



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. 概要
 - 3.1. MasterBuilderとは？
 - 3.2. メリット
- 4. 基本
 - 4.1. MasterBuilder定義
 - 4.1.1. 概要
 - 4.1.2. 基本設定
 - 4.1.3. 項目設定
 - 4.1.4. 拡張設定
 - 4.1.5. サーフィス設定
 - 4.1.6. メンテナンス設定
 - 4.1.7. 検索設定
 - 4.1.8. インポート設定
 - 4.1.9. エクスポート設定
 - 4.1.10. Sample
 - 4.2. 検索用SQL
 - 4.2.1. 概要
 - 4.2.2. 一覧検索用SQL
 - 4.2.3. 対象検索用SQL
 - 4.2.4. 期間検索用SQL
 - 4.2.5. マスタ検索用SQL
 - 4.2.6. エクスポート用SQL
 - 4.2.7. Sample
 - 4.3. 設定の流れ
 - 4.3.1. テーブル定義
 - 4.3.2. Service、Entity作成
 - 4.3.3. MasterBuilder定義作成
 - 4.3.4. 検索用SQL作成
 - 4.3.5. 動作確認
- 5. 詳細
 - 5.1. メンテナンス
 - 5.1.1. クライアントサイド
 - 5.1.2. サーバーサイド
 - 5.2. 検索
 - 5.2.1. クライアントサイド
 - 5.2.2. サーバーサイド
 - 5.3. インポート
 - 5.3.1. インポート機能
 - 5.3.2. サーバーサイド
 - 5.4. エクスポート
 - 5.4.1. エクスポート機能
 - 5.4.2. サーバーサイド
 - 5.5. バリデート
 - 5.5.1. バリデート機能
 - 5.5.2. サーバーサイド
- 6. API
 - 6.1. REST API
 - 6.1.1. REST APIについて
 - 6.1.2. REST APIで利用可能なマスター一覧
 - 6.1.3. エンドポイントプレフィックス
 - 6.1.4. リクエストパラメータ共通仕様
 - 6.1.5. レスポンスデータ共通仕様
 - 6.1.6. エンドポイント
- 7. リファレンス
 - 7.1. マスタ拡張IDの利用
 - 7.1.1. 概要
 - 7.1.2. カスタマイズ方法

- 7.2. 指定可能なエレメントタイプと属性値
- 7.3. 項目値のコンバート
- 7.4. 指定可能なジョブパラメータ
 - 7.4.1. インポートジョブで指定可能なパラメータ
 - 7.4.2. エクスポートジョブで指定可能なパラメータ
 - 7.4.3. パラメータの優先度
 - 7.4.4. ファイルパスの置換文字列
- 7.5. インポートファイルのカラム判定について
 - 7.5.1. 「xxxxCd,xxxxName」が設定された場合
 - 7.5.2. 「companyCd,xxxxCd,xxxxName」が設定された場合
 - 7.5.3. 「companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName」が設定された場合
- 7.6. マスタ検索、バリデーションのみを使用するマスタの作成
 - 7.6.1. 概要
 - 7.6.2. マスタ検索、バリデーションのみを使用する場合
 - 7.6.3. Enumクラスを使用した検索部品を作成する場合
 - 7.6.4. XMLの固定値を使用した検索部品を作成する場合
- 7.7. 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
 - 7.7.1. 概要
 - 7.7.2. Service、Entity
 - 7.7.3. MasterBuilder定義
 - 7.7.4. 検索用SQL
 - 7.7.5. 動作確認
 - 7.7.6. Java、JSPソース

改訂情報

変更年月日	変更内容
2016-08-01	初版
2017-10-01	第2版 以下の変更を行いました。 本書の構成や記載内容を整備
2018-04-01	第3版 以下の変更を行いました。 本書の構成や記載内容を整備
2019-04-01	第4版 以下の変更を行いました。 本書の構成や記載内容を整備
2019-08-01	第5版 以下の変更を行いました。 本書の構成や記載内容を整備
2019-12-01	第6版 以下の変更を行いました。 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 「 設定の流れ 」の kaiden_base.dicon
2020-08-01	第7版 以下の変更を行いました。 本書の記載内容を整備
2022-12-01	第8版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ はじめにを修正 ■ 基本を修正 ■ 詳細を修正 ■ APIを追加 ■ 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を追加
2023-04-01	第9版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ REST API で利用可能なマスター一覧を修正
2023-06-30	第10版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ REST API で利用可能なマスター一覧を修正 ■ マスタ検索、バリデーションのみを使用するマスタの作成を追加
2023-11-01	第11版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ MasterBuilder定義を修正 ■ 検索を修正 ■ REST API で利用可能なマスター一覧を修正
2023-12-22	第12版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ REST API で利用可能なマスター一覧を修正
2024-04-01	第13版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ MasterBuilder定義を修正 ■ REST API で利用可能なマスター一覧を修正

変更年月日	変更内容
2024-08-31	第14版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ MasterBuilder定義の「拡張設定」を修正 ■ MasterBuilder定義の「検索設定」を修正 ■ メンテナンスの「Logic」と「シーケンス」を修正 ■ 検索の「Logic」と「シーケンス」を修正 ■ インポートの「Logic」と「シーケンス」を修正 ■ エクスポートの「Logic」と「シーケンス」を修正 ■ バリデートの「Logic」を修正 ■ REST APIで利用可能なマスター一覧を修正
2025-06-30	第15版 以下の変更を行いました。 ■ 本書の記載内容を整備 ■ REST APIで利用可能なマスター一覧を修正

はじめに

本書の内容

本書では、次の様な構成でMasterBuilderを使用したマスタ関連機能の構築方法や拡張方法を記載しています。

- [概要](#)
MasterBuilderの概要やコンセプトを記載しています。
- [基本](#)
MasterBuilderの設定方法を記載しています。
- [詳細](#)
MasterBuilderの機能詳細を記載しています。
- [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

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! の開発者

その他

- 本書内で記載されている外部URLは、2025年6月現在のものとなります。

概要

本項では、MasterBuilderの概要を説明します。

- [MasterBuilderとは？](#)
- [メリット](#)

MasterBuilderとは？

MasterBuilderとは、次のマスタ関連機能を「設定ファイル+検索用のSQL」で実現するための機能です。

- マスタメンテナンス機能
- 検索機能
- バリデーション
- インポート機能（ジョブ）
- エクスポート機能（ジョブ）

「設定ファイル+検索用のSQL」以外にも各所に拡張ポイントを設け、カスタマイズ性も考慮しています。
次項以降では、基本的な設定方法や検索用SQLの作成方法、各機能に応じたカスタマイズポイント等を解説します。

メリット

MasterBuilderを利用することで次の様なメリットが生まれます。

- **マスタ関連機能の構築工数の低下**
MasterBuilderでは、「設定ファイル+検索用のSQL」でマスタ関連機能を実現するため、各マスタ関連機能を個別に作成する場合に比べ、大幅に工数が削減できます。
また、カスタマイズを行った場合でもカスタマイズ範囲を局所化可能なため、カスタマイズ工数の削減が見込めます。
- **仕様変更への柔軟な対応**
構築後に仕様変更があった場合でも、設定により各機能の対応が行えるため、再構築による手戻りを低減できます。
また、仕様変更に伴う反映漏れ（メンテナンス機能には反映したが、インポート機能には反映が漏れるなど）も、統一した設定により防止可能です。
- **提供機能の統一**
マスタメンテナンス機能は提供しているがインポート機能は未提供など、マスタによる提供機能のばらつきを抑制できます。
また、共通基盤により実装方法が統一化されるため、開発者による実装方法のばらつきも抑制できます。

基本

本項では、MasterBuilderの設定方法を説明します。

MasterBuilder定義

本項では、MasterBuilderの設定ファイル（以降、MasterBuilder定義）の作成方法を説明します。

- 概要
 - MasterBuilder定義
 - MasterBuilder定義の配置先
 - マスタ特性
- 基本設定
- 項目設定
- 拡張設定
- サーフィス設定
- メンテナンス設定
- 検索設定
- インポート設定
- エクスポート設定
- Sample

概要

MasterBuilder定義

MasterBuilder定義は、XML形式で次の設定を行います。

設定名	概要
基本設定	マスタの基本となる設定です。（ID、名称など）
項目設定	マスタの項目設定です。本設定を使用して各機能別設定の簡略化を実現しています。
拡張設定	マスタを拡張する場合の設定です。（設定だけでは実現できない場合などに使用します。）
サーフィス設定	マスタ名や項目名などの設定です。
マスタメンテナンス設定	マスタメンテナンス機能の個別設定です。
検索設定	検索部品・セレクトボックス機能の個別設定です。
インポート設定	インポート機能の個別設定です。
エクスポート設定	エクスポート機能の個別設定です。

i コラム

MasterBuilder定義の全体イメージは「[Sample](#)」を参照してください。

i コラム

Ver2.0.16で、MasterBuilder定義の編集・検証が可能な機能を追加しました。
詳細は『[intra-mart Accel Kaiden! 経費旅費 / 管理者操作ガイド](#)』 - 「運用時の操作」 - 「Builder編集」を参照してください。

MasterBuilder定義の配置先

MasterBuilder定義は次のパスに配置します。
「%ストレージパス%/kaiden/generic/master/%マスタID%.xml」

ストレージはシステムストレージ、パブリックストレージのどちらでも配置可能で、パブリックストレージが優先的に読み込まれます。
intra-mart Accel Kaiden!のMasterBuilder定義は、システムストレージに配置されます。
上記の読み込み順を使用して、独自のMasterBuilder定義を優先的に読み込ませることが可能です。
（テナント別にMasterBuilder定義を作成したい場合も、パブリックストレージを使用してください。）

また、マスタ拡張IDを使用することで、使用箇所別のカスタマイズも可能です。
マスタ拡張IDを使用する場合は、MasterBuilder定義ファイルの名称が「%マスタID_%マスタ拡張ID%.xml」となる様に設定します。

例）経費区分マスタメンテナンスで使用する、勘定科目マスタ検索部品の検索結果に予備項目Aを追加

- intra-mart Accel Kaiden!の勘定科目マスタ設定の配置先
「%システムストレージパス%/kaiden/generic/master/account.xml」
- カスタマイズの勘定科目マスタ設定の配置先（予備項目Aを表示するように設定変更）
「%パブリックストレージパス%/kaiden/generic/master/account_exp_type.xml」

コラム

マスタIDとは、構築対象のマスタを一意に判定するためのIDです。
マスタ拡張IDとは、構築したマスタ機能の拡張を行う場合に使用するIDです。
詳細は、「[マスタ拡張IDの利用](#)」を参照してください。

マスタ特性

MasterBuilderでは、基本設定で設定するサービスクラスに関連するEntityに保持する項目を参照し、各マスタの特性を判断します。
判断した特性に合わせて、様々な処理を行います。

特性	判定方法	主な利用用途
会社化	Entityに会社コード(companyCd)が存在	会社認可判定、ジョブの処理単位
期間化	Entityに期間コード(termCd)が存在	期間一覧表示、期間処理（有効無効）
国際化	EntityにロケールID(localeId)が存在	ロケール分割

コラム

マスタの特性に応じて、設定が必要なコラムがあります。
詳細は、「[テーブル定義](#)」を参照してください。

基本設定

基本設定では、MasterBuilder定義の基本となる値を設定します。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">
  <masterId>account</masterId>
  <surfaceKey>master_name</surfaceKey>
  <service>k30mAccountService</service>
  <resources>service://kaiden/master/account</resources>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- **マスタID (genericMasterSettings / masterId) : 必須**
定義を行うマスタのIDを設定します。（MasterBuilder定義のファイル名と必ず同一の値を設定します。）
未設定の場合、MasterBuilder定義読み込み時に読み込みエラーが発生します。
- **サーフィスキー (genericMasterSettings / surfaceKey) : 必須**
定義を行うマスタの名称（サーフィスキー）を設定します。（実際のサーフィスは、後述のサーフィス定義で設定します。）
未設定の場合、MasterBuilder定義読み込み時に読み込みエラーが発生します。
- **サービスクラス (genericMasterSettings / service) : 任意**
定義を行うマスタのサービスクラス（コンポーネント名）を設定します。
標準設定のまま各機能を使用する場合は設定必須です。（各機能の呼出し時にエラーが発生します。）

コラム

APIを使用した検索

検索部品やセレクトボックス、バリデーション機能のみを使用し、APIによる検索を行う場合はサービスクラスの設定は不要です。
後述する拡張設定でLogicクラスを拡張し、APIを使用した検索処理を追加してください。

- **認可リソース (genericMasterSettings / resources) : 任意**
定義を行うマスタの認可リソース（URL）を設定します。
マスタメンテナンスの各処理時に使用しますので、マスタメンテナンス機能を使用しない場合は設定不要です。

コラム

認可判定

MasterBuilderの認可判定は次の2段階で実行されます。

1. 「MasterBuilder : service://kaiden/generic/master」に対する認可有無
2. 設定された認可リソース（URL）に対する認可有無

会社認可の判定は本設定の有無に限らず、マスタ特性が会社化のマスタ（Entityに「companyCd」を含む）の場合に実行されます。
（表示等の処理の場合は参照権限、登録・更新・削除等の処理の場合は編集権限）

項目設定

項目設定では、各マスタ関連機能で使用する項目定義を行います。

本項目に共通的に項目設定を行うことで、各機能設定では設定の簡略化を実現しています。

項目設定は、「genericMasterSettings / items / item」階層に設定を行います。

「genericMasterSettings / items / item」階層は、1件以上の複数設定が可能です。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <items>

    <!-- 略 -->

    <!-- 勘定科目コード -->
    <item name="accountCd" surfaceKey="account_code" type="TEXT_CODE" primary="true">
      <args>
        <arg key="style" value="width:300px;"/>
        <arg key="maxlength" value="100"/>
      </args>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'kaidenId' }"/>
        <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'100' }"/>
      </validates>
    </item>

    <!-- 勘定科目名 -->
    <item name="accountName" surfaceKey="account_name" type="TEXT_LOCALES">
      <args>
        <arg key="style" value="width:300px;"/>
        <arg key="maxlength" value="250"/>
        <arg key="imeMode" value="on"/>
      </args>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
      </validates>
    </item>

    <!-- 略 -->

  </items>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- 項目名 (**genericMasterSettings / items / item @ name**) : 必須
定義を行う項目の項目名 (HTMLのname属性値) を設定します。
未設定の場合、MasterBuilder定義読み込み時に読み込みエラーが発生します。
- サーフィスキー (**genericMasterSettings / items / item @ surfaceKey**) : 必須
定義を行う項目の名称 (サーフィスキー) を設定します。(実際のサーフィスは、後述のラベル定義で設定します。)
未設定の場合、MasterBuilder定義読み込み時に読み込みエラーが発生します。
- プライマリーキー (**genericMasterSettings / items / item @ primary**) : 任意
定義を行う項目がプライマリーキーか否かを設定します。(true:プライマリーキー、false:非プライマリーキー)
- エレメントタイプ (**genericMasterSettings / items / item @ type**) : 任意
定義を行う項目のエレメントタイプを設定します。
設定値に応じて、指定可能なエレメント引数が変わります。
エレメントタイプと設定可能なエレメント属性は「[指定可能なエレメントタイプと属性値](#)」を参照してください。
- エレメント属性 (**genericMasterSettings / items / item / args / arg**) : 任意
定義を行う項目のエレメントタイプに合わせた属性値を設定します。
- エレメント属性キー (**genericMasterSettings / items / item / args / arg @ key**) : 必須
属性値のキーとなる項目です。エレメント引数の値と合わせて設定します。
- エレメント属性値 (**genericMasterSettings / items / item / args / arg @ value**) : 任意
属性値の値となる項目です。エレメント引数のキーと合わせて設定します。
- バリデート設定 (**genericMasterSettings / items / item / validates / validate**) : 任意
定義を行う項目のバリデート方法を設定します。
- バリデートタイプ (**genericMasterSettings / items / item / validates / validate @ type**) : 必須
定義を行う項目のバリデート実行タイミングを設定します。
設定可能なバリデートの種類は次の通りです。

設定値	解説
DEFAULT	デフォルトバリデートです。登録、更新、インポートの各バリデート時に合わせて実行します。

ENTRY	メンテナンス機能で新規登録を行う際に実行されるバリデートです。
RENEW	メンテナンス機能で更新を行う際に実行されるバリデートです。
DELETE	メンテナンス機能で削除を行う際に実行されるバリデートです。
IMPORT	インポート時に実行されるバリデートです。

- バリデート引数 (**genericMasterSettings / items / item / validates / validate @ args**) : 必須
定義を行う項目のバリデート引数をJSON形式で設定します。
「type」にはバリデーションIDを設定します。
「args」にはバリデーションの引数を設定します。(MasterBuilderバリデーションの場合のみMap形式となる様に設定します。)
設定可能なバリデートは、『[intra-mart Accel Kaiden! GadgetBuilder 設定ガイド](#)』-「リファレンス」-「バリデート処理」を参照してください。

例) 必須チェック

```
<validate type="DEFAULT" args="{ 'type':'required' }"/>
```

例) 最大文字数

```
<validate type="DEFAULT" args="{ 'type':'maxLength', 'args':{'250'}}"/>
```

例) 勘定科目マスタチェック (MasterBuilderバリデーション)

```
<validate type="DEFAULT" args="{ 'type':'masterBuilderExists', 'args':{'masterId':'tax_type', 'extensionId':'exp_type', 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate', 'taxTypeCd':'taxTypeCd', 'includeDisabled':'true'}}"/>
```

- コンバート設定 (**genericMasterSettings / items / item / convert**) : 任意
画面とロジックの間で行う値のコンバート形式を設定します。
コンバート内容の詳細は「[項目値のコンバート](#)」を参照してください。
- HTMLコンバート (**genericMasterSettings / items / item / convert @ display**) : 必須
Javaのロジックで取得した値 (DBから取得した値) から画面で使用するための値にコンバートする方法を設定します。
- Javaコンバート (**genericMasterSettings / items / item / convert @ process**) : 必須
画面からのリクエストデータをJavaのロジックで使用するための値 (DB登録・更新するための値) にコンバートする方法を設定します。

拡張設定

拡張設定では、MasterBuilderの標準機能を拡張 (カスタマイズ) する場合の拡張先の設定を行います。
拡張 (カスタマイズ) を行わない場合は設定不要です。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <extension>
    <logic>jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSearchLogic</logic>
  </extension>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- **Logic**クラス (**genericMasterSettings / extension / logic**) : 任意
拡張用のLogicクラスを設定します。
未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。
マスタ検索、バリデーションのみを使用する場合は、次のLogicクラスを設定します。
「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSearchLogic」
指定できる拡張用クラスについて「[マスタ検索、バリデーションのみを使用するマスタの作成](#)」を併せて参照してください。
- **Manager**クラス (**genericMasterSettings / extension / manager**) : 任意
拡張用のManagerクラスのコンポーネント名を設定します。
未設定の場合は「masterManager」のコンポーネントを使用します。
- **Converter**クラス (**genericMasterSettings / extension / converter**) : 任意
拡張用のConverterクラスを設定します。
未設定の場合はConverterクラスを使用しません。
Converterクラスを作成する際は、以下クラスを継承してください。
「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.converter.AbstractMasterConverter」
- **一覧検索用SQL** (**genericMasterSettings / extension / searchSql**) : 任意
拡張用の一覧検索用SQLパスを設定します。(マスタメンテナンスの一覧検索で使用するSQL)
未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。
- **対象検索用SQL** (**genericMasterSettings / extension / selectSql**) : 任意
拡張用の対象検索用SQLパスを設定します。(マスタメンテナンスの更新対象検索で使用するSQL)

未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。

- **期間検索用SQL (genericMasterSettings / extension / termSql) : 任意**
拡張用の期間検索用SQLパスを設定します。(マスタメンテナンスの期間検索で使用するSQL)
未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。
- **マスタ検索用SQL (genericMasterSettings / extension / masterSql) : 任意**
拡張用のマスタ検索用SQLパスを設定します。(マスタ検索で使用するSQL)
未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。
- **エクスポート用SQL (genericMasterSettings / extension / exportSql) : 任意**
拡張用のエクスポート用SQLパスを設定します。(エクスポートの対象抽出で使用するSQL)
未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。
- **マスタメンテナンス画面のJSPパス (genericMasterSettings / extension / mainteJsp) : 任意**
拡張用のマスタメンテナンス画面のJSPパスを設定します。(マスタメンテナンス画面のメインJSP)
未設定の場合は「/kaiden/v2/base/generic/master/maintenance.jsp」を使用します。
- **マスタメンテナンス画面(検索部品)のJSPパス (genericMasterSettings / extension / mainteSearchJsp) : 任意**
拡張用のマスタメンテナンス画面(検索部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceSearch.jsp」を使用します。
- **マスタメンテナンス画面(一覧部品)のJSPパス (genericMasterSettings / extension / mainteListJsp) : 任意**
拡張用のマスタメンテナンス画面(一覧部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceList.jsp」を使用します。
- **マスタメンテナンス画面(期間部品)のJSPパス (genericMasterSettings / extension / mainteTermJsp) : 任意**
拡張用のマスタメンテナンス画面(期間部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceTerm.jsp」を使用します。
- **マスタメンテナンス画面(編集部品)のJSPパス (genericMasterSettings / extension / mainteEditJsp) : 任意**
拡張用のマスタメンテナンス画面(編集部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceEdit.jsp」を使用します。
- **マスタメンテナンス画面(ボタン部品)のJSPパス (genericMasterSettings / extension / mainteButtonJsp) : 任意**
拡張用のマスタメンテナンス画面(ボタン部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceButton.jsp」を使用します。
- **マスタメンテナンス画面(拡張部品)のJSPパス (genericMasterSettings / extension / mainteExtensionJsp) : 任意**
拡張用のマスタメンテナンス画面(拡張部品)のJSPパスを設定します。(メインJSPでインクルードして使用します。)
未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceExtension.jsp」を使用します。
- **マスタメンテナンス用の処理URL (genericMasterSettings / extension / mainteUrl) : 任意**
拡張用のマスタメンテナンス画面からのAjaxパスを設定します。
未設定の場合は「kaiden/v2/generic/master」を使用します。
- **検索用の処理URL (genericMasterSettings / extension / searchUrl) : 任意**
拡張用の検索機能からのAjaxパスを設定します。
未設定の場合は「kaiden/v2/generic/master/masterSearch」を使用します。
- **検索結果画面のJSPパス (genericMasterSettings / extension / searchJsp) : 任意**
拡張用の検索結果画面のJSPパスを設定します。
未設定の場合は「/kaiden/v2/base/foundation/master_search/genericMasterSearch.jsp」を使用します。
「/kaiden/v2/base/foundation/master_search/genericMasterSearch_v02.jsp」を設定すると、次のように検索結果画面にページングが表示されます。

No	経費区分コード	経費区分名	勘定科目	税区分	ゾートキー
1	20100002	近距離交通費	旅費交通費	課税(10%)	20100002
2	20100003	国内出張	旅費交通費	課税(10%)	20100003
3	20100004	国内宿泊日当	旅費交通費	課税(10%)	20100004
4	20100005	国内宿泊代・実費精算	旅費交通費	課税(10%)	20100005
5	30100001	海外出張 国内移動	旅費交通費	課税(10%)	30100001
6	30100002	海外出張 航空費	旅費交通費	課税(10%)	30100002
7	30100003	海外宿泊日当	旅費交通費	不課税	30100003
8	30100008	海外出張 海外移動	旅費交通費	不課税	30100008
9	30100009	海外宿泊代・実費精算	旅費交通費	不課税	30100009
10	30200001	社内研修時の旅費	教育訓練費	課税(10%)	30200001

