



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. その他
- 3. 基本
 - 3.1. 事前準備
 - 3.1.1. 開発環境構築
 - 3.2. 仕様解説
 - 3.2.1. Gadget
 - 3.2.2. TMTTable
 - 3.2.3. Accessor
 - 3.2.4. Ajax
 - 3.2.5. Mode&Formated
 - 3.2.6. Dialog
 - 3.2.7. ListData
 - 3.2.8. FS2List
 - 3.2.9. Manager
 - 3.2.10. Conductor
 - 3.2.11. Log
 - 3.2.12. MultiLocales
 - 3.2.13. Helper&Util
 - 3.2.14. Exception
- 4. 応用
 - 4.1. ガジェット（経費旅費）
 - 4.1.1. 機能説明
 - 4.1.2. クライアントサイド
 - 4.1.3. サーバサイド
 - 4.2. ガジェット（勤務管理）
 - 4.2.1. 機能説明
 - 4.2.2. チュートリアル
 - 4.3. その他
 - 4.3.1. 機能説明
- 5. リファレンス
 - 5.1. バリデーション
 - 5.1.1. 項目
 - 5.1.2. マスタ
 - 5.1.3. その他
 - 5.2. タグライブラリ
 - 5.2.1. エレメント
 - 5.2.2. リストデータ
 - 5.2.3. その他
 - 5.3. Service、Entity作成
 - 5.3.1. テーブル定義
 - 5.3.2. Entityの作成
 - 5.3.3. Serviceの作成
 - 5.4. カスタマイズ時の注意事項
 - 5.4.1. 2.0.12以前のバージョン(SAStruts+S2JDBC)から2.0.13以降のバージョン(SpringMVC+Mirage-SQL)へアップデートする際の注意点

改訂情報

変更年月日	変更内容
2013-11-21	初版
2014-06-09	第2版 以下を追加しました。 intra-mart Accel Kaiden! 勤務管理 の記載を追加
2015-02-27	第3版 以下の追加・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
2016-08-01	第4版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 ガジェットデータ補完を修正 集計項目の集計方法 を追加 集計項目を新たに追加するの定義 を修正 集計項目を新たに追加するの実装 を修正 照会画面等に新たな集計項目を表示する を追加 集計チェックを新たに追加するの集計チェック処理 を修正
2017-04-01	第5版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 チュートリアル を修正 機能説明 を修正
2017-07-01	第6版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 「ガジェット（ICカードデータ検索ガジェットと連携する）」を追加 「ロジックフロー（「transit manager」と連携して自動申請する）」を追加 「ガジェット（法人カードデータ検索ガジェットと連携する）」を追加
2017-10-01	第7版 以下の追加・変更を行いました。 サーバサイド を修正
2018-04-01	第8版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 機能説明 を修正
2019-04-01	第9版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 機能説明 を修正 カスタマイズ時の注意事項 を追加
2019-08-01	第10版 以下の追加・変更を行いました。 本書の構成や記載内容を整備
2019-12-01	第11版 以下の追加・変更を行いました。 diconファイルに記述されている DTD ファイルの参照先を http から https に変更 実装 を修正 集計チェック処理 を修正 「diconファイルの作成」を修正 「diconファイルの作成」を修正 実装 を修正 「diconファイルの作成」を修正 事前準備 を修正
2020-04-01	第12版 以下の追加・変更を行いました。 本書の記載内容を整備 定義 を修正 機能説明 を修正

変更年月日	変更内容
2020-12-01	第13版 以下の追加・変更を行いました。 本書の記載内容を整備 機能説明を修正
2021-04-30	第14版 以下の追加・変更を行いました。 本書の記載内容を整備
2021-12-01	第15版 以下の追加・変更を行いました。 本書の記載内容を整備 機能説明を修正
2022-03-31	第16版 以下の追加・変更を行いました。 本書の記載内容を整備 機能説明を修正
2022-12-21	第17版 以下の追加・変更を行いました。 本書の構成や記載内容を整備 はじめにを修正 事前準備を修正 基本の「初めてのプログラミング」を削除 Gadgetを修正 Ajaxを修正 Mode&Formatedを修正 ListDataを修正 FS2Listを修正 Managerを修正 Conductorを修正 Logを修正 MultiLocalesを修正 Helper&Utilを修正 Exceptionを修正 ガジェット（経費旅費）を修正 ガジェット（経費旅費）の「チュートリアル」を削除 ガジェット（勤務管理）を修正 応用の「マスタメンテナンス」を削除 応用の「マスタ検索」を削除 応用の「ガジェット（ICカードデータ検索ガジェットと連携する）」を削除 応用の「ロジックフロー（「transit manager」と連携して自動申請する）」を削除 応用の「ガジェット（法人カードデータ検索ガジェットと連携する）」を削除 バリデーションの マスタを修正 バリデーションの その他を修正 タグライブラリの エレメントを修正 タグライブラリの リストデータを修正 タグライブラリの その他を修正 タグライブラリの「マスタ検索」を削除 リファレンスに Service、Entity作成を追加 リファレンスに カスタマイズ時の注意事項を追加 「サンプル」を削除

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。
- **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 2022 Winter
- intra-mart Accel Kaiden! Core Module 2.0.13
- intra-mart Accel Kaiden! 経費旅費 2.0.13
- intra-mart Accel Kaiden! 勤務管理 2.0.13

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TERASOLUNA Server Framework for Java (5.x)(SpringMVC)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
- Mirage-SQLでの開発の知識/経験

コラム

intra-mart Accel Kaiden!はSpringMVC+Mirage-SQL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、『[TERASOLUNA Server Framework for Java \(5.x\) 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。

その他

- 本書内で記載されている外部URLは、2022年12月現在のものです。

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
 - [applicationContext.xml](#)の作成

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

applicationContext.xmlの作成

モジュール・プロジェクトの作成後、applicationContext.xmlの作成をします。
「applicationContext-kaiden_tutorial.xml」を「src/main/resources/META-INF/spring」に作成し、次のように編集します。
この設定により、指定パッケージ配下のクラスがSpringのコンポーネント対象として扱われます。
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.kaiden.v2.tutorial」と設定したものと記載しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.1.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.1.xsd">

  <!-- DIコンポーネントの対象とする要素のトップレベルパッケージ -->
  <context:component-scan base-package="jp.co.slcs.kaiden.v2.tutorial" />

</beans>
```

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) 吉柳辰巳

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	片道金額 *
経路検索連携	金額 *	

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  日当金額再取得

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 *	金額 *
☐ 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（メンテナンス）（プロジェクトマスタ）
ガジェットの色合せはプログラムで固定です。

プロジェクトメンテナンス

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
英語 ☐ D Company Workflow System Restructuring Projec
中国語(中華人民共和国) ☐ 信息系統公司工作流系統重新构筑项目

上位プロジェクト 

更新 削除 一覧に戻る

- 勤務表画面
ガジェットの色合せはプログラムで固定です。

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

ガジェット組合せはプログラムで固定です。

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

勤務時間 09:00 - 17:00 時間外 0:00 摘要

+ 休憩・控除・時間年休を追加

09:00 12:00 - 13:00 : 休憩 17:00

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

即時更新

勤務表PDF

ガジェット組合せはプログラムで固定です。

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)				
14 (月)	出勤	09:00 - 17:00 (07:00)		

ガジェットの見出し

ガジェットは次のような識別子で構成されています。

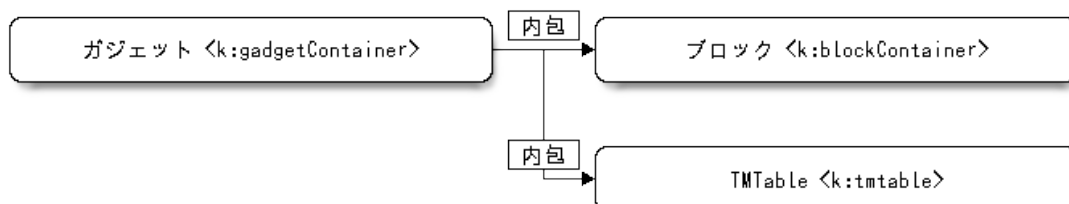
識別子	説明
ガジェットクラス	ガジェットの種別を表します。
ガジェットバリエーション	同一ガジェット種別で画面レイアウト違いを表します。
ガジェットインスタンス	特定のガジェットを同時に複数利用する場合に使用します。 ガジェットクラス、ガジェットバリエーションでガジェットを特定することが可能ですが、申請書などで、特定のガジェットを同時に複数利用する可能性があるため連番を設けています。同時複数利用を行わない画面では、原則「0」となり、同時複数利用を行う画面では、「0」からの連番です。
ガジェットID	ガジェットクラス、ガジェットバリエーション、ガジェットインスタンスの3つをアンダースコア（_）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

ガジェットの構造

ガジェットは次のような構造です。
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:tmtable>`



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エLEMENT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 **TMTTable** <k:tmtable> を配置してください。
テキストボックス <k:text>などのELEMENTは、必ず **ブロック** <k:blockContainer>、または **TMTTable** <k:tmtable> 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 **TMTTable** <k:tmtable> を内包
- **TMTTable** <k:tmtable> 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です。

```
<%@ 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/v2/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。

i 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**

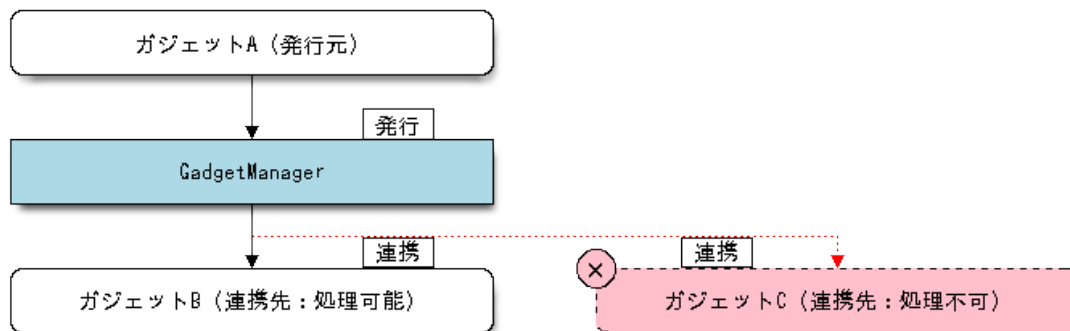
エレメントのname属性およびid属性は、**ガジェット <k:gadgetContainer>**内で一意な値を設定してください。

ブロック <k:blockContainer>や**TMTTable <k:tmttable>**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でイベントを発行し、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。

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

TMTTableとは？

TMTTable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）

- 一覧画面（マスタメンテナンス）（プロジェクトマスタ）

TMTTableのプログラミング

TMTTableは、タグライブラリの `TMTTable <k:tmtable>` を使用します。

`TMTTable <k:tmtable>` は、`<table>` に変換されますので、基本的なプログラミング方法は `<table>` と同様ですが、`<thead>` と `<tbody>` 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>

ヘッダ行として扱われますので、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>）を作成しておくと、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>にあらかじめ<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。

TMTTable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

タプルIDは、タプルに対して付与される一意な連番です。

タプル削除がされると欠番が発生しますし、「上に挿入」などを行った場合には画面表示順とタプルIDの順序は一致しません。

TMTTableの操作は、このタプルIDを利用して操作します。

タプル操作のプログラミング

JavaScriptにてタプ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、タプ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 tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTTableID"];
// 最下行に1行（1タプル）追加
tmtable.append();
```

■ 下に挿入

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

■ 上に挿入

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

- タブルを1つ削除

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

- タブルをすべて削除

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
// タブルをすべて削除
tmtable.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。

TMTable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[TMTable 操作パネル](#) `<k:tmtableOperationPanel>`を使用してください。
次の様に[TMTable 操作パネル](#) `<k:tmtableOperationPanel>`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ます。

- プログラミング例

```
<k:tmtable id="sampleTmtable" class="imui-form">
  <thead>
    <tr>
      <td>
        <k:tmtableOperationPanel 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:tmtableOperationPanel 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:tmtable>
```

TMTableの関連API

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

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"];
//タプル追加（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プルID
    //引数：tmtableAccessor : 当該TMtableへのAccessor
    //引数：data          : 挿入されたタプルに設定しようとしている値
});
```

- タプル追加（挿入）直後（データ設定後）

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
//タプル追加（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プルID
    //引数：tmtableAccessor : 当該TMtableへのAccessor
    //引数：data          : 挿入されたタプルに設定された値
});
```

- タプル削除時

```
// TMtableの取り出し
var tmtable = KAIDEN.getGadget("ガジェットID").tmtables["TMTableID"];
//タプル削除時イベント
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・TMTable）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"];
```

