションなし)
ID (Java)	advanceBalance
出力メッセージ	精算予定額は、仮払残高以下で入力してください。

- 引数

本バリデーションの引数は次の通りです。

設定順	説明	必須
1	精算予定額を保持するフィールドの「name」属性を設定します。	○
2	事前申請書のシステム案件IDを保持するフィールドの「name」属性を設定します。	○
3	事前申請書のユーザデータIDを保持するフィールドの「name」属性を設定します。	○
4	事前申請書のガジェットクラスを保持するフィールドの「name」属性を設定します。	○
5	事前申請書のガジェットインスタンスを保持するフィールドの「name」属性を設定します。	○

- はじめに
 - 本タグライブラリを使用する際の設定について
 - 本項に未記載の属性について
 - 属性「formatted」について
 - ガジェット上での属性「name」について
- intra-mart Accel Kaiden! 基盤モジュール
 - サーフィス <k:surface>
 - テキストボックス <k:text>
 - テキストボックス(数値) <k:numericText>
 - テキストボックス(コード) <k:codeText>
 - テキストボックス(Hidden) <k:hidden>
 - テキストボックス(日付) <k:date>
 - テキストボックス(時間) <k:time>
 - テキストボックス(国際化対応) <k:text-locales>
 - テキストボックス(パスワード) <k:password>
 - テキストエリア <k:textarea>
 - テキストエリア(国際化対応) <k:textarea-locales>
 - ドロップダウンリスト <k:select>
 - ドロップダウンリスト(シンプル) <k:simpleDropdownMenu>
 - チェックボックス <k:checkbox>
 - アンカー <k:a>
 - ボタン <k:button>
 - ラジオボタン <k:radio>

はじめに

本タグライブラリを使用する際の設定について

本タグライブラリを使用するJSPのページで、次の様な taglib ディレクティブを指定する必要があります。

```
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
```

本項に未記載の属性について

本項で記載していない次の様な属性も各タグライブラリに設定可能です。

通常のエレメントに対する属性を設定してください。

- class
- value
- style
- maxlength
- readonly

属性「formatted」について

属性「formatted」は、データベースに登録されている項目値とは別の値を表示したい場合に使用します。

例①) 金額の項目値 (カンマ編集)

- 未フォーマットの項目値: 999999999
- フォーマット済の項目値: 999,999,999

入力時 (modeが"0") は項目値の編集・登録を行う必要があるため「未フォーマットの項目値」を使用し、参照時 (modeが"1") は項目値の編集・登録を行う必要がないため「フォーマット済の項目値」を使用したい場合、属性「formatted」に"view"を設定し、次の様な実装を行います。

- formatPlainValueメソッドで金額の項目値に「999999999」を設定
- formatFormattedValueメソッドで金額の項目値に「999,999,999」を設定

例②) 数値の項目値 (スケール編集)

- 未フォーマットの項目値: 999.0000000000
- フォーマット済の項目値: 999

数値 (18,10) の項目の場合、データベースによっては小数部の値が「0」で補完される場合があります。入力時 (modeが"0")、参照時 (modeが"1") とともに「0」を除去した値を使用したい場合、属性「formatted」に"both"を設定し、次の様な実装を行います。

- formatPlainValueメソッドで数値の項目値に「999.0000000000」を設定
- formatFormattedValueメソッドで数値の項目値に「999」を設定



コラム

実際の未フォーマット／フォーマット済の項目値は、次のConvertorクラスの処理により設定可能です。ガジェット (経費旅費) の場合は、Action (基底) クラスで実装していますので、サンプルソースを参照してください。

- ConvertForJsonSource#formatPlainValue

- 未フォーマットの項目値を返却
 - ConvertForJsonSource#formatFormattedValue
- フォーマット済の項目値を返却

ガジェット上での属性「name」について

ガジェット内のエレメントに指定した「name」属性に設定された値は、タグライブラリの展開時に「name」属性に設定しますが、ガジェットの初期化処理で「name」属性に設定された値はガジェットやブロックを考慮した値に再設定します。タグライブラリの展開時に「name」属性に設定された値は、「data-kaiden-org-name」属性に保持します。

intra-mart Accel Kaiden! 基盤モジュール

サーフェイス <k:surface>

- 概要
サーフェイス(ラベル)用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	
surfaceKey	ラベル表示用のサーフェスキーを設定します。 サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	○
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view" :modeが"1"の場合にフォーマットされた項目値を表示します。 "both" :modeにかかわらずフォーマットされた項目値を表示します。	
requiredMark	必須マークの表示有無を設定します。 "true": 必須マークを表示します。 "false": 必須マークを表示しません。 ※省略時は"false"が設定されたものとします。	

テキストボックス <k:text>

- 概要
テキストボックス用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <input type="text"> "1": <label>	○
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view" :modeが"1"の場合にフォーマットされた項目値を表示します。 "both" :modeにかかわらずフォーマットされた項目値を表示します。	
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on": フォーカス時にIMEを「active」に設定します。 "off": フォーカス時にIMEを「inactive」に設定します。 "disabled": フォーカス時にIMEを「disabled」に設定します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス(数値) <k:numericText>

- 概要
テキストボックス(数値)用のタグライブラリです。
本タグライブラリでは、IMEモードを「disabled」に設定します。(※IMEモードの対応ブラウザのみ)
また、数値入力用に文字配置を右寄せに設定します。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○

属性名	説明	必須
mode	表示モードを設定します。 “0”:<input type=“text”> “1”:<label>	○
formatted	フォーマットされた項目値の表示方法を設定します。 “input”:modeが“0”の場合にフォーマットされた項目値を表示します。 “view”:modeが“1”の場合にフォーマットされた項目値を表示します。 “both”:modeにかかわらずフォーマットされた項目値を表示します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス(コード) <k:codeText>

- 概要
テキストボックス(コード)用のタグライブラリです。
本タグライブラリでは、IMEモードを「disabled」に設定します。(※IMEモードの対応ブラウザのみ)
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”:<input type=“text”> “1”:<label>	○
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス(Hidden) <k:hidden>

- 概要
テキストボックス(Hidden)用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”:<input type=“hidden”> “1”:<label>	○
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス(日付) <k:date>

- 概要
テキストボックス(日付)用のタグライブラリです。
テキストボックスの横にカレンダーアイコンが表示され、カレンダーと連動できるテキストボックスです。
本タグライブラリでは、IMEモードを「disabled」に設定します。(※IMEモードの対応ブラウザのみ)
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”:<input type=“text”> “1”:<label>	○
formatted	フォーマットされた項目値の表示方法を設定します。 “input”:modeが“0”の場合にフォーマットされた項目値を表示します。 “view”:modeが“1”の場合にフォーマットされた項目値を表示します。 “both”:modeにかかわらずフォーマットされた項目値を表示します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス(時間) <k:time>

- 概要**
テキストボックス(時間)用のタグライブラリです。
1桁、または2桁の数値が入力された場合は、入力値をH、mmには00を補完して「H:mm」形式に変換します。
3桁から5桁の数値が入力された場合は、最後の2桁以外をHに変換、最後の2桁をmmに変換して「H:mm」形式に変換します。
また、↑キー、↓キーで、時間を最小入力単位(分)に設定された値分、増減できます。

例①) 入力値「9」→「9:00」
例②) 入力値「911」→「9:11」



注意

テキストボックス(時間)は、入力した値によって"0"および"."を自動補完するイベントをliveで発行しています。
テキストボックス(時間)のイベント処理を編集する場合は、必ず **liveイベント** を使用してください。

- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0":<input type="text"> "1":<label>	○
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	
maxHour	時間の最大値を設定します。 設定値は「data-kaiden-max-hour」属性に設定します。 ※省略時は「data-kaiden-max-hour」属性を設定しません。	
timeFormat	時刻表現を指定します。 "H24":24時までを表現します。 "H99":24時以降も表現します。(午前2時は「26時」と表現します。) 設定値は「data-kaiden-timeFormat」属性に設定します。 ※省略時は「BaseSettings.xml」から取得した時刻表現(timeFormat)を使用します。	
minutesUnit	時刻の最小入力単位(分)を設定します 設定値は「data-kaiden-minutes-unit」属性に設定します。 ※省略時は「data-kaiden-minutes-unit」属性を設定しません。	

テキストボックス(国際化対応) <k:text-locales>

- 概要**
テキストボックス(国際化対応)用のタグライブラリです。
本タグライブラリで、テキストボックスをintra-mart Accel Platformが使用するロケール数分表示します。
intra-mart Accel Platformが使用するロケールが、日本語・英語・中国語3つの場合、次の様なテキストボックスを表示します。

日本語	
英語	
中国語(中華人民共和国)	

システムロケールが一つの場合は、[テキストボックス <k:text>](#)と同じように表示します。
ログインユーザのロケールに対応したテキストボックスを先頭に表示し、その他のロケールはIDの昇順で表示します。

- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0":<input type="text"> "1":<label>	○
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view":modeが"1"の場合にフォーマットされた項目値を表示します。 "both":modeにかかわらずフォーマットされた項目値を表示します。	
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on":フォーカス時にIMEを「active」に設定します。 "off":フォーカス時にIMEを「inactive」に設定します。 "disabled":フォーカス時にIMEを「disabled」に設定します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストボックス (パスワード) <k:password>

- 概要
テキストボックス (パスワード) 用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <input type="password"> "1": <label>	○
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストエリア <k:textarea>