1 - 10 / 32 件

最初へ 前へ 1 2 3 4 次へ 最後へ

- **セレクトボックス用の処理URL (genericMasterSettings / extension / selectUrl) : 任意**
拡張用のセレクトボックス機能からのAjaxパスを設定します。
未設定の場合は「kaiden/v2/generic/master/masterSelect」を使用します。

サーフィス設定

サーフィス設定では、画面やメッセージなどに使用するサーフィスの設定を行います。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <surfaces>

    <!-- 略 -->

    <surface surfaceKey="account_code" surface="Account Code" >
      <name localeId="ja" surface="勘定科目コード" />
      <name localeId="en" surface="Account Code" />
      <name localeId="zh_CN" surface="[]科目代[]" />
    </surface>
    <surface surfaceKey="account_name" surface="Account Name" >
      <name localeId="ja" surface="勘定科目名" />
      <name localeId="en" surface="Account Name" />
      <name localeId="zh_CN" surface="[]科目名" />
    </surface>

    <!-- 略 -->

  </surfaces>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- サーフィスキー (**genericMasterSettings / surfaces / surface @ surfaceKey**) : 必須
サーフィスキーを設定します。
- サーフィス (**genericMasterSettings / surfaces / surface @ surface**) : 必須
デフォルトのサーフィスを設定します。(後述の言語設定に対象のロケールが存在しない場合に使用します。)
- 言語設定 (**genericMasterSettings / surfaces / surface / name**) : 任意
各言語(ロケール)に合わせた設定を行います。
- 言語設定・ロケール (**genericMasterSettings / surfaces / surface / name @ localeId**) : 必須
言語設定を行うロケールを設定します。
- 言語設定・サーフィス (**genericMasterSettings / surfaces / surface / name @ surface**) : 必須
各言語(ロケール)に合わせたサーフィスを設定します。

メンテナンス設定

メンテナンス設定ではマスタメンテナンス機能で使用する項目の設定を行います。
マスタメンテナンス機能を使用しない場合は設定不要です。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <mainte>
    <listItems>
      <item name="accountCd" width="250px"/>
      <item name="accountName"/>
    </listItems>
    <editItems>
      <item name="accountCd"/>
      <item name="accountName"/>
    </editItems>
  </mainte>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- メンテナンス(一覧)項目設定 (**genericMasterSettings / mainte / listItems / item**) : 任意
一覧画面の項目を設定します。
- メンテナンス(一覧)項目設定・項目名 (**genericMasterSettings / mainte / listItems / item @ name**) : 必須
一覧画面に表示する項目名(項目設定で設定した項目名)を設定します。
- メンテナンス(一覧)項目設定・表示幅 (**genericMasterSettings / mainte / listItems / item @ width**) : 任意
一覧画面に表示した際の表示幅を設定します。
未設定の場合は、表示幅の指定を行いません。
- メンテナンス(一覧)項目設定・表示位置 (**genericMasterSettings / mainte / listItems / item @ align**) : 任意
一覧画面に表示した際の表示位置(left,center,right)を設定します。

未設定の場合は、左寄せ（left）を設定します。

- メンテナンス（一覧）項目設定・ソート有無（**genericMasterSettings / mainte / listItems / item @ sort**）：任意
一覧画面でのソート有無（true,false）を設定します。
未設定の場合は、ソート有（true）を設定します。
- メンテナンス（一覧）項目設定・ソート方法（**genericMasterSettings / mainte / listItems / item @ order**）：任意
一覧画面での初期ソート方法（asc,desc）を設定します。
未設定の場合は、昇順（asc）を設定します。
- メンテナンス（編集）項目設定（**genericMasterSettings / mainte / editItems / item**）：任意
編集画面の項目を設定します。
- メンテナンス（編集）項目設定・項目名（**genericMasterSettings / mainte / editItems / item @ name**）：必須
編集画面に表示する項目名（項目設定で設定した項目名）を設定します。
- メンテナンス（編集）項目設定・結合（**genericMasterSettings / mainte / editItems / item @ join**）：任意
編集画面に表示する項目のうち、結合を行う場合の結合キーを設定します。
未設定の場合は、項目の結合を行いません。



コラム

結合キーとは

結合設定を行わない場合は、各項目がそれぞれの行として表示されます。

結合を行う項目に同じ結合キーを設定することで、複数項目を一つの行として表示されます。

この時、項目名として表示されるサーフィスは、設定順で一番初めに設定された項目のサーフィスが使用されます。

例）予備項目A、予備項目Bの結合

結合キー（**extension**）を設定した場合

予備項目A	

結合キーを設定しない場合

予備項目A	
予備項目B	

- マスタメンテナンス画面（一覧部品）の表示件数（**genericMasterSettings / mainte / limit**）：任意
一覧画面の表示件数を設定します。
未設定の場合は、20件を設定します。
- マスタメンテナンス画面の登録・更新後の一覧画面遷移有無（**genericMasterSettings / mainte / truns**）：任意
登録・更新処理後の一覧画面遷移有無（true,false）を設定します。
未設定の場合は、遷移しない（false）を設定します。
- マスタメンテナンス画面の一覧画面遷移時の再検索有無（**genericMasterSettings / mainte / reload**）：任意
登録・更新処理後、一覧画面に遷移した際の再検索有無（true,false）を設定します。
未設定の場合は、再検索する（true）を設定します。

検索設定

検索設定では、検索部品やセレクトボックスで使用する項目の設定を行います。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <search type="SELECT">
    <items>
      <item name="accountCd" type="CODE"/>
      <item name="accountName" type="NAME" width="250px"/>
    </items>
  </search>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- 検索タイプ（**genericMasterSettings / search @ type**）：必須
検索タイプの初期値を設定します。
 - **SEARCH**
初期値を検索部品とする場合に設定します。

No	勘定科目コード	勘定科目名
1	31000010	給料賃金
2	31000012	修繕費
3	31000013	通信費
4	31000014	会議費
5	31000015	広告宣伝費
6	31000016	販売管理費
7	31000017	接待交際費
8	31000018	荷造運賃
9	31000019	福利厚生費

■ SELECT

初期値をセレクトボックス（リストデータ）とする場合に設定します。

■ RADIO

初期値をラジオボタンとする場合に設定します。

検索結果に紐づくデータがラジオボタンですべて表示されるため、検索結果が少数で固定のマスタに対して使用することを推奨します。

- 検索項目設定 (**genericMasterSettings / search / items / item**) : 任意
検索に使用する項目を設定します。
- 項目名 (**genericMasterSettings / search / items / item @ name**) : 必須
検索に使用する項目名（項目設定で設定した項目名）を設定します。
- 表示幅 (**genericMasterSettings / search / items / item @ width**) : 任意
検索部品での検索結果表示時の表示幅を設定します。
- 表示位置 (**genericMasterSettings / search / items / item @ align**) : 任意
検索部品での検索結果表示時の表示位置（left,center,right）を設定します。
未設定の場合は、左寄せ（left）を設定します。
- 項目タイプ (**genericMasterSettings / search / items / item @ type**) : 任意
セレクトボックス、ラジオボタンでの項目タイプを設定します。
 - CODE
セレクトボックス、ラジオボタンのコード値とする項目に設定します。
 - NAME
セレクトボックス、ラジオボタンの表示値とする項目に設定します。
- 検索結果の表示件数 (**genericMasterSettings / search / searchLimit**) : 任意
検索部品を使用した検索を行う場合の表示件数を設定します。

i コラム

検索結果の表示件数について

MasterBuilderで定義されている全マスタの表示件数を変更したい場合

『intra-mart Accel Kaiden! プログラミングガイド』-「応用」-「その他」-「機能説明」を参照して定数ファイルの「マスタ検索部品（PC版）の表示上限件数」と「マスタ検索部品（SP版）の表示上限件数」を変更してください。

PC版

検索結果の表示件数は以下の優先順位で設定します。

1. 検索結果の表示件数を設定した場合
検索結果の表示件数を設定します。
2. 定数ファイルの「マスタ検索部品（PC版）の表示上限件数」の値を設定した場合
検索結果の表示件数を「マスタ検索部品（PC版）の表示上限件数」に設定します。
3. いずれも未設定の場合
10件を設定します。

SP版

検索結果の表示件数は以下の優先順位で設定します。

1. 検索結果の表示件数を設定した場合
検索結果の表示件数を設定します。
2. 定数ファイルの「マスタ検索部品（SP版）の表示上限件数」の値を設定した場合
検索結果の表示件数を「マスタ検索部品（SP版）の表示上限件数」に設定します。
3. いずれも未設定の場合
20件を設定します。

インポート設定

インポート設定では、インポートジョブで使用する項目の設定を行います。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <importJob>
    <items>
      <item name="companyCd" />
      <item name="accountCd" />
      <item name="startDate" />
      <item name="endDate" />
      <item name="deleteFlag" />
      <item name="localeId" />
      <item name="accountName" />
    </items>
  </importJob>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- インポートジョブ項目設定（**genericMasterSettings / importJob / items / item**）：任意
インポートジョブで使用する項目を設定します。
- インポートジョブ項目設定・項目名（**genericMasterSettings / importJob / items / item @ name**）：任意
インポートジョブで使用する項目名（項目設定で設定した項目名）を設定します。
設定順に応じてインポートファイルの列順序が変動します。
詳細は、「[インポートファイルの列順序判定について](#)」を参照してください。
- インポートジョブパラメータ（**genericMasterSettings / importJob / parameters / parameter**）：任意
ジョブパラメータを設定します。
詳細は、「[指定可能なジョブパラメータ](#)」を参照してください。
- インポートジョブパラメータキー（**genericMasterSettings / importJob / parameters / parameter @ key**）：任意
ジョブパラメータのキーを設定します。
- インポートジョブパラメータ値（**genericMasterSettings / importJob / parameters / parameter @ value**）：任意
ジョブパラメータの値を設定します。

エクスポート設定

エクスポート設定では、エクスポートジョブで使用する項目の設定を行います。

設定


```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 略 -->

  <exportJob>
    <items>
      <item name="companyCd" />
      <item name="accountCd" />
      <item name="startDate" />
      <item name="endDate" />
      <item name="deleteFlag" />
      <item name="localeId" />
      <item name="accountName" />
    </items>
  </exportJob>

  <!-- 略 -->

</genericMasterSettings>
```

解説

- エクスポートジョブ項目設定 (**genericMasterSettings / exportJob / items / item**) : 任意
エクスポートジョブで使用する項目を設定します。
- エクスポートジョブ項目設定・項目名 (**genericMasterSettings / exportJob / items / item @ name**) : 任意
エクスポートジョブで使用する項目名（項目設定で設定した項目名）を設定します。
設定順でエクスポートファイルのカラム順とします。
- エクスポートジョブ項目設定・ソート有無 (**genericMasterSettings / exportJob / items / item @ sort**) : 任意
エクスポートデータのソート有無 (true,false) を設定します。
未設定の場合は、ソート有 (true) を設定します。
- エクスポートジョブ項目設定・ソート方法 (**genericMasterSettings / exportJob / items / item @ order**) : 任意
エクスポートデータのソート方法 (asc,desc) を設定します。
未設定の場合は、昇順 (asc) を設定します。
- エクスポートジョブパラメータ (**genericMasterSettings / exportJob / parameters / parameter**) : 任意
ジョブパラメータを設定します。
詳細は、「[指定可能なジョブパラメータ](#)」を参照してください。
- エクスポートジョブパラメータキー (**genericMasterSettings / exportJob / parameters / parameter @ key**) : 任意
ジョブパラメータのキーを設定します。
- エクスポートジョブパラメータ値 (**genericMasterSettings / exportJob / parameters / parameter @ value**) : 任意
ジョブパラメータの値を設定します。

Sample

次のMasterBuilder定義は勘定科目マスタの設定を行った場合のサンプルです。

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">
  <!-- 勘定科目 -->
  <masterId>account</masterId>
  <surfaceKey>master_name</surfaceKey>
  <service>k30mAccountService</service>
  <resources>service://kaiden/master/account</resources>

  <!-- 項目設定 -->
  <items>
    <!-- 会社コード -->
    <item name="companyCd" surfaceKey="company_code" primary="true">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'companyExists', 'args':{'companyCd','searchCriteriaDate','true'}}"/>
      </validates>
    </item>
    <!-- 会社名 -->
    <item name="companyName" surfaceKey="company_name"/>
    <!-- 期間コード -->
    <item name="termCd" surfaceKey="term_code" primary="true"/>
    <!-- 開始日 -->
    <item name="startDate" surfaceKey="start_date">
      <convert display="STRING_DATE" process="TIMESTAMP"/>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'date' }"/>
      </validates>
    </item>
    <!-- 終了日 -->
    <item name="endDate" surfaceKey="end_date">
      <convert display="STRING_DATE" process="TIMESTAMP_SHIFT"/>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
      </validates>
    </item>
  </items>

```

```

<validate type="DEFAULT" args="{ 'type':'date' }"/>
<validate type="DEFAULT" args="{ 'type':'dateMin', 'args':'startDate' }"/>
</validates>
</item>
<!-- 削除フラグ -->
<item name="deleteFlag" surfaceKey="delete_flag">
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'flag' }"/>
  </validates>
</item>
<!-- ロケールID -->
<item name="localeId" surfaceKey="locale_id" primary="true">
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'locale' }"/>
  </validates>
</item>

<!-- 勘定科目コード -->
<item name="accountCd" surfaceKey="account_code" type="TEXT_CODE" primary="true">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="100"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'kaidenId' }"/>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'100' }"/>
  </validates>
</item>
<!-- 勘定科目名 -->
<item name="accountName" surfaceKey="account_name" type="TEXT_LOCALES">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>

<!-- 予備項目A -->
<item name="extensionA" surfaceKey="extension_a" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目B -->
<item name="extensionB" surfaceKey="extension_b" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目C -->
<item name="extensionC" surfaceKey="extension_c" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目D -->
<item name="extensionD" surfaceKey="extension_d" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>

```

```

</validates>
</item>
<!-- 予備項目E -->
<item name="extensionE" surfaceKey="extension_e" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目F -->
<item name="extensionF" surfaceKey="extension_f" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目G -->
<item name="extensionG" surfaceKey="extension_g" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目H -->
<item name="extensionH" surfaceKey="extension_h" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目I -->
<item name="extensionI" surfaceKey="extension_i" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目J -->
<item name="extensionJ" surfaceKey="extension_j" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 登録日 -->
<item name="entryTs" surfaceKey="entry_ts"><convert display="STRING_TIMESTAMP" process="TIMESTAMP"/></item>
<!-- 登録者 -->
<item name="entryUserCd" surfaceKey="entry_user_cd"/>
<!-- 更新カウント -->
<item name="renewCnt" surfaceKey="renew_cnt"><convert display="STRING" process="LONG"/></item>
<!-- 更新日 -->
<item name="renewTs" surfaceKey="renew_ts"><convert display="STRING_TIMESTAMP" process="TIMESTAMP"/></item>
<!-- 更新者 -->
<item name="renewUserCd" surfaceKey="renew_user_cd"/>
</items>

<!-- メンテナンス設定 -->
<mainte>
  <listItems>
    <item name="accountCd" width="250px"/>
    <item name="accountName"/>
  </listItems>
  <listItems>

```

```

<editItems>
  <item name="accountCd"/>
  <item name="accountName"/>
</editItems>
<truns>true</truns>
<reload>true</reload>
</mainte>

<!-- 検索設定 -->
<search type="SEARCH">
  <items>
    <item name="accountCd" type="CODE"/>
    <item name="accountName" type="NAME" width="250px"/>
  </items>
</search>

<!-- インポート設定 -->
<importJob>
  <items>
    <item name="companyCd" />
    <item name="accountCd" />
    <item name="localeId" />
    <item name="startDate" />
    <item name="endDate" />
    <item name="accountName" />
    <item name="deleteFlag" />
  </items>
</importJob>

<!-- エクスポート設定 -->
<exportJob>
  <items>
    <item name="companyCd" />
    <item name="accountCd" />
    <item name="localeId" />
    <item name="startDate" />
    <item name="endDate" />
    <item name="accountName" />
    <item name="deleteFlag" />
  </items>
</exportJob>

<!-- ラベル設定 -->
<surfaces>
  <surface surfaceKey="master_name" surface="Account">
    <name localeId="ja" surface="勘定科目" />
    <name localeId="en" surface="Account" />
    <name localeId="zh_CN" surface="科目" />
  </surface>

  <surface surfaceKey="company_code" surface="Company Code">
    <name localeId="ja" surface="会社コード" />
    <name localeId="en" surface="Company Code" />
    <name localeId="zh_CN" surface="公司代" />
  </surface>

  <surface surfaceKey="company_name" surface="Company Name">
    <name localeId="ja" surface="会社名" />
    <name localeId="en" surface="Company Name" />
    <name localeId="zh_CN" surface="公司名" />
  </surface>

  <surface surfaceKey="term_code" surface="Term Code">
    <name localeId="ja" surface="期間コード" />
    <name localeId="en" surface="Term Code" />
    <name localeId="zh_CN" surface="期代" />
  </surface>

  <surface surfaceKey="start_date" surface="Start date">
    <name localeId="ja" surface="開始日" />
    <name localeId="en" surface="Start date" />
    <name localeId="zh_CN" surface="开始日" />
  </surface>

  <surface surfaceKey="end_date" surface="End Date">
    <name localeId="ja" surface="終了日" />
    <name localeId="en" surface="End Date" />
    <name localeId="zh_CN" surface="结束日" />
  </surface>

  <surface surfaceKey="delete_flag" surface="Delete Flag">
    <name localeId="ja" surface="削除フラグ" />
    <name localeId="en" surface="Delete Flag" />
    <name localeId="zh_CN" surface="删除志" />
  </surface>

  <surface surfaceKey="locale_id" surface="Locale ID">
    <name localeId="ja" surface="ローカルID" />
    <name localeId="en" surface="Locale ID" />
    <name localeId="zh_CN" surface="区域ID" />
  </surface>

  <surface surfaceKey="account_code" surface="Account Code">

```

```

<surface surfaceKey="account_code" surface="Account Code">
  <name localeId="ja" surface="勘定科目コード" />
  <name localeId="en" surface="Account Code" />
  <name localeId="zh_CN" surface="[]科目代[]" />
</surface>
<surface surfaceKey="account_name" surface="Account Name">
  <name localeId="ja" surface="勘定科目名" />
  <name localeId="en" surface="Account Name" />
  <name localeId="zh_CN" surface="[]科目名" />
</surface>