```
// TTableに対するAccessor
```

```
var tmtableAccessor = KAIDEN.getGadget("ガジェットID").tmtables["TTableID"].tmtableAccessor;
```

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、TTable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
```

■ TTableの場合

```
// TTableに対するAccessor
```

```
var tmtableAccessor = KAIDEN.getGadget("ガジェットID").tmtables["TTableID"].tmtableAccessor;
```

```
var findName = tmtableAccessor("name属性", "タプルID"); //name属性でのアクセス
```

```
var findId = tmtableAccessor("#id属性", "タプル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はタプルが動的に増減するため、タプルが存在しない（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、タプルIDを指定して対象エレメントを特定しますが、ここではあえてタプルIDを指定せずに「tmttableAccessor("name属性")」とすることで、全てのタプ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TMTableの同一タプル内の別要素を隣接要素と呼びます。

たとえば、以下のようにTMTable内にテキストボックスが2つある場合、“text1”と“text2”が隣接要素です。

```
<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。

```
// TMtable 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、TMTable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通信が行う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。

- **Controllerクラス**
「jp.co.slcs.kaiden.v2.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.kaiden.v2.base.foundation.model.dto.LogicResultDto;
import jp.co.slcs.kaiden.v2.base.foundation.util.ResponseWriter;
import jp.co.slcs.kaiden.v2.base.foundation.conf.LogicResultStatus;
import net.arnx.jsonic.JSON;
import org.springframework.web.bind.annotation.RequestMapping;

@RequestMapping(value = {"", "/index"})
public String index(HttpServletRequest request, HttpServletResponse response,
    Form form, Model model){

    /**～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(response, 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()は、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して扱われ、"1"を設定した場合は表示モードとして扱われます。

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>

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.kaiden.v2.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[TMTTable <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ページ全体で一意的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"}]);
});
```

i 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"}]);
});
```

i 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"});
```

i 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

タグライブラリを使用してリストデータを作成可能です。

■ プログラミング例

```
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/ui" %>
<%@ taglib prefix="klist_labor" uri="http://kaiden.slcs.co.jp/taglib/v2/listdata/labormgr" %>

<klist_labor:listDataWorkStatusCat name="workStatusCatData" blank="false" />
<k:select name="workStatusCat" mode="0" dataName="workStatusCatData"/>
```

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、Controller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.kaiden.v2.base.foundation.fs2.AbstractFs2ListForm を継承します。

- プログラミング例(**Form**)

```
package jp.co.slcs.kaiden.v2.base.feature.form.kaiden;

import jp.co.slcs.kaiden.v2.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.kaiden.v2.base.foundation.fs2.AbstractFs2ListDto を継承します。

- プログラミング例(Dto)

```
package jp.co.slcs.kaiden.v2.base.feature.dto.kaiden;

import jp.co.slcs.kaiden.v2.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Controllerクラスを作成します。

FS2ListのControllerクラスを実装する場合、jp.co.slcs.kaiden.v2.base.foundation.fs2.AbstractFs2ListController を継承します。

- プログラミング例(Controller)

```
package jp.co.slcs.kaiden.v2.base.feature.action.kaiden;

import java.util.HashMap;
import java.util.Map;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import jp.co.slcs.kaiden.v2.base.feature.dto.kaiden.Fs2ListDto;
import jp.co.slcs.kaiden.v2.base.feature.form.kaiden.Fs2ListForm;
import jp.co.slcs.kaiden.v2.base.foundation.exception.KaidenSystemException;
import jp.co.slcs.kaiden.v2.base.foundation.fs2.AbstractFs2ListController;
import jp.co.slcs.kaiden.v2.base.foundation.fs2.AbstractFs2ListDto;
import jp.co.slcs.kaiden.v2.base.foundation.fs2.AbstractFs2ListForm;

/**
 * FS2ListのControllerクラスです。
 */
@Controller
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
@RequestMapping("kaiden/v2/labor/dailyApprove")
public class Fs2ListController extends AbstractFs2ListController {

    /** Form. */
    public Fs2ListForm fs2ListForm;

    /** Dto. */
    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
*/
@RequestMapping(value = "search")
public String search(HttpServletRequest request, HttpServletResponse response, Fs2ListForm form,
    Model model) throws KaidenSystemException {
    setForm(form);
    super.index(request, response, fs2ListForm, model);
}

/**
 * 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;
}

/**
 * Formを設定.
 *
 * @param form Form
 */
protected void setForm(Fs2ListForm form) {
    fs2ListForm = form;
}
}

```

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)**
特定のタプルのデータを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。

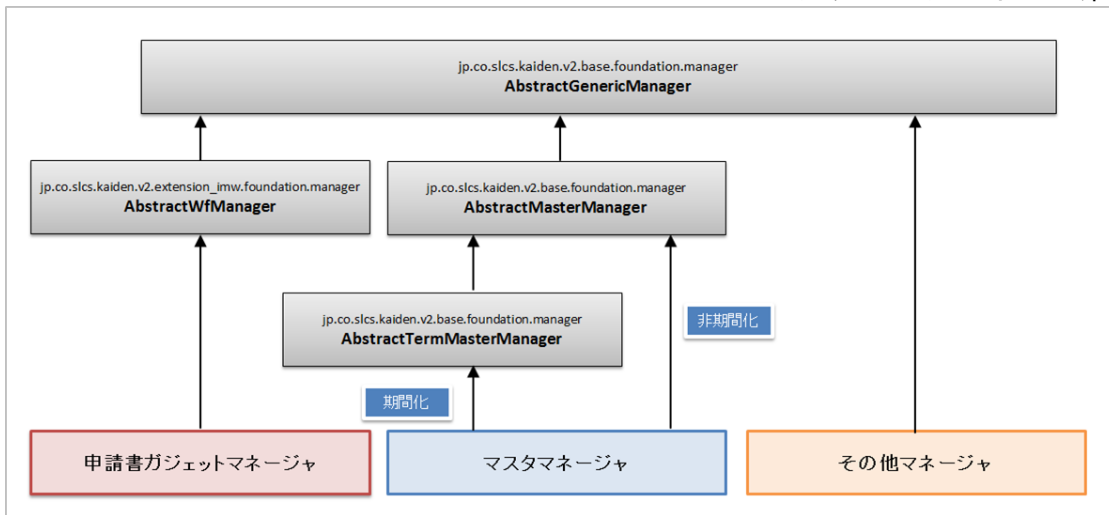
- マネージャとは？
- マネージャの構成
- マネージャの呼び出し
- マネージャの作成
 - 会社認可
 - SQLファイル検索
 - キーワード検索

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!の項目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。

- @Autowiredアノテーションによるコンポーネント取得

```

@Autowired
public ItemManager itemManager;

public void xxxMethod() {
    itemManager.xxxx();
}

```

- コンポーネント名を指定してコンポーネント取得

```

public ItemManager itemManager;

public void xxxMethod() {
    itemManager = ApplicationContextHolder.getApplicationContext().getBean(ItemManager.class);
    itemManager.xxxx();
}

```

コラム

intra-mart Accel Kaiden!のマネージャクラスでは、@Autowired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.kaiden.v2.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.kaiden.v2.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。

- @Autowiredアノテーションによるコンポーネント取得

```
@Autowired
public WorkReportConductor workReportConductor;

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

- 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 = ApplicationContextHolder.getApplicationContext().getBean(WorkReportConductor.class);
    workReportConductor.xxxx();
}
```

 コラム

intra-mart Accel Kaiden!のコンダクタークラスでは、@Autowired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.kaiden.v2.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。

MultiLocales

intra-mart Accel Kaiden!でのメッセージ、サーフィスはjp.co.slcs.kaiden.v2.base.foundation.helper.PropertyHelperクラスを使用して取得できます。

```

/** PropertyHelper. */
@Autowired
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」、サーフィス設定ファイル名は「モジュールID_surfaces_言語ID.xml」です。
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ィ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
 - 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。

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。
CompanyGroupManagerHelper	intra-mart Accel PlatformのCompanyGroupManagerクラスをラッピングしていません。
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。
ConvertPdfHelper	PDF変換に関する操作を行うクラスです。
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。
DocumentDataSettingsHelper	ドキュメントデータ設定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。
ImageViewHelper	画像参照に関連する操作を行うクラスです。
ImportHelper	Import処理を扱う機能を扱うクラスです。
JobnetManagerHelper	intra-mart Accel PlatformのJobSchedulerManagerクラスをラッピングしています。
LockHelper	intra-mart Accel PlatformのNewLockクラスをラッピングしています。
NumberHelper	数値に関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。
SessionScopeStorageFileHelper	セッションスコープストレージのファイル操作を扱うクラスです。
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。
TaskManagerHelper	intra-mart Accel PlatformのTaskManagerクラスをラッピングしています。
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。

クラス名	説明
XmlSettingsHelper	XML設定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
ExtensionNumberingSettingsHelper	拡張採番設定ファイル読み込み機能を扱うクラスです。
NumberingRuleReplacePatternHelper	採番ルール定義における文字列置換機能を扱うクラスです。
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。

- @Autowiredアノテーションによるコンポーネント取得

```
@Autowired
public CurrencyHelper currencyHelper;

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

- コンポーネント名を指定してコンポーネント取得

```
public CurrencyHelper currencyHelper;

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

コラム

intra-mart Accel Kaiden!のヘルパークラスでは、@Autowired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。

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。
BarcodeUtil	バーコード生成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。
GadgetBuilderUtil	GadgetBuilderの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。
HttpUtil	HTTP通信のユーティリティクラスです。
JsonUtil	JSON操作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。
MasterBuilderUtil	MasterBuilderのユーティリティクラスです。
ParametersUtil	パラメータ保持用のユーティリティクラスです。
PDFUtil	PDF操作のユーティリティクラスです。
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。
ReportBuilderUtil	ReportBuilderの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。
RestApiUtil	RestAPIのユーティリティクラスです。
SettingsUtil	設定情報に関するユーティリティクラスです。
SimpleFileUtil	SimpleFileのユーティリティクラスです。

クラス名	説明
StringUtil	String型オブジェクトのユーティリティクラスです。

IM-Workflow連携モジュール

クラス名	説明
AppUtil	申請書マスタのユーティリティクラスです。

ワークフローモジュール

クラス名	説明
BudgetUtil	予算のユーティリティクラスです。

勤務管理モジュール

クラス名	説明
LaborTaskUtil	intra-mart Accel Kaiden! 勤務管理モジュールのタスクに関するユーティリティクラスです。
PaidLeaveGrantUtil	有給休暇付与に関する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。
KaidenAuthorizeDenyException	対象機能の認可を持たない場合にスローされます。
KaidenDuplicationEntryException	重複登録が行われた際にスローされます。
KaidenException	基底の例外クラスです。
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。
KaidenHaventFuncAuthorityException	機能権限を持たない場合にスローされます。
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。
KaidenJobExecuteException	ジョブ実行時にエラーが発生した場合にスローされます。
KaidenMultiValidateException	サーバサイドバリデートエラーが発生した場合にスローされます。 複数のバリデート実行結果を格納することに対応しています。

クラス名	説明
KaidenNotAvailableException	対象機能が利用できない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。
KaidenTaskStopParametersException	タスクがストップした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。

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、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ィスマスタ](#)
 - [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です。

- 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

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
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です。

- 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。

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ィスマスタ
- 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

GadgetBuilderで申請書ガジェットを作成しインポートすると、申請書ガジェットマスタデータにデータが投入されます。

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」です。
----------	--

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」です。
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。
サーフィスキー	ガジェットのタイトルとなるサーフィスキーを設定します。 後述するガジェットサーフィスマスタのサーフィ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です。 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

以下のタグが存在し、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以下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です。
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です。
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です。
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です。
SPサポート	spSupport	SP(スマートフォン)画面をサポートする場合に設定します。
SPサポート対象外	spNoSupport	SP(スマートフォン)画面をサポートしない場合に設定します。
チェックルール	checkRule	申請書マスタメンテナンスのチェックルール画面で入力チェックの設定を可能にする場合設定します。 申請書マスタメンテナンスの詳細は、『 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。
検索画面のSPサポート	spSearch	SP(スマートフォン)画面をサポートする検索画面の場合に設定します。

申請書ガジェットサーフィスマスタ

ガジェットのサーフィス情報を保持するマスタです。
サーフィスキーに対して、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、サーフィ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。
サーフィスキー	ガジェットのタイトルとなるサーフィ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。
役割としては以下の通りです。

- 「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。
サーフィスキー	ガジェットサーフィスマスタのサーフィスキーに紐づけて設定します。 ここで紐づけたサーフィ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に設定されます。
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です。 「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です。 「0:設定変更可」 「a:ラベル(変更不可)」 「b:テキスト(変更不可)」 「c:テキスト(処理時入力不可・変更不可)」
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。

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。
サーフィスキー	ガジェットサーフィスマスタのサーフィスキーに紐づけて設定します。 ここで紐づけたサーフィスは申請書ガジェットのラベルに設定されます。
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です。 「0:変更可能(デフォルト任意)」将来拡張用の項目です。 「1:変更可能(デフォルト必須)」将来拡張用の項目です。 「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です。 「0:設定変更可」将来拡張用の項目です。 「a:ラベル(変更不可)」 「b:テキスト(変更不可)」 「c:テキスト(処理時入力不可・変更不可)」 「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。

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。

- [パラメータ、申請書ガジェットの設定内容取得](#)

パラメータ、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、IMW・Kaiden!のパラメータや「申請書マスタメンテナンス」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.kaiden.v2.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