- 概要
テキストエリア用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <textarea> "1": <label>	○
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on": フォーカス時にIMEを「active」に設定します。 "off": フォーカス時にIMEを「inactive」に設定します。 "disabled": フォーカス時にIMEを「disabled」に設定します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

テキストエリア (国際化対応) <k:textarea-locale>

- 概要
テキストエリア (国際化対応) 用のタグライブラリです。
本タグライブラリで、テキストエリアをintra-mart Accel Platformが使用するロケール数分表示します。
intra-mart Accel Platformが使用するロケールが、日本語・英語・中国語3つの場合、次の様なテキストエリアを表示します。

日本語	
英語	
中国語 (中華人民共和国)	

システムロケールが一つの場合は、[テキストエリア <k:textarea>](#)と同じように表示します。
ログインユーザのロケールに対応したテキストエリアを先頭に表示し、その他のロケールはIDの昇順で表示します。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <textarea> "1": <label>	○
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on": フォーカス時にIMEを「active」に設定します。 "off": フォーカス時にIMEを「inactive」に設定します。 "disabled": フォーカス時にIMEを「disabled」に設定します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

ドロップダウンリスト <k:select>

- **概要**
ドロップダウンリスト用のタグライブラリです。
- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”: <select> “1”: <label>	○
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。)サーフェスキーを設定することで、プロパティファイルから文字列を取得します。サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	
viewModeName	「mode」属性値が“1”の場合の「name」属性を設定します。	
dataName	「option」タグを生成するためのデータ名を設定します。リストデータタグなどで生成したデータを指定する場合は、リストデータの「name」属性の値を設定します。	
includeDisabled	無効(論理削除)データの「option」タグを生成有無を設定します。 “true”: データ中の「deleteFlag」が「1」のデータも「option」タグを生成します。 “false”: データ中の「deleteFlag」が「1」のデータは「option」タグを生成しません。	
selectChain	値の変更イベントで、連動する項目を設定します。 ※設定方法: selectChain={%コード%:'%キー%', %コード%:'%キー%', ...} 値の変更イベントで、設定したコードを「name」属性に持つ項目の「value」属性に、データ中の設定したキーに紐づく値を設定します。	

ドロップダウンリスト(シンプル) <k:simpleDropDownMenu>

- **概要**
ドロップダウンリスト(シンプル)用のタグライブラリです。
、の組み合わせにより、一階層のみのシンプルなドロップダウンメニューを表示します。
メニュークリックなどのイベント処理はありませんので、\$(“#id”).on(“click”, function(){...});などでイベント処理を実装してください。
- **使用例**

```
<k:simpleDropDownMenu>
<li><k:a href="javascript:void(0)" mode="0" text="メニュー"/>
  <ul>
    <li><k:a href="javascript:void(0)" mode="0" text="メニュー1" /></li>
    <li><k:a href="javascript:void(0)" mode="0" text="メニュー2" /></li>
    <li><k:a href="javascript:void(0)" mode="0" text="メニュー3" /></li>
    <li><k:a href="javascript:void(0)" mode="0" text="メニュー4" /></li>
  </ul>
</li>
</k:simpleDropDownMenu>
```

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
dropWidth	横幅を指定します。	

チェックボックス <k:checkbox>

- **概要**
チェックボックス用のタグライブラリです。
- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”: <input type="text"> “1”: <label>	○
label	ラベル用のサーフェスキーを設定します。 本属性に値を設定した場合は、次の様に出力します。 <label><input type="checkbox" />本属性値</label>	

属性名	説明	必須
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。)サーフェスキーを設定することで、プロパティファイルから文字列を取得します。サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

アンカー <k:a>

- 概要
アンカー用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
mode	表示モードを設定します。 "0": <a> "1": 非表示 (何も出力しません)	○
text	リンク名用のサーフェスキーを設定します。サーフェスキーを設定することで、プロパティファイルから文字列を取得します。サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

ボタン <k:button>

- 概要
ボタン用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <input type="button"> "1": 非表示 (何も出力しません)	○
value	ボタン名用のサーフェスキーを設定します。サーフェスキーを設定することで、プロパティファイルから文字列を取得します。サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

ラジオボタン <k:radio>

- 概要
ラジオボタン用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 "0": <input type="radio"> "1": <label>	○
label	ラベル用のサーフェスキーを設定します。本属性に値を設定した場合は、次の様に出力します。 <label><input type="radio" />属性値</label>	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。)サーフェスキーを設定することで、プロパティファイルから文字列を取得します。サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	

リストデータ

- はじめに
 - 本タグライブラリを使用する際の設定について
 - リストデータについて
- intra-mart Accel Kaiden! 基盤モジュール
 - 会社マスタ(認可参照) <klist_base:listDataCompany>
 - 通貨マスタ <klist_base:listDataCurrency>
 - ロールマスタ <klist_base:listDataRole>
 - システムロケール <klist_base:listDataSystemLocale>
 - 項目マスタ <klist_base:listDataItem>
 - 項目区分マスタ <klist_base:listDataItemType>
 - 会社口座マスタ <klist_base:listDataCompanyBank>
 - 月 <klist_base:listDataMonth>
 - 月(略) <klist_base:listDataMonthShort>
 - 日 <klist_base:listDataDayOfMonth>
 - 曜日 <klist_base:listDataWeekday>
 - 曜日(略) <klist_base:listDataWeekdayShort>
- intra-mart Accel Kaiden! ワークフローモジュール
 - 税区分マスタ <klist_workflow:listDataTaxType>
 - 支払方法マスタ <klist_workflow:listDataPayMethod>
 - 支払分類マスタ <klist_workflow:listDataPayType>
 - 役職(日当関連)マスタ <klist_workflow:listDataAllowancePost>
 - 事前申請書 <klist_workflow:listDataAdvApp>
 - 精算申請書 <klist_workflow:listDataStlApp>
- intra-mart Accel Kaiden! 勤務管理モジュール
 - 勤務体系カテゴリマスタ <klist_labor:listDataWsCategory>
 - 勤務状況カテゴリマスタ <klist_labor:listDataWorkStatusCat>
 - 勤務状況マスタ <klist_labor:listDataWorkStatus>
 - 勤務日付区分マスタ <klist_labor:listDataLbrDayType>
 - 休憩・控除・時間年休マスタ <klist_labor:listDataDeduction>
 - 休憩・控除・時間年休中分類マスタ <klist_labor:listDataDeduMcategory>
 - 代理処理種別 <klist_labor:listDataLbrAgentAuthAct>
 - 勤務カレンダールール指定パターン <klist_labor:listDataSettingPattern>
 - 代理元ユーザ <klist_labor:listDataSubstituteAgency>
 - 締め期間 <klist_labor:listDataLaborPeriodList>
 - 権限セット <klist_labor:listDataAuthoritySet>
 - 管理組織 <klist_labor:listDataDepartmentAdmin>
 - 下位組織 <klist_labor:listDataDepartmentTree>
 - 振替出勤日(法定休日) <klist_labor:listDataUnProcessSubWorkInLeagalHoliday>
 - 振替出勤日(所定休日) <klist_labor:listDataUnProcessSubWorkOutLeagalHoliday>
 - 振替休日(法定休日) <klist_labor:listDataUnProcessSubHolidayInLeagalHoliday>
 - 振替休日(所定休日) <klist_labor:listDataUnProcessSubHolidayOutLeagalHoliday>
 - 日(28日未満) <klist_labor:listDataDayOfMonthU28>

はじめに

本タグライブラリを使用する際の設定について

本タグライブラリを使用するJSPのページで、次の様な taglib ディレクティブを指定する必要があります。

- intra-mart Accel Kaiden! 基盤モジュール
<%@ taglib prefix="klist_base" uri="http://kaiden.slcs.co.jp/taglib/listdata/base" %>
- intra-mart Accel Kaiden! ワークフローモジュール
<%@ taglib prefix="klist_workflow" uri="http://kaiden.slcs.co.jp/taglib/listdata/workflow" %>
- intra-mart Accel Kaiden! 勤務管理モジュール
<%@ taglib prefix="klist_labor" uri="http://kaiden.slcs.co.jp/taglib/listdata/labormgr"%>

リストデータについて

本項に記載されているリストデータは**ドロップダウンリスト** <k:select>などで「option」タグを生成するためのデータとして使用します。
特に記載のない場合、リストデータには次の値が含まれます。

- 項目キー:key
- 項目値:value
- 削除フラグ:deleteFlag

intra-mart Accel Kaiden! 基盤モジュール

会社マスタ(認可参照) <klist_base:listDataCompany>

- 概要
会社マスタ(IM-共通マスタ)のリストデータ作成用タグライブラリです。

出力対象は「authCriteriaUserCd」に設定されたユーザが「type」に設定された権限を持ち、

「searchCriteriaDate」に設定された日付時点で有効な会社となります。
出力順は会社のソートキー順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
type	リストデータの内容を設定します。 "readParmitCompany": 参照権のある会社のリストデータ "writeParmitCompany": 更新権のある会社のリストデータ	○
authCriteriaUserCd	認可判定時のユーザコードを設定します。 ※省略時はログインユーザのユーザコードが設定されたものとします。	
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

通貨マスタ <klist_base:listDataCurrency>

- 概要
通貨マスタ（IM-共通マスタ）のリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の通貨となります。
出力順は通貨のソートキー順（昇順）となります。

本リストデータは、通常の出力項目（key, value, deleteFlag）に加え、「整数部桁数:intLen」「小数部桁数:decLen」を出力します。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○

ロールマスタ <klist_base:listDataRole>

- 概要
ロールマスタのリストデータ作成用タグライブラリです。

出力対象はカテゴリが「KAIDEN」に設定されたロールとなります。
出力順はロールのID順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