<surface surfaceKey="extension_a" surface="Preliminary Item A">
  <name localeId="ja" surface="予備項目A" />
  <name localeId="en" surface="Preliminary Item A" />
  <name localeId="zh_CN" surface="[]項目A" />
</surface>
<surface surfaceKey="extension_b" surface="Preliminary Item B">
  <name localeId="ja" surface="予備項目B" />
  <name localeId="en" surface="Preliminary Item B" />
  <name localeId="zh_CN" surface="[]項目B" />
</surface>
<surface surfaceKey="extension_c" surface="Preliminary Item C">
  <name localeId="ja" surface="予備項目C" />
  <name localeId="en" surface="Preliminary Item C" />
  <name localeId="zh_CN" surface="[]項目C" />
</surface>
<surface surfaceKey="extension_d" surface="Preliminary Item D">
  <name localeId="ja" surface="予備項目D" />
  <name localeId="en" surface="Preliminary Item D" />
  <name localeId="zh_CN" surface="[]項目D" />
</surface>
<surface surfaceKey="extension_e" surface="Preliminary Item E">
  <name localeId="ja" surface="予備項目E" />
  <name localeId="en" surface="Preliminary Item E" />
  <name localeId="zh_CN" surface="[]項目E" />
</surface>
<surface surfaceKey="extension_f" surface="Preliminary Item F">
  <name localeId="ja" surface="予備項目F" />
  <name localeId="en" surface="Preliminary Item F" />
  <name localeId="zh_CN" surface="[]項目F" />
</surface>
<surface surfaceKey="extension_g" surface="Preliminary Item G">
  <name localeId="ja" surface="予備項目G" />
  <name localeId="en" surface="Preliminary Item G" />
  <name localeId="zh_CN" surface="[]項目G" />
</surface>
<surface surfaceKey="extension_h" surface="Preliminary Item H">
  <name localeId="ja" surface="予備項目H" />
  <name localeId="en" surface="Preliminary Item H" />
  <name localeId="zh_CN" surface="[]項目H" />
</surface>
<surface surfaceKey="extension_i" surface="Preliminary Item I">
  <name localeId="ja" surface="予備項目I" />
  <name localeId="en" surface="Preliminary Item I" />
  <name localeId="zh_CN" surface="[]項目I" />
</surface>
<surface surfaceKey="extension_j" surface="Preliminary Item J">
  <name localeId="ja" surface="予備項目J" />
  <name localeId="en" surface="Preliminary Item J" />
  <name localeId="zh_CN" surface="[]項目J" />
</surface>
<surface surfaceKey="entry_ts" surface="Entry Date">
  <name localeId="ja" surface="登録日" />
  <name localeId="en" surface="Entry Date" />
  <name localeId="zh_CN" surface="注册日" />
</surface>
<surface surfaceKey="entry_user_cd" surface="Entry User">
  <name localeId="ja" surface="登録者" />
  <name localeId="en" surface="Entry User" />
  <name localeId="zh_CN" surface="注册者" />
</surface>
<surface surfaceKey="renew_cnt" surface="Renew Count">
  <name localeId="ja" surface="更新カウント" />
  <name localeId="en" surface="Renew Count" />
  <name localeId="zh_CN" surface="更新[]数" />
</surface>
<surface surfaceKey="renew_ts" surface="Renew Date">
  <name localeId="ja" surface="更新日" />
  <name localeId="en" surface="Renew Date" />
  <name localeId="zh_CN" surface="更新日" />
</surface>
<surface surfaceKey="renew_user_cd" surface="Renew User">
  <name localeId="ja" surface="更新者" />
  <name localeId="en" surface="Renew User" />
  <name localeId="zh_CN" surface="更新者" />
</surface>
</surfaces>

```

</src/main/resources/>
</genericMasterSettings>

検索用SQL

本項では、MasterBuilderで使用するSQLの作成方法を説明します。

- 概要
- 一覧検索用SQL
- 対象検索用SQL
- 期間検索用SQL
- マスタ検索用SQL
- エクスポート用SQL
- Sample

概要

MasterBuilderでSQLを使用する処理は次の通りです。

- マスタメンテナンス機能（一覧部品）の一覧表示データ抽出
- マスタメンテナンス機能（編集部品）の更新対象データ抽出
- マスタメンテナンス機能（期間部品）の更新対象期間抽出
- マスタ検索部品やセレクトボックスへのデータ抽出
- エクスポート機能のエクスポート対象データ抽出

「[拡張設定](#)」を使用することで、上記それぞれの処理用のSQLを作成することも可能ですが、カスタマイズ性や保守性を考慮し、単一SQLでの実現を推奨します。

「[拡張設定](#)」を使用しない場合のMasterBuilderで使用するSQLの配置先は次の通りです。

%ModuleProject%/src/main/resources/META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql

（マスタIDには、MasterBuilder定義で設定したマスタIDを指定してください。）

単一のSQLで各検索処理を実現するために次の様な機能を用意しています。

- **検索基準となる値の補完**
検索基準日・検索基準ロケール・検索基準ユーザはどの処理でも補完し、SQLに引き渡されます。

名称	SQLで使用する値	補完順
検索基準日	searchCriteriaDate	リクエストパラメータ > システム日付
検索基準ロケール	searchCriteriaLocale	リクエストパラメータ > ログインユーザのロケール（ジョブの場合はシステムロケール）
検索基準ユーザ	searchCriteriaUser	リクエストパラメータ > ログインユーザ

- **検索用途の連携**
各検索用途に応じた値を「searchType」に設定してSQLに引き渡されます。
これにより、Where句のIF分岐が可能となります

対象	SQLで使用する値	設定値
一覧検索	searchType	search
対象検索	searchType	select
期間検索	searchType	term
マスタ検索	searchType	master
エクスポート	searchType	export

一覧検索用SQL

一覧検索用のSQLでは、メンテナンス機能の一覧画面に表示するマスタデータを抽出します。

一覧表示時に単一のロケールデータを表示する必要がある為、国際化されたマスタの場合は注意が必要です。

また、本検索の結果が「[対象検索用SQL](#)」、「[期間検索用SQL](#)」のリクエストパラメータとして引き渡されます。

- 検索結果の表示先

No.	▲ 設定科目コード	設定科目名
1	31000010	給料賃金
2	31000012	修繕費
3	31000013	運賃費
4	31000014	会議費
5	31000015	広告宣伝費
6	31000016	販売管理費
7	31000017	防災対策費
8	31000018	物流運賃
9	31000019	福利厚生費
10	31000021	水道光熱費
11	31000022	消耗品費
12	31000023	印刷複写費
13	31000024	新聞図書費
14	31000025	支払報酬
15	31000030	雑費
16	31000038	教育訓練費
17	31001011	災害対策費

■ SQLに引き渡される検索条件

検索条件	設定値
searchCriteriaCompany	検索条件に設定された会社のコード値 会社化特性の場合のみ設定 非会社化特性の場合は「null」
searchCriteriaDate	検索条件に設定された検索基準日
searchCriteriaUser	ログインユーザのユーザコード
searchCriteriaLocale	検索条件に設定されたロケールのID
searchWithDisabled	検索条件に設定された無効データの検索有無 チェックなしの場合は「0」 チェックありの場合は「null」
searchCriteriaKeyword	検索条件に設定されたキーワード 未設定の場合は「null」
orderBy	一覧表示のソート条件 初回検索時は「メンテナンス設定」に設定されたソート順
searchType	固定値「search」

対象検索用SQL

対象検索用のSQLでは、メンテナンス機能の編集画面（編集部品）に表示するマスタデータを抽出します。

■ 検索結果の表示先

開始日	終了日
1900/01/01	2999/12/31

会社	設定科目コード	設定科目名
サンプル会社	31000010	給料賃金
		Salary and Wages
		工資薪金

■ SQLに引き渡される検索条件

対象検索時は、次の検索条件に加え、一覧検索時に返却したデータがすべて検索条件（リクエストパラメータ）となります。次の検索条件がリクエストパラメータに含まれる場合は上書きします。

検索条件	設定値
searchCriteriaDate	期間化特性、かつリクエストパラメータに「startDate」が含まれる場合はその値 上記以外はシステム日付
searchCriteriaUser	リクエストパラメータに「searchCriteriaUser」が含まれる場合はその値 上記以外はログインユーザのユーザコード
searchCriteriaLocale	国際化特性、かつリクエストパラメータに「localeId」が含まれる場合はその値 上記以外はログインユーザのロケール
searchType	固定値「select」

期間検索用SQL

期間検索用のSQLでは、メンテナンス機能の編集画面（期間部品）に表示するマスタデータを抽出します。

- 検索結果の表示先

- SQLに引き渡される検索条件

期間検索時は、次の検索条件に加え、一覧検索時に返却したデータがすべて検索条件（リクエストパラメータ）となります。次の検索条件がリクエストパラメータに含まれる場合は上書きします。

検索条件	設定値
searchCriteriaDate	期間化特性、かつリクエストパラメータに「startDate」が含まれる場合はその値 上記以外はシステム日付
searchCriteriaUser	リクエストパラメータに「searchCriteriaUser」が含まれる場合はその値 上記以外はログインユーザのユーザコード
searchCriteriaLocale	国際化特性、かつリクエストパラメータに「localeId」が含まれる場合はその値 上記以外はログインユーザのロケール
searchType	固定値「term」

マスタ検索用SQL

マスタ検索用のSQLでは、検索部品やセレクトボックスに表示するマスタデータを抽出します。

- 検索結果の表示先

- SQLに引き渡される検索条件

マスタ検索時は、次の検索条件に加え、項目の検索条件に指定されたデータがすべて検索条件（リクエストパラメータ）となります。次の検索条件がリクエストパラメータに含まれる場合は上書きします。

検索条件	設定値
searchCriteriaDate	リクエストパラメータに「searchCriteriaDate」が含まれる場合はその値 上記以外はシステム日付
searchCriteriaUser	リクエストパラメータに「searchCriteriaUser」が含まれる場合はその値 上記以外はログインユーザのユーザコード
searchCriteriaLocale	リクエストパラメータに「searchCriteriaLocale」が含まれる場合はその値 上記以外はログインユーザのロケール
deleteFlag	リクエストパラメータ「includeDisabled」に「true」以外が設定されていた場合は「0」 上記以外の場合は「null」
searchCriteriaKeyword	検索部品の場合は入力されたキーワード 上記以外、または未設定の場合は「null」
searchType	固定値「master」

エクスポート用SQL

エクスポート用のSQLでは、エクスポートを行うマスタデータを抽出します。

- SQLに引き渡される検索条件

エクスポート時は、次の検索条件に加え、「searchCriteria」から始まるジョブのパラメータが検索条件（リクエストパラメータ）となります。

検索条件	設定値
searchCriteriaCompany	会社化特性の場合は処理中の会社のコード値 非会社化特性の場合は「null」
searchCriteriaDate	システム日付
searchCriteriaUser	リクエストパラメータに「searchCriteriaUser」が含まれる場合はその値 上記以外はログインユーザのユーザコード
searchCriteriaLocale	リクエストパラメータに「searchCriteriaLocale」が含まれる場合はその値 上記以外はログインユーザのロケール
targetDate	期間モードでエクスポートを実行した場合は検索対象日 上記以外は「null」
renewTs	更新日モードでエクスポートを実行した場合は検索対象日 上記以外は「null」
searchType	固定値「export」

Sample

次のSQLは勘定科目マスタの設定を行った場合のサンプルです。

```
select
  target.company_cd
, dept.department_name as company_name
, target.account_cd
, target.term_cd
, target.locale_id
, target.account_name
, target.start_date
, target.end_date
, target.delete_flag
, target.extension_a
, target.extension_b
, target.extension_c
, target.extension_d
, target.extension_e
, target.extension_f
, target.extension_g
, target.extension_h
, target.extension_i
, target.extension_j
, target.entry_ts
, target.entry_user_cd
, target.renew_cnt
, target.renew_ts
, target.renew_user_cd
from
  k30m_account target
left join (
  select
    imm_department.company_cd
  , imm_department.department_set_cd
  , imm_department.department_cd
  , imm_department.locale_id
  , imm_department.start_date
  , imm_department.end_date
  , imm_department.department_name
  from
    imm_department
) dept
on target.company_cd = dept.company_cd
and target.company_cd = dept.department_set_cd
and target.company_cd = dept.department_cd
and target.locale_id = dept.locale_id
/*IF searchCriteriaDate != null*/
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'
/*END*/

/*IF searchType == "search"*/
/*BEGIN*/
where
  /*IF searchCriteriaCompany != null*/
  target.company_cd = /*searchCriteriaCompany*/'searchCriteriaCompany'
  /*END*/
  /*IF searchCriteriaDate != null*/
  and target.start_date <= /*searchCriteriaDate*/2000/01/01'
  and target.end_date > /*searchCriteriaDate*/2000/01/01'
  /*END*/
  /*IF searchWithDisabled != null*/
  and target.delete_flag = /*searchWithDisabled*/'searchWithDisabled'
```

```

/*END*/
/*IF searchCriteriaLocale != null*/
and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
/*END*/
/*IF searchCriteriaKeyword != null*/
and (
  target.account_cd      like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
or target.account_name   like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
)
/*END*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/account_cd
/*END*/
/*END*/

/*IF searchType == "select"*/
/*BEGIN*/
where
  /*IF accountCd != null*/
  target.account_cd = /*accountCd*/accountCd'
  /*END*/
  /*IF companyCd != null*/
  and target.company_cd = /*companyCd*/companyCd'
  /*END*/
  /*IF termCd != null*/
  and target.term_cd = /*termCd*/termCd'
  /*END*/
/*END*/
order by account_cd
/*END*/

/*IF searchType == "term"*/
/*BEGIN*/
where
  /*IF accountCd != null*/
  target.account_cd = /*accountCd*/accountCd'
  /*END*/
  /*IF companyCd != null*/
  and target.company_cd = /*companyCd*/companyCd'
  /*END*/
  /*IF localeId != null*/
  and target.locale_id = /*localeId*/localeId'
  /*END*/
/*END*/
order by account_cd, start_date
/*END*/

/*IF searchType == "master"*/
/*BEGIN*/
where
  /*IF accountCd != null*/
  target.account_cd = /*accountCd*/accountCd'
  /*END*/
  /*IF searchCriteriaCompany != null*/
  and target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
  /*END*/
  /*IF searchCriteriaDate != null*/
  and target.start_date <= /*searchCriteriaDate*/2000/01/01'
  and target.end_date   > /*searchCriteriaDate*/2000/01/01'
  /*END*/
  /*IF deleteFlag != null*/
  and target.delete_flag = /*deleteFlag*/deleteFlag'
  /*END*/
  /*IF searchCriteriaLocale != null*/
  and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
  /*END*/
  /*IF searchCriteriaKeyword != null*/
  and (
    target.account_cd      like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
  or target.account_name   like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
  )
  /*END*/
/*END*/
order by account_cd
/*END*/

/*IF searchType == "export"*/
/*BEGIN*/
where
  /*IF searchCriteriaCompany != null*/
  target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
  /*END*/
  /*IF targetDate != null*/
  and target.start_date <= /*targetDate*/2000/01/01'
  and target.end_date   > /*targetDate*/2000/01/01'

```

```
/*END*/
/*IF renewTs != null*/
and target.renew_ts >= /*renewTs*/'2000/01/01'
/*END*/
/*END*/
order by account_cd, start_date, locale_id
/*END*/
```

設定の流れ

本項では、MasterBuilderの設定の流れを説明します。

- テーブル定義
 - 必須カラム
 - 特性カラム
 - その他カラム
 - 定義サンプル
- Service、Entity作成
 - Service、Entityの作成
 - Service、Entityの配置
- MasterBuilder定義作成
 - MasterBuilder定義ファイル作成
 - 基本設定
 - 項目設定
 - 拡張設定
 - サーフィス設定
 - メンテナンス設定
 - 検索・リストデータ設定
 - インポート設定
 - エクスポート設定
 - MasterBuilder定義
- 検索用SQL作成
 - 検索項目設定
 - 一覧検索条件設定
 - 対象検索条件設定
 - 期間検索条件設定
 - マスタ検索条件設定
 - エクスポート条件設定
 - 検索用SQL
- 動作確認
 - メンテナンス機能
 - 検索部品・セレクトボックス
 - インポートジョブ
 - エクスポートジョブ

テーブル定義

はじめに、対象となるマスタテーブルの定義を行います。
要件に合わせて次の各カラムを組み合わせて定義を行ってください。

必須カラム

MasterBuilderを使用する場合は次のカラムは必ず設定してください。（全て「not null」制約）

論理名	物理名	PostgreSQL	Oracle	SqlServer
登録日時	entry_ts	timestamp without time zone	timestamp	datetime2
登録ユーザ	entry_user_cd	varchar(100)	varchar2(100)	nvarchar(100)
更新カウント	renew_cnt	numeric(18,0)	numeric(18,0)	numeric(18,0)
更新日時	renew_ts	timestamp without time zone	timestamp	datetime2
更新ユーザ	renew_user_cd	varchar(100)	varchar2(100)	nvarchar(100)

特性カラム

会社化特性

会社化特性のマスタを作成する場合は次のカラムを設定してください。（全て「not null」制約）

論理名	物理名	PostgreSQL	Oracle	SqlServer
会社コード	company_cd	varchar(100)	varchar2(100)	nvarchar(100)

期間化特性

期間化特性のマスタを作成する場合は次のカラムを設定してください。（全て「not null」制約）

論理名	物理名	PostgreSQL	Oracle	SqlServer
期間コード	term_cd	varchar(50)	varchar2(50)	nvarchar(50)
開始日	start_date	timestamp without time zone	timestamp	datetime2
終了日	end_date	timestamp without time zone	timestamp	datetime2
削除フラグ	delete_flag	varchar(1)	varchar2(1)	nvarchar(1)

国際化特性

国際化特性のマスタを作成する場合は次のカラムを設定してください。（全て「not null」制約）

論理名	物理名	PostgreSQL	Oracle	SqlServer
ロケールID	locale_id	varchar(50)	varchar2(50)	nvarchar(50)

その他カラム

拡張用の項目として次のカラムを用意することでカスタマイズ性が向上します。

次のカラムはintra-mart Accel Kaiden!が提供するすべてのマスタに設定されています。

論理名	物理名	PostgreSQL	Oracle	SqlServer
予備項目A	extension_a	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目B	extension_b	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目C	extension_c	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目D	extension_d	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目E	extension_e	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目F	extension_f	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目G	extension_g	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目H	extension_h	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目I	extension_i	varchar(1000)	varchar2(1000)	nvarchar(1000)
予備項目J	extension_j	varchar(1000)	varchar2(1000)	nvarchar(1000)

定義サンプル

次項以降では、次の定義を利用した場合の設定の流れを説明します。

- PostgreSQL

```
create table k99m_sample
(
  company_cd          varchar(100) not null,
  term_cd             varchar(50) not null,
  start_date          timestamp without time zone not null,
  end_date            timestamp without time zone not null,
  delete_flag         varchar(1) not null,
  locale_id           varchar(50) not null,
  sample_cd           varchar(100) not null,
  sample_name         varchar(1000) not null,
  extension_a         varchar(1000),
  extension_b         varchar(1000),
  extension_c         varchar(1000),
  extension_d         varchar(1000),
  extension_e         varchar(1000),
  extension_f         varchar(1000),
  extension_g         varchar(1000),
  extension_h         varchar(1000),
  extension_i         varchar(1000),
  extension_j         varchar(1000),
  entry_ts            timestamp without time zone not null,
  entry_user_cd       varchar(100) not null,
  renew_cnt           numeric(18,0) not null,
  renew_ts            timestamp without time zone not null,
  renew_user_cd       varchar(100) not null
);
alter table k99m_sample add primary key (company_cd, sample_cd, locale_id, term_cd);
```

- Oracle

```
create table k99m_sample
(
  company_cd          varchar2(100) not null,
  term_cd             varchar2(50)  not null,
  start_date          timestamp not null,
  end_date            timestamp not null,
  delete_flag         varchar2(1)  not null,
  locale_id           varchar2(50)  not null,
  sample_cd           varchar2(100) not null,
  sample_name         varchar2(1000) not null,
  extension_a         varchar2(1000),
  extension_b         varchar2(1000),
  extension_c         varchar2(1000),
  extension_d         varchar2(1000),
  extension_e         varchar2(1000),
  extension_f         varchar2(1000),
  extension_g         varchar2(1000),
  extension_h         varchar2(1000),
  extension_i         varchar2(1000),
  extension_j         varchar2(1000),
  entry_ts            timestamp not null,
  entry_user_cd       varchar2(100) not null,
  renew_cnt           numeric(18,0) not null,
  renew_ts            timestamp not null,
  renew_user_cd       varchar2(100) not null
);
alter table k99m_sample add primary key (company_cd, sample_cd, locale_id, term_cd);
```

- SqlServer

```
create table k99m_sample
(
  company_cd          nvarchar(100) not null,
  term_cd             nvarchar(50)  not null,
  start_date          datetime2 not null,
  end_date            datetime2 not null,
  delete_flag         nvarchar(1)  not null,
  locale_id           nvarchar(50)  not null,
  sample_cd           nvarchar(100) not null,
  sample_name         nvarchar(1000) not null,
  extension_a         nvarchar(1000),
  extension_b         nvarchar(1000),
  extension_c         nvarchar(1000),
  extension_d         nvarchar(1000),
  extension_e         nvarchar(1000),
  extension_f         nvarchar(1000),
  extension_g         nvarchar(1000),
  extension_h         nvarchar(1000),
  extension_i         nvarchar(1000),
  extension_j         nvarchar(1000),
  entry_ts            datetime2 not null,
  entry_user_cd       nvarchar(100) not null,
  renew_cnt           numeric(18,0) not null,
  renew_ts            datetime2 not null,
  renew_user_cd       nvarchar(100) not null
);
alter table k99m_sample add primary key (company_cd, sample_cd, locale_id, term_cd);
```

Service、Entity作成

次にService、Entityの作成を行います。

Service、Entityの作成

Service、Entityを作成します。

作成方法の詳細は『[intra-mart Accel Kaiden! プログラミングガイド](#)』の「Service、Entity作成」を参照してください。

Service、Entityの配置

作成したService、Entityをプロジェクトに配置します。

配置先のパス、およびパッケージは必要に応じて変更してください。

また、必要に応じて「applicationContext.xml」を作成してください。

作成方法の詳細は『[intra-mart Accel Kaiden! プログラミングガイド](#)』の「基本 - 事前準備 - 開発環境構築 - applicationContext.xmlの作成」を参照してください。

(本説明では、intra-mart Accel Kaiden! 基盤モジュールに配置した想定で作成しています。)

- K99mSample.java**

%ModuleProject%/src/main/generated/jp/co/slcs/kaiden/v2/base/foundation/model/entity/K99mSample.java

```

package jp.co.slcs.kaiden.v2.base.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;

/**
 * サンプルマスタ.
 *
 * @author XXX
 */
@Table(name = "k99m_sample")
public final class K99mSample 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[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";

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

    /** フィールド名[サンプルコード]. */
    public static final String SAMPLE_CD = "sampleCd";

    /** フィールド名[サンプル名]. */
    public static final String SAMPLE_NAME = "sampleName";

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

    /** 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;

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

    /** サンプルコード. */
    @PrimaryKey(generationType = GenerationType.APPLICATION)
    @Column(name = "sample_cd")
    public String sampleCd;

    /** サンプル名. */
    @Column(name = "sample_name")
    public String sampleName;
}

```

- K99mSampleService.java

```
%ModuleProject%/src/main/generated/jp/co/slcs/kaiden/v2/base/foundation/model/service/K99mSampleService.java
```

```
package jp.co.slcs.kaiden.v2.base.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.entity.K99mSample;

/**
 * {@link K99mSample} のサービスクラス.
 *
 * @author XXX
 */
@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public final class K99mSampleService extends GenerateService<K99mSample> {
    //
}
```

MasterBuilder定義作成

続いて、MasterBuilder定義を作成します。
本説明では、マスタIDを「sample」とします。

MasterBuilder定義ファイル作成

定義用のファイルを作成します。
ファイル名は必ず「%マスタID%.xml (sample.xml)」とします。

配置先

%ModuleProject%/src/main/storage/system/kaiden/generic/master/sample.xml

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">
</genericMasterSettings>
```

基本設定

マスタIDや作成したServiceを基に基本設定を行います。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <masterId>sample</masterId>
  <surfaceKey>master_name</surfaceKey>
  <service>k99mSampleService</service>
  <resources>service://kaiden/master/sample</resources>
</genericMasterSettings>
```

- マスタID (**genericMasterSettings** / **masterId**)
「sample」を設定します。
- サーフィスキー (**genericMasterSettings** / **surfaceKey**) : 必須
「master_name」を設定します。
実際のサーフィスは、後述のサーフィス設定で行います。
- サービスクラス (**genericMasterSettings** / **service**) : 任意
「[Service](#)、[Entity作成](#)」で作成したServiceのコンポーネント名「k99mSampleService」を設定します。
- 認可リソース (**genericMasterSettings** / **resources**) : 任意
「service://kaiden/master/sample」を設定します。
合わせて、認可設定を行います。

項目設定

作成したEntityを基に項目設定を行います。
Entityに保持していない項目（会社名など）もSQLで取得する場合は設定します。
次の例では、カスタマイズ用の予備項目はテキストボックスとして設定しています。

また、作成したEntityに直接保持はしていませんが、GenerateEntityに保持する次の項目の設定します。

- 登録日 (entryTs)
- 登録者 (entryUserCd)
- 更新カウント (renewCnt)

- 更新日（renewTs）
- 更新者（renewUserCd）
- 予備項目A～J（extensionA～J）

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <items>
    <!-- 会社コード -->
    <item name="companyCd" surfaceKey="company_code" primary="true">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'companyExists', 'args':['companyCd','searchCriteriaDate','true'] }"/>
      </validates>
    </item>
    <!-- 会社名 -->
    <item name="companyName" surfaceKey="company_name"/>
    <!-- 期間コード -->
    <item name="termCd" surfaceKey="term_code" primary="true"/>
    <!-- 開始日 -->
    <item name="startDate" surfaceKey="start_date">
      <convert display="STRING_DATE" process="TIMESTAMP"/>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'date' }"/>
      </validates>
    </item>
    <!-- 終了日 -->
    <item name="endDate" surfaceKey="end_date">
      <convert display="STRING_DATE" process="TIMESTAMP_SHIFT"/>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'date' }"/>
        <validate type="DEFAULT" args="{ 'type':'dateMin', 'args':['startDate'] }"/>
      </validates>
    </item>
    <!-- 削除フラグ -->
    <item name="deleteFlag" surfaceKey="delete_flag">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'flag' }"/>
      </validates>
    </item>
    <!-- ロケールID -->
    <item name="localeId" surfaceKey="locale_id" primary="true">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'locale' }"/>
      </validates>
    </item>
    <!-- サンプルコード -->
    <item name="sampleCd" surfaceKey="sample_code" type="TEXT_CODE" primary="true">
      <args>
        <arg key="style" value="width:300px;"/>
        <arg key="maxlength" value="100"/>
      </args>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'kaidenId' }"/>
        <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':['100'] }"/>
      </validates>
    </item>
    <!-- サンプル名 -->
    <item name="sampleName" surfaceKey="sample_name" type="TEXT_LOCALES">
      <args>
        <arg key="style" value="width:300px;"/>
        <arg key="maxlength" value="250"/>
        <arg key="imeMode" value="on"/>
      </args>
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':['250'] }"/>
      </validates>
    </item>
    <!-- 予備項目A -->
    <item name="extensionA" surfaceKey="extension_a" type="TEXT">
      <args>
        <arg key="style" value="width:300px;"/>
        <arg key="maxlength" value="250"/>
      </args>
    </item>
  </items>
</genericMasterSettings>
```



```

    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目B -->
<item name="extensionB" surfaceKey="extension_b" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目C -->
<item name="extensionC" surfaceKey="extension_c" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目D -->
<item name="extensionD" surfaceKey="extension_d" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目E -->
<item name="extensionE" surfaceKey="extension_e" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目F -->
<item name="extensionF" surfaceKey="extension_f" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目G -->
<item name="extensionG" surfaceKey="extension_g" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目H -->
<item name="extensionH" surfaceKey="extension_h" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目I -->
<item name="extensionI" surfaceKey="extension_i" type="TEXT">
  <args>