```

<!-- パラメータの取り出し -->
<c:set var="searchCriteriaCompany" value="${appDto.searchCriteriaCompany}" />
<c:set var="imwApplyBaseDate" value="${appDto.imwApplyBaseDate}" />
<c:set var="applyAuthUserCd" value="${appDto.applyAuthUserCd}" />

<!-- 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]}" />

<!--サーフィスの取り出し-->
<%-- 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>

```

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。

- IM-Workflow案件番号採番機能の作成
 - 案件番号採番処理
 - applicationContext.xmlの設定
 - 案件番号採番処理の設定ファイル
 - メッセージ定義ファイルの作成

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.kaiden.v2.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。

以下のサンプルでは、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.kaiden.v2.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.kaiden.v2.base.foundation.exception.KaidenRuntimeException;
import jp.co.slcs.kaiden.v2.extension_imw.foundation.model.dto.ActionProcessParameterDto;
import jp.co.slcs.kaiden.v2.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;
    }
}
}
```

applicationContext.xmlの設定

applicationContext.xmlに設定を追加します。

本チュートリアルでは、「applicationContext-kaiden_tutorial.xml」を「src/main/resources/META-INF/spring」に作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-4.1.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-4.1.xsd">

    <!-- DIコンポーネントの対象とする要素のトップレベルパッケージ -->
    <context:component-scan base-package="jp.co.slcs.kaiden.v2.tutorial" />

    <bean id="imwNumberingExtensionSample"
        class="jp.co.slcs.kaiden.v2.tutorial.foundation.util.imw.ImwNumberingExtensionSample" />

</beans>
```

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/kaiden/v2/extension_imw/ImwNumberingSettings.xml」です。

idタグにDIコンテナへの登録名、messageIdには申請書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">
  <!-- 案件番号採番号クラス -->
  <workflowMatterNumberingClasses>
    <workflowMatterNumberingClass>
      <!-- 案件番号採番号クラス (DIコンテナへの登録名) -->
      <id>imwNumbering</id>
      <!-- 説明 (extension_imw_message.xmlのエントリ)-->
      <messageId>EIMW.defaultNumberingClass</messageId>
    </workflowMatterNumberingClass>
    <workflowMatterNumberingClass>
      <!-- 拡張案件番号採番号クラス (DIコンテナへの登録名) -->
      <id>extensionNumberingImpl</id>
      <!-- 説明 (extension_imw_message.xmlのエントリ)-->
      <messageId>EIMW.extensionNumberingClass</messageId>
    </workflowMatterNumberingClass>
    <workflowMatterNumberingClass>
      <!-- フローid単位でのKaiden独自案件番号採番号クラス (DIコンテナへの登録名) -->
      <id>kaidenNumberingByFlow</id>
      <!-- 説明 (extension_imw_message.xmlのエントリ)-->
      <messageId>EIMW.kaidenNumberingByFlowClass</messageId>
    </workflowMatterNumberingClass>
    <workflowMatterNumberingClass>
      <!-- チュートリアル案件番号採番号クラス (DIコンテナへの登録名) -->
      <id>imwNumberingExtensionSample</id>
      <!-- 説明 (tutorial_messages.xmlのエントリ)-->
      <messageId>TTRL.imwNumberingTutorial</messageId>
    </workflowMatterNumberingClass>
  </workflowMatterNumberingClasses>
</imwNumberingSettings>
```

メッセージ定義ファイルの作成

メッセージ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_messages_ja.xml」「tutorial_messages_en.xml」「tutorial_messages_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.imwNumberingTutorial">チュートリアル(すべての申請書で通し番号)</entry>

</properties>
```

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。

機能説明

本項では、勤務表画面を参考にガジェット（勤務管理）機能の概要や構成を解説します。

- 概要
- 構成

概要

intra-mart Accel Kaiden!にて提供しているガジェット（勤務管理）は、勤務表の一覧表示や入力処理を行う機能です。

集計値表示は勤務体系カテゴリごとにガジェットを分けており、勤務体系カテゴリに特化した集計値を表示しています。

intra-mart Accel Kaiden!の勤務表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です。

- intra-mart Accel Kaiden! 基盤モジュール
- intra-mart Accel Kaiden! 勤務管理モジュール

構成

intra-mart Accel Kaiden!のガジェット（勤務管理）は次のような画面構成です。

勤務表や勤務入力画面の各ガジェットの組合せはプログラム固定です。

勤務表画面

The screenshot displays the '勤務表' (Work Schedule) screen for employee 青柳辰巳 (100001). The interface includes a header with a date range (2014/05/21 - 2014/06/20) and a '提出' (Submit) button. Below the header, there are several sections:

- 通常(一般)**: A table showing work status for dates from 5/21 to 6/1. A red box labeled '勤務表リスト部ガジェット' (Work Schedule List Section Widget) highlights this table.
- 労働時間**: A table showing working hours. A red box labeled '勤務体系カテゴリ別集計値表示ガジェット' (Work System Category-wise Summary Value Display Widget) highlights this table.
- 変形(月)**: A table showing work status for dates from 6/1 to 6/19. A red box labeled '勤務表リスト部ガジェット' (Work Schedule List Section Widget) highlights this table.
- 労働時間**: Another table showing working hours. A red box labeled '勤務体系カテゴリ別集計値表示ガジェット' (Work System Category-wise Summary Value Display Widget) highlights this table.
- 出勤日数**: A table showing attendance counts. A red box labeled '勤務体系カテゴリ共通集計値表示ガジェット' (Work System Category Common Summary Value Display Widget) highlights this table.

At the bottom of the screen, there is a '提出' (Submit) button and a footer with copyright information: Copyright © 2012 NTT DATA INTRAMART CORPORATION. The page is powered by intra-mart.

コラム

同一期間に異なる複数の勤務体系に所属した場合の画面です。

複数の勤務体系に所属した場合は、勤務体系カテゴリ別の集計値を表示します。

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 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

休憩・控除・時間年休を追加

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 労働時間内(深夜) 法定休日 所定休日 休憩時間 控除 控除免除 時間年休 時間外

7:00 7:00(-) (-) (-) 0:00 0:00 0:00 0:00 0:00 0:00

同時更新

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

2015

日 月 火 水 木 金 土

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 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

休憩・控除・時間年休を追加

09:00

12:00 - 13:00 : 休憩

17:00

日次集計値表示ガジェット

勤務情報一括入力ガジェット

PDF

勤務表

青柳辰巳(100001)

2014.04.21 ~ 2014.05.20

サンプル会社 サンプル課 1

通電(一般)

日付	勤務状況	勤務時間	時間外	備考
4/21(月)	出勤	09:00 ~ 17:00 (07:00)		
22(火)	出勤	09:00 ~ 17:00 (07:00)		
23(水)	出勤	09:00 ~ 17:00 (07:00)		
24(木)	出勤	09:00 ~ 17:00 (07:00)		
25(金)	出勤	09:00 ~ 17:00 (07:00)		
26(土)	休日出勤(予定)	09:00 ~ 20:00 (10:00)	2:00	
27(日)				
28(月)				
29(火)				
30(水)				
5/1(木)				
2(金)				
3(土)				
4(日)				
5(月)				
6(火)				
7(水)				
8(木)				
9(金)				
10(土)				
11(日)				
12(月)				
13(火)				
14(水)				
15(木)				
16(金)				
17(土)				
18(日)				
19(月)				
20(火)				

PDF集計値表示ガジェット



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です。

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)に紐づいています。

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
摘要ガジェット	dailyRecordNote	lbrDailyRecordNote	v01
勤務手当ガジェット	dailyAllowanceData	dailyAllowanceDataDetail	v01
PDF集計値表示ガジェット	wrSummaryPdf	wrSummaryPdfNormal	v01
PDF勤務表表示ガジェット	workReportListPdf	workReportListPdfNormal	v01
勤務表一括入力ガジェット	dailyRecordLump	lbrDailyRecordLump	v01
作業実績入力ガジェット	dailyWorkData	dailyWorkDataDetail	v01

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
作業実績お気に入りガジェット	dailyWorkData	personalTemp	v01
勤務情報入力ガジェット(スマートフォン)	dailyRecordSp	lbrDailyRecordSp	v01
日次集計値表示ガジェット(スマートフォン)	dailySummarySp	dailySummaryNormalSp	v01
摘要ガジェット(スマートフォン)	dailyRecordNoteSp	lbrDailyRecordNoteSp	v01
勤務手当ガジェット(スマートフォン)	dailyAllowanceDataSp	dailyAllowanceDataDetailSp	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」として扱われます。

日次集計項目は、「週次」「締め期間」「月次」を集計する際の元データです。

■ 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。

集計項目の集計方法

集計項目の集計方法を、日次集計を例に説明します。

勤務表に24時を超えた時刻を入力した場合、翌日の勤務時間として扱われます。

例えば所定労働時間が7時間の従業員が、所定労働日に勤務時間を09:00から18:00、休憩時間を12:00から13:00と登録したとします。

日次の集計は以下の通りです。

- 所定労働日の労働時間：8時間
- 休憩時間：1時間
- 時間外の法定内時間：1時間

また、所定労働時間が7時間の従業員が、所定労働日に勤務時間を09:00から25:00、休憩時間を12:00から13:00と登録したとします。

登録した日の翌日が所定休日の場合、日次の集計は以下の通りです。

- 所定労働日の労働時間：14時間
- 休憩時間：1時間
- 所定休日の労働時間：1時間
- 時間外の法定内時間：1時間
- 時間外の法定外時間：7時間

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
- R：勤務手当マスタ由来

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
- 勤務手当コード

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
-----	-----	-----	---	----	----	----	----

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		
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です。

■ 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	パラメータID	parameter_data_id	varchar2	100	Y	PK	
2	パラメータSEQ	parameter_seq	number	15,0	Y	PK	
3	枝番号	parameter_seq	number	18,0	Y	PK	
4	パラメータ値	parameter_value	varchar2	4000			

■ 集計キー属性マスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	集計キー	sum_key	varchar2	30	Y	PK	
2	属性キー	attribute_key	varchar2	100	Y	PK	
3	属性値	attribute_val	varchar2	4000			
4	ソートキー	sort_key	number	15,0	Y		

集計時に管理者に係る集計値を集計対象から外す場合は、集計キー属性マスタに値を登録します。



コラム

属性キーには「management_ignore_sumkey」を設定してください。
属性値には値を設定しないでください。

- 集計キー勤務体系カテゴリ別属性マスタ

No.	論理名	物理名	型	長さ	必須	PK	備考
1	集計キー	sum_key	varchar2	30	Y	PK	
2	属性キー	attribute_key	varchar2	100	Y	PK	
3	勤務体系カテゴリコード	ws_category_cd	varchar2	20	Y	PK	
4	属性値	attribute_val	varchar2	4000			
5	ソートキー	sort_key	number	15,0	Y		

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ます。
(週次、月次では行っていません。)



コラム

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。
その場合、集計キー勤務体系カテゴリ別属性マスタに値を登録してください。
属性キーには「exclude_simple_sum_daily」を設定してください。
属性値には値を設定しないでください。

-- 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', 10, 'daily', '', null, null, null, null, null, null, null,
null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
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', 10, 'period', 'periodLateEarlyTimeParam', null, null,
null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 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', 'tutorial2');
```

-- 集計キー属性マスタ

```
insert into k31m_sumkey_att values ('DA_late_early_time_M', 'management_ignore_sumkey', null, 1, null, null, null, null,
null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 集計キー勤務体系カテゴリ別属性マスタ

```
insert into k31m_sumkey_wscat_att values ('DA_late_early_time_M', 'exclude_simple_sum_daily', 'normal', null, 1, null,
null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```



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。

i コラム

日次集計のみ管理者を集計対象から外す設定としています。
必要に応じて集計キーを休憩キー属性マスタに登録します。

実装

集計処理を作成します。

i コラム

集計クラス名を「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。

日次集計クラス作成

日次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.kaiden.v2.tutorial2.foundation.sum.util.calc.daily;

import static jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.conf.KlsKey.D;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.process.model.dto.SumDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.process.model.dto.SumDto.DailySumDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.SumCalcIF;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.model.dto.GenericSumCalcDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.model.dto.SumCalcDefDto;
import
jp.co.slcs.kaiden.v2.tutorial2.foundation.sum.util.calc.daily.DailyLateEarlyTimeCalc.DailyLateEarlyTimeCalcDto;