システムロケール <klist_base:listDataSystemLocale>

- 概要
システムロケールのリストデータ作成用タグライブラリです。

出力対象はintra-mart Accel Platformが使用するロケールとなります。
出力順はロケールのID順（昇順）となります。
但し、ログインユーザのロケールは先頭になります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

項目マスタ <klist_base:listDataItem>

- 概要**
項目マスタのリストデータ作成用タグライブラリです。

出力対象は「itemTypeCd」に設定された項目区分に属し、「companyCd」に設定された会社の項目となります。
出力順は項目のソートキー順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。 ※省略時は項目マスタのビルトインデータのみ取得します。	
itemTypeCd	項目区分コードを設定します。	○

項目区分マスタ <klist_base:listDataItemType>

- 概要**
項目区分マスタのリストデータ作成用タグライブラリです。

出力対象は項目区分すべてとなります。
出力順は項目区分のコード順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

会社口座マスタ <klist_base:listDataCompanyBank>

- 概要**
会社口座マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の、「searchCriteriaDate」に設定された日付時点で有効な口座となります。
出力順は口座のソートキー順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

月 <klist_base:listDataMonth>

- 概要**
月のリストデータ作成用タグライブラリです。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

月(略) <klist_base:listDataMonthShort>

- 概要
月(略)のリストデータ作成用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

日 <klist_base:listDataDayOfMonth>

- 概要
日のリストデータ作成用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

曜日 <klist_base:listDataWeekday>

- 概要
曜日のリストデータ作成用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

曜日(略) <klist_base:listDataWeekdayShort>

- 概要
曜日(略)のリストデータ作成用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○

属性名	説明	必須
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

intra-mart Accel Kaiden! ワークフローモジュール

税区分マスタ <klist_workflow:listDataTaxType>

- 概要
税区分マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の、「searchCriteriaDate」に設定された日付時点で有効な税区分となります。
出力順は税区分のコード順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

支払方法マスタ <klist_workflow:listDataPayMethod>

- 概要
支払方法マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の、「searchCriteriaDate」に設定された日付時点で有効な支払方法となります。
出力順は支払方法のソートキー順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

支払分類マスタ <klist_workflow:listDataPayType>

- 概要
支払分類マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の、「searchCriteriaDate」に設定された日付時点で有効な支払分類となります。
出力順は支払分類のソートキー順（昇順）となります。

本リストデータは、通常の出力項目（key, value, deleteFlag）に加え、「立替フラグ: paymentFlag」を出力します。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

役職（日当関連）マスタ <klist_workflow:listDataAllowancePost>

- **概要**
役職（日当関連）マスタのリストデータ作成用タグライブラリです。

出力対象はProductWorkflowSettings.xmlに設定された組織セットの、「searchCriteriaDate」に設定された日付時点で有効な役職となります。
出力順は役職のソートキー順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

事前申請書 <klist_workflow:listDataAdvApp>

- **概要**
事前申請書のリストデータ作成用タグライブラリです。

出力対象は事前申請書のコンテンツ定義すべてとなります。
出力順はコンテンツのID順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

精算申請書 <klist_workflow:listDataStlApp>

- **概要**
精算申請書のリストデータ作成用タグライブラリです。

出力対象は精算申請書のコンテンツ定義すべてとなります。
出力順はコンテンツのID順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

intra-mart Accel Kaiden! 勤務管理モジュール

勤務体系カテゴリマスタ <klist_labor:listDataWsCategory>

- **概要**
勤務体系カテゴリマスタのリストデータ作成用タグライブラリです。

出力対象は勤務体系カテゴリすべてとなります。
出力順は勤務体系カテゴリのソートキー順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

勤務状況カテゴリマスタ <klist_labor:listDataWorkStatusCat>

- 概要**
勤務状況カテゴリマスタのリストデータ作成用タグライブラリです。

出力対象は勤務状況カテゴリすべてとなります。
出力順は勤務状況カテゴリのソートキー順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

勤務状況マスタ <klist_labor:listDataWorkStatus>

- 概要**
勤務状況マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の、「searchCriteriaDate」に設定された日付時点で有効な勤務状況となります。
「wsAgreementCd」を設定した場合は、上記に加え「wsAgreementCd」を設定した勤務体系に紐づく勤務状況となります。
出力順は勤務状況のソートキー順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
wsAgreementCd	勤務体系コードを設定します。	

勤務日付区分マスタ <klist_labor:listDataLbrDayType>

- 概要**
勤務日付区分マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の勤務日付区分となります。
「lbrDayTypeCatCd」を設定した場合は、上記に加え「lbrDayTypeCatCd」を設定した日付区分カテゴリに紐づく勤務日付区分となります。
出力順は勤務日付区分のソートキー順（昇順）となります。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
lbrDayTypeCatCd	日付区分カテゴリを設定します。	

休憩・控除・時間年休マスタ <klist_labor:listDataDeduction>

- **概要**
休憩・控除・時間年休マスタのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の休憩・控除・時間年休となります。
「deductionMcategoryCds」を設定した場合は、上記に加え「deductionMcategoryCds」を設定した休憩・控除・時間年休中分類に紐づく休憩・控除・時間年休となります。
出力順は休憩・控除・時間年休のソートキー順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
deductionMcategoryCds	休憩・控除・時間年休中分類コードを設定します。	

休憩・控除・時間年休中分類マスタ <klis_t_labor:listDataDeduMcategory>

- **概要**
休憩・控除・時間年休中分類マスタのリストデータ作成用タグライブラリです。

出力対象は休憩・控除・時間年休の中分類すべてとなります。
出力順は休憩・控除・時間年休中分類のソートキー順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

代理処理種別 <klis_t_labor:listDataLbrAgentAuthAct>

- **概要**
代理処理種別のリストデータ作成用タグライブラリです。

出力対象は代理処理種別すべてとなります。
出力順は「日次申請」「日次承認」「日次点検」「勤務表提出」「勤務表点検」の順となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

勤務カレンダールール指定パターン <klis_t_labor:listDataSettingPattern>

- **概要**
勤務カレンダールール指定パターンのリストデータ作成用タグライブラリです。

出力対象は勤務カレンダールール指定パターンすべてとなります。
出力順は「週次曜日指定」「月次曜日指定」「年次日付指定」「年次曜日指定」「個別日付指定」の順となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

属性名	説明	必須

代理元ユーザ <klist_labor:listDataSubstituteAgency>

- 概要
代理元ユーザのリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の「lbrProcessType」に設定された代理処理対象権限を持つユーザとなります。
「substitutePlaceUserCd」を設定した場合は、上記に加え「substitutePlaceUserCd」を設定したユーザの代理権限をもつユーザを対象にします。
出力順はユーザのコード順（昇順）となります。

i コラム

代理対象期間の検索条件について

- 代理対象期間検索基準日1、代理対象期間検索基準日2がともに未指定の場合、代理対象期間を検索条件に含めません。
- 代理対象期間検索基準日1、代理対象期間検索基準日2の両方が設定された場合、基準日1から基準日2の期間と重複する代理対象期間を持つ代理元ユーザが対象となります。
- 代理対象期間検索基準日1のみを設定した場合は、「代理対象期間開始日 <= 基準日1 < 代理対象期間終了日」に一致する代理対象期間を持つ代理元ユーザが対象となります。
- 代理対象期間検索基準日2のみ指定された場合は、代理対象期間を検索条件に含めません。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含めない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
lbrProcessType	代理処理対象を設定します。	○
substitutePlaceUserCd	代理先ユーザコードを設定します。 ※省略時はログインユーザのユーザコードが設定されたものとします。	
availableCriteriaDate	代理有効期間検索基準日を設定します。 ※省略時はシステム日付が設定されたものとします。	
targetCriteriaDate1	代理対象期間検索基準日1を設定します。	
targetCriteriaDate2	代理対象期間検索基準日2を設定します。	

締め期間 <klist_labor:listDataLaborPeriodList>

- 概要
締め期間(yyyy/MM/dd - yyyy/MM/dd)のリストデータ作成用タグライブラリです。

出力対象は「companyCd」に設定された会社の「searchCriteriaDate」に設定された日付時点で有効な締め期間となります。
「wsAgreementCd」「openedFlag」「suspendFlag」「companyClosedFlag」「archivedFlag」「fromDate」「toDate」を設定した場合は、上記に加え各条件に一致した締め期間を対象にします。
出力順は締め期間開始日順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含めない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
wsAgreementCd	勤務体系コードを設定します。	
openedFlag	オープンフラグを設定します。	
suspendFlag	一時中断フラグを設定します。	
companyClosedFlag	会社締め済フラグを設定します。	

属性名	説明	必須
archivedFlag	アーカイブフラグを設定します。	
fromDate	検索期間開始日を設定します。	
toDate	検索期間終了日を設定します。	

権限セット <klist_labor:listDataAuthoritySet>

- 概要**
権限セットのリストデータ作成用タグライブラリです。

出力対象は権限セットすべてとなります。
出力順は権限セットのソートキー順（昇順）となります。

- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

管理組織 <klist_labor:listDataDepartmentAdmin>

- 概要**
管理組織のリストデータ作成用タグライブラリです。

出力対象は「searchCriteriaDate」に設定された日付時点で有効な、ログインユーザが管理者に設定されている組織となります。
出力順は会社、および組織のソートキー順（昇順）となります。

- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

下位組織 <klist_labor:listDataDepartmentTree>

- 概要**
下位組織のリストデータ作成用タグライブラリです。

出力対象は「companyCd」「departmentSetCd」「departmentCd」に設定された組織とその下位組織となります。
「searchCriteriaDate」「includeDisabled」を設定した場合は、上記に加え各条件に一致した組織とその下位組織を対象にします。
出力順は会社、および組織のソートキー順（昇順）となります。

- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
departmentSetCd	組織セットコードを設定します。	○
departmentCd	組織コードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
includeDisabled	無効（論理削除）データを対象とするかを設定します。 "true": 無効（論理削除）データを対象とします。 "false": 無効（論理削除）データを対象としません。 ※省略時は"false"が設定されたものとします。	

振替出勤日（法定休日）<klist_labor:listDataUnProcessSubWorkInLeagalHoliday>

- 概要
振替出勤日（法定休日）のリストデータ作成用タグライブラリです。

出力対象は「userCd」に設定されたユーザの「companyCd」に設定された会社で未取得の振替出勤日（法定休日分）となります。
「searchCriteriaDate」を設定した場合は、上記に加え「searchCriteriaDate」に設定された日付時点で未取得の振替出勤日（法定休日分）が対象となります。
出力順は日付順（昇順）となります。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
userCd	ユーザコードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

振替出勤日（所定休日）<klist_labor:listDataUnProcessSubWorkOutLeagalHoliday>

- 概要
振替出勤日（所定休日）のリストデータ作成用タグライブラリです。

出力対象は「userCd」に設定されたユーザの「companyCd」に設定された会社で未取得の振替出勤日（所定休日分）となります。
「searchCriteriaDate」を設定した場合は、上記に加え「searchCriteriaDate」に設定された日付時点で未取得の振替出勤日（所定休日分）が対象となります。
出力順は日付順（昇順）となります。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
userCd	ユーザコードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

振替休日（法定休日）<klist_labor:listDataUnProcessSubHolidayInLeagalHoliday>

- 概要
振替休日（法定休日）のリストデータ作成用タグライブラリです。

出力対象は「userCd」に設定されたユーザの「companyCd」に設定された会社で未取得の振替休日（法定休日分）となります。
「searchCriteriaDate」を設定した場合は、上記に加え「searchCriteriaDate」に設定された日付時点で未取得の振替休日（法定休日分）が対象となります。
出力順は日付順（昇順）となります。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 "true": リストデータの先頭に空白データを含む。 "false": リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
userCd	ユーザコードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

振替休日と紐づけられていない法定休日振替休日を一覧します。
振替休日順に出力します。

振替休日（所定休日）<klist_labor:listDataUnProcessSubHolidayOutLeagalHoliday>

- **概要**
振替休日（所定休日）のリストデータ作成用タグライブラリです。

出力対象は「userCd」に設定されたユーザの「companyCd」に設定された会社で未取得の振替休日（所定休日分）となります。
「searchCriteriaDate」を設定した場合は、上記に加え「searchCriteriaDate」に設定された日付時点で未取得の振替休日（所定休日分）が対象となります。
出力順は日付順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○
companyCd	会社コードを設定します。	○
userCd	ユーザコードを設定します。	○
searchCriteriaDate	検索時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	

日(28日未満) <klist_labor:listDataDayOfMonthU28>

- **概要**
日(28日未満)のリストデータ作成用タグライブラリです。

出力対象は1日から27日と末日となります。
出力順は日付順（昇順）となります。

- **属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
blank	リストデータに空データ（空白行）を含むか否かを設定します。 “true”: リストデータの先頭に空白データを含む。 “false”: リストデータの先頭に空白データを含まない。	○
name	リストデータの名前を設定します。 HTMLページ全体で一意的な名前として扱われます。	○

- はじめに
 - 本タグライブラリを使用する際の設定について
 - 本項に未記載の属性について
 - マスタデータ取得について
 - マスタ検索の後処理について
 - 属性「formatted」について
 - 属性「resultValue」について
 - 属性「resultChain」について
 - 属性「primaryTarget」について
 - 属性「searchCriteriaXXXX」について
- intra-mart Accel Kaiden! 基盤モジュール
 - ユーザ <kmsb:user>
 - 組織 <kmsb:department>
 - 金融機関 <kmsb:bankFacilJp>
- intra-mart Accel Kaiden! ワークフローモジュール
 - プロジェクト <kmsw:project>
 - 経費区分 <kmsw:expType>
 - 勘定科目 <kmsw:account>
 - 仕入先 <kmsw:vendor>
 - 日当 <kmsw:allowance>
- intra-mart Accel Kaiden! 勤務管理モジュール
 - 組織 <kmsl:department>
 - 勤務体系 <kmsl:wsAgreement>

はじめに

本タグライブラリを使用する際の設定について

本タグライブラリを使用するJSPのページで、次の様な taglib ディレクティブを指定する必要があります。

- intra-mart Accel Kaiden! 基盤モジュール
`<%@ taglib prefix="kmsb" uri="http://kaiden.slcs.co.jp/taglib/master_search/base" %>`
- intra-mart Accel Kaiden! ワークフローモジュール
`<%@ taglib prefix="kmsw" uri="http://kaiden.slcs.co.jp/taglib/master_search/workflow" %>`
- intra-mart Accel Kaiden! 勤務管理モジュール
`<%@ taglib prefix="kmsl" uri="http://kaiden.slcs.co.jp/taglib/master_search/labormgr" %>`

本項に未記載の属性について

本項で記載していない次の様な属性も各タグライブラリに設定可能です。
 通常のエレメントに対する属性を設定してください。

- class
- value
- style
- maxlength
- readonly

マスタデータ取得について

画面表示時に「primaryTarget」設定したエレメントに値が設定されている場合に、
 名称表示のためにマスタデータを再度取得する場合があります。
 マスタデータ取得時には、属性の説明に「マスタデータ取得時に～」と記載している属性を検索条件として使用します。

- 「検索・マスタデータ取得時に～」: 検索時、マスタデータ取得時に検索条件として使用します。
- 「マスタデータ取得時に～」: 検索時には検索条件として使用せず、マスタデータ取得時にのみ検索条件として使用します。

マスタ検索の後処理について

マスタ検索の各処理（選択、クリア、マスタデータ取得）後に、後処理を追加することが可能です。
 後処理の詳細は『intra-mart Accel Kaiden! JsDoc』を参照してください。

- 選択処理の後処理
`KAIDEN.base.foundation.MasterSearch#addEventPostSelectedData ( String gadgetId, String containerId, String fieldKey, Function fnc )`
- クリア処理の後処理
`KAIDEN.base.foundation.MasterSearch#addEventPostClearData ( String gadgetId, String containerId, String fieldKey, Function fnc )`
- マスタデータ取得処理の後処理
`KAIDEN.base.foundation.MasterSearch#addEventPostRefreshData ( String gadgetId, String containerId, String fieldKey, Function fnc )`

属性「formatted」について

属性「formatted」は、データベースに登録されている項目値とは別の値を表示したい場合に使用します。

例①) 金額の項目値 (カンマ編集)

- 未フォーマットの項目値:999999999
- フォーマット済の項目値:999,999,999

入力時(modeが"0")は項目値の編集・登録を行う必要があるため「未フォーマットの項目値」を使用し、参照時(modeが"1")は項目値の編集・登録を行う必要がないため「フォーマット済の項目値」を使用したい場合、属性「formatted」に"view"を設定し、次の様な実装を行います。

- formatPlainValueメソッドで金額の項目値に「999999999」を設定
- formatFormattedValueメソッドで金額の項目値に「999,999,999」を設定

例②)数値の項目値(スケール編集)

- 未フォーマットの項目値:999.0000000000
- フォーマット済の項目値:999

数値(18,10)の項目の場合、データベースによっては小数部の値が「0」で補完される場合があります。入力時(modeが"0")、参照時(modeが"1")ともに「0」を除去した値を使用したい場合、属性「formatted」に"both"を設定し、次の様な実装を行います。

- formatPlainValueメソッドで数値の項目値に「999.0000000000」を設定
- formatFormattedValueメソッドで数値の項目値に「999」を設定



コラム

実際の未フォーマット／フォーマット済の項目値は、次のConvertorクラスの処理により設定可能です。ガジェット(経費旅費)の場合は、Action(基底)クラスで実装していますので、サンプルソースを参照してください。

- ConvertForJsonSource#formatPlainValue
未フォーマットの項目値を返却
- ConvertForJsonSource#formatFormattedValue
フォーマット済の項目値を返却

属性「resultValue」について

対象選択時にタグライブラリで生成された要素の「value」に設定するデータのキーを設定します。設定可能なキーは各タグライブラリの検索結果を参照してください。

- 例
次の様に設定することで、対象選択時に経費区分名(expTypeName)がタグライブラリで生成された要素の「value」に設定します。

```
<kmsw:expType
  name="targetExpTypeName"
  resultValue="expTypeName"
/>
```

属性「resultChain」について

対象選択時に複数の要素の「value」に設定するデータのキーを設定します。設定可能なキーは各タグライブラリを検索結果を参照してください。

- 例
次の様に設定することで、対象選択時に経費区分名(expTypeName)がタグライブラリで生成された要素の「value」に設定します。

```
<k:hidden
  name="targetExpTypeCd"
/>
<kmsw:expType
  name="targetExpTypeName"
  resultValue="expTypeName"
  resultChain="{targetExpTypeCd:'expTypeCd', targetAccountCd:'accountCd', targetAccountName:'accountName', targetTaxTypeCd:'taxTypeCd', targetTaxTypeName:'taxTypeName'}"
  primaryTarget="targetExpTypeCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaExpTypeCd="targetExpTypeCd"
/>
<k:hidden
  name="targetAccountCd"
/>
<k:hidden
  name="targetAccountName"
/>
<k:hidden
  name="targetTaxTypeCd"
/>
<k:hidden
  name="targetTaxTypeName"
/>
```