```

```

<arg key="style" value="width:300px;"/>
<arg key="maxlength" value="250"/>
<arg key="imeMode" value="on"/>
</args>
<validates>
  <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
</validates>
</item>
<!-- 予備項目 -->
<item name="extensionJ" surfaceKey="extension_j" type="TEXT_LOCALES">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 登録日 -->
<item name="entryTs" surfaceKey="entry_ts">
  <convert display="STRING_TIMESTAMP" process="TIMESTAMP"/>
</item>
<!-- 登録者 -->
<item name="entryUserCd" surfaceKey="entry_user_cd"/>
<!-- 更新カウント -->
<item name="renewCnt" surfaceKey="renew_cnt">
  <convert display="STRING" process="LONG"/>
</item>
<!-- 更新日 -->
<item name="renewTs" surfaceKey="renew_ts">
  <convert display="STRING_TIMESTAMP" process="TIMESTAMP"/>
</item>
<!-- 更新者 -->
<item name="renewUserCd" surfaceKey="renew_user_cd"/>
</items>
</genericMasterSettings>

```

拡張設定

本説明では、拡張設定は行いません。

MasterBuilderの標準機能を拡張（カスタマイズ）する場合は必要に応じて設定してください。

サーフィス設定

項目設定で設定した各項目のサーフィスを設定します。

次の例では、デフォルトのサーフィスのみ設定を行っていますが、

複数ロケール用の設定を行う場合は、「[サーフィス設定](#)」を参考に設定を行ってください。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <!-- 略 -->

  <!-- サーフィス設定 -->
  <surfaces>
    <surface surfaceKey="master_name" surface="サンプル" />
    <surface surfaceKey="company_code" surface="会社コード" />
    <surface surfaceKey="company_name" surface="会社名" />
    <surface surfaceKey="term_code" surface="期間コード" />
    <surface surfaceKey="start_date" surface="開始日" />
    <surface surfaceKey="end_date" surface="終了日" />
    <surface surfaceKey="delete_flag" surface="削除フラグ" />
    <surface surfaceKey="locale_id" surface="ロケールID" />
    <surface surfaceKey="sample_code" surface="サンプルコード" />
    <surface surfaceKey="sample_name" surface="サンプル名" />
    <surface surfaceKey="extension_a" surface="予備項目A" />
    <surface surfaceKey="extension_b" surface="予備項目B" />
    <surface surfaceKey="extension_c" surface="予備項目C" />
    <surface surfaceKey="extension_d" surface="予備項目D" />
    <surface surfaceKey="extension_e" surface="予備項目E" />
    <surface surfaceKey="extension_f" surface="予備項目F" />
    <surface surfaceKey="extension_g" surface="予備項目G" />
    <surface surfaceKey="extension_h" surface="予備項目H" />
    <surface surfaceKey="extension_i" surface="予備項目I" />
    <surface surfaceKey="extension_j" surface="予備項目J" />
    <surface surfaceKey="entry_ts" surface="登録日" />
    <surface surfaceKey="entry_user_cd" surface="登録者" />
    <surface surfaceKey="renew_cnt" surface="更新カウント" />
    <surface surfaceKey="renew_ts" surface="更新日" />
    <surface surfaceKey="renew_user_cd" surface="更新者" />
  </surfaces>
</genericMasterSettings>
```

メンテナンス設定

項目設定で設定した各項目の内、メンテナンス機能で利用する項目を設定します。
次の例では、サンプルコード・サンプル名を一覧画面、編集画面に表示されるように設定しています。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <!-- 略 -->

  <!-- サーフィス設定 -->
  <!-- 略 -->

  <!-- メンテナンス設定 -->
  <mainte>
    <listItems>
      <item name="sampleCd" width="250px"/>
      <item name="sampleName"/>
    </listItems>
    <editItems>
      <item name="sampleCd"/>
      <item name="sampleName"/>
    </editItems>
  </mainte>
</genericMasterSettings>
```

検索・リストデータ設定

項目設定で設定した各項目の内、検索・リストデータ機能で利用する項目を設定します。
次の例では、サンプルコード・サンプル名を検索結果画面に、サンプル名をセレクトボックスに表示されるように設定しています。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <!-- 略 -->

  <!-- サーフィス設定 -->
  <!-- 略 -->

  <!-- メンテナンス設定 -->
  <!-- 略 -->

  <!-- 検索設定 -->
  <search type="SEARCH">
    <items>
      <item name="sampleCd" type="CODE"/>
      <item name="sampleName" type="NAME"/>
    </items>
  </search>

</genericMasterSettings>
```

インポート設定

項目設定で設定した各項目の内、インポート機能で利用する項目を設定します。

次の例では、インポートファイルのレイアウトが「会社コード,開始日,終了日,削除フラグ,ロケールID,サンプルコード,サンプル名」となる様に設定しています。インポートファイルのレイアウトを変更する場合は、設定順を変更してください。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <!-- 略 -->

  <!-- サーフィス設定 -->
  <!-- 略 -->

  <!-- メンテナンス設定 -->
  <!-- 略 -->

  <!-- 検索設定 -->
  <!-- 略 -->

  <!-- インポート設定 -->
  <importJob>
    <items>
      <item name="companyCd"/>
      <item name="startDate"/>
      <item name="endDate"/>
      <item name="deleteFlag"/>
      <item name="localeId"/>
      <item name="sampleCd"/>
      <item name="sampleName"/>
    </items>
  </importJob>
</genericMasterSettings>
```

エクスポート設定

項目設定で設定した各項目の内、エクスポート機能で利用する項目を設定します。

次の例では、エクスポートファイルのレイアウトが「会社コード,開始日,終了日,削除フラグ,ロケールID,サンプルコード,サンプル名」となる様に設定しています。エクスポートファイルのレイアウトを変更する場合は、設定順を変更してください。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <!-- 略 -->

  <!-- 項目設定 -->
  <!-- 略 -->

  <!-- サーフィス設定 -->
  <!-- 略 -->

  <!-- メンテナンス設定 -->
  <!-- 略 -->

  <!-- 検索設定 -->
  <!-- 略 -->

  <!-- インポート設定 -->
  <!-- 略 -->

  <!-- エクスポート設定 -->
  <exportJob>
    <items>
      <item name="companyCd"/>
      <item name="startDate"/>
      <item name="endDate"/>
      <item name="deleteFlag"/>
      <item name="localeId"/>
      <item name="sampleCd"/>
      <item name="sampleName"/>
    </items>
  </exportJob>
</genericMasterSettings>
```

MasterBuilder定義

前項までの設定を行ったMasterBuilder定義は次の様になります。

内容

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">

  <!-- 基本設定 -->
  <masterId>sample</masterId>
  <surfaceKey>master_name</surfaceKey>
  <service>k99mSampleService</service>
  <resources>service://kaiden/master/sample</resources>

  <!-- 項目設定 -->
  <items>
    <!-- 会社コード -->
    <item name="companyCd" surfaceKey="company_code" primary="true">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
        <validate type="DEFAULT" args="{ 'type':'companyExists', 'args':['companyCd','searchCriteriaDate','true'] }"/>
      </validates>
    </item>
    <!-- 会社名 -->
    <item name="companyName" surfaceKey="company_name"/>
    <!-- 期間コード -->
    <item name="termCd" surfaceKey="term_code" primary="true">
      <!-- 開始日 -->
      <item name="startDate" surfaceKey="start_date">
        <convert display="STRING_DATE" process="TIMESTAMP"/>
        <validates>
          <validate type="DEFAULT" args="{ 'type':'required' }"/>
          <validate type="DEFAULT" args="{ 'type':'date' }"/>
        </validates>
      </item>
      <!-- 終了日 -->
      <item name="endDate" surfaceKey="end_date">
        <convert display="STRING_DATE" process="TIMESTAMP_SHIFT"/>
        <validates>
          <validate type="DEFAULT" args="{ 'type':'required' }"/>
          <validate type="DEFAULT" args="{ 'type':'date' }"/>
          <validate type="DEFAULT" args="{ 'type':'dateMin', 'args':['startDate'] }"/>
        </validates>
      </item>
    <!-- 削除フラグ -->
    <item name="deleteFlag" surfaceKey="delete_flag">
      <validates>
        <validate type="DEFAULT" args="{ 'type':'required' }"/>
```

```

<validate type="DEFAULT" args="{ 'type':'required' }"/>
<validate type="DEFAULT" args="{ 'type':'flag' }"/>
</validates>
</item>
<!-- ロケールID -->
<item name="localeId" surfaceKey="locale_id" primary="true">
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'locale' }"/>
  </validates>
</item>
<!-- サンプルコード -->
<item name="sampleCd" surfaceKey="sample_code" type="TEXT_CODE" primary="true">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="100"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'kaidenId' }"/>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'100' }"/>
  </validates>
</item>
<!-- サンプル名 -->
<item name="sampleName" surfaceKey="sample_name" type="TEXT_LOCALES">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'required' }"/>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目A -->
<item name="extensionA" surfaceKey="extension_a" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目B -->
<item name="extensionB" surfaceKey="extension_b" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目C -->
<item name="extensionC" surfaceKey="extension_c" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目D -->
<item name="extensionD" surfaceKey="extension_d" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type':'maxLength', 'args':'250' }"/>
  </validates>
</item>
<!-- 予備項目E -->
<item name="extensionE" surfaceKey="extension_e" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>

```

```

<validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
</validates>
</item>
<!-- 予備項目F -->
<item name="extensionF" surfaceKey="extension_f" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目G -->
<item name="extensionG" surfaceKey="extension_g" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目H -->
<item name="extensionH" surfaceKey="extension_h" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目I -->
<item name="extensionI" surfaceKey="extension_i" type="TEXT">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 予備項目J -->
<item name="extensionJ" surfaceKey="extension_j" type="TEXT_LOCALES">
  <args>
    <arg key="style" value="width:300px;"/>
    <arg key="maxlength" value="250"/>
    <arg key="imeMode" value="on"/>
  </args>
  <validates>
    <validate type="DEFAULT" args="{ 'type': 'maxLength', 'args': '250' }"/>
  </validates>
</item>
<!-- 登録日 -->
<item name="entryTs" surfaceKey="entry_ts">
  <convert display="STRING_TIMESTAMP" process="TIMESTAMP"/>
</item>
<!-- 登録者 -->
<item name="entryUserCd" surfaceKey="entry_user_cd"/>
<!-- 更新カウント -->
<item name="renewCnt" surfaceKey="renew_cnt">
  <convert display="STRING" process="LONG"/>
</item>
<!-- 更新日 -->
<item name="renewTs" surfaceKey="renew_ts">
  <convert display="STRING_TIMESTAMP" process="TIMESTAMP"/>
</item>
<!-- 更新者 -->
<item name="renewUserCd" surfaceKey="renew_user_cd"/>
</items>

<!-- サーフイス設定 -->
<surfaces>
  <surface surfaceKey="master_name" surface="サンプル" />
  <surface surfaceKey="company_code" surface="会社コード"/>
  <surface surfaceKey="company_name" surface="会社名"/>
  <surface surfaceKey="term_code" surface="期間コード"/>
  <surface surfaceKey="start_date" surface="開始日"/>
  <surface surfaceKey="end_date" surface="終了日"/>
  <surface surfaceKey="delete_flag" surface="削除フラグ"/>
  <surface surfaceKey="locale_id" surface="ロケールID"/>

```

```

<surface surfaceKey="sample_code" surface="サンプルコード"/>
<surface surfaceKey="sample_name" surface="サンプル名"/>
<surface surfaceKey="extension_a" surface="予備項目A"/>
<surface surfaceKey="extension_b" surface="予備項目B"/>
<surface surfaceKey="extension_c" surface="予備項目C"/>
<surface surfaceKey="extension_d" surface="予備項目D"/>
<surface surfaceKey="extension_e" surface="予備項目E"/>
<surface surfaceKey="extension_f" surface="予備項目F"/>
<surface surfaceKey="extension_g" surface="予備項目G"/>
<surface surfaceKey="extension_h" surface="予備項目H"/>
<surface surfaceKey="extension_i" surface="予備項目I"/>
<surface surfaceKey="extension_j" surface="予備項目J"/>
<surface surfaceKey="entry_ts" surface="登録日"/>
<surface surfaceKey="entry_user_cd" surface="登録者"/>
<surface surfaceKey="renew_cnt" surface="更新カウント"/>
<surface surfaceKey="renew_ts" surface="更新日"/>
<surface surfaceKey="renew_user_cd" surface="更新者"/>
</surfaces>

<!-- メンテナンス設定 -->
<mainte>
<listItems>
  <item name="sampleCd" width="250px"/>
  <item name="sampleName"/>
</listItems>
<editItems>
  <item name="sampleCd"/>
  <item name="sampleName"/>
</editItems>
</mainte>

<!-- 検索設定 -->
<search type="SEARCH">
<items>
  <item name="sampleCd" type="CODE"/>
  <item name="sampleName" type="NAME"/>
</items>
</search>

<!-- インポート設定 -->
<importJob>
<items>
  <item name="companyCd"/>
  <item name="startDate"/>
  <item name="endDate"/>
  <item name="deleteFlag"/>
  <item name="localeId"/>
  <item name="sampleCd"/>
  <item name="sampleName"/>
</items>
</importJob>

<!-- エクスポート設定 -->
<exportJob>
<items>
  <item name="companyCd"/>
  <item name="startDate"/>
  <item name="endDate"/>
  <item name="deleteFlag"/>
  <item name="localeId"/>
  <item name="sampleCd"/>
  <item name="sampleName"/>
</items>
</exportJob>
</genericMasterSettings>

```

検索用SQL作成

最後に、MasterBuilderで使用する検索用のSQLを作成します。

配置先

```
%ModuleProject%/src/main/resources/META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/sample.sql
```

検索項目設定

一覧検索、対象検索、期間検索、マスタ検索、エクスポートで使用する検索項目を設定します。
次の例では、「[テーブル定義](#)」で定義した項目に加え、会社名を抽出する様に設定しています。

内容


```

select
    target.company_cd
, dept.department_name as company_name
, target.term_cd
, target.start_date
, target.end_date
, target.delete_flag
, target.locale_id
, target.sample_cd
, target.sample_name
, target.extension_a
, target.extension_b
, target.extension_c
, target.extension_d
, target.extension_e
, target.extension_f
, target.extension_g
, target.extension_h
, target.extension_i
, target.extension_j
, target.entry_ts
, target.entry_user_cd
, target.renew_cnt
, target.renew_ts
, target.renew_user_cd
from
    k99m_sample target
left join
    imm_department dept
on dept.company_cd = target.company_cd
and dept.department_set_cd = target.company_cd
and dept.department_cd = target.company_cd
and dept.locale_id = /*searchCriteriaLocale*/ja'
and dept.start_date <= /*searchCriteriaDate*/2000/01/01'
and dept.end_date > /*searchCriteriaDate*/2000/01/01'

```

一覧検索条件設定

メンテナンス機能の一覧画面表示用の条件を設定します。

SQLに引き渡される検索条件は、「[一覧検索用SQL](#)」を参照してください。

内容

```

/* 略 */

/*IF searchType == "search"*/
/*BEGIN*/
where
    /*IF searchCriteriaCompany != null*/
    target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
    /*END*/
    /*IF searchCriteriaDate != null*/
    and target.start_date <= /*searchCriteriaDate*/2000/01/01'
    and target.end_date > /*searchCriteriaDate*/2000/01/01'
    /*END*/
    /*IF searchWithDisabled != null*/
    and target.delete_flag = /*searchWithDisabled*/searchWithDisabled'
    /*END*/
    /*IF searchCriteriaLocale != null*/
    and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
    /*END*/
    /*IF searchCriteriaKeyword != null*/
    and (
        target.sample_cd like /*searchCriteriaKeyword*/%' ESCAPE '$'
        or target.sample_name like /*searchCriteriaKeyword*/%' ESCAPE '$'
    )
    /*END*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/sample_cd
/*END*/
/*END*/

```

対象検索条件設定

メンテナンス機能の編集画面（編集部品）表示用の条件を設定します。

SQLに引き渡される検索条件は、「[対象検索用SQL](#)」を参照してください。

内容

```

/* 略 */

/*IF searchType == "search"*/
/* 略 */
/*END*/

/*IF searchType == "select"*/
/*BEGIN*/
where
/*IF sampleCd != null*/
target.sample_cd = /*sampleCd*/sampleCd'
/*END*/
/*IF companyCd != null*/
and target.company_cd = /*companyCd*/companyCd'
/*END*/
/*IF termCd != null*/
and target.term_cd = /*termCd*/termCd'
/*END*/
/*END*/
order by sample_cd
/*END*/

```

期間検索条件設定

メンテナンス機能の編集画面（期間部品）表示用の条件を設定します。
SQLに引き渡される検索条件は、「[期間検索用SQL](#)」を参照してください。

内容

```

/* 略 */

/*IF searchType == "search"*/
/* 略 */
/*END*/

/*IF searchType == "select"*/
/* 略 */
/*END*/

/*IF searchType == "term"*/
/*BEGIN*/
where
/*IF sampleCd != null*/
target.sample_cd = /*sampleCd*/sampleCd'
/*END*/
/*IF companyCd != null*/
and target.company_cd = /*companyCd*/companyCd'
/*END*/
/*IF localeId != null*/
and target.locale_id = /*localeId*/localeId'
/*END*/
/*END*/
order by sample_cd, start_date
/*END*/

```

マスタ検索条件設定

検索部品やセレクトボックス表示用の条件を設定します。
SQLに引き渡される検索条件は、「[マスタ検索用SQL](#)」を参照してください。

内容

```

/* 略 */

/*IF searchType == "search"*/
/* 略 */
/*END*/

/*IF searchType == "select"*/
/* 略 */
/*END*/

/*IF searchType == "term"*/
/* 略 */
/*END*/

/*IF searchType == "master"*/
/*BEGIN*/
where
/*IF sampleCd != null*/
target.sample_cd = /*sampleCd*/sampleCd'
/*END*/
/*IF searchCriteriaCompany != null*/
and target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
/*END*/
/*IF searchCriteriaDate != null*/
and target.start_date <= /*searchCriteriaDate*/2000/01/01'
and target.end_date > /*searchCriteriaDate*/2000/01/01'
/*END*/
/*IF deleteFlag != null*/
and target.delete_flag = /*deleteFlag*/deleteFlag'
/*END*/
/*IF searchCriteriaLocale != null*/
and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
/*END*/
/*IF searchCriteriaKeyword != null*/
and (
target.sample_cd like /*searchCriteriaKeyword*/%'ESCAPE '$'
or target.sample_name like /*searchCriteriaKeyword*/%'ESCAPE '$'
)
/*END*/
/*END*/
order by sample_cd
/*END*/

```

エクスポート条件設定

エクスポート用の条件を設定します。
SQLに引き渡される検索条件は、「[エクスポート用SQL](#)」を参照してください。

内容

```

/* 略 */

/*IF searchType == "search"*/
/* 略 */
/*END*/

/*IF searchType == "select"*/
/* 略 */
/*END*/

/*IF searchType == "term"*/
/* 略 */
/*END*/

/*IF searchType == "master"*/
/* 略 */
/*END*/

/*IF searchType == "export"*/
/*BEGIN*/
where
/*IF searchCriteriaCompany != null*/
target.company_cd = /*searchCriteriaCompany*/'searchCriteriaCompany'
/*END*/
/*IF targetDate != null*/
and target.start_date <= /*targetDate*/'2000/01/01'
and target.end_date > /*targetDate*/'2000/01/01'
/*END*/
/*IF renewTs != null*/
and target.renew_ts >= /*renewTs*/'2000/01/01'
/*END*/
/*END*/
order by sample_cd, start_date, locale_id
/*END*/

```

検索用SQL

前項までの設定を行った検索用SQLは次のようになります。

内容

```

select
  target.company_cd
, dept.department_name as company_name
, target.term_cd
, target.start_date
, target.end_date
, target.delete_flag
, target.locale_id
, target.sample_cd
, target.sample_name
, target.extension_a
, target.extension_b
, target.extension_c
, target.extension_d
, target.extension_e
, target.extension_f
, target.extension_g
, target.extension_h
, target.extension_i
, target.extension_j
, target.entry_ts
, target.entry_user_cd
, target.renew_cnt
, target.renew_ts
, target.renew_user_cd
from
  k99m_sample target
left join
  imm_department dept
    on dept.company_cd = target.company_cd
    and dept.department_set_cd = target.company_cd
    and dept.department_cd = target.company_cd
    and dept.locale_id = /*searchCriteriaLocale*/'ja'
    and dept.start_date <= /*searchCriteriaDate*/'2000/01/01'
    and dept.end_date > /*searchCriteriaDate*/'2000/01/01'

/*IF searchType == "search"*/
/*BEGIN*/
where
/*IF searchCriteriaCompany != null*/
target.company_cd = /*searchCriteriaCompany*/'searchCriteriaCompany'
/*END*/
/*IF searchCriteriaDate != null*/

```

```

/*IF searchCriteriaDate != null*/
and target.start_date <= /*searchCriteriaDate*/2000/01/01'
and target.end_date > /*searchCriteriaDate*/2000/01/01'
/*END*/
/*IF searchWithDisabled != null*/
and target.delete_flag = /*searchWithDisabled*/searchWithDisabled'
/*END*/
/*IF searchCriteriaLocale != null*/
and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
/*END*/
/*IF searchCriteriaKeyword != null*/
and (
  target.sample_cd like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
  or target.sample_name like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
)
/*END*/
/*END*/
/*IF orderBy != null*/
order by /*$orderBy*/sample_cd
/*END*/
/*END*/

/*IF searchType == "select"*/
/*BEGIN*/
where
  /*IF sampleCd != null*/
  target.sample_cd = /*sampleCd*/sampleCd'
  /*END*/
  /*IF companyCd != null*/
  and target.company_cd = /*companyCd*/companyCd'
  /*END*/
  /*IF termCd != null*/
  and target.term_cd = /*termCd*/termCd'
  /*END*/
/*END*/
order by sample_cd
/*END*/

/*IF searchType == "term"*/
/*BEGIN*/
where
  /*IF sampleCd != null*/
  target.sample_cd = /*sampleCd*/sampleCd'
  /*END*/
  /*IF companyCd != null*/
  and target.company_cd = /*companyCd*/companyCd'
  /*END*/
  /*IF localeId != null*/
  and target.locale_id = /*localeId*/localeId'
  /*END*/
/*END*/
order by sample_cd, start_date
/*END*/

/*IF searchType == "master"*/
/*BEGIN*/
where
  /*IF sampleCd != null*/
  target.sample_cd = /*sampleCd*/sampleCd'
  /*END*/
  /*IF searchCriteriaCompany != null*/
  and target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
  /*END*/
  /*IF searchCriteriaDate != null*/
  and target.start_date <= /*searchCriteriaDate*/2000/01/01'
  and target.end_date > /*searchCriteriaDate*/2000/01/01'
  /*END*/
  /*IF deleteFlag != null*/
  and target.delete_flag = /*deleteFlag*/deleteFlag'
  /*END*/
  /*IF searchCriteriaLocale != null*/
  and target.locale_id = /*searchCriteriaLocale*/searchCriteriaLocale'
  /*END*/
  /*IF searchCriteriaKeyword != null*/
  and (
    target.sample_cd like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
    or target.sample_name like /*searchCriteriaKeyword*/'% ' ESCAPE '$'
  )
  /*END*/
/*END*/
order by sample_cd
/*END*/

/*IF searchType == "export"*/
/*BEGIN*/
where
  /*IF searchCriteriaCompany != null*/