/**
 * 遅刻早退合計時間算出クラス。
 */
@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
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とします。

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.kaiden.v2.tutorial2.foundation.sum.util.calc.multiperiod.period;

import static jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import static jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil.toBigDecimal;
import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.manager.sum.SummaryManager;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.process.model.dto.SumDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.process.model.dto.SumDto.MultiPeriodSumDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.SumCalcIF;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.model.dto.GenericSumCalcDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.calc.model.dto.SumCalcDefDto;
import
jp.co.slcs.kaiden.v2.tutorial2.foundation.sum.util.calc.multiperiod.period.PeriodLateEarlyTimeCalc.PeriodLateEa

/**
 * 締め期間遅刻早退時間算出クラス。
 */
@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
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. */
@Autowired
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.getPeriodEndDate(), "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;
}
}

```

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。

applicationContext.xmlの設定

applicationContext.xmlの設定を追加します。

「src/main/resources/META-INF/spring」に設定した「applicationContext-kaiden_tutorial.xml」を次のように編集します。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.1.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.1.xsd">

  <!-- DIコンポーネントの対象とする要素のトップレベルパッケージ -->
  <context:component-scan base-package="jp.co.slcs.kaiden.v2.tutorial" />
  <context:component-scan base-package="jp.co.slcs.kaiden.v2.tutorial2" />

</beans>
```

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。

照会画面等に新たな集計項目を表示する

照会画面等に新たな集計項目を表示する手順を解説します。

はじめに - 照会画面等に新たな集計項目を表示するには

以下画面では、検索速度を上げるため集計値の検索用テーブルからデータを取得しています。
そのため、新たに作成した集計項目を以下画面の検索結果として表示するには、検索用テーブルに集計値を登録する設定を行う必要があります。
運用中に設定を変更することは想定していません。

集計値の検索用テーブルからデータを取得している画面

- 日次承認
- 日次点検
- 日次承認（管理者）
- 勤務表照会
- 勤務表点検
- 勤務表照会（管理者）

次項以降、新たに追加した集計項目を照会画面等に表示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ます。

- 前項で作成した「遅刻早退時間」を日次承認画面、勤務表照会画面で表示したい

これに対応した設定を実施していきます。

定義

検索用設定ファイルを設定する

検索用設定ファイルに、新しく作成した「日次集計項目」と「締め期間集計項目」を設定します。
検索用設定ファイルでは、集計キーに基づく値を集計用テーブルのどのカラムに格納するのかを設定します。

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<summaryIxMappingSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/summaryIxMappingSettings"
xmlns:tns="http://kaiden.slcs.co.jp/xmlschema/summaryIxMappingSettings">
  <defaultSetting>
    <divSettings>
      <divSetting>
        <sumDiv>daily</sumDiv>
        <mappingSettings>
          <mappingSetting>
            <fieldKey>extra_col_001</fieldKey>
            <sumKey>DA_late_early_time_M</sumKey>
          </mappingSetting>
        </mappingSettings>
      </divSetting>
      <divSetting>
        <sumDiv>period</sumDiv>
        <mappingSettings>
          <mappingSetting>
            <fieldKey>extra_col_001</fieldKey>
            <sumKey>PA_late_early_time_M</sumKey>
          </mappingSetting>
        </mappingSettings>
      </divSetting>
    </divSettings>
  </defaultSetting>
  <!--
  <companySettings>
    <companySetting>
      <companyCd>comp_other_01</companyCd>
      <setting>
        <divSettings>
          <divSetting>
            <sumDiv>daily</sumDiv>
            <mappingSettings>
              <mappingSetting>
                <fieldKey>extra_col_001</fieldKey>
                <sumKey>DA_late_early_time_M</sumKey>
              </mappingSetting>
            </mappingSettings>
          </divSetting>
          <divSetting>
            <sumDiv>period</sumDiv>
            <mappingSettings>
              <mappingSetting>
                <fieldKey>extra_col_001</fieldKey>
                <sumKey>PA_late_early_time_M</sumKey>
              </mappingSetting>
            </mappingSettings>
          </divSetting>
        </divSettings>
      </setting>
    </companySetting>
  </companySettings>
  -->
</summaryIxMappingSettings>

```

「daily」に「日次集計項目」を、「period」に「締め期間集計項目」を設定します。

フィールドキーには集計キーを設定します。

合計キーにはカラム名を指定します。

日次集計項目の値は、日次集計テーブル（検索用）の指定したカラムに格納されます。

締め期間集計項目の値は、締め期間集計データテーブル（検索用）の指定したカラムに格納されます。

カラム名は「extra_col_001」から「extra_col_050」を指定してください。

サーフィス定義を作成する

ファイル名をtutorial2_surfaces.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.DA_late_early_time_M">遅刻早退時間（日次）</entry>
  <entry key="TTRL.summaryItem.V.PA_late_early_time_M">遅刻早退時間</entry>
</properties>
```

日次承認画面、勤務表照会画面の定義ファイルを設定する

日次承認画面の定義ファイルに、新しく作成した「日次集計項目」を追加します。

- 日次承認画面の定義ファイル

```
<?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>
    <!-- 検索画面の行番号ラベルID -->
    <rownumberSurfaceKey></rownumberSurfaceKey> <!-- デフォルトを利用 -->
    <!-- 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>
    <colsHeight>32px</colsHeight>
    <!-- 検索画面の固定列カラム設定 -->
    <fixedCols>

    ... 略 ...

    <!-- 検索画面の可変列カラム設定 -->
    <variableCols>

    ... 略 ...

    <variableCol>
      <fieldKey>DA_late_early_time_M</fieldKey>
      <width>120px</width>
      <align></align>
      <headerSurfaceKey>TTRL.summaryItem.V.DA_late_early_time_M</headerSurfaceKey>
      <colPosition></colPosition>
      <colspan></colspan>
      <rowspan>2</rowspan>
      <sortable>true</sortable>
    </variableCol>

    ... 略 ...

  </variableCols>
  <!-- 検索画面のソート順 -->
  <orders>
    <order>
      <fieldKey>userName</fieldKey>
      <type>asc</type>
    </order>
    <order>
      <fieldKey>targetDate</fieldKey>
      <type>asc</type>
    </order>
  </orders>
</defaultSetting>
<!-- 会社ごとの設定 -->
</fs2ListSettings>
```

勤務表照会画面の定義ファイルに、新しく作成した「締め期間集計項目」を追加します。

- 勤務表照会画面の定義ファイル

```

<?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>
    <!-- 検索画面の行番号ラベルID -->
    <rownumberSurfaceKey></rownumberSurfaceKey> <!-- デフォルトを利用 -->
    <!-- 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>
    <colsHeight>32px</colsHeight>
    <!-- 検索画面の固定列カラム設定 -->
    <fixedCols>

    ... 略 ...

    <!-- 検索画面の可変列カラム設定 -->
    <variableCols>

    ... 略 ...

    <variableCol>
      <fieldKey>PA_late_early_time_M</fieldKey>
      <width>120px</width>
      <align></align>
      <headerSurfaceKey>TTRL.summaryItem.V.PA_late_early_time_M</headerSurfaceKey>
      <colPosition></colPosition>
      <colspan></colspan>
      <rowspan>2</rowspan>
      <sortable>true</sortable>
    </variableCol>

    ... 略 ...

  </variableCols>
  <!-- 検索画面のソート順 -->
  <orders>
    <order>
      <fieldKey>periodStartDate</fieldKey>
      <type>desc</type>
    </order>
    <order>
      <fieldKey>userName</fieldKey>
      <type>asc</type>
    </order>
  </orders>
</defaultSetting>
<!-- 会社ごとの設定 -->
</fs2ListSettings>

```

検索画面の可変列カラム設定に、新しく作成した「日次集計項目」と「締め期間集計項目」を追加します。

フィールドキーには集計キーを設定します。

ヘッダーサーフィスキーにはサーフィス定義で定義したサーフィスキーを設定します。

集計値の検索用テーブルからデータを取得している画面の定義ファイルは以下の通りです。

日次の集計値の検索用テーブルからデータを取得

画面名	定義ファイル名
日次承認	DailyApproveSettings.xml
日次点検	DailyInspectSettings.xml
日次承認（管理者）	DailyApproveSupSettings.xml

日次以外の集計値の検索用テーブルからデータを取得

画面名	定義ファイル名
勤務表照会	WorkReportSearchEmpSettings.xml
勤務表点検	WorkReportInspectSettings.xml
勤務表照会（管理者）	WorkReportSearchSupSettings.xml

確認

日次承認画面、勤務表照会画面を表示して、追加した項目が表示され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ます。

- 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

No.	論理名	物理名	型	長さ	必須	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	サーフィスキー	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/views/kaiden/v2/tutorial2/gadget/klsLateEarlyPeriodTime/klsLateEarlyPeriodTimeV01.jsp
Javaパス	lateEarlyPeriodTime



コラム

「Javaパス+「Checker」」がコンポーネントIDです。

■ 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

■ ガジェットサーフィスマスタ

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	サーフィスキー	surface_key	varchar2	100	Y	PK
4	ロケールID	locale_id	varchar2	50	Y	PK
5	サーフィス	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

No.	論理名	物理名	型	長さ	必須	PK	備考
6	サーフィスキー	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	
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です。

i コラム

チェック可否には「"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。

i 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です。

INSERT文は以下の通りです。

-- ガジェットマスタ

```
insert into k10m_gadget values ('klsLateEarlyTime', 'v01', '/WEB-INF/views/kaiden/v2/tutorial2/gadget/klsLateEarlyPeriodTime/klsLateEarlyPeriodTimeV01.jsp', 'lateEarlyPeriodTime', 1, 'chapterTitle', null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- ガジェットタグマスタ

```
insert into k10m_gadget_tags values ('klsLateEarlyTime', 'v01', 'period', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k10m_gadget_tags values ('klsLateEarlyTime', 'v01', 'sumCheck', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- ガジェットサーフェスマスタ

```
insert into k10m_gadget_surface values ('klsLateEarlyTime', 'v01', 'chapterTitle', 'ja', '遅刻早退時間チェック', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k10m_gadget_surface values ('klsLateEarlyTime', 'v01', 'chapterTitle', 'en', '遅刻早退時間チェック', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k10m_gadget_surface values ('klsLateEarlyTime', 'v01', 'chapterTitle', 'zh_CN', '遅刻早退時間チェック', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 集計チェックAPIパラメータマスタ

```
insert into k31m_check_prm_def values ('klsLateEarlyTime', 'v01', 'criteriaValue', 'ja', '2', '遅刻早退時間', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_def values ('klsLateEarlyTime', 'v01', 'criteriaValue', 'en', '2', '遅刻早退時間', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_def values ('klsLateEarlyTime', 'v01', 'criteriaValue', 'zh_CN', '2', '遅刻早退時間', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 集計チェックAPIパラメータ内訳マスタ

```
insert into k31m_check_prm_def_dtl values ('klsLateEarlyTime', 'v01', 'criteriaValue', 1, '5', null, '1', null, '1', '1', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_def_dtl values ('klsLateEarlyTime', 'v01', 'criteriaValue', 2, '5', null, '2', null, '1', '1', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_def_dtl values ('klsLateEarlyTime', 'v01', 'criteriaValue', 3, '5', null, '3', null, '1', '1', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_def_dtl values ('klsLateEarlyTime', 'v01', 'criteriaValue', 4, '5', null, '4', null, '1', '1', null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 集計チェックマスタ

```
insert into k31m_ws_use_check values ('normal', '#default#', 'default', 'klsLateEarlyTime', 'v01', 1, 'default', 'lateEarlyTime_param_001', null, '0', null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```

-- 集計チェックパラメータマスタ

```
insert into k31m_check_prm_value values ('lateEarlyTime_param_001', 'criteriaValue', 1, null, null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_value values ('lateEarlyTime_param_001', 'criteriaValue', 2, null, null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_value values ('lateEarlyTime_param_001', 'criteriaValue', 3, '60', null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
insert into k31m_check_prm_value values ('lateEarlyTime_param_001', 'criteriaValue', 4, '120', null, null, null, null, null, null, null, null, null, null, '2014-04-01 00:00:00', 'tutorial2', 1, '2014-04-01 00:00:00', 'tutorial2');
```