コラム

上記のソースでは、対象選択時に次の様な値を設定します。

- 属性「resultChain」の設定により、「targetExpTypeCd」に「expTypeCd」（経費区分コード）を設定
- 属性「resultValue」の設定により、「targetExpTypeName」に「expTypeName」（経費区分名）を設定
- 属性「resultChain」の設定により、「targetAccountCd」に「accountCd」（勘定科目コード）を設定
- 属性「resultChain」の設定により、「targetAccountName」に「accountName」（勘定科目名）を設定
- 属性「resultChain」の設定により、「targetTaxTypeCd」に「taxTypeCd」（税区分コード）を設定
- 属性「resultChain」の設定により、「targetTaxTypeName」に「taxTypeName」（税区分名）を設定

属性「primaryTarget」について

検索状態を判定するエレメントの「name」を設定します。

設定したエレメントに値が設定されている場合に、検索済み（選択済み）と判定します。

未検索（未選択）の状態で、バリデーションが実行されると、「未確定です。」というエラーが表示されます。

- 例**
次の様に設定することで、「targetDepartmentCd」、「targetDepartmentSetCd」に値が設定されている場合に、検索済み（選択済み）と判定します。

```
<k:hidden
  name="targetDepartmentSetCd"
/>
<k:hidden
  name="targetDepartmentCd"
/>
<kmsb:department
  name="targetDepartmentName"
  resultValue="departmentName"
  resultChain="{targetDepartmentSetCd:'departmentSetCd', targetDepartmentCd:'departmentCd'}"
  primaryTarget="targetDepartmentCd,targetDepartmentSetCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaDepartmentCd="targetDepartmentCd"
  searchCriteriaDepartmentSetCd="targetDepartmentSetCd"
/>
```

属性「searchCriteriaXXXX」について

検索、またはマスタデータ取得時に使用する値を保持するエレメントの「name」を設定します。

本属性には値を直接設定することも可能です。

- 例**
次の様に設定することで、検索、またはマスタデータ取得時に「searchCriteriaDate」、「searchCriteriaCompany」、「targetDepartmentCd」、「targetDepartmentSetCd」の値を使用します。

```
<k:hidden
  name="targetDepartmentSetCd"
/>
<k:hidden
  name="targetDepartmentCd"
/>
<kmsb:department
  name="targetDepartmentName"
  resultValue="departmentName"
  resultChain="{targetDepartmentSetCd:'departmentSetCd', targetDepartmentCd:'departmentCd'}"
  primaryTarget="targetDepartmentCd,targetDepartmentSetCd"
  searchCriteriaDate="searchCriteriaDate"
  searchCriteriaCompany="searchCriteriaCompany"
  searchCriteriaDepartmentCd="targetDepartmentCd"
  searchCriteriaDepartmentSetCd="targetDepartmentSetCd"
/>
```

コラム

ガジェットでのsearchCriteriaDate、searchCriteriaCompanyについて

searchCriteriaDate、searchCriteriaCompanyは申請画面や処理画面自体で保持する項目となります。

上記の例の様に直接「searchCriteriaDate」、「searchCriteriaCompany」を指定した場合は、
「searchCriteriaDate」は検索基準日、「searchCriteriaCompany」は検索基準会社のコード値を使用します。
※検索基準日：申請一覧（IM-Workflow）で選択した申請基準日
※検索基準会社：会社選択画面で選択した会社（複数会社の所属していない場合は、所属会社）

次の様な形式でリクエストパラメータから取得することも可能です。

```
<c:set var="searchCriteriaCompany">${param.searchCriteriaCompany}</c:set>
<c:set var="searchCriteriaDate">${param.searchCriteriaDate}</c:set>
```


■ 概要

ユーザマスタ (IM-共通マスタ) 検索用タグライブラリです。

■ 属性

本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 “0” : <input type="text"> “1” : <label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) “on” : フォーカス時にIMEを「active」に設定します。 “off” : フォーカス時にIMEを「inactive」に設定します。 “disabled” : フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 “input” : modeが“0”の場合にフォーマットされた項目値を表示します。 “view” : modeが“1”の場合にフォーマットされた項目値を表示します。 “both” : modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 “true” : 無効(論理削除)データを含む。 “false” : 無効(論理削除)データを含まない。 ※省略時は“false”が設定されたものとします。
effectiveAuthCompanyCondition	検索者の認可設定に応じた検索対象の制限有無を設定します。 “true” : 検索対象を認可設定(参照権)のある組織に制限する。 “false” : 検索対象を制限しない。 ※省略時は“false”が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaUserCd	マスタデータ取得時に指定するユーザコードを保持する項目の「name」属性を設定します。

■ 検索結果

本タグライブラリの検索結果は、次のキーで取得可能です。

各キーの名称については、IM-共通マスタのドキュメントを参照してください。

- userCd
- userName
- userSearchName
- sex
- contryCd
- zipCode
- address1
- address2
- address3
- telephoneNumber
- extensionNumber
- faxNumber
- extensionFaxNumber
- mobileNumber
- emailAddress1
- emailAddress2
- mobileEmailAddress
- notes
- sortKey
- url
- startDate
- endDate
- termCd
- deleteFlag
- localeId

組織 <kmsb:department>

- 概要
組織マスタ(IM-共通マスタ)検索用タグライブラリです。

i コラム

組織マスタ検索の違いについて

組織マスタ検索には、「組織 <kmsb:department>」(intra-mart Accel Kaiden! 基盤モジュール)」と「組織 <kmsl:department>」(intra-mart Accel Kaiden! 勤務管理モジュール)があります。

各検索には次の違いがあります。

- 組織 <kmsb:department> (intra-mart Accel Kaiden! 基盤モジュール)
検索時や、またはマスタデータ取得時に該当エレメントの値 (value) がない場合の検索条件 (組織セットコード) は「BaseSettings.xml」から取得した組織セットコード (defaultDepartmentSetCd) を使用します。
- 組織 <kmsl:department> (intra-mart Accel Kaiden! 勤務管理モジュール)
検索時、またはマスタデータ取得時に該当エレメントの値 (value) がない場合の検索条件 (組織セットコード) は「ProductLabormgrSettings.xml」から取得した組織セ (decisionAttachDepartmentSetCd) を使用します。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 “0”: <input type="text"> “1”: <label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) “on”: フォーカス時にIMEを「active」に設定します。 “off”: フォーカス時にIMEを「inactive」に設定します。 “disabled”: フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 “input”: modeが“0”の場合にフォーマットされた項目値を表示します。 “view”: modeが“1”の場合にフォーマットされた項目値を表示します。 “both”: modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 “true”: 無効(論理削除)データを含む。 “false”: 無効(論理削除)データを含まない。 ※省略時は“false”が設定されたものとします。
effectiveAuthCompanyCondition	検索者の認可設定に応じた検索対象の制限有無を設定します。 “true”: 検索対象を認可設定(参照権)のある組織に制限する。 “false”: 検索対象を制限しない。 ※省略時は“false”が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaDepartmentSetCd	マスタデータ取得時に指定する組織セットコードを保持する項目の「name」属性を設定します。 ※検索時、またはマスタデータ取得時に該当エレメントの値 (value) がない場合は「BaseSettings.xml」から取得した組織セットコード (defaultDepartmentSetCd) を使用します。
searchCriteriaDepartmentCd	マスタデータ取得時に指定する組織コードを保持する項目の「name」属性を設定します。

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。
各キーの名称については、IM-共通マスタのドキュメントを参照してください。

- companyCd
- companyName
- departmentSetCd
- departmentSetName
- departmentCd
- departmentName
- departmentShortName
- departmentSearchName
- countryCd

- zipCode
- address1
- address2
- address3
- telephoneNumber
- extensionNumber
- faxNumber
- extensionFaxNumber
- emailAddress1
- emailAddress2
- url
- notes
- sortKey
- startDate
- endDate
- termCd
- deleteFlag
- localeId

金融機関 <kmsb:bankFacilJp>

- 概要**
金融機関マスタ検索用タグライブラリです。
- 属性**
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 "0":<input type="text"> "1":<label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on":フォーカス時にIMEを「active」に設定します。 "off":フォーカス時にIMEを「inactive」に設定します。 "disabled":フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view":modeが"1"の場合にフォーマットされた項目値を表示します。 "both":modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
searchCriteriaBankFacilCd	マスタデータ取得時に指定する金融機関コードを保持する項目の「name」属性を設定します。
searchCriteriaBranchNo	マスタデータ取得時に指定する支店番号を保持する項目の「name」属性を設定します。

- 検索結果**
本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- bankFacilCd (金融機関コード)
- bankFacilNameKj (金融機関名)
- bankFacilNameKn (金融機関名(カナ))
- branchNo番号 (支店番号)
- branchNameKj (支店名)
- branchNameKn (支店名(カナ))
- zipCd (郵便番号)
- branchAddressKj (支店所在地)
- branchAddressKn (支店所在地(カナ))
- telephoneNumber (電話番号)
- clearingHouseNo (手形交換所番号)
- lineCd (金融機関並びコード)
- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)
- extensionD (予備項目D)

- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

intra-mart Accel Kaiden! ワークフローモジュール

プロジェクト <kmsw:project>

- 概要
プロジェクトマスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”: <input type=“text”> “1”: <label>	○
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) “on”: フォーカス時にIMEを「active」に設定します。 “off”: フォーカス時にIMEを「inactive」に設定します。 “disabled”: フォーカス時にIMEを「disabled」に設定します。	
formatted	フォーマットされた項目値の表示方法を設定します。 “input”: modeが“0”の場合にフォーマットされた項目値を表示します。 “view”: modeが“1”の場合にフォーマットされた項目値を表示します。 “both”: modeにかかわらずフォーマットされた項目値を表示します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。	○
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。	○
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."	○
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 “true”: 無効(論理削除)データを含む。 “false”: 無効(論理削除)データを含まない。 ※省略時は“false”が設定されたものとします。	
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。	○
searchCriteriaProjectCd	マスタデータ取得時に指定するプロジェクトコードを保持する項目の「name」属性を設定します。	○

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- companyCd (会社コード)
- companyName (会社名)
- projectCd (プロジェクトコード)
- projectName (プロジェクト名)
- parentCompanyCd (会社コード(上位))
- parentProjectCd (プロジェクトコード(上位))
- parentProjectName (プロジェクト名(上位))
- localeId (ロケールID)
- startDate (開始日)
- endDate (終了日)
- termCd (期間コード)
- deleteFlag (削除フラグ)
- extensionA (予備項目A)
- extensionB (予備項目B)

- extensionC (予備項目C)
- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

経費区分 <kmsw:expType>

- 概要
経費区分マスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 "0": <input type="text"> "1": <label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on": フォーカス時にIMEを「active」に設定します。 "off": フォーカス時にIMEを「inactive」に設定します。 "disabled": フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 "input": modeが"0"の場合にフォーマットされた項目値を表示します。 "view" : modeが"1"の場合にフォーマットされた項目値を表示します。 "both" : modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 "true": 無効(論理削除)データを含む。 "false": 無効(論理削除)データを含まない。 ※省略時は"false"が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaExpTypeCd	マスタデータ取得時に指定する経費区分コードを保持する項目の「name」属性を設定します。

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- companyCd (会社コード)
- companyName (会社名)
- expTypeCd (経費区分コード)
- expTypeName (経費区分名)
- expTypeCategory (経費区分カテゴリ)
- accountCd (勘定科目コード)
- accountName (勘定科目名)
- taxTypeCd (税区分コード)
- taxTypeName (税区分名)
- sortKey (ソートキー)
- localeId (ロケールID)
- startDate (開始日)
- endDate (終了日)
- termCd (期間コード)

- deleteFlag (削除フラグ)
- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)
- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

勘定科目 <kmsw:account>

- 概要
勘定科目マスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
name	「name」属性を設定します。	○
mode	表示モードを設定します。 “0”: <input type="text"> “1”: <label>	○
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) “on”: フォーカス時にIMEを「active」に設定します。 “off”: フォーカス時にIMEを「inactive」に設定します。 “disabled”: フォーカス時にIMEを「disabled」に設定します。	
formatted	フォーマットされた項目値の表示方法を設定します。 “input”: modeが“0”の場合にフォーマットされた項目値を表示します。 “view”: modeが“1”の場合にフォーマットされた項目値を表示します。 “both”: modeにかかわらずフォーマットされた項目値を表示します。	
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。	○
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。	○
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."	○
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 “true”: 無効(論理削除)データを含む。 “false”: 無効(論理削除)データを含まない。 ※省略時は“false”が設定されたものとします。	
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。	
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。	○
searchCriteriaAccountCd	マスタデータ取得時に指定する勘定科目コードを保持する項目の「name」属性を設定します。	○

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- companyCd (会社コード)
- companyName (会社名)
- accountCd (勘定科目コード)
- accountName (勘定科目名)
- localeId (ロケールID)
- startDate (開始日)
- endDate (終了日)
- termCd (期間コード)
- deleteFlag (削除フラグ)
- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)

- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

仕入先 <kmsw:vendor>

- 概要
仕入先マスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 "0":<input type="text"> "1":<label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on":フォーカス時にIMEを「active」に設定します。 "off":フォーカス時にIMEを「inactive」に設定します。 "disabled":フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view" :modeが"1"の場合にフォーマットされた項目値を表示します。 "both" :modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 "true":無効(論理削除)データを含む。 "false":無効(論理削除)データを含まない。 ※省略時は"false"が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaCustomerCd	マスタデータ取得時に指定する仕入先コードを保持する項目の「name」属性を設定します。

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。

- companyCd (会社コード)
- companyName (会社名)
- customerCd (仕入先コード)
- customerName (仕入先名)
- vendorKbn (支払先分類)
- accountCdPayable (債務勘定科目コード)
- withholdingTaxFlag (源泉税利用有無)
- accountCdWithholdingTax (源泉税科目コード)
- taxTypele (外税/内税区分)
- taxRoundTypeCd (消費税端数処理区分)
- searchKeyword (検索キーワード)
- startDate (開始日)
- endDate (終了日)
- termCd (期間コード)
- deleteFlag (削除フラグ)
- localeId (ロケールID)

- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)
- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

日当 <kmsw:allowance>

- 概要
日当マスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 "0":<input type="text"> "1":<label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on":フォーカス時にIMEを「active」に設定します。 "off":フォーカス時にIMEを「inactive」に設定します。 "disabled":フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 "input":modeが"0"の場合にフォーマットされた項目値を表示します。 "view" :modeが"1"の場合にフォーマットされた項目値を表示します。 "both" :modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="%コード%:'%キー%', %コード%:'%キー%', ..." 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 "true":無効(論理削除)データを含む。 "false":無効(論理削除)データを含まない。 ※省略時は"false"が設定されたものとします。
currencyCd	検索・マスタデータ取得時に指定する通貨コードを保持する項目の「name」属性を設定します。
allowanceCriteriaUserCd	検索・マスタデータ取得時に指定する日当対象ユーザコードを保持する項目の「name」属性を設定します。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaAllowanceCd	マスタデータ取得時に指定する日当コードを保持する項目の「name」属性を設定します。

- 検索結果
本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- companyCd (会社コード)
- companyName (会社名)
- allowanceCd (日当コード)
- allowanceName (日当名)
- currencyCd (通貨コード)
- currencyName (通貨名)
- amount (日当額)
- amountFormatted (日当額 (カンマ区切り))
- localeId (ロケールID)
- startDate (開始日)

- endDate (終了日)
- termCd (期間コード)
- deleteFlag (削除フラグ)
- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)
- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

intra-mart Accel Kaiden! 勤務管理モジュール

組織 <kmsl:department>

- 概要
組織マスタ (IM-共通マスタ) 検索用タグライブラリです。

 コラム

組織マスタ検索の違いについて

組織マスタ検索には、「[組織 <kmsb:department>](#) (intra-mart Accel Kaiden! 基盤モジュール)」と「[組織 <kmsl:department>](#) (intra-mart Accel Kaiden! 勤務管理モジュール)」があります。

各検索には次の違いがあります。

- [組織 <kmsb:department>](#) (intra-mart Accel Kaiden! 基盤モジュール)
検索時や、またはマスタデータ取得時に該当エレメントの値 (value) がない場合の検索条件 (組織セットコード) は「BaseSettings.xml」から取得した組織セットコード (defaultDepartmentSetCd) を使用します。
- [組織 <kmsl:department>](#) (intra-mart Accel Kaiden! 勤務管理モジュール)
検索時、またはマスタデータ取得時に該当エレメントの値 (value) がない場合の検索条件 (組織セットコード) は「ProductLabormgrSettings.xml」から取得した組織セットコード (decisionAttachDepartmentSetCd) を使用します。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 “0” : <input type="text"> “1” : <label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) “on” : フォーカス時にIMEを「active」に設定します。 “off” : フォーカス時にIMEを「inactive」に設定します。 “disabled” : フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 “input” : modeが“0”の場合にフォーマットされた項目値を表示します。 “view” : modeが“1”の場合にフォーマットされた項目値を表示します。 “both” : modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:%キー%, %コード%:%キー%, ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、 検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効 (論理削除) データを含むか否かを設定します。 “true” : 無効 (論理削除) データを含む。 “false” : 無効 (論理削除) データを含まない。 ※省略時は“false”が設定されたものとします。

属性名	説明
effectiveAuthCompanyCondition	検索者の認可設定に応じた検索対象の制限有無を設定します。 "true": 検索対象を認可設定 (参照権) のある組織に制限する。 "false": 検索対象を制限しない。 ※省略時は"false"が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaDepartmentSetCd	マスタデータ取得時に指定する組織セットコードを保持する項目の「name」属性を設定します。 ※検索時、またはマスタデータ取得時に該当要素の値 (value) がない場合は「ProductLabormgrSettings.xml」から取得した組織セットコード (decisionAttachDepartmentSetCd) を使用します。
searchCriteriaDepartmentCd	マスタデータ取得時に指定する組織コードを保持する項目の「name」属性を設定します。

検索結果

本タグライブラリの検索結果は、次のキーで取得可能です。
各キーの名称については、IM-共通マスタのドキュメントを参照してください。

- companyCd
- companyName
- departmentSetCd
- departmentSetName
- departmentCd
- departmentName
- departmentShortName
- departmentSearchName
- countryCd
- zipCode
- address1
- address2
- address3
- telephoneNumber
- extensionNumber
- faxNumber
- extensionFaxNumber
- emailAddress1
- emailAddress2
- url
- notes
- sortKey
- startDate
- endDate
- termCd
- deleteFlag
- localeId