```

```

/*searchCriteriaCompany*/
target.company_cd = /*searchCriteriaCompany*/searchCriteriaCompany'
/*END*/
/*IF targetDate != null*/
and target.start_date <= /*targetDate*/2000/01/01'
and target.end_date > /*targetDate*/2000/01/01'
/*END*/
/*IF renewTs != null*/
and target.renew_ts >= /*renewTs*/2000/01/01'
/*END*/
/*END*/
order by sample_cd, start_date, locale_id
/*END*/

```

動作確認

これまでに設定した内容を動作確認します。

配置先

```

src
├── main
│   ├── generated
│   │   ├── jp
│   │   │   ├── co
│   │   │   │   ├── slcs
│   │   │   │   │   ├── kaiden
│   │   │   │   │   │   ├── v2
│   │   │   │   │   │   │   ├── base
│   │   │   │   │   │   │   │   ├── foundation
│   │   │   │   │   │   │   │   │   ├── model
│   │   │   │   │   │   │   │   │   │   ├── entity
│   │   │   │   │   │   │   │   │   │   │   ├── K99mSample.java
│   │   │   │   │   │   │   │   │   │   │   └── service
│   │   │   │   │   │   │   │   │   │   │       └── K99mSampleService.java
│   │   │   │   │   │   │   │   │   │   └── K99mSampleService.java
│   │   │   │   │   │   │   │   │   └── K99mSampleService.java
│   │   │   │   │   │   │   │   └── K99mSampleService.java
│   │   │   │   │   │   │   └── K99mSampleService.java
│   │   │   │   │   │   └── K99mSampleService.java
│   │   │   │   │   └── K99mSampleService.java
│   │   │   │   └── K99mSampleService.java
│   │   │   └── K99mSampleService.java
│   │   └── K99mSampleService.java
│   └── K99mSampleService.java
├── resources
│   ├── META-INF
│   │   ├── sql
│   │   │   ├── jp
│   │   │   │   ├── co
│   │   │   │   │   ├── slcs
│   │   │   │   │   │   ├── kaiden
│   │   │   │   │   │   │   ├── v2
│   │   │   │   │   │   │   │   ├── generic
│   │   │   │   │   │   │   │   │   ├── master
│   │   │   │   │   │   │   │   │   └── sample.sql
│   │   │   │   │   │   │   └── sample.sql
│   │   │   │   │   │   └── sample.sql
│   │   │   │   │   └── sample.sql
│   │   │   │   └── sample.sql
│   │   │   └── sample.sql
│   │   └── sample.sql
│   └── sample.sql
├── storage
│   ├── system
│   │   ├── kaiden
│   │   │   ├── generic
│   │   │   │   ├── master
│   │   │   │   │   └── sample.xml
│   │   │   └── sample.xml
│   │   └── sample.xml
│   └── sample.xml
└── sample.xml

```



コラム

配置先には記載していませんが、「[テーブル定義](#)」で定義したテーブルを作成してください。

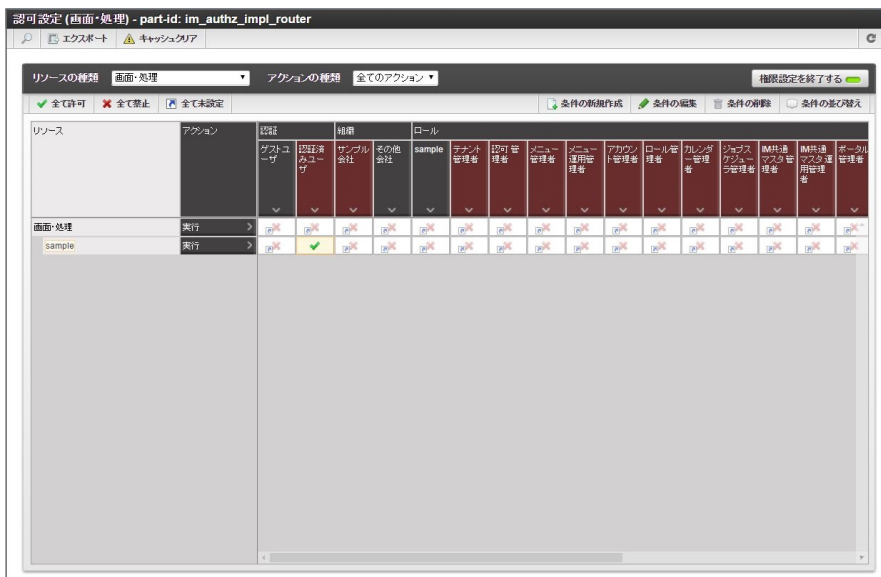
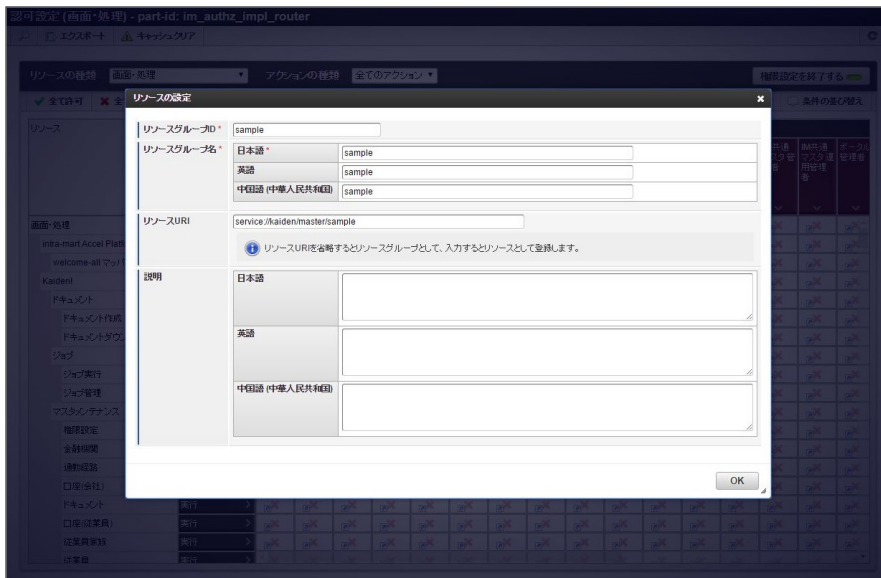
メンテナンス機能

メンテナンス機能は、次の手順で動作確認を行います。

1. 認可リソース作成・設定

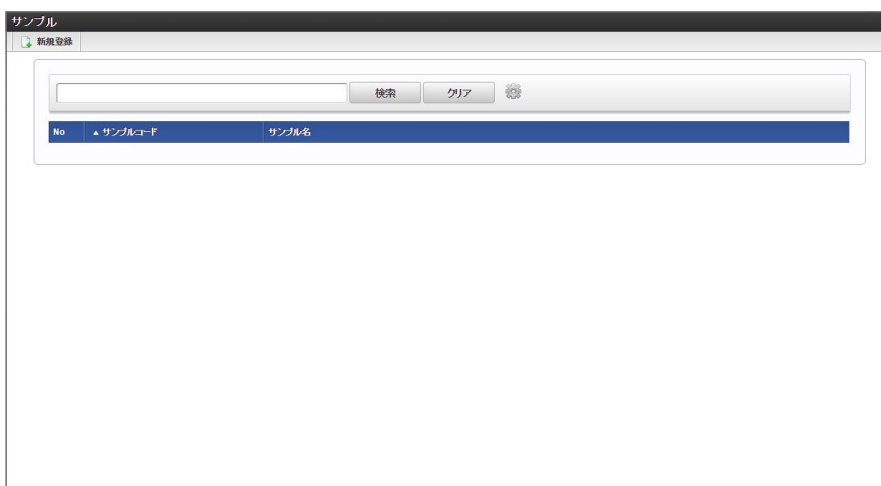
MasterBuilder定義で設定した認可リソース「service://kaiden/master/sample」を設定し、検証用のユーザに設定します。

また、認可リソース「service://kaiden/generic/master」を検証用のユーザに設定します。



2. メンテナンス機能表示

「http://<HOST>:<PORT>/<CONTEXT_PATH>/kaiden/v2/generic/master?masterId=sample」にアクセスし、マスタメンテナンス画面を表示します。



3. 動作確認

一覧検索や新規登録、更新、削除などが正しく行われることを確認します。

サンプル

期間一覧

1900/01/01 - 2999/12/31

期間分割 期間変更

開始日 * 終了日 * 1900/01/01 2999/12/31

会社 * サンプル会社

サンプルコード * sample01

サンプル名 *

日本語 sample01

英語 sample01

中国語 (中国) sample01

更新 一覧に戻る

サンプル

新規登録

検索 クリア

No	▲ サンプルコード	サンプル名
1	sample01	sample01
2	sample02	sample02
3	sample03	sample03
4	sample04	sample04
5	sample05	sample05

最初へ 前へ 1 次へ 最後へ

i コラム

メニューに設定する場合は、次の様に設定します。

メニューアイテムの新規作成

メニューアイテムID * sample

メニューアイテム名 *

日本語 * sample

英語 sample

中国語 (中国) sample

URL * kaiden/v2/generic/master 権限設定

呼び出し方法 POST

引数

+ 行追加 - 選択行削除

キー	値
☐ masterid	sample

アイコン画像

標準 ☒ ファイル/パス コンテキストパス配下のURLを入力してください。

☐ CSS Sprites imui://csssprites/ クラス名を入力してください。

16px

32px

新規作成

- **メニューアイテムID**
任意の値を設定してください。
- **メニューアイテム名**
任意の値を設定してください。
- **URL**
kaiden/v2/generic/master
- **呼び出し方法**
POST
- **引数**
値には、マスタIDを設定します。

キー	値
masterid	sample

検索部品・セレクトボックス

検索部品・セレクトボックスは、次の手順で動作確認を行います。
事前にマスタメンテナンス機能を利用して、検索対象のマスタデータを作成してください。

1. 確認用アプリケーション作成

Controller

%ModuleProject%/src/main/java/jp/co/slcs/kaiden/v2/base/feature/controller/kaiden/sample/SampleController.java

```

package jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.sample;

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;

/**
 * Sample(Controller).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Controller
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
@RequestMapping("kaiden/v2/sample/sample")
public class SampleController {

    /**
     * index.
     *
     * @return /kaiden/v2/base/sample/sample.jsp
     */
    @RequestMapping(value = {"", "/index"})
    public String index() {
        return "/kaiden/v2/base/sample/sample.jsp";
    }
}

```

JSP

%ModuleProject%/src/main/webapp/WEB-INF/views/kaiden/v2/base/sample/sample.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="k" uri="http://kaiden.slcs.co.jp/taglib/v2/ui" %>
<%@ taglib prefix="k_generic" uri="http://kaiden.slcs.co.jp/taglib/v2/generic" %>
<%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
<%@ taglib prefix="imart" uri="http://www.intra-mart.co.jp/taglib/im-tenant" %>

<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}_${gadgetInstance}</c:set>
<c:set var="blockId">inputArea</c:set>
<c:set var="extensionId">${gadgetClass}_${gadgetVariation}</c:set>

<imart type="head">
  <title>サンプル</title>
  <jsp:include page="/WEB-INF/views/kaiden/v2/base/foundation/commonIncludeHeader.jsp"></jsp:include>
</imart>

<div class="imui-title">
  <h1>サンプル</h1>
</div>

<div class="imui-form-container-wide" style="overflow:auto;min-width:930px">
  <div id="kaiden-contents-container" style="overflow:hidden">
    <div id="kaiden-system-container" style="display:none">
      <jsp:include page="/WEB-INF/views/kaiden/v2/base/foundation/genericErrorContainer.jsp"></jsp:include>
      <jsp:include page="/WEB-INF/views/kaiden/v2/base/foundation/dateTextController.jsp"></jsp:include>
    </div>
    <jsp:include page="/WEB-INF/views/kaiden/v2/base/foundation/tmtable/tmtableOperationPanel.jsp"></jsp:include>
    <div id="kaiden-mainte-controller">
      <k:gadgetContainer gadgetId="KaidenMainteController_v01_0"
        gadgetClass="KaidenMainteController"
        gadgetVariation="v01"
        gadgetInstance="0">
      </k:gadgetContainer>
    </div>
    <div id="kaiden-mainte-mainte-container">
      <k:gadgetContainer
        gadgetClass="${gadgetClass}"
        gadgetVariation="${gadgetVariation}"
        gadgetInstance="${gadgetInstance}"
        gadgetTitle="Sample"
        gadgetId="${gadgetId}">
      <k:blockContainer id="${blockId}">
        <table class="imui-form" style="width:100%">

```

```

<colgroup>
  <col style="width:200px;" />
</colgroup>
<tr>
  <th><k:surface surfaceKey="検索基準日"/></th>
  <td><k:date name="searchCriteriaDate" mode="0"/></td>
</tr>
<tr>
  <th><k:surface surfaceKey="検索基準口ケール"/></th>
  <td>
    <k_generic:master
      masterId="locale"
      extensionId="{extensionId}"
      searchType="search"
      mode="0"
      name="searchCriteriaLocale"
      viewName="searchCriteriaLocaleName"
      surfaceKey=""
      criteria="{ 'localeId':'searchCriteriaLocale' }"
      resultChain="{ 'searchCriteriaLocale':'localeId', 'searchCriteriaLocaleName':'localeName' }"
      resultValue="localeName"
      primaryTarget="searchCriteriaLocale"
      style="width:200px;"
    />
  </td>
</tr>

<tr>
  <th><k:surface surfaceKey="検索基準会社"/></th>
  <td>
    <k_generic:master
      masterId="company"
      extensionId="{extensionId}"
      searchType="search"
      mode="0"
      name="searchCriteriaCompany"
      viewName="searchCriteriaCompanyName"
      surfaceKey=""
      criteria="{ 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate',
'searchCriteriaLocale':'searchCriteriaLocale', 'includeDisabled':'true' }"
      resultChain="{ 'searchCriteriaCompany':'companyCd', 'searchCriteriaCompanyName':'companyName' }"
      resultValue="companyName"
      primaryTarget="searchCriteriaCompany"

      refreshTarget="searchCriteriaDate,searchCriteriaLocale"
      gadgetId="{gadgetId}"
      gadgetBlockId="{blockId}"
      style="width:200px;"
    />
  </td>
</tr>
</table>
<br/>
<table class="imui-form" style="width:100%">
  <colgroup>
    <col />
    <col />
    <col />
  </colgroup>
  <tr>
    <th><k:surface surfaceKey="指定なし"/></th>
    <th><k:surface surfaceKey="検索部品指定"/></th>
    <th><k:surface surfaceKey="セレクトボックス指定"/></th>
  </tr>
  <tr>
    <td>
      <k_generic:master
        masterId="sample"
        extensionId="{extensionId}"
        mode="0"
        name="sampleCd"
        viewName="sampleName"
        surfaceKey=""
        criteria="{ 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate',
'searchCriteriaLocale':'searchCriteriaLocale', 'sampleCd':'sampleCd', 'includeDisabled':'true', 'includeBlank':'true' }"
        resultChain="{ 'sampleCd':'sampleCd', 'sampleName':'sampleName' }"
        resultValue="sampleName"
        primaryTarget="sampleCd"

        refreshTarget="searchCriteriaCompany,searchCriteriaDate,searchCriteriaLocale"
        gadgetId="{gadgetId}"
        gadgetBlockId="{blockId}"
        style="width:200px;"
      />
    </td>
  </tr>

```

```

<td>
  <k_generic:master
    masterId="sample"
    extensionId="{extensionId}"
    searchType="search"
    mode="0"
    name="search_sampleCd"
    viewName="search_sampleName"
    surfaceKey=""
    criteria="{ 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate',
'searchCriteriaLocale':'searchCriteriaLocale', 'sampleCd':'search_sampleCd', 'includeDisabled':'true', 'includeBlank':'true'}"
    resultChain="{ 'search_sampleCd':'sampleCd', 'search_sampleName':'sampleName'}"
    resultValue="sampleName"
    primaryTarget="search_sampleCd"

    refreshTarget="searchCriteriaCompany,searchCriteriaDate,searchCriteriaLocale"
    gadgetId="{gadgetId}"
    gadgetBlockId="{blockId}"
    style="width:200px;"
  />
</td>
<td>
  <k_generic:master
    masterId="sample"
    extensionId="{extensionId}"
    searchType="select"
    mode="0"
    name="select_sampleCd"
    viewName="select_sampleName"
    surfaceKey=""
    criteria="{ 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate',
'searchCriteriaLocale':'searchCriteriaLocale', 'sampleCd':'select_sampleCd', 'includeDisabled':'true', 'includeBlank':'true'}"
    resultChain="{ 'select_sampleCd':'sampleCd', 'select_sampleName':'sampleName'}"
    resultValue="sampleName"
    primaryTarget="select_sampleCd"

    refreshTarget="searchCriteriaCompany,searchCriteriaDate,searchCriteriaLocale"
    gadgetId="{gadgetId}"
    gadgetBlockId="{blockId}"
    style="width:200px;"
  />
</td>
</tr>
</table>
</k:blockContainer>
</k:gadgetContainer>
</div>
</div>
<script type="text/javascript">
jQuery(function($) {
  var accessor = KAIDEN.getGadget("{gadgetId}").blockAccessors["{blockId}"];
  var eventManager = KAIDEN.getGadget("{gadgetId}").eventManager;

  eventManager.registPushListener("initialize", function(**String*/args) {

    // 検索基準日
    var _nowDate = KAIDEN.Util.nowDate();
    accessor("searchCriteriaDate").setValue(_nowDate);

    // 検索基準ロケール
    accessor("searchCriteriaLocale").setValue("ja");
    KAIDEN.MasterSearch.refresh(accessor("searchCriteriaLocaleName").elem(true), true);

    // 検索基準会社
    accessor("searchCriteriaCompany").setValue("comp_sample_01");
    KAIDEN.MasterSearch.refresh(accessor("searchCriteriaCompanyName").elem(true), true);

    // リストデータの更新
    $("[data-kaiden-select-list]").each(function() {
      KAIDEN.SelectListData.refresh(this);
    });
  });
  KAIDEN.gadgetMan.execPush("initialize");
});
</script>
</div>

```

2. 確認用アプリケーション表示

「http://<HOST>:<PORT>/<CONTEXT_PATH>/kaiden/v2/sample/sample」にアクセスし、確認用アプリケーションを表示します。

サンプル

Sample

検索基準日 2016/05/09

検索基準ロケール ☒ 日本語

検索基準会社 ☒ サンプル会社

指定なし

検索部品指定

セレクトボックス指定

3. 動作確認

検索部品・セレクトボックスが正しく動作することを確認します。

サンプル

Sample

検索基準日 2016/05/09

検索基準ロケール ☒ 日本語

検索基準会社 ☒ サンプル会社

指定なし

検索部品指定

セレクトボックス指定

No	サンプルコード	サンプル名
1	sample01	sample01
2	sample02	sample02
3	sample03	sample03
4	sample04	sample04
5	sample05	sample05

サンプル

Sample

検索基準日 2016/05/09

検索基準ロケール ☒ 日本語

検索基準会社 ☒ サンプル会社

指定なし

検索部品指定

セレクトボックス指定

sample01
sample02
sample03
sample04
sample05

また、MasterBuilder定義の「検索タイプ (genericMasterSettings / search @ type)」に設定した値を切り替えることで、検索部品・セレクトボックスが切り替わることを確認します。

指定なし

sample01
sample02
sample03
sample04
sample05

指定なし

No	サンプルコード	サンプル名
1	sample01	sample01
2	sample02	sample02
3	sample03	sample03
4	sample04	sample04
5	sample05	sample05

インポートジョブ

インポートジョブ機能は、次の手順で動作確認を行います。

1. インポートジョブ作成

「サイトマップ」→「ジョブ管理」→「ジョブ設定」を表示し、インポート用のジョブを作成します。

ジョブ作成

基本情報

ジョブカテゴリ

sample

ツリーから選択

クリア

ジョブID *

import_sample

ジョブ名 *

日本語

インポート

英語

インポート

中国語 (中国)

インポート

ジョブの説明

実行時の情報

実行言語 *

Java

実行プログラム *

jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterImportJob

実行パラメータ

+ パラメータ追加 - すべて削除 ジョブ定義から取得

パラメータリスト (追加後にクリックして入力してください)

キー	値	削除
masterId	sample	×

新規作成

- **ジョブカテゴリ**
任意の値を設定してください。
- **ジョブID**
任意の値を設定してください。
- **ジョブ名**
任意の値を設定してください。
- **実行言語**
Java
- **実行プログラム**
jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterImportJob
- **実行パラメータ**
値には、マスタIDを設定します。

キー	値
masterId	sample

2. インポートジョブネット作成
- 「サイトマップ」→「ジョブ管理」→「ジョブネット設定」を表示し、インポート用のジョブネットを作成します。

ジョブネット作成

基本情報

ジョブネットカテゴリ

sample

ツリーから選択

クリア

ジョブネットID *

import_sample

ジョブネット名 *

日本語

インポート

英語

インポート

中国語 (中華人民共和国)

インポート

ジョブネットの説明

実行時の情報

並列実行

☐ 並列実行を許可する

実行ジョブ

+ ジョブを追加

- すべて削除

ジョブリスト

ジョブID	ジョブ名	前鋒
import_sample	インポート	×

実行パラメータ

+ パラメータ追加

- すべて削除

パラメータリスト(追加時にクリックして入力してください)

キー	値	前鋒
----	---	----

トリガ設定

日時指定

新規登録

日時指定	
繰り返し指定	
営業日指定	

新規作成

- ジョブネットカテゴリ
任意の値を設定してください。
- ジョブネットID
任意の値を設定してください。
- ジョブネット名
任意の値を設定してください。
- 実行ジョブ
「インポートジョブ作成」で作成したジョブ

3. インポートファイル作成

「%Public Storage%/kaiden/generic/master/sample/import.csv」にインポートファイルを作成します。