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。

SQLServerの主キー最大長の900byteを超える可能性があります。



注意

「パラメータID」は一意的値を設定してください。

1個のパラメータを複数の集計チェックで使用しないでください。

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.kaiden.v2.tutorial2.foundation.sum.util.check.multiperiod.period;

import static jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil.nullToZero;
import static jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil.toMsgHHMM;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import jp.co.slcs.kaiden.v2.base.foundation.util.BeanUtil;
import jp.co.slcs.kaiden.v2.base.foundation.util.StringUtil;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.conf.SevereLevel;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.process.model.dto.SumDto.MultiPeriodSumDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.SummaryUtil;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.AbstractSummaryChecker;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.model.dto.GenericBorderCheckDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckDefDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckDto;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckResult;
import jp.co.slcs.kaiden.v2.product_labormgr.foundation.sum.util.check.model.dto.SumCheckerGenericDto;

@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
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.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 getMessageld() {
    /* *****
     * 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);
}
}

```

コラム

クラス名を「ガジェットマスタJavaパス+”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。

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>
```

サーフィス定義ファイル作成

照会画面等に新たな集計項目を表示するの定義で使したサーフィス定義ファイルを使します。

applicationContext.xmlの設定

照会画面等に新たな集計項目を表示するの実装で使したapplicationContext.xmlを使します。

集計チェック定義登録ガジェット

Controllerクラス作成

コントローラークラスを作成します。

```
package jp.co.slcs.kaiden.v2.tutorial2.feature.action.kaiden.gadget;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import
jp.co.slcs.kaiden.v2.product_labormgr.feature.controller.kaiden.gadget.AbstractKlsSettingControllerBase;
import jp.co.slcs.kaiden.v2.product_labormgr.feature.form.kaiden.gadget.AbstractKlsSettingForm;
import jp.co.slcs.kaiden.v2.product_labormgr.feature.logic.kaiden.gadget.AbstractKlsSettingLogic;
import jp.co.slcs.kaiden.v2.tutorial2.feature.form.kaiden.gadget.KlsLateEarlyPeriodTimeV01Form;
import jp.co.slcs.kaiden.v2.tutorial2.feature.logic.kaiden.gadget.KlsLateEarlyPeriodTimeV01Logic;

@Controller
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
@RequestMapping("kaiden/v2/gadget/klsLateEarlyPeriodTimeV01")
public class KlsLateEarlyPeriodTimeV01Controller extends AbstractKlsSettingControllerBase {

    public KlsLateEarlyPeriodTimeV01Form klsLateEarlyPeriodTimeV01Form;

    @Autowired
    public KlsLateEarlyPeriodTimeV01Logic klsLateEarlyPeriodTimeV01Logic;

    @Override
    protected AbstractKlsSettingForm getForm() {
        return klsLateEarlyPeriodTimeV01Form;
    }

    @Override
    protected void setForm(AbstractKlsSettingForm form) {
        klsLateEarlyPeriodTimeV01Form = (KlsLateEarlyPeriodTimeV01Form) form;
    }

    @Override
    protected AbstractKlsSettingLogic getLogic() {
        return klsLateEarlyPeriodTimeV01Logic;
    }
}
```

Formクラス作成

フォームクラスを作成します。

```
package jp.co.slcs.kaiden.v2.tutorial2.feature.form.kaiden.gadget;

import jp.co.slcs.kaiden.v2.product_labormgr.feature.form.kaiden.gadget.AbstractKlsSettingForm;

public class KlsLateEarlyPeriodTimeV01Form extends AbstractKlsSettingForm {
}
```

Logicクラス作成

ロジッククラスを作成します。

```
package jp.co.slcs.kaiden.v2.tutorial2.feature.logic.kaiden.gadget;

import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import jp.co.slcs.kaiden.v2.product_labormgr.feature.logic.kaiden.gadget.AbstractKlsSettingLogic;

@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public class KlsLateEarlyPeriodTimeV01Logic extends AbstractKlsSettingLogic {
}
```

JSP作成

JSPファイル名をklsLateEarlyPeriodTimeV01.jspとします。

「/WEB-INF/views/kaiden/v2/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/v2/ui" %>
<%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
<%@ page import="jp.co.slcs.kaide.vn2.base.foundation.util.PropertyJspUtil"%>
<%@ page import="jp.co.slcs.kaiden.v2.base.foundation.conf.BaseProp" %>
<%@ page import="jp.co.slcs.kaiden.v2.product_labormgr.foundation.conf.ProductLabormgrProp" %>
<%@ page import="jp.co.slcs.kaiden.v2.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="wsCategoryCd">${param.wsCategoryCd}</c:set>
<c:set var="companyCd">${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="viewMode"><%= ViewMode.VIEW.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_setings">{k_time:['], k_${gadgetId}TimeMax4Caution1:['criteriaValue|2', 'criteriaValue|3',
'criteriaValue|4'], k_timeMax2:['1000:00']}</c:set>
<c:set var="validCriteriaValue_2_setings">{k_time:['], k_${gadgetId}TimeMax4Caution2:['criteriaValue|3',
'criteriaValue|4'], k_timeMax2:['1000:00']}</c:set>
<c:set var="validCriteriaValue_3_setings">{k_time:['], k_timeMax:['criteriaValue|4'], k_timeMax2:['1000:00']}</c:set>
<c:set var="validCriteriaValue_4_setings">{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="${sumCheckBlockId}" style="padding:0px 20px;">
<table class="imui-table-mixed" style="table-layout:fixed;margin-bottom:0.5em;">
<colgroup>
<col style="width:100%;"/>
</colgroup>
<tr>
<th>
<k:surface name="checkRequiredFlagOn" surfaceKey="" class="kaiden-ui-wsA_sumCheck kaiden-ui-icon-enable16
mr-5 hidden kaiden-ui-wsA_sumCheck-label" title="${surfaceCheckRequiredFlagOff_text}"/>
<k:surface name="checkRequiredFlagOff" surfaceKey="" class="kaiden-ui-wsA_sumCheck kaiden-ui-icon-disable16

```

```

mr-5 hidden kaiden-ui-wsA_sumCheck-label" title="{surfaceCheckRequiredFlagOn_text}"/>
  <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>
<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-ll16 mr-5 valign-M"></span><k:surface
surfaceKey="{surfaceLevel2}" class="valign-M"/></th>
                  <th><span class="kaiden-ui-icon-warn-lll16 mr-5 valign-M"></span><k:surface
surfaceKey="{surfaceLevel3}" class="valign-M"/></th>
                  <th><span class="kaiden-ui-icon-warn-iv16 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>
        </tr>
      </table>
    </div>
  </td>
</tr>
</thead>
</table>

```

```

    </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 ajax = KAIDEN.Ajax()
    , criteria = {}
    criteria.wsCategoryCd = "{wsCategoryCd}";
    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 = KAIKEN Ajax()

```

```

var ajax = KAIIDEN.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 = KAIIDEN.gadgetMan.getDataSet("${gadgetId}");
updateData.wsUseSumCheck = wsUseSumCheck

data.updateDataJson = KAIIDEN.Util.stringifyJSON(updateData);
//勤務体系情報より、更新カウンタを要求
data.wsaRenewCnt = KAIIDEN.gadgetMan.execPull("claimWsaBasicInfo", undefined, "${gadgetId}")[0].result.renewCnt;
data.termCd = "${termCd}";
data.wsaTermRenewCnt = KAIIDEN.gadgetMan.execPull("claimWsaTermInfo", undefined, "${gadgetId}")
[0].result.renewCnt;
ajax.postSyncAuto("${accessUrl}/renewCheckFlag", data, function(returnData) {
    KAIIDEN.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 = KAIIDEN.Ajax()
    , data = {}
    , wsUseSumCheck = {}
    , updateData = {};

    if (!KAIIDEN.gadgetMan.preSingleValidate()) {return}; //単項目チェック前
    if ("${debug}" || KAIIDEN.gadgetMan.validate("settingsklsdaily36limitv01", false)) { //バリデーション
        imuiConfirm("${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 = KAIIDEN.gadgetMan.getDataSet("${gadgetId}");
            updateData.wsUseSumCheck = wsUseSumCheck

            data.updateDataJson = KAIIDEN.Util.stringifyJSON(updateData);
            //勤務体系情報より、更新カウンタを要求

```

```

data.wsaRenewCnt = KAIDEN.gadgetMan.execPull("claimWsaBasicInfo", undefined, "${gadgetId}")
[0].result.renewCnt;
  data.termCd = "${termCd}";
  data.wsaTermRenewCnt = KAIDEN.gadgetMan.execPull("claimWsaTermInfo", undefined, "${gadgetId}")
[0].result.renewCnt;
  ajax.postSyncAuto("${accessUrl}/renewParamValues", data, function(returnData) {
    KAIDEN.getGadget("${gadgetId}").variable.loadSumCheckSetting(); //設定内容を検索
  });
});
} else {
  kaidenErrorDisplay("show");
}
});

KAIDEN.getGadget("${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1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」として操作できます。

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の通りです。

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。
勤務表一括（組織）	管理ユーザの日々の勤務予定／実績を一括で入力することができます。

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 標準版では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.kaiden.v2.product_labormgr.foundation.manager.access_data_filter.LaborSupAuthorizedAccessUserFilter`

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.kaiden.v2.product_labormgr.foundation.manager.access_data_filter.LaborEmpAuthorizedAccessUserFilter`

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);
```



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.kaiden.v2.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。

マスタメンテナンス機能 Controller

Controller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.kaiden.v2.base.foundation.master.AbstractTermMasterA2MaintenanceController
クラスを継承し、
非期間化マスタの場合はjp.co.slcs.kaiden.v2.base.foundation.master.AbstractMasterA2MaintenanceController
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です。

- 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.kaiden.v2.base.foundation.master.AbstractTermMasterA2MaintenanceController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.AbstractMasterA2MaintenanceControllerクラスを継承
 * します。
 * ***** */
public class DeptAdminFuncController extends AbstractTermMasterA2MaintenanceController {

    @Autowired
    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。
 * ***** */
@Override
public String search(HttpServletRequest request, HttpServletResponse response, Form form, Model model) {
    /* *****
     * 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。
     * ***** */
    a2CriteriaDate = dateHelper.toTimestamp("%基準日%");
    return super.search(request, response, form, model);
}
// .
// .
// .
// .
// 以降、イベントに対応するメソッドを記述
}

```

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.kaiden.v2.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.kaiden.v2.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.kaiden.v2.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.kaiden.v2.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

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です。
詳細はjp.co.slcs.kaiden.v2.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.kaiden.v2.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.kaiden.v2.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.kaiden.v2.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.kaiden.v2.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。
    // .
    // .