勤務体系 <kmsl:wsAgreement>

- 概要
勤務体系マスタ検索用タグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
name	「name」属性を設定します。
mode	表示モードを設定します。 "0": <input type="text"> "1": <label>
imeMode	IMEモードを設定します。(※IMEモードの対応ブラウザのみ本属性の値が有効になります。) "on": フォーカス時にIMEを「active」に設定します。 "off": フォーカス時にIMEを「inactive」に設定します。 "disabled": フォーカス時にIMEを「disabled」に設定します。
formatted	フォーマットされた項目値の表示方法を設定します。 "input": modeが"0"の場合にフォーマットされた項目値を表示します。 "view" : modeが"1"の場合にフォーマットされた項目値を表示します。 "both" : modeにかかわらずフォーマットされた項目値を表示します。
surfaceKey	項目名用のサーフェスキーを設定します。(エラーメッセージなどで使用します。) サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
resultValue	対象選択時に本項目の「value」属性に設定するキーを設定します。 対象選択時、本項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。

属性名	説明
resultChain	対象選択時に連動する項目を設定します。 ※設定方法:resultChain="{%コード%:'%キー%', %コード%:'%キー%', ...}" 対象選択時、設定したコードを「name」属性に持つ項目の「value」属性に、検索結果の設定したキーに紐づく値を設定します。
primaryTarget	未選択状態を判定する項目の「name」属性を設定します。 ※設定方法:primaryTarget="%コード%, %コード%, ..."
includeDisabled	検索対象に無効(論理削除)データを含むか否かを設定します。 "true":無効(論理削除)データを含む。 "false":無効(論理削除)データを含まない。 ※省略時は"false"が設定されたものとします。
searchCriteriaDate	検索・マスタデータ取得時に指定する基準日を保持する項目の「name」属性を設定します。 ※省略時はシステム日付が設定されたものとします。
searchCriteriaCompany	検索・マスタデータ取得時に指定する会社コードを保持する項目の「name」属性を設定します。
searchCriteriaWsAgreementCd	マスタデータ取得時に指定する勤務体系コードを保持する項目の「name」属性を設定します。

■ 検索結果

本タグライブラリの検索結果は、次のキーで取得可能です。
括弧内は各キーの名称になります、実際の設定値ではありませんのでご注意ください。

- companyCd (会社コード)
- companyName (会社名)
- wsAgreementCd (勤務体系コード)
- wsAgreementName (勤務体系名)
- wsCategoryCd (勤務体系カテゴリコード)
- wsCategoryName (勤務体系カテゴリ名)
- lbrYearStartM (年度開始月)
- lbrYearStartD (年度開始日)
- lbrWeekStartCd (起算曜日コード)
- lbrCloseCycleCd (締め期間単位コード)
- weeklyWorkingHoursCd (週の所定労働時間区分)
- note (備考)
- objectiveTimeType (客観時間種別)
- managerFlag (管理者フラグ)
- lbrCloseD1 (締め日1)
- lbrCloseD2 (締め日2)
- lbrCloseD3 (締め日3)
- lbrCloseCycleWeeks (締め期間週単位)
- variableWorkTypeCd (変形労働種別)
- variablePeriodStartDate (変形期間起算日)
- variablePeriodTypeCd (変形期間単位区分)
- variablePeriodNum (変形期間単位数)
- shortageCarryFwdCd (不足分繰越区分)
- openedMaxStartDate (オープン済開始日(最大))
- localeId (ロケールID)
- extensionA (予備項目A)
- extensionB (予備項目B)
- extensionC (予備項目C)
- extensionD (予備項目D)
- extensionE (予備項目E)
- extensionF (予備項目F)
- extensionG (予備項目G)
- extensionH (予備項目H)
- extensionI (予備項目I)
- extensionJ (予備項目J)
- entryTs (登録日)
- entryUserCd (登録者)
- renewCnt (更新回数)
- renewTs (更新日)
- renewUserCd (更新者)

項目

- はじめに
 - 本タグライブラリを使用する際の設定について
- intra-mart Accel Kaiden! 基盤モジュール
 - ガジェット <k:gadgetContainer>
 - ブロック <k:blockContainer>
 - TMTTable <k:tmtable>
 - TMTTable操作パネル <k:tmtableOperationPanel>
 - ダイアログ <k:dialog>
 - フローティングボックス <k:floatingBox>
 - 2段ヘッダリスト <k:fs2List>

はじめに

本タグライブラリを使用する際の設定について

本タグライブラリを使用するJSPのページで、次の様な taglib ディレクティブを指定する必要があります。
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>

intra-mart Accel Kaiden! 基盤モジュール

ガジェット <k:gadgetContainer>

- 概要
ガジェット用のタグライブラリです。
本タグライブラリで囲んだ内容がガジェットとなります。
ガジェット内の要素「[ブロック <k:blockContainer>](#)」「[TMTTable <k:tmtable>](#)」については、各項目を参照してください。
- 使用例

```
<c:set var="gadgetClass">sampleGadget</c:set>
<c:set var="gadgetVariation">v01</c:set>
<c:set var="gadgetInstance">${param.gadget_instance}</c:set>
<c:set var="gadgetId">${gadgetClass}_${gadgetVariation}_${gadgetInstance}</c:set>

<k:gadgetContainer gadgetClass="${gadgetClass}"
    gadgetVariation="${gadgetVariation}"
    gadgetInstance="${gadgetInstance}"
    gadgetTitle="サンプル"
    gadgetId="${gadgetId}"
    style="width:850px;overflow:hidden;">
    <!-- ガジェットの内包要素 -->
</k:gadgetContainer>
```



コラム

ガジェットの幅 (width) は他のガジェットと合わせるため、極力「850px」としてください。

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明
gadgetClass	ガジェットクラスを設定します。
gadgetVariation	ガジェットバリエーションを設定します。
gadgetInstance	ガジェットインスタンスを設定します。 「gadgetInstance」は申請書マスタメンテナンスで申請書を作成した際に生成します。 リクエストパラメータから「\${param.gadget_instance}」の様に取得可能です。
gadgetId	ガジェットIDを設定します。 ガジェットクラス、ガジェットバリエーション、ガジェットインスタンスの3つをアンダースコア(_) でつなげたものをガジェットIDといい、このガジェットIDが、 ジェットを一意に示す値になります。
gadgetTitle	ガジェットタイトル用のサーフェスキーを設定します。 サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。
hideGadgetTitle	タイトルの非表示有無を設定します。 “true”: ガジェットのタイトルを表示しません。 “false”: ガジェットのタイトルを表示します。 ※省略時は“false”が設定されたものとします。 本設定の有無に関わらず、エラーダイアログ内のタイトルは表示します。

属性名	説明
noScriptForDebug	ガジェット初期化用javascriptの出力有無を設定します。 “true”: ガジェット初期化用javascriptを出力しません。 “false”: ガジェット初期化用javascriptを出力します。 ※省略時は“false”が設定されたものとします。 本設定は開発時に使用するもので、機能リリース時は省略してください。

 **注意**
「gadgetClass」、「gadgetVariation」はガジェットマスタデータと同じ値を設定してください。

ブロック <k:blockContainer>

- 概要
ブロック用のタグライブラリです。
ガジェット内で単票形式で表現する場合に使用するタグライブラリです。

 **コラム**
<k:blockContainer>はHTML出力時には<div>に変換されますので、<div>の属性を指定できます。

使用例

経費旅費情報

支出目的 *	
備考	

```
<k:blockContainer id="sampleBlock">
  <table class="imui-form">
    <tr>
      <th><k:surface surfaceKey="支出目的" mode="0" requiredMark="true"/></th>
      <td><k:text name="summary1" mode="0" /></td>
    </tr>
    <tr>
      <th><k:surface surfaceKey="備考" mode="0" /></th>
      <td><k:text name="summary2" mode="0" /></td>
    </tr>
  </table>
</k:blockContainer>
```

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
id	「id」属性を設定します。	○

TMTable <k:tmtable>

- 概要
TMTable用のタグライブラリです。
ガジェット内で帳票形式で表現する場合に使用するタグライブラリです。

 **コラム**
<k:tmtable>はHTML出力時には<table>に変換されますので、<table>の属性を指定できます。
<tbody>内の内容が動的に行追加/行削除される対象となります。
[TMTable操作パネル <k:tmtableOperationPanel>](#)にて、9つの操作パネルを追加でき、この操作パネルによって行追加/行削除などの操作が行えます。

使用例

	入力欄1 *	入力欄2 *
		
		
		

```
<k:ttable id="sampleTmtable" class="imui-form">
<thead>
<tr>
<td>
<k:ttableOperationPanel position="header" allowAdd="true" allowDel="true" />
</td>
<th><k:surface surfaceKey="入力欄1" requiredMark="true"/></th>
<th><k:surface surfaceKey="入力欄2" requiredMark="true"/></th>
</tr>
</thead>
<tbody>
<tr>
<td>
<k:ttableOperationPanel position="detail" allowAdd="true" allowDel="true" />
</td>
<td><k:text name="inputArea1" mode="0"/></td>
<td><k:text name="inputArea2" mode="0"/></td>
</tr>
</tbody>
</k:ttable>
```

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
id	「id」属性を設定します。	○
rowNumberTarget	行番号を表示する項目の「name」属性を設定します。 本属性に「 <i>TTable操作パネル</i> <i><k:ttableOperationPanel></i> 」のnameを設定することで、行番号を表示します。	