```
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample05","sample05"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample05","sample05"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample05","sample05"
```

4. 動作確認

「サイトマップ」→「ジョブ管理」→「ジョブネット設定」を表示し、インポート用のジョブネットを即時実行します。

ジョブネット情報

基本情報

ジョブネットカテゴリ

sample

ジョブネットID

import_sample

ジョブネット名

日本語	インポート
英語	インポート
中国語 (中華人民共和国)	インポート

ジョブネットの説明

実行時の情報

並列実行

☐ 並列実行を許可する

実行ジョブ

ジョブリスト

ジョブID	ジョブ名
import_sample	インポート

実行パラメータ

パラメータリスト

キー	値
----	---

トリガ設定

日時指定

繰り返し指定

営業日指定

即時実行

編集

削除

実行後、メンテナンス機能を利用して、インポートが正しく実行されたことを確認します。

エクスポートジョブ

エクスポートジョブ機能は、次の手順で動作確認を行います。

事前にマスタメンテナンス機能を利用して、エクスポート対象のマスタデータを作成してください。

1. エクスポートジョブ作成

「サイトマップ」→「ジョブ管理」→「ジョブ設定」を表示し、エクスポート用のジョブを作成します。

ジョブ作成

基本情報

ジョブカテゴリ

sample

ツリーから選択

クリア

ジョブID *

export_sample

ジョブ名 *

日本語

英語

中国語 (中国)

エクスポート

エクスポート

エクスポート

ジョブの説明

実行時の情報

実行言語 *

Java

実行プログラム *

jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterExportJob

実行パラメータ

+ パラメータ追加 - すべて削除 設定 ジョブ定義から取得

パラメータリスト (追加後にクリックして入力してください)

キー	値	削除
masterId	sample	✖

新規作成

- **ジョブカテゴリ**
任意の値を設定してください。
- **ジョブID**
任意の値を設定してください。
- **ジョブ名**
任意の値を設定してください。
- **実行言語**
Java
- **実行プログラム**
jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterExportJob
- **実行パラメータ**
値には、マスタIDを設定します。

キー	値
masterId	sample

2. エクスポートジョブネット作成
- 「サイトマップ」→「ジョブ管理」→「ジョブネット設定」を表示し、エクスポート用のジョブネットを作成します。

ジョブネット作成

基本情報

ジョブネットカテゴリ

sample

ツリーから選択

クリア

ジョブネットID *

export_sample

ジョブネット名 *

日本語

エクスポート

英語

エクスポート

中国語 (中華人民共和国)

エクスポート

ジョブネットの説明

実行時の情報

並列実行

☐ 並列実行を許可する

実行ジョブ

+ ジョブを追加

- すべて削除

ジョブリスト

ジョブID	ジョブ名	削除
export_sample	エクスポート	

実行パラメータ

+ パラメータ追加

- すべて削除

パラメータリスト (追加欄をクリックして入力してください)

キー	値	削除
----	---	----

トリガ設定

日時指定

新規登録

日時指定

繰り返し指定

営業日指定

新規作成

- ジョブネットカテゴリ
任意の値を設定してください。
- ジョブネットID
任意の値を設定してください。
- ジョブネット名
任意の値を設定してください。
- 実行ジョブ
「エクスポートジョブ作成」で作成したジョブ

3. 動作確認

「サイトマップ」→「ジョブ管理」→「ジョブネット設定」を表示し、エクスポート用のジョブネットを即時実行します。

ジョブネット情報

基本情報

ジョブネットカテゴリ

sample

ジョブネットID

export_sample

ジョブネット名

日本語	エクスポート
英語	エクスポート
中国語 (中華人民共和国)	エクスポート

ジョブネットの説明

実行時の情報

並列実行

☐ 並列実行を許可する

実行ジョブ

ジョブリスト

ジョブID	ジョブ名
export_sample	エクスポート

実行パラメータ

パラメータリスト

キー	値
----	---

トリガ設定

日時指定

繰り返し指定

営業日指定

即時実行

編集

削除

「%Public Storage%/kaiden/generic/master/sample/export.csv」にマスタデータがエクスポートされることを確認します。