}

```

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

No.	論理名	型
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](#)
 - [DataDelivery連携モジュール](#)
 - [SAStruts版 アーカイブ\(ファイル連携モジュール\)](#)

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。

**注意**

各モジュールの定数ファイルでモジュールのバージョンを定義しています。
定数ファイルを変更した環境をアップデートする場合、モジュールバージョンの値を修正してください。
また、アップデートにより定数ファイルの定義に変更がないか確認してください。

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.xx	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です。
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。

Key	初期値	項目名	説明
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)
BASE.documentData.zip.count	100	Zip化実施時の ファイル数	Zip化実施時のファイル数 を定義しています。
BASE.masterSearchSp.displayUpperLimitNumber	20	マスタ検索部品 (SP版)の表示上 限件数	マスタ検索部品 (SP版) の表示上限件数を定義して います。
BASE.masterCache.mode	0	マスタキャッシュ モード	以下のとおりMaster Builder部品で検索時のマ スタキャッシュモードを適 用します。 0:ガジェット単位でマスタ データのキャッシュを行いま す。 1:申請書単位でマスタデー タのキャッシュを行いま す。 9:マスタデータのキャッ シュを行いません。
BASE.settingMaster.mode	false	設定マスタモード	以下のとおり intra-mart Accel Kaiden! の設定の 読込方法を適用します。 true: intra-mart Accel Kaiden! の設定を設定マ スタから読み込みます。 false: intra-mart Accel Kaiden! の設定を設定 ファイルから読み込みま す。

コラム

マスタキャッシュモードで「0:ガジェット単位でキャッシュ」、「1:申請書単位でキャッシュ」を設定すると、MasterBuilder部品で検索時に「extensionId」を含めた検索条件と結果をキャッシュに保持します。再検索時に検索条件が同じ場合はキャッシュした検索結果を使用します。ただし、標準機能のガジェットはガジェットごとに「extensionId」の設定が異なるため、「1:申請書単位でキャッシュ」を指定する場合は「extensionId」の設定を統一する必要があります。「1:申請書単位でキャッシュ」を指定し、申請書内の違うガジェットで同じ検索条件のMasterBuilder部品が存在する場合はレスポンスが改善する可能性があります。

注意

設定マスタモードで「true: intra-mart Accel Kaiden! の設定を設定マスタから読み込み」を設定した場合、設定マスタの登録が必須となります。設定マスタの設定は、『[intra-mart Accel Kaiden! 経費旅費 / 管理者操作ガイド](#)』-「マスタ設定」-「設定マスタ」を参照してください。

■ Path

|src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/extension_imw_constants.xml

Key	初期値	項目名	説明
EIMW.version	2.0.xx	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_cplmatterdelete	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	/kaiden/v2/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.select_company	/kaiden/v2/extension_imw/workflow/select_company/select_company_sp.jsp	会社(sp)選択画面JSPパス	会社(sp)選択画面のJSPを定義しています。
EIMW.jsp_path.apply	/kaiden/v2/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	/kaiden/v2/extension_imw/workflow/apply/apply_sp.jsp	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	/kaiden/v2/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	/kaiden/v2/extension_imw/workflow/temporary_save/temporary_save_sp.jsp	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	/kaiden/v2/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。
EIMW.jsp_path.unapply_sp	/kaiden/v2/extension_imw/workflow/unapply/unapply_sp.jsp	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	/kaiden/v2/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	/kaiden/v2/extension_imw/workflow/reapply/reapply_sp.jsp	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。

Key	初期値	項目名	説明
EIMW.jsp_path.process	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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	/kaiden/v2/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.sp_nosupport_detail	/kaiden/v2/extension_imw /workflow/sp_nosupport /sp_nosupport_detail.jsp	モバイルサ ポート外画 面JSPパス (詳細用)	モバイルサポート 外画面の詳細用の JSPを定義してい ます。
EIMW.jsp_path.history	/kaiden/v2/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	/kaiden/v2/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。

Key	初期値	項目名	説明
EIMW.jsp_path.matter_search	/kaiden/v2/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.jsp_path.matter_search	/kaiden/v2/extension_imw /workflow/matter_search /matter_search_sp.jsp	申請書検索 (sp)画面JSPパス	申請書検索(sp)画面のJSPを定義しています。
EIMW.jsp_path.public_matter_search	/kaiden/v2/extension_imw /workflow/public_matter_search /public_matter_search.jsp	申請書検索 (公開申請書コピー) 画面JSPパス	申請書検索（公開申請書コピー）画面のJSPを定義しています。
EIMW.jsp_path.public_matter_search_sp	/kaiden/v2/extension_imw /workflow/public_matter_search /public_matter_search_sp.jsp	申請書検索 (公開申請書コピー) (sp)画面JSPパス	申請書検索（公開申請書コピー）(sp)画面のJSPを定義しています。
EIMW.sendback_regist	0	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実施しない 1:登録する、及びサーバサイド入力チェックを実施する
EIMW.reserve_regist	0	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実施しない 1:登録する、及びサーバサイド入力チェックを実施する
EIMW.reserve_cancel_regist	0	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実施しない 1:登録する、及びサーバサイド入力チェックを実施する
EIMW.jsp_include_path.matter_info	/WEB-INF/views/kaiden/v2/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。

Key	初期値	項目名	説明
EIMW.jsp_include_path.matter_info	/WEB-INF/views/kaiden/v2/extension_imw/workflow/matter_info/matter_info_sp.jsp	案件情報 (sp)JSPパス	案件情報(sp)のJSPを定義しています。
EIMW.jsp_include_path.matter_info_print	/WEB-INF/views/kaiden/v2/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。
EIMW.jsp_path.print	/kaiden/v2/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/views/kaiden/v2/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。
EIMW.jsp_include_path.extension	/WEB-INF/views/kaiden/v2/extension_imw/workflow/extension/include.jsp	拡張JSPパス (画面共通)	GadgetBuilder版ガジェットの拡張JSPパスを定義しています。
EIMW.master.app.dispPagePathFlag	0	申請書マスタのページパス入力欄の表示	以下の条件で申請書マスタのページパス入力欄の表示有無を判定します。 0:表示しない(標準) 1:表示する(カスタマイズ向け)
EIMW.master.app.processInputSp	0	申請書マスタの承認入力時のsp画面利用可否	以下の条件で申請書マスタの承認入力時のsp画面利用可否を判定します。 0:利用不可(標準) 1:利用可(カスタマイズ向け)

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.xx	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。

Key	初期値	項目名	説明
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表示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)
PWKF.jNavi.method.validData	TicketIsExistResponse	J'sNAVI Jr.有効なチケット伝票データの存在有無情報取得メソッド	J'sNAVI Jr.有効なチケット伝票データの存在有無情報取得メソッドを定義しています。
PWKF.jNavi.method.ticketAmount	TicketAmountResponse	J'sNAVI Jr.チケット発注金額合計取得メソッド	J'sNAVI Jr.チケット発注金額合計取得メソッドを定義しています。
PWKF.File.numbering.enabled	true	ファイルID発番設定	以下の定義によってファイルIDの発番有無を切替えます。 true:ファイルIDを「prefix + 連番(formatで指定した形式) + suffix」形式で発番します。 false:ユニークなIDを自動発番します。
PWKF.File.numbering.prefix	ファイルID接頭辞	「ファイル管理」で発番するファイルIDの接頭辞を定義しています。	「ファイル管理」で発番するファイルIDの接頭辞を定義しています。
PWKF.File.numbering.format	%05d	ファイルID連番フォーマット	「ファイル管理」で採番する連番のフォーマットを定義しています。

Key	初期値	項目名	説明
PWKF.File.numbering.suffix	なし	ファイルID 接尾辞	「ファイル管理」で発番するファイルIDの接頭辞を定義しています。
PWKF.File.upload.path	kaiden/upload /{COMPANY_CD}/{FILE_ID}/{PROC_NO}	ファイル アップロード先	「ファイル管理」で登録したファイルのアップロード先を定義しています。
PWKF.File.upload.extension	.pdf,.bmp,.gif,.jpg,.jpeg,.png,.tif,.tiff	ファイル アップロード可能な ファイルの 拡張子	「ファイル管理」で登録可能なファイルの拡張子を定義しています。

コラム

標準の申請書マスタに対する「申請書マスタのページパス入力欄の表示」の設定は、「extension_imw_constants.xml」に移動しました。

「product_workflow_constants.xml」の設定はSAStruts版 アーカイブ(ワークフローモジュール)の申請書マスタ用の定義です。

コラム

ファイルIDの接頭辞、接尾辞で使用可能な置換文字列は次の通りです。
置換文字列を設定する際は、「{ }」で囲んでください。

置換文字列	説明
COMPANY_CD	会社コードに置換します。
CATEGORY_CD	ファイル分類コードに置換します。
PROC_CD	処理方法コードに置換します。
RECEIVED_CD	受領方法コードに置換します。
PROC_DATE_Y1	処理日時「西暦年(yyyy)」に置換します。
PROC_DATE_Y2	処理日時「西暦年上2桁(yy)」に置換します。
PROC_DATE_Y3	処理日時「西暦年下2桁(yy)」に置換します。
PROC_DATE_G3	処理日時「和暦元号(G)」に置換します。
PROC_DATE_GY1	処理日時「和暦年(y)」に置換します。
PROC_DATE_GY2	処理日時「和暦年(yy)」に置換します。
PROC_DATE_M1	処理日時「月(M)」に置換します。
PROC_DATE_M2	処理日時「月(MM)」に置換します。
PROC_DATE_D1	処理日時「日(d)」に置換します。
PROC_DATE_D2	処理日時「日(dd)」に置換します。
PROC_DATE	処理日時「年月日(yyyyMMdd)」に置換します。

コラム

ファイルアップロード先で使用可能な置換文字列は次の通りです。
置換文字列を設定する際は、「{ }」で囲んでください。

置換文字列	説明
COMPANY_CD	会社コードに置換します。
FILE_ID	ファイルIDに置換します。
PROC_NO	処理番号に置換します。
CATEGORY_CD	ファイル分類コードに置換します。
PROC_CD	処理方法コードに置換します。
RECEIVED_CD	受領方法コードに置換します。
PROC_DATE_Y1	処理日時「西暦年(yyyy)」に置換します。
PROC_DATE_Y2	処理日時「西暦年上2桁(yy)」に置換します。
PROC_DATE_Y3	処理日時「西暦年下2桁(yy)」に置換します。
PROC_DATE_G3	処理日時「和暦元号(G)」に置換します。
PROC_DATE_GY1	処理日時「和暦年(y)」に置換します。
PROC_DATE_GY2	処理日時「和暦年(yy)」に置換します。
PROC_DATE_M1	処理日時「月(M)」に置換します。
PROC_DATE_M2	処理日時「月(MM)」に置換します。
PROC_DATE_D1	処理日時「日(d)」に置換します。
PROC_DATE_D2	処理日時「日(dd)」に置換します。
PROC_DATE	処理日時「年月日(yyyyMMdd)」に置換します。

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.xx	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
PLBR.LumpUpload.maxCount	1000	作業実績一括アップロード時の最大アップロード件数	作業実績一括アップロード時の最大アップロード件数を定義しています。

Key	初期値	項目名	説明
PLBR.workDataFixPeriod.canOpenBefore	2	会社組織締め 可能期間 現在 月より前	現在月よりどのくら い前の期間の締め・ 締め解除が可能かを 定義しています。 ex. 1 = 現在月～1 か月前の期間まで締 め・締め解除可能
PLBR.workDataFixPeriod.canOpenAfter	1	会社組織締め 可能期間 現在 月より後	現在月よりどのくら い後の期間の締め・ 締め解除が可能かを 定義しています。 ex. 1 = 現在月～1 か月後の期間まで締 め・締め解除可能
PLBR.workDataConfirm.limit	20	作業実績確認 一覧の検索画 面に表示する 最大件数	作業実績確認一覧の 検索画面に表示する 最大件数を定義して います。
PLBR.workDataConfirm.targetMonthBefore	12	作業実績確認 一覧の対象年 月の表示範囲	作業実績確認一覧の 対象年月の表示範囲 を定義しています。
PLBR.workReport.leaveType.limit	4	勤務表に表示 する休暇種別 の最大件数	勤務表の集計値の残 数管理に表示する休 暇種別の最大件数を 定義しています。
PLBR.workReport. intervalBetweenTheWorkGadget	klsIntervalBetweenTheWork _v01_1	勤務表 勤務間 インターバル 設定有無判定 用ガジェット	勤務間インターバル 設定有無を判定する ガジェットを定義し ています。
PLBR.workReport.leaveManagement. adjustHours.auth.edit	1	調整後時間数 の編集可能項 目コード ※利用不可	休暇管理マスタで調 整後時間数の編集が 可能な項目コードを 定義しています。 カンマ区切りで複数 設定可能です。
PLBR.dailyApproveSp.limit	20	日次承認画面 に表示する最 大件数	スマートフォン用画 面の日次承認・日次 承認(管理者)画面に 表示する最大件数を 定義しています。
PLBR.formatSumValPattern	1	集計値表示パ ターン（時刻 形式）	「0」を指定した場 合、時刻形式で表示 する集計値がマイナ スの場合ハイフ ン“-“で表示します。 「0」以外を指定し た場合、マイナスの 集計値をマイナス値 のまま表示します。

Key	初期値	項目名	説明
PLBR.workDataInputDetail.defaultRowExists	1	作業実績入力 初期表示時行 追加有無	「0」を指定した場合、初期表示時空白行が表示されません。 「1」を指定した場合、初期表示時空白行が表示されます。
PLBR.sumApi.summaryUseScaleLimit	3	集計処理で平均値を算出する際のスケールの最大値	intra-mart Accel Kaiden! 勤務管理モジュールのモジュール設定の「平均値算出時のスケール」で設定可能な値の最大値を設定します。 「3」を指定した場合、「平均値算出時のスケール」に0から3の整数を指定可能です。
PLBR.nonWorkingHoursChecker.deductionMcategoryCd	0001	非労働時間 チェック対象 の休憩・控 除・時間年 休中分類コード	非労働時間チェックで非労働時間とする 休憩・控除・時間年 休中分類コードを定義しています。 カンマ区切りで複数 設定可能です。



コラム

調整後時間数の編集可能項目コードの「PLBR.workReport.leaveManagement.adjustHours.auth.edit」は利用不可です。

調整後時間数の編集可能項目コードを設定する場合は、intra-mart Accel Kaiden! 勤務管理モジュールのモジュール設定で設定します。

モジュール設定の詳細は『[intra-mart Accel Kaiden! セットアップガイド](#)』 - 「WARファイルの作成」 - 「intra-mart Accel Kaiden! の設定ファイル」 - 「intra-mart Accel Kaiden! 勤務管理モジュール」 - 「モジュール設定」を参照してください。

DataDelivery連携モジュール

Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/relation_data_delivery_constants.xml

Key	初期値	項目名	説明
RDDV.version	2.0.xx	DataDelivery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。
RDDV.File.Conductor. DataDelivery.Path	kaiden/DataDelivery	PDF／CSV出力用パス	PDF／CSV出力用のパスを定義しています。

Key	初期値	項目名	説明
RDDV.File.Conductor. Default.PDF.Path	/home/PDF_IN/ {COMPANY_CD}/{FILE_ID}/ {PROC_NO}/{PDF_FILE_NAME}	PDFファイルパス	PDFの出力先のパスを定義しています。 「PDF／CSV出力用パス」＋本設定値のパスにPDFを出力します。 連携データには本パスのみ出力します。
RDDV.File.Conductor. Default.CSV.Path	/home/CSV_IN/ {COMPANY_CD}_ {FILE_CATEGORY_CD}_ {RECEIVED_CATEGORY_CD}_ {OUTPUT_DATE}.csv	CSVファイルパス	CSVの出力先のパスを定義しています。 「PDF／CSV出力用パス」＋本設定値のパスにCSVを出力します。
RDDV.File.Conductor. Default.CSV.Columns	matterNumber,matterName,applyBaseDate, authUserCd,authUserName,executeUserCd, executeUserName,executeDate, processStatusCd,processStatusName, linkageStatusCd,linkageStatusName, companyCd,fileId,procNo,systemMatterId, userDataId,gadgetId,blockId,tupleId, rowNumber,procUserCd,procUserName, procTs,procCd,procName,fileCategoryCd, fileCategoryName,receivedCategoryCd, receivedCategoryName,receivedCd, receivedName,receivedTs,receivedUserCd, receivedUserName,checkedTs, checkedAuthUserCd,checkedAuthUserName, checkedExecuteUserCd, checkedExecuteUserName, tradingDate,customerCd,customerName, transCurrencyCd,transAmount,summary, fullFileId,resolutionCheck, timestampSetting,timestampTs, pdfFilePath,pdfFileName,orgFileName, registeredNumber	CSV出力カラム	CSVに出力する連携データを定義しています。
RDDV.File.Conductor. Default.CSV.Character	UTF-8	CSV文字コード	CSVの文字コードを定義しています。
RDDV.File.Conductor. Default.CSV.Delimiter	,	CSV区切り文字	CSVの区切り文字を定義しています。
RDDV.File.Conductor. Default.CSV.Enclosing	“	CSV囲み文字	CSVの囲み文字を定義しています。

コラム

PDFファイルパス、CSVファイルパスで使用可能な置換文字列は次の通りです。
置換文字列を設定する際は、「{ }」で囲んでください。

置換文字列	説明
COMPANY_CD	会社コードに置換します。
FILE_ID	ファイルIDに置換します。
PROC_NO	処理番号に置換します。
PROC_USER_CD	処理ユーザコードに置換します。
PROC_USER_NAME	処理ユーザ名に置換します。
PROC_TS	処理日時（yyyyMMdd）に置換します。
PROC_CD	処理方法コードに置換します。
PROC_NAME	処理方法名に置換します。
FILE_CATEGORY_CD	ファイル分類コードに置換します。
FILE_CATEGORY_NAME	ファイル分類名に置換します。
RECEIVED_CATEGORY_CD	受領方法カテゴリコードに置換します。
RECEIVED_CATEGORY_NAME	受領方法カテゴリ名に置換します。
RECEIVED_CD	受領方法コードに置換します。
RECEIVED_NAME	受領方法名に置換します。
RECEIVED_TS	受領日時（yyyyMMdd）に置換します。
RECEIVED_USER_CD	受領者コードに置換します。
RECEIVED_USER_NAME	受領者名に置換します。
TRADING_DATE	取引日（yyyyMMdd）に置換します。
CUSTOMER_CD	取引先コードに置換します。
CUSTOMER_NAME	取引先名に置換します。
TRANS_CURRENCY_CD	取引通貨コードに置換します。
TRANS_AMOUNT	取引通貨額に置換します。
PDF_FILE_NAME	PDFファイル名に置換します。
OUTPUT_DATE	出力日（yyyyMMdd）に置換します。
OUTPUT_TS	出力日（yyyyMMddHHmmssSS）に置換します。

コラム

以下、CSVへ出力可能なカラムの一覧です。
CSV出力カラムで使います。

CSV出力カラム一覧

キー	値
matterNumber	案件番号
matterName	案件名
applyBaseDate	申請基準日
authUserCd	処理権限者コード

キー	値
authUserName	処理権限者名
executeUserCd	処理実行者コード
executeUserName	処理実行者名
executeDate	処理実行日時
processStatusCd	処理ステータスコード
processStatusName	処理ステータス名
linkageStatusCd	連携ステータスコード
linkageStatusName	連携ステータス名
companyCd	会社コード
fileId	ファイルID
procNo	処理番号
systemMatterId	システム案件ID
userDataId	ユーザデータID
gadgetId	ガジェットID
blockId	ブロックID
tupleId	タプルID
rowNumber	行番号
procUserCd	処理ユーザコード
procUserName	処理ユーザ名
procTs	処理日時
procCd	処理方法コード
procName	処理方法名
fileCategoryCd	ファイル分類コード
fileCategoryName	ファイル分類名
receivedCategoryCd	受領方法カテゴリコード
receivedCategoryName	受領方法カテゴリ名
receivedCd	受領方法コード
receivedName	受領方法名
receivedTs	受領日時
receivedUserCd	受領者コード
receivedUserName	受領者名
checkedTs	確認日時
checkedAuthUserCd	確認権限者コード
checkedAuthUserName	確認権限者名
checkedExecuteUserCd	確認実行者コード
checkedExecuteUserName	確認実行者名
tradingDate	取引日
customerCd	取引先コード

キー	値
customerName	取引先名
transCurrencyCd	取引通貨コード
transAmount	取引通貨額
summary	摘要
fullFileId	ファイルID&処理番号
resolutionCheck	解像度チェック結果
timestampSetting	タイムスタンプ付与結果
timestampTs	タイムスタンプ日時
pdfFilePath	PDFファイルパス
pdfFileName	PDFファイル名
orgFileName	オリジナルファイル名
registratedNumber	登録番号

SAStruts版 アーカイブ(ファイル連携モジュール)

■ Path

src/main/webapp/WEB-INF/classes/jp/co/slcs/kaiden2/relation_file_constants.xml

Key	初期値	項目名	説明
RFIL.version	2.0.xx	SAStruts版 アーカイブ (ファイル連携モジュール)バージョン	モジュールのバージョンを定義しています。
RFIL.logic_designer_sortnumber	1000	IM-LogicDesignerカテゴリソート番号	IM-LogicDesignerタスクの「ファイル連携」のソート番号を定義しています。
RFIL.logic_designer_output_path	kaiden/file	ファイル配置先	「ファイル設定」、「ファイル連携」で登録したファイルの配置先を定義しています。
RFIL.file_link_id_numbering_prefix	{PROC_DATE}-	ファイル連携ID 発番文字列接頭辞	「ファイル設定」、「ファイル連携」で採番するファイル連携IDの接頭辞を定義しています。
RFIL.file_link_id_numbering_format	%05d	ファイル連携ID 発番文字列フォーマット	「ファイル設定」、「ファイル連携」で採番するファイル連携IDのフォーマット形式を定義しています。
RFIL.file_link_id_numbering_suffix	なし	ファイル連携ID 発番文字列接尾辞	「ファイル設定」、「ファイル連携」で採番するファイル連携IDの接尾辞を定義しています。
RFIL.proc_number_path_format	%02d	処理番号フォーマット	「ファイル設定」、「ファイル連携」で採番する処理番号のフォーマット形式を定義しています。

 コラム

ファイル連携IDの発番文字列接頭辞、接尾辞で使用可能な置換文字列は次の通りです。
置換文字列を設定する際は、「{ }」で囲んでください。

置換文字列	説明
COMPANY_CD	会社コードに置換します。
PROC_USER_CD	処理ユーザコードに置換します。
PROC_DATE_Y1	処理日時「西暦年(yyyy)」に置換します。
PROC_DATE_Y2	処理日時「西暦年上2桁(yy)」に置換します。
PROC_DATE_Y3	処理日時「西暦年下2桁(yy)」に置換します。
PROC_DATE_G3	処理日時「和暦元号(G)」に置換します。
PROC_DATE_GY1	処理日時「和暦年(y)」に置換します。
PROC_DATE_GY2	処理日時「和暦年(yy)」に置換します。
PROC_DATE_M1	処理日時「月(M)」に置換します。
PROC_DATE_M2	処理日時「月(MM)」に置換します。
PROC_DATE_D1	処理日時「日(d)」に置換します。
PROC_DATE_D2	処理日時「日(dd)」に置換します。
PROC_DATE	処理日時「年月日(yyyyMMdd)」に置換します。
CATEGORY	ファイル分類コードに置換します。

リファレンス

バリデーション

項目

- 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 $\leq$ 入力値 $\leq$ 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 $\leq$ 入力値 $\leq$ 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 $\leq$ 入力値 $\leq$ 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 $\leq$ （または $<$ ） 入力値 $\leq$ （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 $\leq$ （または $<$ ） 入力値 $\leq$ （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 $\leq$ （または $<$ ） 入力値 $\leq$ （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。	○
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。	○
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。	

時刻

■ 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。	○

設定順	説明	必須
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。	

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からスケールを引いた数値を出力します。 {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。

i 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 : ˆ ˚ (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。

i 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・時間年休中分類
 - 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/views/kaiden/v2/product_workflow/foundation/commonInclude.jsp"></jsp:include>
  <c:set var="pwkfCommonIncluded" scope="request">true</c:set>
</c:if>
```