TTable操作パネル *<k:ttableOperationPanel>*

- 概要
TTable操作パネル用のタグライブラリです。
- 使用例

```
<k:ttable id="sampleTmtable" class="imui-form" rowNumberTarget="rowNumber">
<thead>
<tr>
<td style="vertical-align:middle;">
<k:ttableOperationPanel
position="header"
allowAdd="true"
allowDel="true"
/>
</td>
<%-- 他のヘッダー要素 --%>
</tr>
</thead>
<tbody>
<tr>
<td>
<k:ttableOperationPanel
position="detail"
allowAdd="true"
allowDel="true"
name="rowNumber"
/>
</td>
<%-- 他の明細要素 --%>
</tr>
</tbody>
</k:ttable>
```

- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
position	操作パネル配置先を設定します。 “header”:「上に挿入」や「行削除」が使用不可になります。 “detail”:操作パネルのすべてが使用可能です。	○
allowAdd	行追加可否を設定します。 “true”:行追加を有効にします。 “false”:行追加を無効にします。	○
allowDel	行削除可否を設定します。 “true”:行削除を有効にします。 “false”:行削除を無効にします。	○

属性名	説明	必須
name	「name」属性を設定します。 本属性に設定した値を「 <i>TMTable <k:tmtable></i> 」のrowNumberTargetに設定することで、行番号を表示します。	

ダイアログ <k:dialog>

- 概要
ダイアログ用のタグライブラリです。
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
id	「id」属性を設定します。	○
title	タイトル表示用のサーフェスキーを設定します。 サーフェスキーを設定することで、プロパティファイルから文字列を取得します。 サーフェス定義に存在しないキーの場合は、設定された内容をそのまま出力します。	○
modal	モーダル表示有無を設定します。 “true”:ダイアログがモーダル表示になります。 “false”:通常のダイアログ表示となります。	

フローティングボックス <k:floatingBox>

- 概要
フローティングボックス用のタグライブラリです。
- 使用例

```
<k:floatingBox id="dailyInputWindow" modal="true" posHorizonCenter="true" posY="5%">  
  <%-- フローティング表示するための内包要素 --%>  
</k:floatingBox>
```

- フローティングボックスを表示する
\$(“#id”).KaidenFloatingBox(“open”);
- フローティングボックスを閉じる
\$(“#id”).KaidenFloatingBox(“close”);
- フローティングボックスの表示時のイベントを指定する
\$(“#id”).on(“open”, function(){...});
- フローティングボックスの閉じる時のイベントを指定する
\$(“#id”).on(“close”, function(){...});
- フローティングボックスの開くときにポジション (top/left)を指定
\$(“#id”).KaidenFloatingBox({top:“10px”, left:“10px”});
- 属性
本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
id	「id」属性を設定します。 フローティング表示時の要素として出力される<div>タグの「id」属性に設定します。	○
modal	モーダル表示モードを設定します。 “true”:モーダル表示になります。 “false”:モーダル表示になりません。	
draggable	移動可否を設定します。 modalに“true”が設定されている場合は、本属性は使用しません。 “true”:フローティング表示した要素がドラッグ可能になります。 “false”:フローティング表示した要素がドラッグ不可になります。	
dragHandle	ドラッグハンドルを設定します。 modalに“true”が設定されている場合は、本属性は使用しません。 jQuery.Draggableのオプション「handle」を設定します。	
posX	表示座標(X)を設定します。	
posY	表示座標(Y)を設定します。	
posHorizonCenter	左右中央表示モードを設定します。 posXに値が設定されている場合は、本属性は使用しません。 “true”:左右中央に表示します。 “false”:左右配置位置を指定しません。	
posVerticalMiddle	上下中央表示モードを設定します。 posYに値が設定されている場合は、本属性は使用しません。 “true”:上下中央に表示します。 “false”:上下配置位置を指定しません。	
closeButton	閉じるボタンの表示有無を設定します。 “true”:閉じるボタンを表示します。 “false”:閉じるボタンを表示しません。 ※省略時は“false”が設定されたものとします。	

2段ヘッダリスト <k:fs2List>

- 概要
2段ヘッダリスト用のタグライブラリです。
横スクロール対象とならない固定列と、横スクロール対象となる可変列を定義できます。
設定は画面設定のXMLファイルで行います。
- 使用例

✓	No	A	B		E	F	
			C	D		G	H

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fs2ListSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/fs2ListSettings">
  <defaultSetting>
    <!-- 検索画面に表示する最大件数 -->
    <limit>20</limit>
    <!-- 検索画面の行番号サーフェスキー -->
    <rownumberSurfaceKey></rownumberSurfaceKey><!-- 未指定時、デフォルトの「No」を表示 -->
    <!-- 検索一覧画面 選択マークカラム 使用/不使用 -->
    <useSelectedMark>true</useSelectedMark>
    <!-- 検索一覧画面 行番号カラム 使用/不使用 -->
    <useRownumber>true</useRownumber>
    <!-- 検索一覧画面の各カラムの表示縦幅 -->
    <colsHeight>32px</colsHeight>
    <!-- 検索一覧画面の表示縦幅 -->
    <fs2ListScrollHeight>350px</fs2ListScrollHeight>
  </--
  固定列を表示する場合はfixedCols, 可変列を表示する場合はvariableColsのタグで要素を作成してください。

  fieldKey: フィールドキー

  width: 表示幅
  横幅を指定します。
  0pxを指定すると隠し項目になります。
  省略時はwidth指定なしになります。

  align: 表示位置(left / center / right)
  left: 左寄せ
  center: 中寄せ
  right: 右寄せ
  省略時はleftが適用されます。

  headerSurfaceKey: ヘッダサーフェスキー
  省略時は項目名が空欄になります。

  colPosition: 表示カラムの位置(1 / 2)
  1: 上段
  2: 下段
  省略時は1が適用されます。
  2段階にしない項目は指定不要です。

  colspan: 水平方向の結合
  colspan="xx"のxxを指定します。
  0を指定すると結合しません。
  省略時は結合しません。

  rowspan: 垂直方向の結合
  colspan="xx"のxxを指定します。
  0の場合は結合しません。
  省略時は結合しません。
  2まで指定可能で、3以上指定すると2が適用されます。

  sortable: ソート可否
  true: ソート可
  false: ソート不可
  2段ある場合の上段項目はtrueを指定しても無効になります。
  省略できません。
-->

  <!-- 検索画面の固定列カラム設定 -->
  <fixedCols>
    <fixedCol>
      <fieldKey>A</fieldKey>
      <width>100px</width>
      <align></align>
      <headerSurfaceKey></headerSurfaceKey>
      <colPosition></colPosition>
      <colspan></colspan>
      <rowspan>2</rowspan>
      <sortable>>false</sortable>
    </fixedCol>
    <fixedCol>
```



```

</fieldKey>B</fieldKey>
<width>100px</width>
<align></align>
<headerSurfaceKey></headerSurfaceKey>
<colPosition>1</colPosition>
<colspan>2</colspan>
<rowspan></rowspan>
<sortable>>false</sortable>
</fixedCol>
<fixedCol>
  <fieldKey>C</fieldKey>
  <width>50px</width>
  <align></align>
  <headerSurfaceKey></headerSurfaceKey>
  <colPosition>2</colPosition>
  <colspan></colspan>
  <rowspan></rowspan>
  <sortable>>true</sortable>
</fixedCol>
<fixedCol>
  <fieldKey>D</fieldKey>
  <width>50px</width>
  <align></align>
  <headerSurfaceKey></headerSurfaceKey>
  <colPosition>2</colPosition>
  <colspan></colspan>
  <rowspan></rowspan>
  <sortable>>true</sortable>
</fixedCol>
</fixedCols>

```

<!-- 検索画面の可変列カラム設定 -->

```

<variableCols>
  <variableCol>
    <fieldKey>E</fieldKey>
    <width>100px</width>
    <align></align>
    <headerSurfaceKey></headerSurfaceKey>
    <colPosition></colPosition>
    <colspan></colspan>
    <rowspan>2</rowspan>
    <sortable>>false</sortable>
  </variableCol>
  <variableCol>
    <fieldKey>F</fieldKey>
    <width>100px</width>
    <align></align>
    <headerSurfaceKey></headerSurfaceKey>
    <colPosition>1</colPosition>
    <colspan>2</colspan>
    <rowspan></rowspan>
    <sortable>>false</sortable>
  </variableCol>
  <variableCol>
    <fieldKey>G</fieldKey>
    <width>50px</width>
    <align></align>
    <headerSurfaceKey></headerSurfaceKey>
    <colPosition>2</colPosition>
    <colspan></colspan>
    <rowspan></rowspan>
    <sortable>>true</sortable>
  </variableCol>
  <variableCol>
    <fieldKey>H</fieldKey>
    <width>50px</width>
    <align></align>
    <headerSurfaceKey></headerSurfaceKey>
    <colPosition>2</colPosition>
    <colspan></colspan>
    <rowspan></rowspan>
    <sortable>>true</sortable>
  </variableCol>
</variableCols>

```

<!-- 検索画面のソート順:2つまで指定可 -->

```

<orders>
  <order>
    <fieldKey>A</fieldKey>
    <type>asc</type>
  </order>
  <order>
    <fieldKey>E</fieldKey>
    <type>asc</type>
  </order>
</orders>
</defaultSetting>

```

</fs2ListSettings>

■ 属性

本タグライブラリには次の属性が設定可能です。

属性名	説明	必須
ajaxAccessUrl	一覧検索を行うajaxAccessUrlを設定します。	○
id	「id」属性を設定します。	○