```
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample01","sample01"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample02","sample02"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample03","sample03"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample04","sample04"
"comp_sample_01","1900/01/01","2999/12/31","0","en","sample05","sample05"
"comp_sample_01","1900/01/01","2999/12/31","0","ja","sample05","sample05"
"comp_sample_01","1900/01/01","2999/12/31","0","zh_CN","sample05","sample05"
```

本項では、メンテナンス機能の機能詳細を説明します。

- ## クライアントサイド

マスタメンテナンス機能では、他の機能と同様にLogicやManagerの拡張に追加し、各JSPの拡張が可能です。各JSPの詳細や、イベントは次の通りです。

- **JSPの構成イメージ**
メンテナンス機能では、一覧画面と登録・更新画面の2画面の構成となりますが、実際には単一のJSP（メイン）が各部品をインクルードする構成となっています。

各部品の構成イメージは次の通りです。

勘定科目

新規登録

検索

クリア

検索条件

2016/05/09

会社

サンプル会社

ロケール

日本語

ステータスが無効のデータも検索

検索部品

No	▲ 勘定科目コード	勘定科目名
1	31000010	給料賃金
2	31000012	俸給費
3	31000013	通達費
4	31000014	会議費
5	31000015	広告宣伝費
6	31000016	販売管理費
7	31000017	接待交際費
8	31000018	荷運運賃
9	31000019	福利厚生費
10	31000021	水電気熱費
11	31000022	消耗品費
12	31000023	印刷製写費
13	31000024	新聞図書費
14	31000025	支払報酬
15	31000030	雑費
16	31000038	教育訓練費
17	31001011	旅費交通費

一覧部品

最初へ

1

次へ

最後へ

勘定科目

期間一覧

○ 1900/01/01 - 1999/12/31

期間分割 期間変更

● 2000/01/01 - 2099/12/31

期間分割 期間変更

● 2100/01/01 - 2199/12/31

期間分割 期間変更

○ 2200/01/01 - 2999/12/31

期間分割 期間変更

開始日 * - 終了日 *

会社 *

勘定科目コード *

勘定科目名 *

2000/01/01 - 2099/12/31

サンプル会社

31000010

日本語 給料賃金

英語 Salary and Wages

中国語 中華人民共和国 工資薪金

更新

一覧に戻る

編集部品

期間部品

ボタン部品

イメージには表記されていませんが、拡張部品もインクルードされています。

- **実行イベント**
インクルードされた各部品では、EventManagerを利用して部品間を跨いだイベントを実行しています。マスタメンテナンス機能で実行されるイベントは次の通りです。

名称	ID	実行元	メイン	検索	一覧	期間	編集	ボタン
初期表示	initialize	JSPロード時	○	○				
画面の切替有 無設定	setTrans	期間検索、対象検索	○					
一覧画面表示	showListArea	初期表示、登録、更新、削除、一覧に戻る	○	○				
編集画面表示	showMainteArea	新規登録、一覧行、登録、更新	○		○			
新規登録画面 遷移	trunsRegist	新規登録				○	○	○
更新画面遷移	trunsRenewal	一覧行、登録、更新				○	○	○
一覧画面遷移	trunsList	一覧に戻る、登録、更新、削除						
一覧クリア	clearList	クリアボタン			○			
編集クリア	clearMainte	一覧画面表示				○	○	
一覧検索	searchList	一覧画面表示、検索ボタン、タイトル行、ページャ		○				

名称	ID	実行元	メイン	検索	一覧	期間	編集	ボタン
一覧検索結果 設定	setSearchListResult	一覧検索			○			
期間検索	searchTerm	更新画面遷移、非対象カーソル				○		
期間検索結果 設定	setSearchTermResult	期間検索、有効化、無効化、期間分割、期間変更				○		
対象検索	searchTarget	期間検索結果設定、更新画面遷移					○	
対象検索結果 設定	setSearchTargetResult	対象検索					○	
登録	entry	登録ボタン					○	
更新	renew	更新ボタン					○	
削除	remove	削除（ツールバー）					○	
検索基準日変更	setSearchCriteriaDate	初期表示、検索基準日変更、クリアボタン		○			○	
検索基準会社 変更	setSearchCriteriaCompany	初期表示、検索基準会社変更、クリアボタン		○			○	

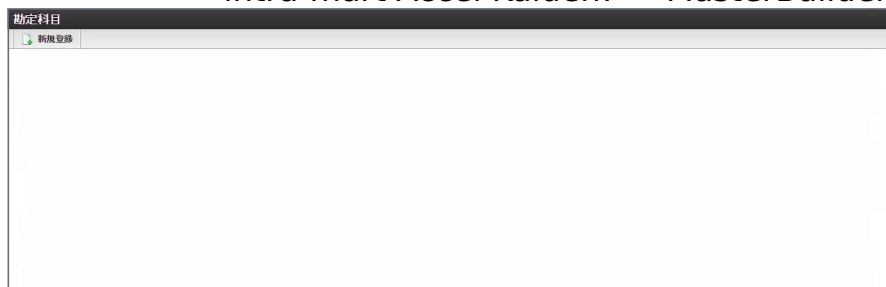
名称	ID	実行元	メイン	検索	一覧	期間	編集	ボタン
検索基準口 ケール変更	setSearchCriteriaLocale	初期表示、検索基準口ケール変更、クリアボタン		○			○	
検索条件保存	setSearchCriteria	一覧検索		○				
ソート条件保存	setSearchOrder	タイトル行		○				
期間有効化	enabledTerm	無効アイコン				○		
期間無効化	disabledTerm	有効アイコン				○		
期間分割	divideTerm	期間分割ダイアログ				○		
期間変更	moveTerm	期間変更ダイアログ				○		

メイン

各部品のインクルード元のJSP（メイン）は、「[拡張設定](#)」の「マスタメンテナンス画面のJSPパス」に設定したJSPが表示されます。
(未設定の場合は「/kaiden/v2/base/generic/master/maintenance.jsp」を使用します。)

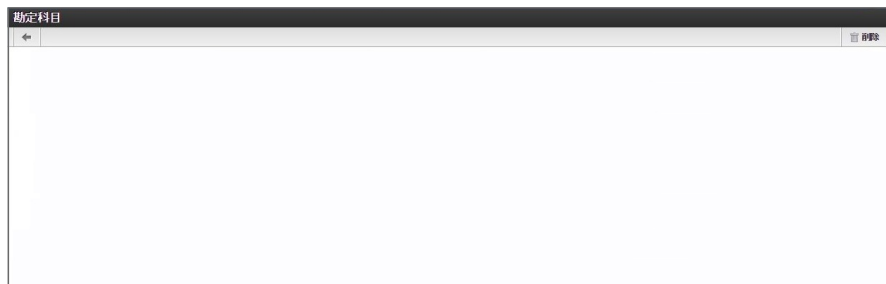
各部品のインクルード元のJSP（メイン）では、各部品のインクルードや一覧画面・登録／更新画面の切り替え、ツールバーなどを実装しています。

一覧画面



- **新規登録アイコン**
クリックすると、登録画面に切り替えます。

登録・更新画面



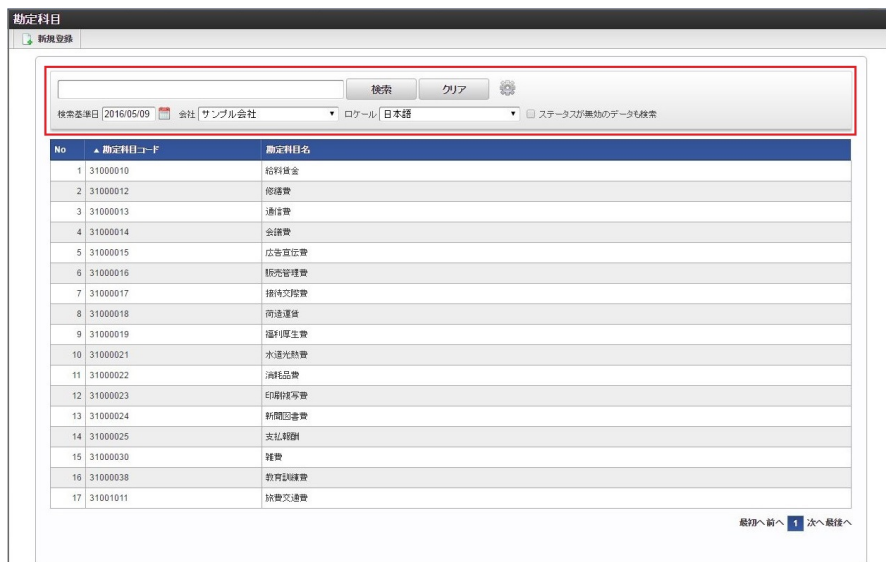
- **一覧に戻るアイコン**
クリックすると、一覧画面に切り替えます。
- **削除アイコン**
クリックすると、表示中のマスターデータを削除します。

検索部品

検索部品は、「[拡張設定](#)」の「マスターメンテナンス画面（検索部品）のJSPパス」に設定したJSPが表示されます。
（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceSearch.jsp」を使用します。）

また、一覧検索時に使用されるSQLは、「[拡張設定](#)」の「一覧検索用SQL」に設定したSQLを使用します。
（未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。）

検索部品では、検索キーワード、検索基準日などの検索時に使用する検索条件と、一覧検索ボタン、および一覧のクリアボタンと各ボタンクリック時の処理を実装しています。



- **検索キーワード**
マスターデータ（コードや名称）に含まれるキーワードを入力します。
未入力の場合は、検索キーワードを検索条件に含みません。
- **検索ボタン**
クリックすると、一覧検索処理を行い一覧部品に検索結果を表示します。
- **クリアボタン**
クリックすると、一覧部品に表示中の一覧検索結果をクリアし、検索条件を画面表示時の値に設定します。
- **検索条件表示／非表示アイコン**
クリックすると、検索条件の表示／非表示を切り替えます。
- **検索基準日**
一覧検索時の検索基準日を設定します。
- **検索基準会社**
一覧検索時の検索基準会社を設定します。

- **検索基準ロケール**
一覧検索時の検索基準ロケールを設定します。
- **無効データ検索有無**
一覧検索結果に無効のデータを含むか否かを設定します。

検索キーワード以外の検索条件は、マスタの特性（会社化、期間化、国際化）に合わせて表示／非表示を切り替えます。
また、各検索条件には、画面表示後すぐに検索可能となる様に初期値が設定されます。

検索条件	対象特性	初期値
検索キーワード	なし	空欄
検索基準日	期間化	システム日付
検索基準会社	会社化	ログインユーザの選択会社
検索基準ロケール	国際化	ログインユーザのロケール
無効データの検索有無	期間化	OFF（無効なデータを検索しない）

各特性に応じた検索部品の表示内容は次の通りです。

- 特性なし

検索

クリア

- 会社化

検索

クリア

会社 サンプル会社

- 期間化

検索

クリア

検索基準日 2016/05/09 ☐ ステータスが無効のデータも検索

- 国際化

検索

クリア

ロケール 日本語

- 会社化、期間化

検索

クリア

検索基準日 2016/05/09 会社 サンプル会社 ☐ ステータスが無効のデータも検索

- 会社化、国際化

検索

クリア

会社 サンプル会社 ロケール 日本語

- 期間化、国際化

検索

クリア

検索基準日 2016/05/09 ロケール 日本語 ☐ ステータスが無効のデータも検索

- 会社化、期間化、国際化

検索

クリア

検索基準日 2016/05/09 会社 サンプル会社 ロケール 日本語 ☐ ステータスが無効のデータも検索

一覧部品

一覧部品は、「[拡張設定](#)」の「マスタメンテナンス画面（一覧部品）のJSP/パス」に設定したJSPが表示されます。
（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceList.jsp」を使用します。）

一覧部品では、一覧検索結果の表示や表示内容のソート、更新画面への遷移などを実装しています。
表示される項目は、「[メンテナンス設定](#)」に設定された項目となります。

- ヘッダー行
クリックすると、一覧検索された検索結果のソート方法を変更します。
- 明細行
クリックすると、対象のマスタデータを更新する画面に切り替えます。
- ページャ
クリックすると、一覧表示の表示ページを切り替えます。

コラム

一覧部品では、「メンテナンス設定」を使用して、項目の制御を行います。

期間部品

期間部品は、「拡張設定」の「マスタメンテナンス画面（期間部品）のJSPパス」に設定したJSPが表示されます。
（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceTerm.jsp」を使用します。）

また、期間検索時に使用されるSQLは、「拡張設定」の「期間検索用SQL」に設定したSQLを使用します。
（未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。）

一覧部品では、対象のマスタデータの期間表示や期間の有効化／無効化処理、期間分割、期間変更等を実装しています。

- 有効化・無効化アイコン
クリックすると、対象期間の有効／無効が切り替わります。
緑丸：有効な期間を表します。
白丸：無効な期間を表します。
- 期間
対象期間の開始日、終了日を表示します。
- 対象・非対象アイコン
クリックすると、編集部品に対象の期間のデータを表示します。
- 期間分割リンク
クリックすると、次の期間分割ダイアログを表示します。
期間分割ダイアログで分割日を入力し、「実行」ボタンをクリックすることで対象の期間を分割します。

- 期間変更リンク
クリックすると、次の期間変更ダイアログを表示します。
期間変更ダイアログで新開始日、新終了日を入力し、「実行」ボタンをクリックすることで対象の期間をコピーし新たな期間として登録します。
この時、新開始日-新終了日と重複する期間が存在する場合は、期間変更や期間削除を行います。

コラム

期間部品は、期間化マスタの場合のみ表示されます。
対象のマスタが期間化マスタではない場合、編集画面は次の様に表示されます。

編集部品

編集部品は、「[拡張設定](#)」の「マスタメンテナンス画面（編集部品）のJSPパス」に設定したJSPが表示されます。
（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceEdit.jsp」を使用します。）

また、対象検索（更新データ検索）時に使用されるSQLは、「[拡張設定](#)」の「対象検索用SQL」に設定したSQLを使用します。
（未設定の場合は「META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql」を使用します。）

編集部品では、新規登録や更新用のマスタデータに対し、登録処理や更新処理、削除処理等を実装しています。
表示される項目は、「[メンテナンス設定](#)」に設定された項目となります。

会社、期間（開始日・終了日）項目は、「[メンテナンス設定](#)」の設定有無に関わらず各特性のマスタの場合に表示されます。

■ 特性なし

■ 会社化

■ 期間化

- 会社化、期間化

開始日 *	終了日 *	1900/01/01	-	2999/12/31
会社 *	サンプル会社			
勘定科目コード *				
勘定科目名 *	日本語			
	英語			
	中国語 (中華人民共和国)			

ボタン部品

ボタン部品は、「[拡張設定](#)」の「マスタメンテナンス画面（ボタン部品）のJSPパス」に設定したJSPが表示されます。（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceButton.jsp」を使用します。）

ボタン部品では、登録、更新、一覧に戻るボタンの処理等を実装しています。

新規登録時

勘定科目	
期間一覧 登録 一覧に戻る	開始日 * 終了日 * 1900/01/01 - 2999/12/31 会社 * サンプル会社 勘定科目コード * 勘定科目名 * 日本語 英語 中国語 (中華人民共和国)

更新時

勘定科目	
期間一覧 1900/01/01 - 1999/12/31 期間分割 期間変更 2000/01/01 - 2099/12/31 期間分割 期間変更 2100/01/01 - 2199/12/31 期間分割 期間変更 2200/01/01 - 2999/12/31 期間分割 期間変更	

- **登録ボタン**
クリックすると、登録処理を実行します。
- **更新ボタン**
クリックすると、更新処理を実行します。
- **一覧に戻るボタン**
クリックすると、一覧画面に切り替えます。

拡張部品

拡張部品は、「[拡張設定](#)」の「マスタメンテナンス画面（拡張部品）のJSPパス」に設定したJSPが表示されます。（未設定の場合は「/WEB-INF/views/kaiden/v2/base/generic/master/maintenanceExtension.jsp」を使用します。）

拡張部品は、次の様な修正を標準の各JSPを修正することなく実装するために用意しています。

- 各項目のクリックイベントやチェンジイベント追加
- ツールバーへのアイコン追加
- 独自クライアントバリデーションの追加

上記の様な拡張を行う際には、各JSPのガジェットIDなどが必要となります。他のJSPを操作するために次の値が連携されます。

連携パラメータ	詳細
<code>\${param.main_gadget_id}</code>	メンテナンス画面のガジェットID
<code>\${param.search_gadget_id}</code>	検索部品のガジェットID
<code>\${param.search_gadget_block_id}</code>	検索部品のブロックID
<code>\${param.list_gadget_id}</code>	一覧部品のガジェットID
<code>\${param.list_gadget_title_block_id}</code>	一覧部品のヘッダーブロックID
<code>\${param.list_gadget_data_block_id}</code>	一覧部品の明細ブロックID

<code>\${param.term_gadget_id}</code>	期間部品のガジェットID
<code>\${param.term_gadget_title_block_id}</code>	期間部品のヘッダーブロックID
<code>\${param.term_gadget_data_block_id}</code>	期間部品の明細ブロックID
<code>\${param.term_gadget_divide_dialog_id}</code>	期間部品の期間分割ダイアログID
<code>\${param.term_gadget_move_dialog_id}</code>	期間部品の期間変更ダイアログID
<code>\${param.edit_gadget_id}</code>	編集部品のガジェットID
<code>\${param.edit_gadget_block_id}</code>	編集部品のブロックID
<code>\${param.button_gadget_id}</code>	ボタン部品のガジェットID
<code>\${param.button_gadget_block_id}</code>	ボタン部品のブロックID
<code>\${param.gadget_class}</code>	拡張部品のガジェットクラス
<code>\${param.gadget_variation}</code>	拡張部品のガジェットバリエーション
<code>\${param.gadget_instance}</code>	拡張部品のガジェットインスタンス
<code>\${param.gadget_id}</code>	拡張部品のガジェットID
<code>\${param.gadget_block_id}</code>	拡張部品のブロックID

サーバーサイド

Controller

初期表示時のControllerクラスは、アクセスしたURLから自動判定されます。

「`http://<HOST>:<PORT>/<CONTEXT_PATH>/kaiden/v2/generic/master?masterId=%マスタID%`」にアクセスした場合は、「`jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.generic.MasterController`」が使用されます。

その他の処理では、「[拡張設定](#)」の「マスタメンテナンス用の処理URL」に設定したURLから自動判定されます。