i 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/views/kaiden/v2/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。	

設定順	説明	必須
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。	

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。	

設定順	説明	必須
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decisionAllowanceDepartmentSetCd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。	○

設定順	説明	必須
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。	○

その他

- 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/views/kaiden/v2/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/views/kaiden/v2/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。	○

タグライブラリ

エレメント

- 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/v2/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

サーフィス <k:surface>

- 概要**
サーフィス（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ィスキーを設定します。 サーフィ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。 サーフィ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です。） “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	項目名用のサーフィ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。） サーフィ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。 サーフィ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。	○
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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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。	○

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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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。	

属性名	説明	必須
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です。） “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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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です。） “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	項目名用のサーフィ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。） サーフィ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。 サーフィ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-locales>

- 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です。） “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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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	ラベル用のサーフィ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>	
surfaceKey	項目名用のサーフィ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。） サーフィ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。 サーフィ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ィスキーを設定します。 サーフィ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。 サーフィ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ィスキーを設定します。 サーフィ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。 サーフィ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	ラベル用のサーフィ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	項目名用のサーフィ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。） サーフィ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。 サーフィ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! 勤務管理モジュール
 - 勤務状況マスタ <klist_labor:listDataWorkStatus>
 - 休憩・控除・時間年休マスタ <klist_labor:listDataDeduction>
 - 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>
 - 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>
 - 点検状態 <klist_labor:listDataSubmissionStatus>
 - 休日出勤 <klist_labor:listDataUnProcessHolidayWork>
 - 代休 <klist_labor:listDataUnProcessCompensation>
 - お気に入り(勤務表) <klist_labor:listDataLbrFavorite>
 - お気に入り(作業実績) <klist_labor:listDataFavorite>
 - 対象年月(作業実績) <klist_labor:listDataTargetMonth>
 - 繰越清算期間 <klist_labor:listDataFlexWorkSettlementPeriod>

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! 勤務管理モジュール
 <%@ taglib prefix="klist_labor" uri="http://kaiden.slcs.co.jp/taglib/v2/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! 勤務管理モジュール

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です。
 「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です。
 出力順は勤務状況のソートキー順（昇順）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。	

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・時間年休です。

「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・時間年休です。

出力順は休憩・控除・時間年休のソートキー順（昇順）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。	○
deductionMcategoryCds	休憩・控除・時間年休中分類コードを設定します。	

代理処理種別 <klist_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です。

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
-----	----	----

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。	○

勤務カレンダールール指定パターン <klst_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です。

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。	○

締め期間 <klst_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です。
「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）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。	

属性名	説明	必須
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。	
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です。

出力順は権限セットのソートキー順（昇順）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。	○

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です。

出力順は会社、および組織のソートキー順（昇順）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。	○
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です。

「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）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。	○
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） <klst_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）です。

「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）が対象です。

出力順は日付順（昇順）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。	○
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） <klst_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）です。
「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）が対象です。
出力順は日付順（昇順）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。	○
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）です。
「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）が対象です。
出力順は日付順（昇順）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。	○
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）です。
「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）が対象です。
出力順は日付順（昇順）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。	○
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) <klst_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日と末日です。

出力順は日付順（昇順）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。	○

点検状態 <klst_labor:listDataSubmissionStatus>

■ 概要

点検状態のリストデータ作成用タグライブラリです。

出力対象は「未提出」、「提出済」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。	○

休日出勤 <klst_labor:listDataUnProcessHolidayWork>

■ 概要

休日出勤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「searchCriteriaDate」に設定された日付時点で未取得の休日出勤日です。

出力順は日付順（昇順）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。	○
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。	○

代休 <klst_labor:listDataUnProcessCompensation>

■ 概要

代休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「searchCriteriaDate」に設定された日付時点で未取得の代休です。

出力順は日付順（昇順）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。	○
userCd	ユーザコードを設定します。	○

属性名	説明	必須
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。	○

お気に入り(勤務表) <klist_labor:listDataLbrFavorite>

■ 概要

勤務表のお気に入り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」、「departmentSetCd」、「departmentCd」に設定された組織で「wsAgreementCd」に設定された勤務体系のお気に入りです。

出力順はお気に入り名順（昇順）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。	○
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。	○
wsAgreementCd	勤務体系コードを設定します。	○
userCd	ユーザコードを設定します。	○

お気に入り(作業実績) <klist_labor:listDataFavorite>

■ 概要

作業実績のお気に入りのリストデータ作成用タグライブラリです。

出力対象は、お気に入りガジェットから取得したユーザ、会社を設定されたお気に入りです。

出力順はお気に入りID順（昇順）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。	○

対象年月(作業実績) <klist_labor:listDataTargetMonth>

■ 概要

作業実績の対象年月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」に設定された会社で「searchCriteriaDate」に設定された日付を基準にオープンしている作業実績の年月です。

「targetMonthBefore」、「targetMonthAfter」に設定された範囲の作業実績が出力対象になります。
出力順は年月順（昇順）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。	○
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。	
targetMonthBefore	検索時の対象年月範囲(過去)を保持する項目の「name」属性を設定します。 ※省略時は勤務管理のモジュール設定が設定されたものとします。	
targetMonthAfter	検索時の対象年月範囲(未来)を保持する項目の「name」属性を設定します。 ※省略時は勤務管理のモジュール設定が設定されたものとします。	

繰越清算期間 <klist_labor:listDataFlexWorkSettlementPeriod>

■ 概要

フレックスの繰越清算期間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ヶ月」、「2ヶ月」、「3ヶ月」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。	○

その他

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/v2/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して扱われます。

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	必須
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。	○

属性名	説明	必須
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がガジェットを一意に示す値です。	○
gadgetTitle	ガジェットタイトル用のサーフィスキーを設定します。 サーフィ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。 サーフィスキー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。	
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/行削除される対象です。

[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

```
<k:tmtable id="sampleTmtable" class="imui-form">
  <thead>
    <tr>
      <td>
        <k:tmtableOperationPanel 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:tmtableOperationPanel 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:tmtable>
```

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「 TMTable操作パネル <k:tmtableOperationPanel> 」のnameを設定することで、行番号を表示します。	

TMTable操作パネル <k:tmtableOperationPanel>

- 概要

TMTable操作パネル用のタグライブラリです。

- 使用例

```
<k:tmtable id="sampleTmtable" class="imui-form" rowNumTarget="rowNumber">
  <thead>
    <tr>
      <td style="vertical-align:middle;">
        <k:tmtableOperationPanel
          position="header"
          allowAdd="true"
          allowDel="true"
        />
      </td>
      <!-- 他のヘッダー要素 --%>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>
        <k:tmtableOperationPanel
          position="detail"
          allowAdd="true"
          allowDel="true"
          name="rowNumber"
        />
      </td>
      <!-- 他の明細要素 --%>
    </tr>
  </tbody>
</k:tmtable>
```

- 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」が使用不可に変化します。 “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。	○
name	「name」属性を設定します。 本属性に設定した値を「 <i>TMTable</i> <k:tmtable>」のrowNumTarget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ィスキーを設定します。 サーフィ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。 サーフィ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されます。 “false”：通常のダイアログ表示です。	

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”：モーダル表示されます。 “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です。 “false”：フローティング表示した要素がドラッグ不可です。	
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。	

属性名	説明	必須
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 : 表示コラムの位置(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。

-->

<!-- 検索画面の固定列コラム設定 -->

<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。	○

本項では、Service、Entityの作成方法を説明します。

テーブル定義

本項では次の様な定義のテーブルを使用します。

- 「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
拡張用の項目として予備項目を用意することでカスタマイズ性が向上します。
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。

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。

Entityの作成

Entityを作成します。

jp.co.slcs.kaiden.v2.base.foundation.model.entity.GenerateEntityを継承してください。

java.io.Serializable.Serializableを実装してください。

Mirage-SQLの実装方法に従い、各カラムの定義を実装してください。

コラム

予備項目A～J、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
上記カラムは、親クラスのGenerateEntity.javaで定義しているため各Entityでの定義は不要です。

コラム

K99mTutorial.javaおよび標準のソースではフィールド名の定数を定義しています。
定数の定義は任意のため、必要に応じて実装してください。

Path

src/main/generated/jp/co/slcs/kaiden/v2/tutorial/foundation/model/entity/K99mTutorial.java

```
package jp.co.slcs.kaiden.v2.tutorial.foundation.model.entity;

import java.io.Serializable;
import java.sql.Timestamp;
import com.miragesql.miragesql.annotation.Column;
import com.miragesql.miragesql.annotation.PrimaryKey;
import com.miragesql.miragesql.annotation.PrimaryKey.GenerationType;
import com.miragesql.miragesql.annotation.Table;
import jp.co.slcs.kaiden.v2.base.foundation.model.entity.GenerateEntity;

/**
 * チュートリアルマスタ.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Table(name = "k99m_tutorial")
public final class K99mTutorial extends GenerateEntity implements Serializable {

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

    /** フィールド名[会社コード]. */
    public static final String COMPANY_CD = "companyCd";

    /** フィールド名[チュートリアルコード]. */
    public static final String TUTORIAL_CD = "tutorialCd";

    /** フィールド名[チュートリアル名称]. */
    public static final String TUTORIAL_NAME = "tutorialName";

    /** フィールド名[ロケールid]. */
    public static final String LOCALE_ID = "localeId";

    /** フィールド名[期間コード]. */
    public static final String TERM_CD = "termCd";

    /** フィールド名[開始日]. */
    public static final String START_DATE = "startDate";

    /** フィールド名[終了日]. */
    public static final String END_DATE = "endDate";

    /** フィールド名[削除フラグ]. */
    public static final String DELETE_FLAG = "deleteFlag";

    /** 会社コード. */
}
```

```

/** 会社コード. */
@PrimaryKey(generationType = GenerationType.APPLICATION)
@Column(name = "company_cd")
public String companyCd;

/** チュートリアルコード. */
@PrimaryKey(generationType = GenerationType.APPLICATION)
@Column(name = "tutorial_cd")
public String tutorialCd;

/** チュートリアル名称. */
@Column(name = "tutorial_name")
public String tutorialName;

/** ロケールid. */
@PrimaryKey(generationType = GenerationType.APPLICATION)
@Column(name = "locale_id")
public String localeId;

/** 期間コード. */
@PrimaryKey(generationType = GenerationType.APPLICATION)
@Column(name = "term_cd")
public String termCd;

/** 開始日. */
@Column(name = "start_date")
public Timestamp startDate;

/** 終了日. */
@Column(name = "end_date")
public Timestamp endDate;

/** 削除フラグ. */
@Column(name = "delete_flag")
public String deleteFlag;
}

```

Serviceの作成

Serviceを作成します。

jp.co.slcs.kaiden.v2.base.foundation.model.service.GenerateService<Entity>を継承してください。

空実装で構いません。

Path

src/main/generated/jp/co/slcs/kaiden/v2/tutorial/foundation/model/service/K99mTutorialService.java

```

package jp.co.slcs.kaiden.v2.tutorial.foundation.model.service;

import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;
import jp.co.slcs.kaiden.v2.base.foundation.model.service.GenerateService;
import jp.co.slcs.kaiden.v2.tutorial.foundation.model.entity.K99mTutorial;

/**
 * {@link K99mTutorial} のサービスクラス.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public final class K99mTutorialService extends GenerateService<K99mTutorial> {
    //
}

```

本項では、カスタマイズ時の注意事項を説明します。

2.0.12以前のバージョン(SAStruts+S2JDBC)から2.0.13以降のバージョン(SpringMVC+Mirage-SQL)へアップデートする際の注意点

2.0.12以前のバージョン(SAStruts+S2JDBCの開発モデル)から2.0.13以降のバージョン(SpringMVC+Mirage-SQLの開発モデル)へアップデートする際の注意点を説明します。

カスタマイズしたモジュールを「SpringMVC+Mirage-SQLの開発モデル」に修正する場合、当項を参考にしてください。

マスタメンテナンス

『[intra-mart Accel Kaiden! MasterBuilder 設定ガイド](#)』を参照してください。

MasterBuilderを使用しないでマスタメンテナンス関連機能を作成した場合、MasterBuilderを使用して作成することを推奨します。

ガジェット（ワークフロー）

『[intra-mart Accel Kaiden! GadgetBuilder 設定ガイド](#)』を参照してください。

GadgetBuilderを使用しないでマスタメンテナンス関連機能を作成した場合、GadgetBuilderを使用して作成することを推奨します。

定数ファイル(constants)

IM-Workflow連携モジュールのextension_imw_constants.xmlを標準から変更している場合、標準のJSPパスを定義している項目の設定値を見直してください。

詳細は「[機能説明](#)」 - 「constants」 - 「IM-Workflow連携モジュール」を参照ください。

Java

カスタマイズしたJavaのソースを修正する際の注意点を説明します。

なお、「SAStruts+S2JDBCの開発モデル」で作成されたJavaのソースを「SpringMVC+Mirage-SQLの開発モデル」に修正する場合の一般的な事項は記載しません。

Kaiden!に関する事項のみ説明します。

- applicationContext.xml
カスタマイズで作成したクラスをSpringのコンポーネント対象として扱うためには、applicationContext.xmlの作成が必要です。
作成方法は「[事前準備](#)」 - 「開発環境構築」 - 「applicationContext.xmlの作成」を参照ください。
- Javaのパス
標準で提供しているJavaのクラスをインポートしている場合、「jp.co.slcs.kaiden2.」の部分を「jp.co.slcs.kaiden.v2.」に変更してください。
- SQLパス
標準で提供しているSQLのパスを記載している場合、「META-INF/sql/jp/co/slcs/kaiden2/」の部分を「META-INF/sql/jp/co/slcs/kaiden/v2/」に変更してください。
- JSPパス
標準で提供しているJSPのパスを記載している場合、「/WEB-INF/view/kaiden2」の部分を「/WEB-INF/views/kaiden/v2」に変更してください。
- コンポーネントの取得
Manager、Conductor、Helperクラスなどを呼び出している場合、呼び出す方法を修正してください。
呼び出す方法は「[仕様解説](#)」を参照ください。

Service・Entity

カスタマイズで作成したテーブルに対するService、Entityを修正してください。

詳細は「[Service、Entity作成](#)」を参照してください。

SQL

Serviceに紐づくSQLをカスタマイズで作成している場合、SQLの格納場所を「META-INF/sql/jp/co/slcs/kaiden2/%Serviceクラスのパス%」から「META-INF/sql/jp/co/slcs/kaiden/v2/%Serviceクラスのパス%」に変更してください。

JSP

- Javaのパス
標準で提供しているJavaのクラスをインポートしている場合、「jp.co.slcs.kaiden2.」の部分を「jp.co.slcs.kaiden.v2.」に変更してください。
- JSPパス
標準で提供しているJSPをインポートしている場合、「/WEB-INF/view/kaiden2」の部分を「/WEB-INF/views/kaiden/v2」に変更してください。
- JSパス、CSSパス
標準で提供しているJS、CSSをインポートしている場合、「kaiden2/」の部分を「kaiden/v2/」に変更してください。
- URL
標準で提供しているURLを記載している場合、「kaiden2/」の部分を「kaiden/v2/」に変更してください。
- タグライブラリ
標準で提供しているタグライブラリを使用する場合、「<http://kaiden.slcs.co.jp/taglib/>」の部分を「<http://kaiden.slcs.co.jp/taglib/v2/>」に変更してください。
- リストデータ
2.0.13で廃止されたリストデータを使用している場合、MasterBuilderの検索機能を使用してください。
使用可能なリストデータは「[リストデータ](#)」を参照ください。
使用方法は、『[intra-mart Accel Kaiden! MasterBuilder 設定ガイド](#)」-「詳細」-「検索」を参照してください。
- マスタ検索
タグライブラリのマスタ検索は2.0.13で廃止しました。
MasterBuilderの検索機能を使用してください。
使用方法は、『[intra-mart Accel Kaiden! MasterBuilder 設定ガイド](#)」-「詳細」-「検索」を参照してください。