(未設定の場合は「`kaiden/v2/generic/master`」を使用します。)

Controllerクラスでは、認可判定とLogicクラスの対象メソッド起動を行います。

Controllerクラスのメソッドと主な起動処理は次の通りです。

処理名	メソッド	主な起動処理
初期表示	<code>index</code>	初期表示時
一覧検索	<code>search</code>	検索部品の検索ボタンクリック時
対象検索	<code>select</code>	一覧部品のデータ行クリック時（更新対象の検索）
登録	<code>entry</code>	ボタン部品の登録ボタンクリック時
更新	<code>renew</code>	ボタン部品の更新ボタンクリック時
削除	<code>remove</code>	ツールバーの削除リンククリック時
有効化	<code>toEnabled</code>	期間部品の無効アイコンクリック時
無効化	<code>toDisabled</code>	期間部品の有効アイコンクリック時
期間検索	<code>selectTerm</code>	一覧部品のデータ行クリック時（更新対象の検索）
期間分割	<code>divideTerm</code>	期間部品の期間分割実行時
期間変更	<code>moveTerm</code>	期間部品の期間変更実行時
バリデート（認可）	<code>authorize</code>	一覧部品のデータ行クリック時、ツールバーの新規登録アイコンクリック時



コラム

追加のメソッドを作成する場合や、独自の認可判定を行う場合はControllerクラスの拡張を行います。
メニューからの遷移URLは通常のSpringMVCの仕組みに合わせて実装してください。
登録や更新などの処理ではAjaxのURLを使用しますが、
このURLを変更する場合は「[拡張設定](#)」の「マスタメンテナンス用の処理URL」を変更してください。

Form

Controllerクラスへアクセスした際のリクエストパラメータは、Formクラス(`jp.co.slcs.kaiden.v2.base.feature.form.kaiden.generic.MasterForm`)に格納されます。

Formクラスでは、マスタID、マスタ拡張ID、検索キーワードの他に、汎用的にデータを取得するためにJSON化したデータを保持しています。

Controllerクラスの各メソッド呼出し時に格納されるJSONデータの内容は次の通りです。

- **初期表示（index）**
マスタメンテナンス画面への遷移時に引き渡したパラメータ（キーが「`json`」のパラメータ）が設定されます。

■ 一覧検索 (search)

パラメータ名	パラメータキー	パラメータの取得元	補足
マスタID	masterId	初期表示時のリクエストパラメータ	
マスタ拡張ID	extensionId	初期表示時のリクエストパラメータ	
検索キーワード	searchKeyword	検索部品の検索キーワード	
検索基準会社コード	searchCriteriaCompany	検索部品の検索基準会社	会社化特性の場合のみ
検索基準日	searchCriteriaDate	検索部品の検索基準日	期間化特性の場合のみ
無効データ検索有無	searchWithDisabled	検索部品の無効データ検索有無	期間化特性の場合のみ - チェックありの場合は「true」 - チェックなしの場合は「false」
検索基準ロケール	searchCriteriaLocale	検索部品の検索基準ロケール	国際化特性の場合のみ
ソート条件	orderBy	MasterBuilder定義	一覧部品でソート順を変更した場合は、変更後のソート条件
ページ番号	numberOfPage	表示中のページ番号	初回検索時は「1」
表示件数	limit	MasterBuilder定義	
表示開始レコード数	offset	「(ページ番号 - 1) x 表示件数」の計算結果	

■ 対象検索 (select)

更新画面遷移時は、対象行の一覧検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。
表示期間変更時は、対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。

■ 登録 (entry)

「KAIDEN.gadgetMan.getDataSet("%編集部品のガジェットID%")」の取得結果（編集部品への入力値）をJSON化して設定します。

■ 更新 (renew)

「KAIDEN.gadgetMan.getDataSet("%編集部品のガジェットID%")」の取得結果（編集部品への入力値）をJSON化して設定します。

■ 削除 (remove)

対象検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。

■ 有効化 (toEnabled)

対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。

■ 無効化 (toDisabled)

対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。

■ 期間検索 (selectTerm)

更新画面遷移時は、対象行の一覧検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。
表示期間変更時は、対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）が設定されます。

■ 期間分割 (divideTerm)

対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）に次の値を追加した内容が設定されます。

パラメータ名	パラメータキー	パラメータの取得元	補足
分割日	divideDate	期間分割ダイアログの分割日	

■ 期間変更 (moveTerm)

対象行の期間検索時のレスポンスパラメータ（キーが「json」のパラメータ）に次の値を追加した内容が設定されます。

パラメータ名	パラメータキー	パラメータの取得元	補足
新開始日	newStartDate	期間変更ダイアログの新開始日	
新終了日	newEndDate	期間変更ダイアログの新終了日	

■ バリデート (認可) (authorize)

JSONデータは設定されません。

Logic

Logicクラスは、「[拡張設定](#)」の「Logicクラス」に設定したクラスを使用します。
(未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。)

Logicクラスでは、検索・登録・更新・削除などの処理を行います。
Logicクラスのメソッドと主な処理内容次の通りです。

処理名	メソッド	主な処理
初期表示	index(GenericMasterSettings settings, AbstractGenericMasterForm form)	メンテナンス画面の各項目設定の読み出し
一覧検索	search(GenericMasterSettings settings, AbstractGenericMasterForm form)	一覧表示用のデータ検索

対象検索	<code>select(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	更新用のデータ検索
登録	<code>entry(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	入力データの新規登録
更新	<code>renew(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	編集データの更新
削除	<code>remove(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	表示データの削除
有効化	<code>toEnabled(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	対象期間データの有効化
無効化	<code>toDisabled(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	対象期間データの無効化
期間検索	<code>selectTerm(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	期間一覧用のデータ検索
期間分割	<code>divideTerm(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	対象期間データの期間分割
期間変更	<code>moveTerm(GenericMasterSettings settings, AbstractGenericMasterForm form)</code>	対象期間データの期間変更

i コラム

期間変更処理の変更点

MasterBuilderの期間変更処理は、従来の期間変更処理と異なります。

- 従来の期間変更処理
既に登録済みのデータの期間を変更し、空白となった期間がある場合は前後の期間を延長する。
(空白となった期間は無効状態)
- MasterBuilderの期間変更処理
既に登録済みのデータをコピーし、入力された新时期間のデータとして追加する。
(期間コードは新たに採番)

i コラム

APIを使用したメンテナンス機能を実現する場合や、独自のロジックを追加する場合はLogicクラスの拡張を行います。
Controllerクラスからの起動を想定するメソッドを追加する場合はControllerクラスの拡張が必要です。

データの補完や成型を行いたい場合は、後述のConverterクラスの拡張でも対応可能です。
Logicクラスの各メソッドへのリクエストパラメータやレスポンスパラメータに対してデータ補完や成型が可能です。

Manager

Managerクラスは、「[拡張設定](#)」の「Managerクラス」に設定したコンポーネントを使用します。
(未設定の場合は「masterManager」を使用します。)

Managerクラスでは、SQLを利用した検索処理を行います。

基本的に拡張の必要はありませんが、標準のManagerクラスでは単一のテーブルに対して処理を実装しているため、複数のテーブルに跨る処理を実装する場合に拡張を行います。

(検索時のJoinなどは複数テーブルに跨りますが、SQLで実現可能なため、拡張の必要はありません。)

処理名	メソッド
一覧検索	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria, int limit, int offset)</code>
対象検索	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)</code>
登録	■ <code>insert(GenerateService<? extends GenerateEntity> service, List<? extends GenerateEntity> entityList)</code>
更新	■ <code>update(GenerateService<? extends GenerateEntity> service, List<? extends GenerateEntity> entityList, String... includes)</code>
削除	■ <code>delete(GenerateService<? extends GenerateEntity> service, Class<? extends GenerateEntity> entityClass, List<Where> criteria)</code>
有効化	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)</code> ■ <code>update(GenerateService<? extends GenerateEntity> service, List<? extends GenerateEntity> entityList, String... includes)</code>
無効化	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)</code> ■ <code>update(GenerateService<? extends GenerateEntity> service, List<? extends GenerateEntity> entityList, String... includes)</code>
期間検索	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)</code>
期間分割	■ <code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)</code> ■ <code>insert(GenerateService<? extends GenerateEntity> service, GenerateEntity entity)</code> ■ <code>update(GenerateService<? extends GenerateEntity> service, GenerateEntity entity, String... includes)</code>

期間変更

- `select(GenerateService<? extends GenerateEntity> service, Class<? extends GenerateEntity> entityClass, List<Where> criteria, List<OrderByItem> order)`
- `select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria)`
- `insert(GenerateService<? extends GenerateEntity> service, GenerateEntity entity)`
- `update(GenerateService<? extends GenerateEntity> service, GenerateEntity entity, String... includes)`
- `delete(GenerateService<? extends GenerateEntity> service, GenerateEntity entity)`

Converter

Converterクラスは、「[拡張設定](#)」の「Converterクラス」に設定したクラスを使用します。
(未設定の場合はConverterクラスを使用しません。)

Converterクラスでは、Logicクラスへのリクエストパラメータ、Logicクラスからのレスポンスパラメータに対し、補完や成型が行えます。
設定を行わない場合は、補完や成型は行われません。

- 画面表示は%表記だが、データ上は数値として保持 (80%→0.8)
- 固定値の補完
- カンマ区切りへの変換

i コラム

文字列と日付の変換や、文字列と日付の変換はConverterクラスではなく、「[項目設定](#)」の「コンバート設定」を利用してください。
詳細は「[項目値のコンバート](#)」を参照してください。

! 注意

Converterクラスの処理は各データに対して行われるため、データベースの特定データを補完する場合はLogicクラスで実装を行ってください。
Converterクラスに実装した場合、一覧検索のレスポンスデータの補完などで一覧行の各行単位で処理が行われるため、SQLを実行する様な処理では、複数回SQLが実行されてしまい、レスポンス悪化が懸念されます。

シーケンス

サーバーサイドの処理シーケンス (概要) は次の通りです。

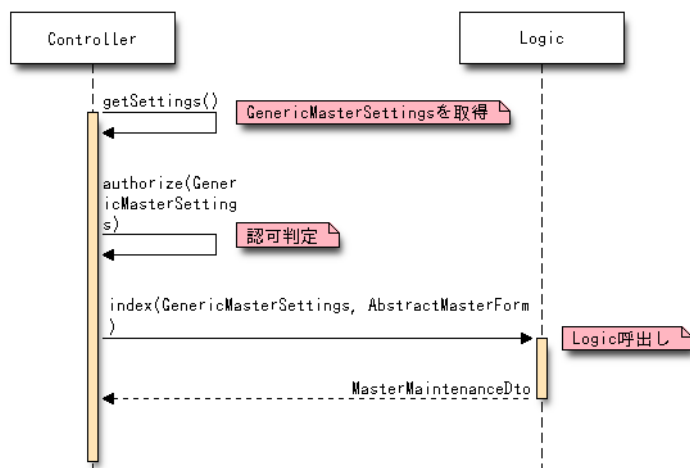
i コラム

シーケンスに記載の「Controller」「Logic」「Manager」の実際のクラスは次の通りです。
シーケンスの各処理は基底クラスで実装しています。

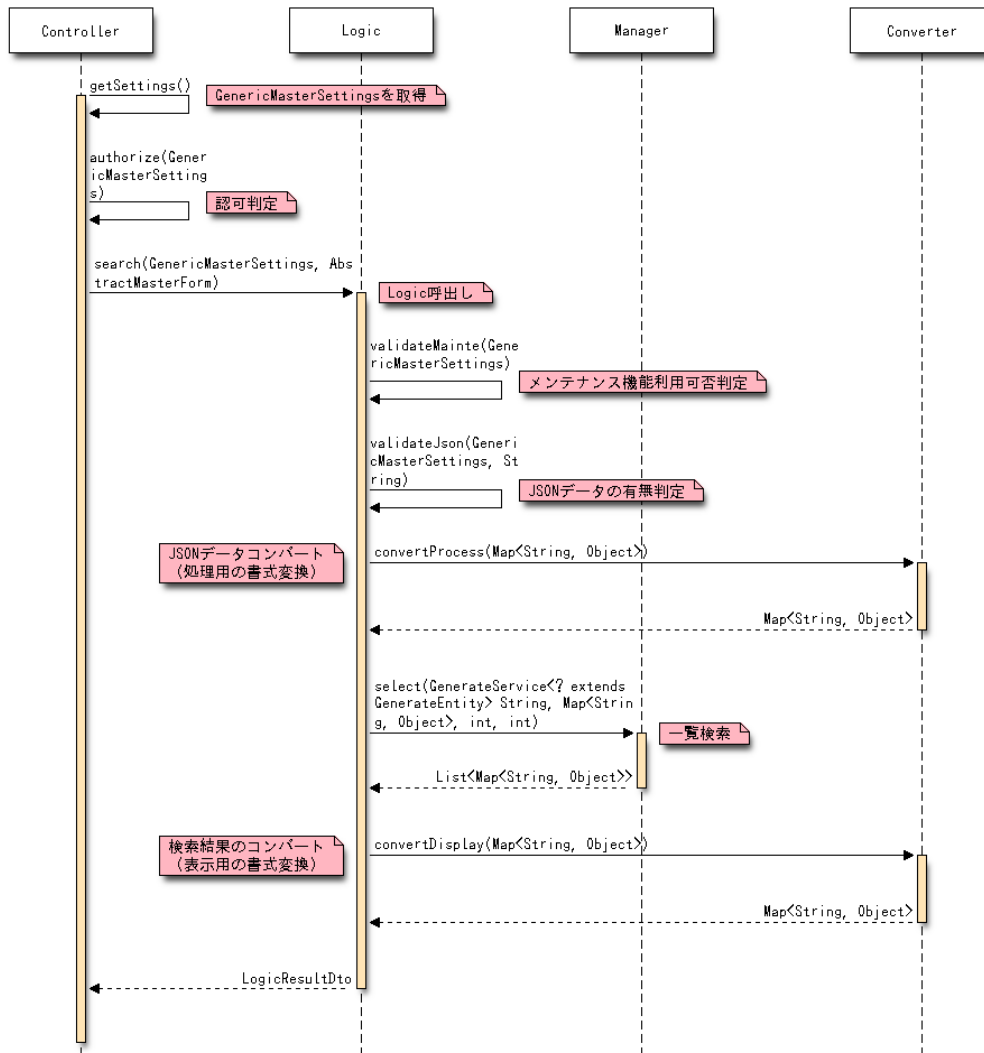
- **Controller**
基底クラス : `jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.generic.AbstractMasterController`
実装クラス : `jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.generic.MasterController`
- **Logic**
基底クラス : `jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.AbstractMasterLogicBase`
実装クラス : `jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic`
- **Manager**
IFクラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.MasterManager`
基底クラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.AbstractMasterManager`
実装クラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.impl.MasterManagerImpl`

「Converter」は、「[拡張設定](#)」の「Converterクラス」に設定していない場合は実行されません。

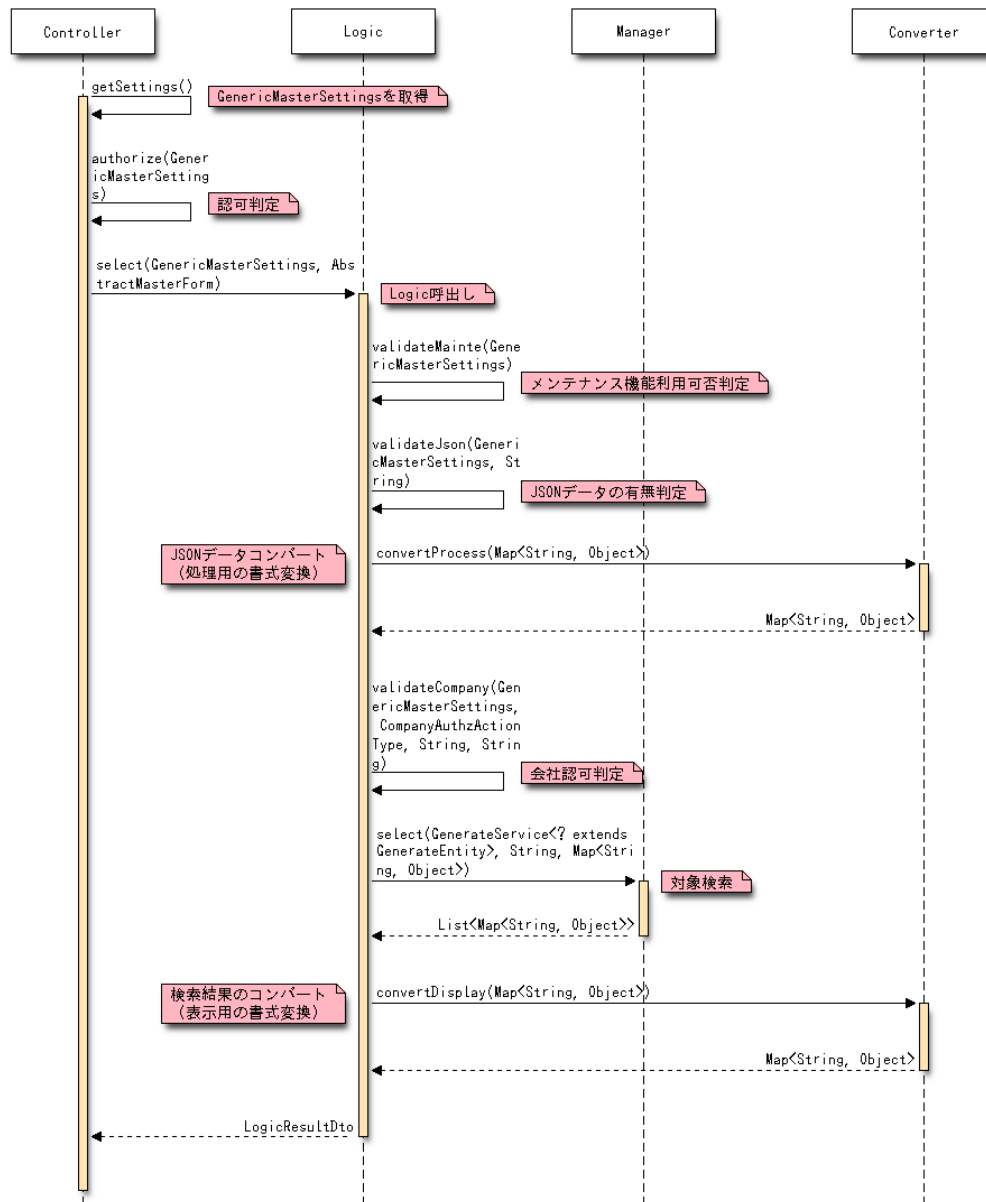
- 初期表示(index)



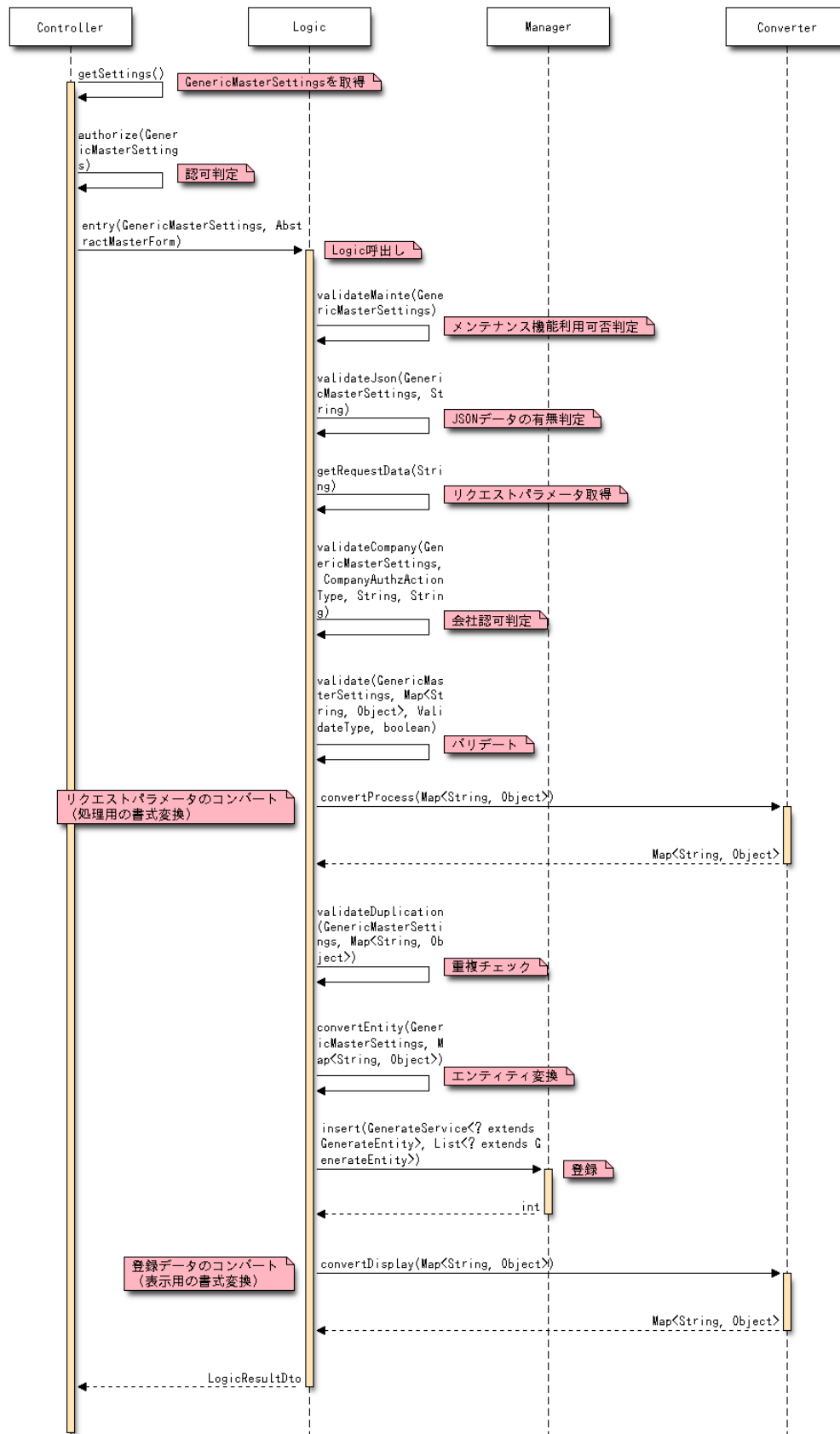
■ 一覧検索(search)



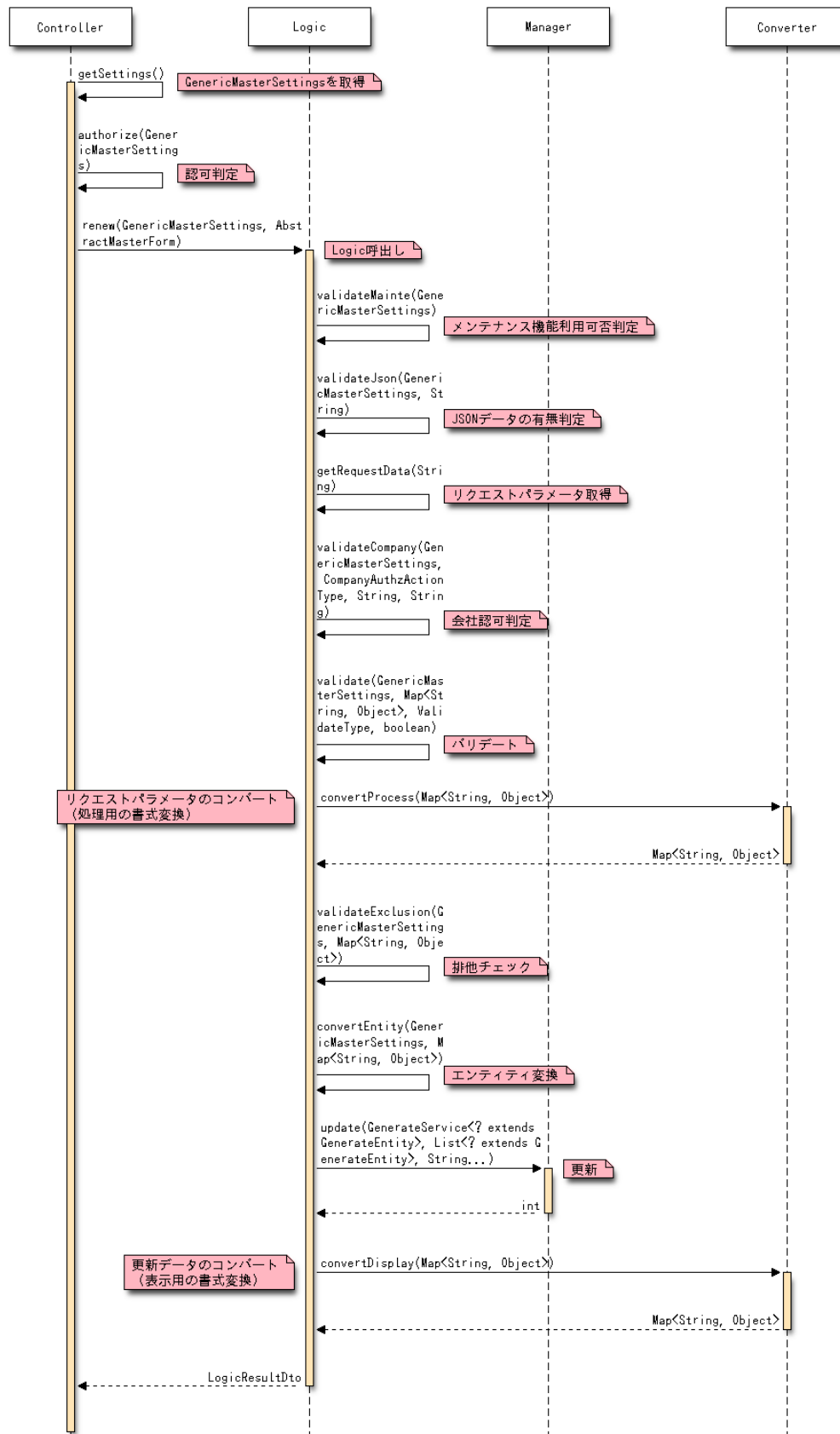
■ 対象検索(select)



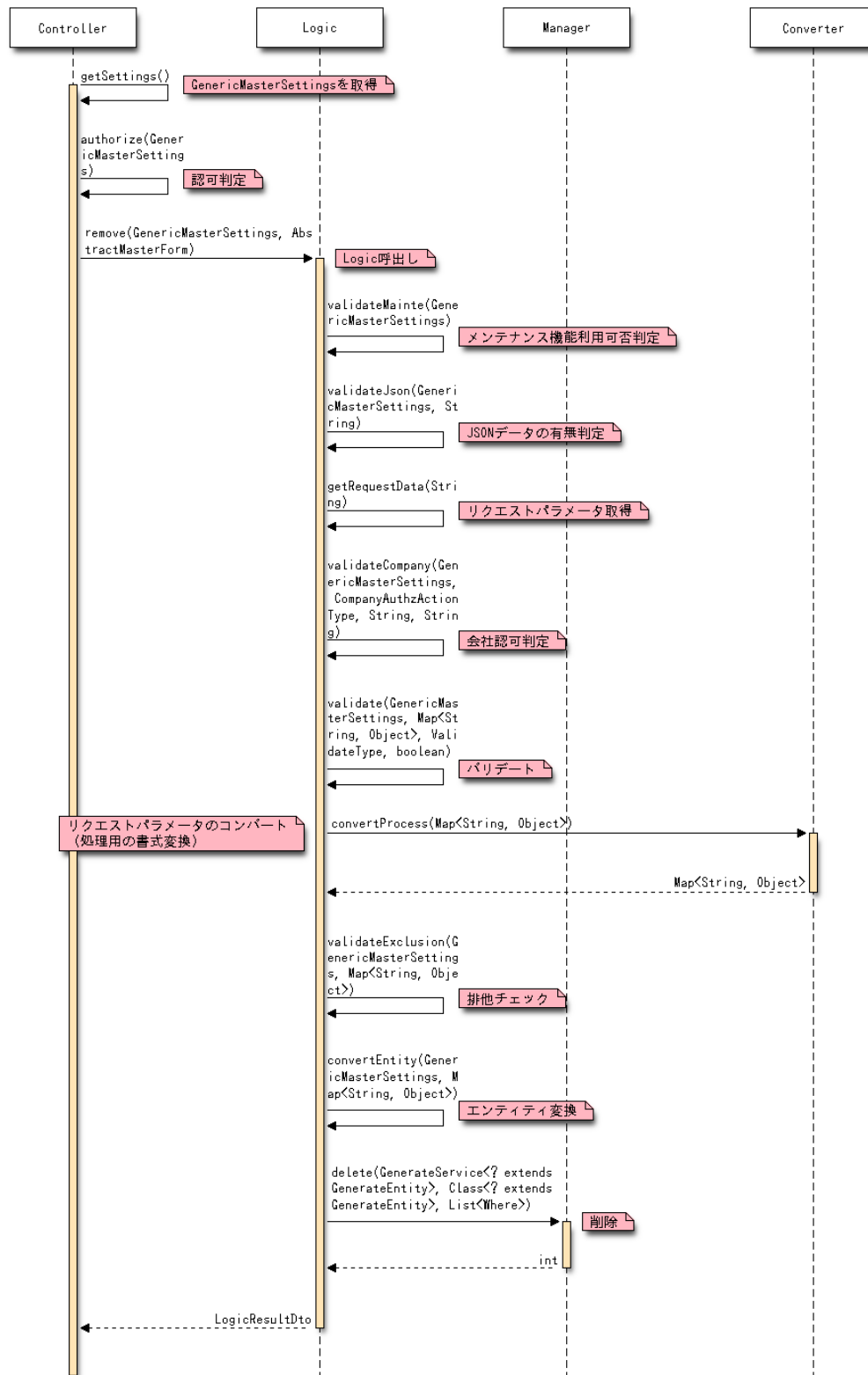
登録(entry)



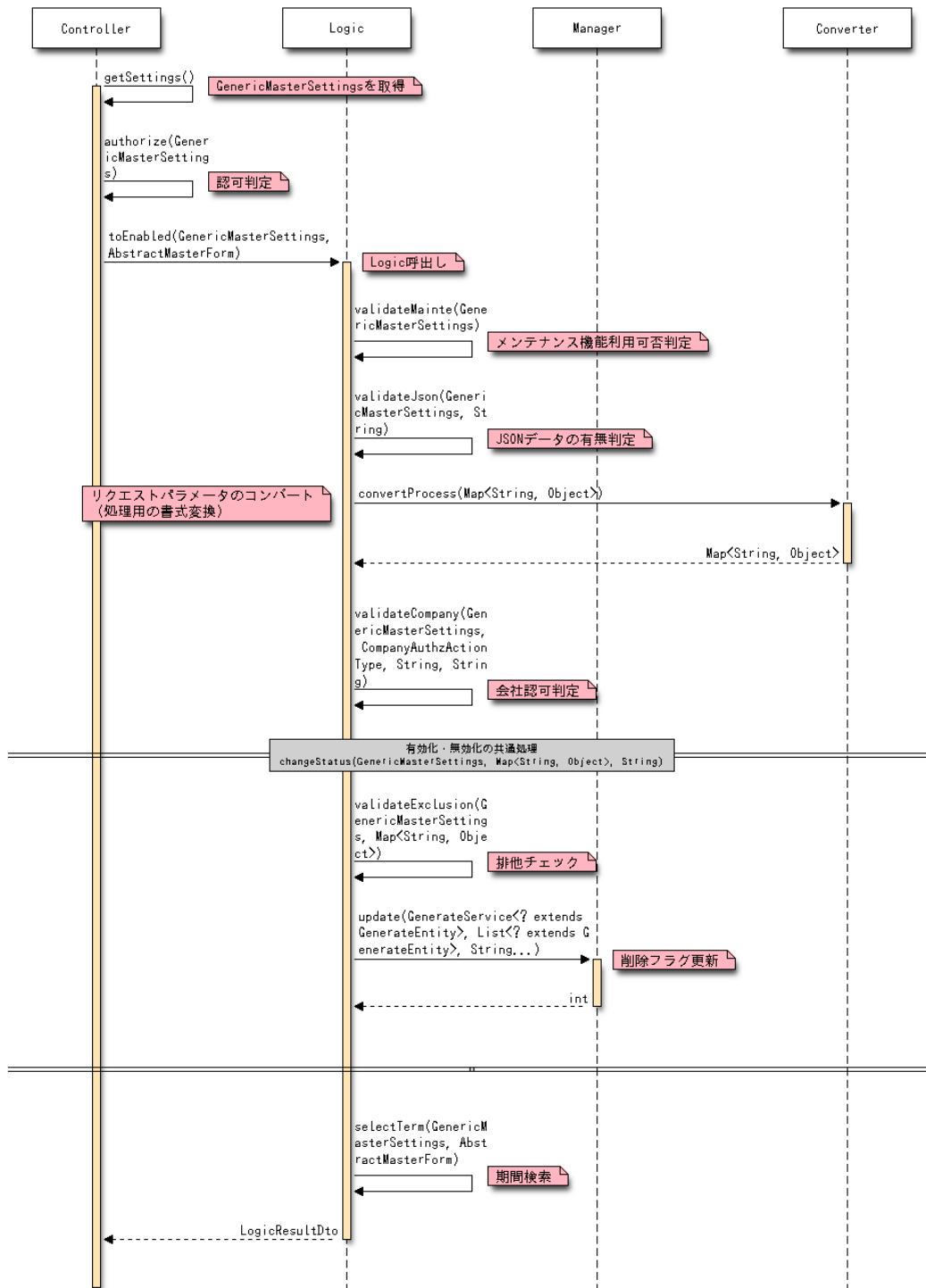
- 更新(renew)



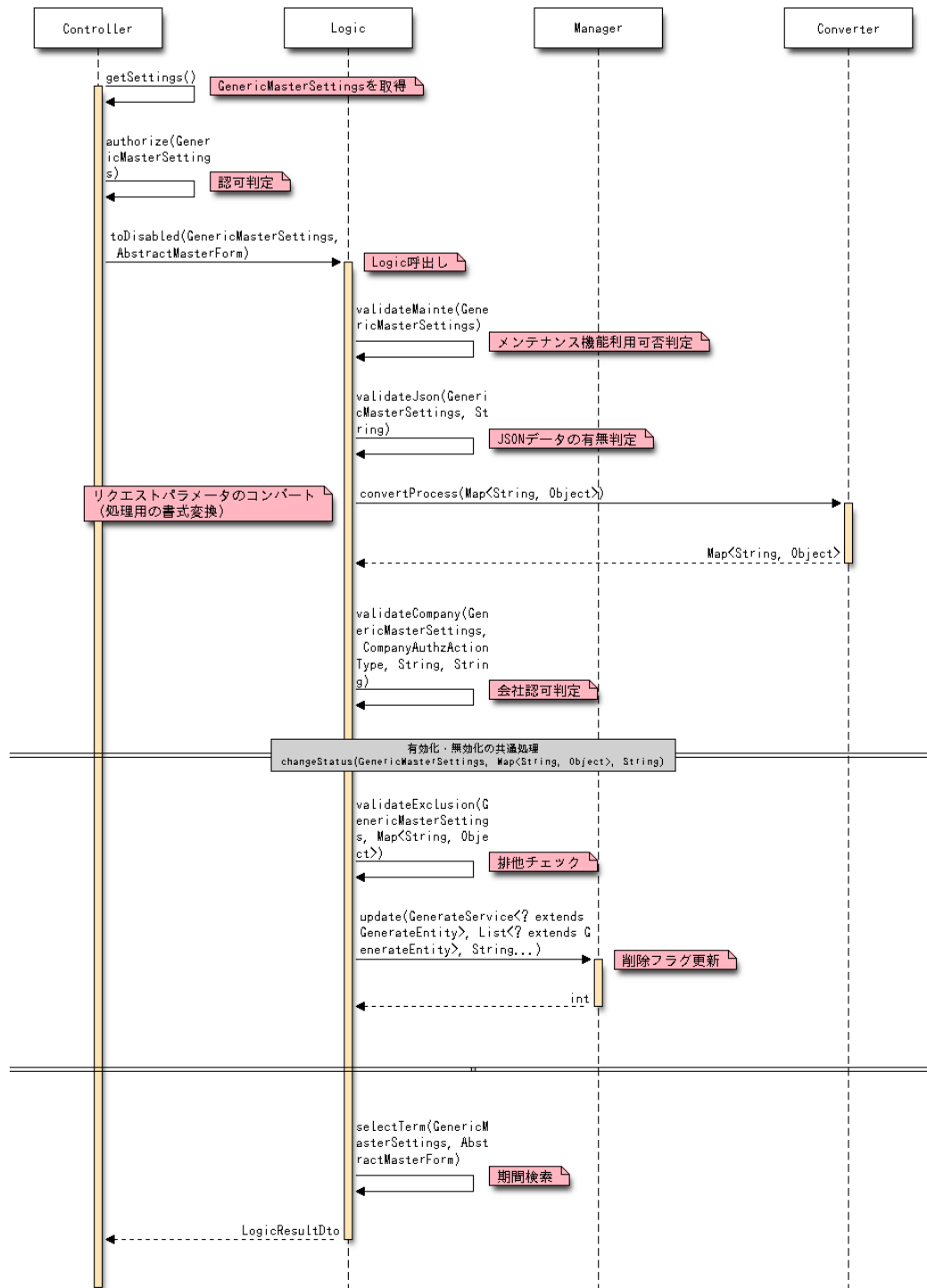
- 削除(remove)



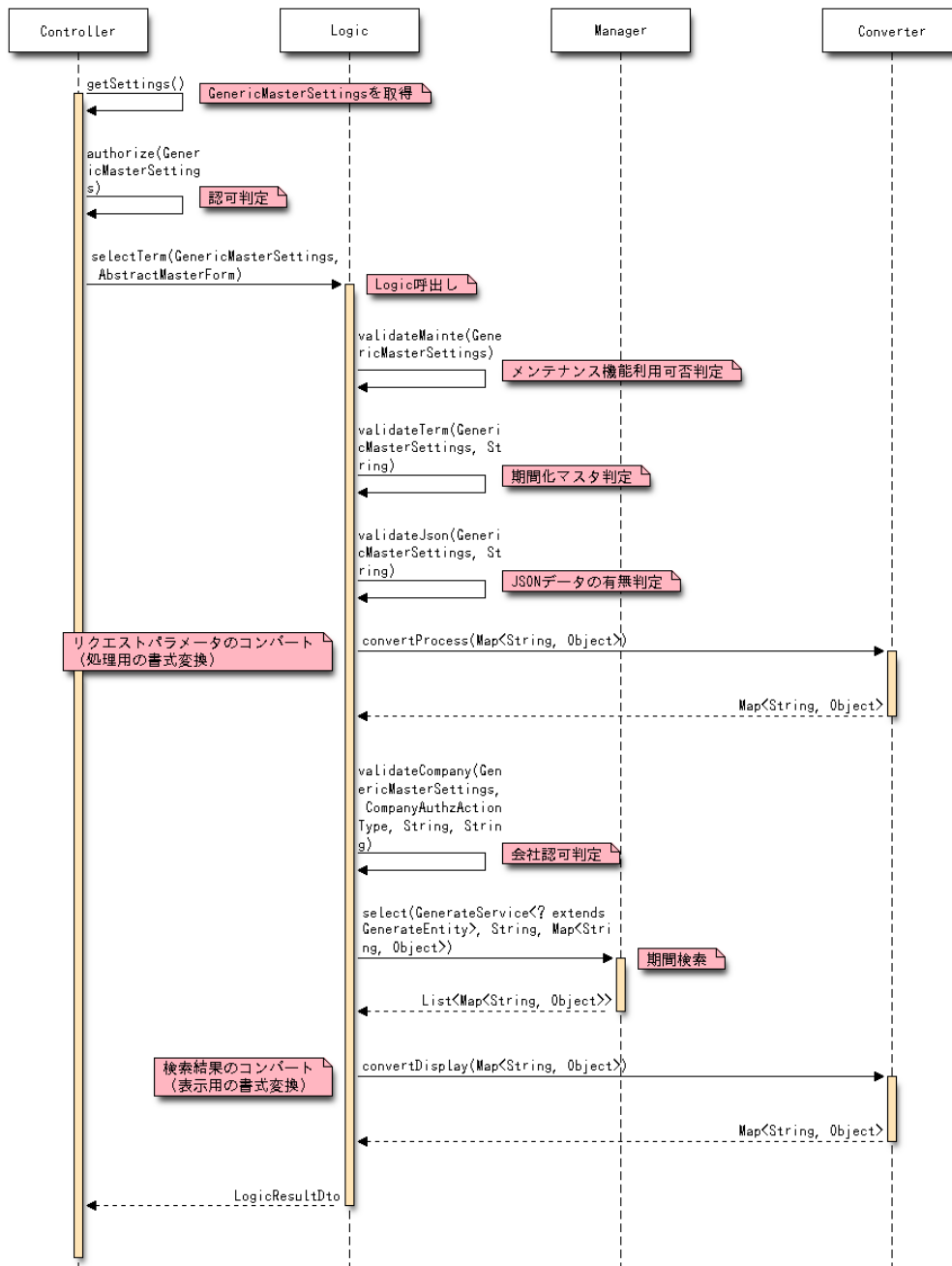
- 有効化(toEnabled)



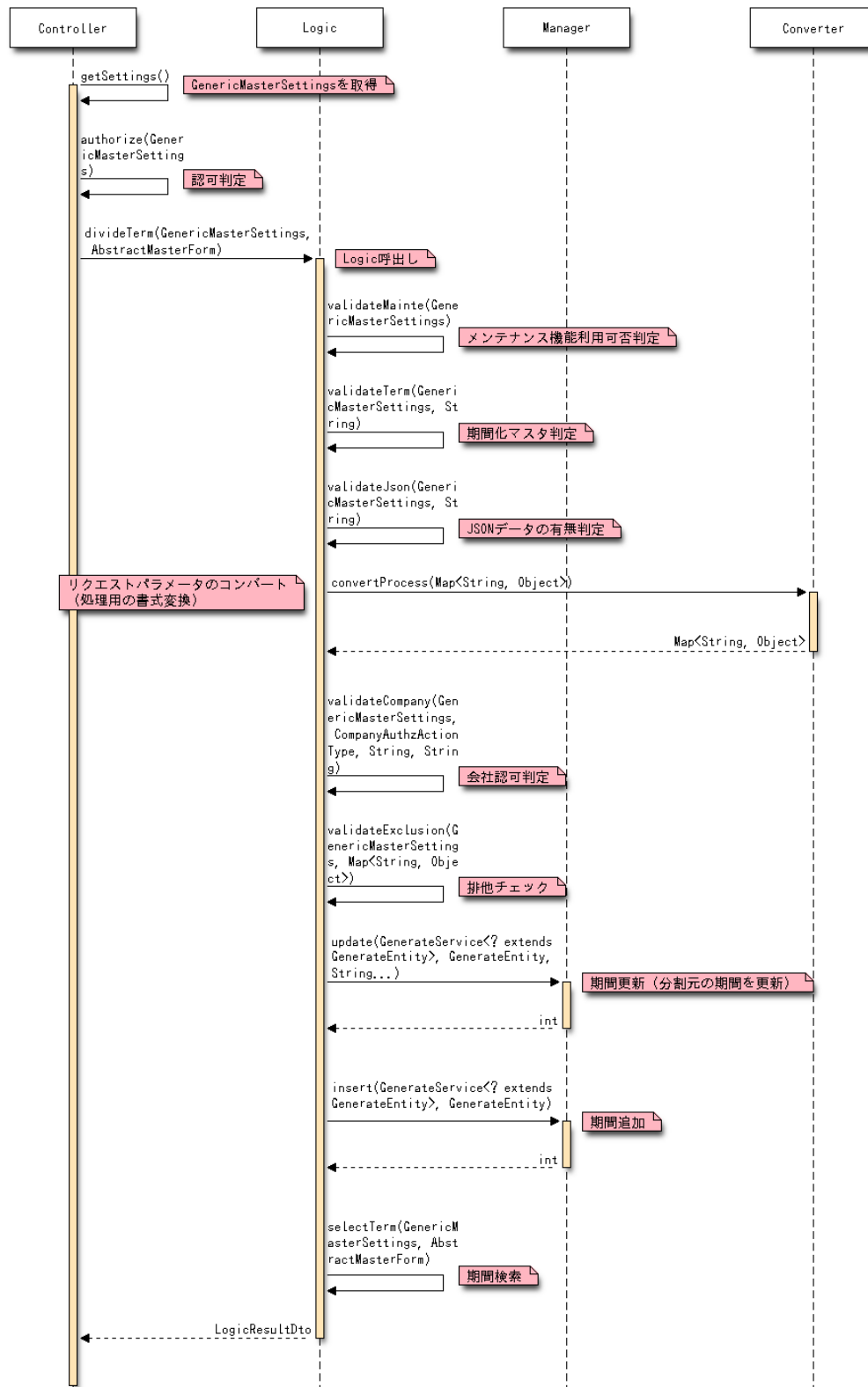
- 無効化(`toDisabled`)



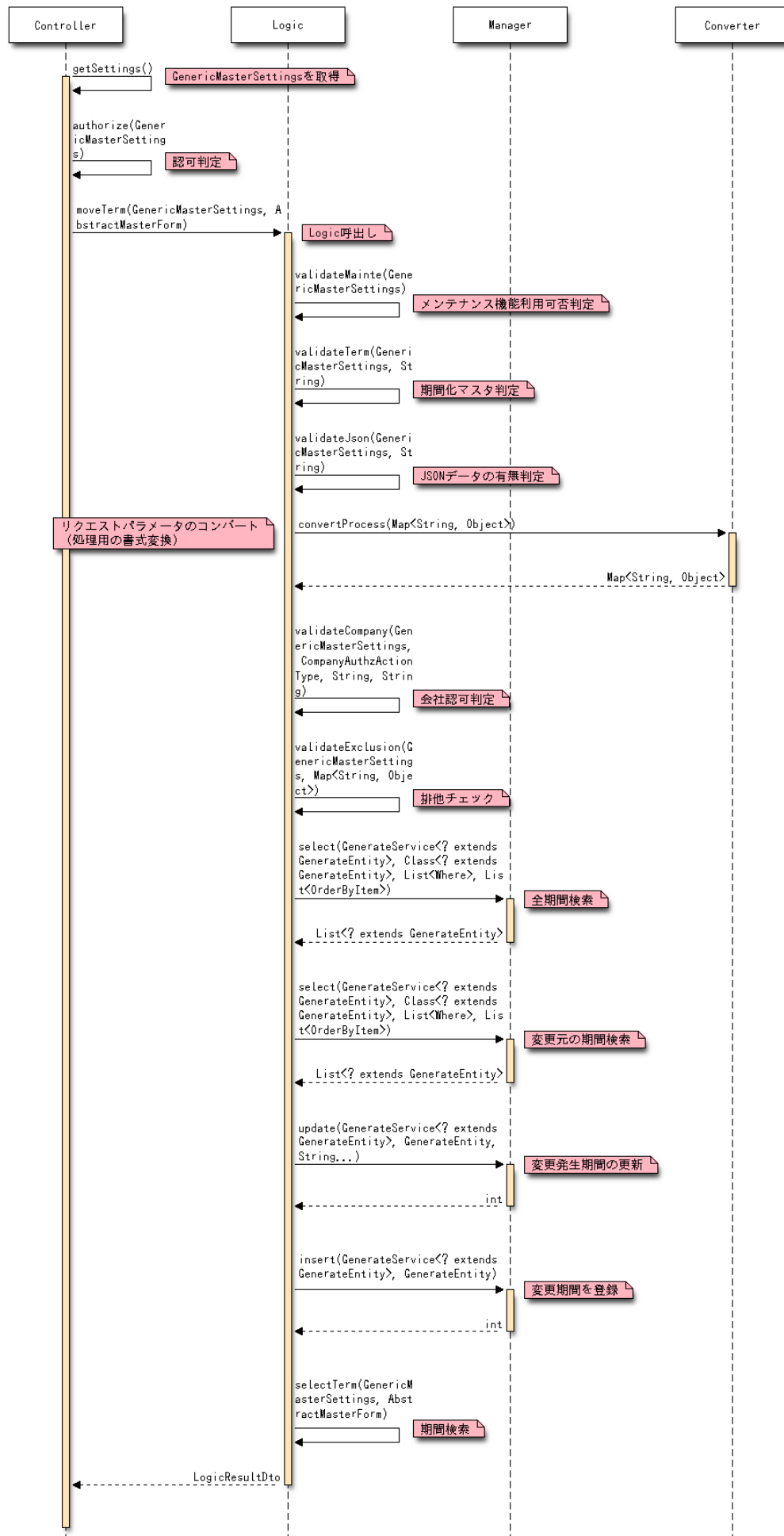
- 期間検索(selectTerm)



- 期間分割(`divideTerm`)



- 期間変更(moveTerm)



本項では、検索機能の機能詳細を説明します。

- クライアントサイド
 - 検索機能の使用方法
 - 初期値の指定方法
 - 関連エレメントの値変更時の再検索方法
 - 明細でセレクトボックス利用時の注意点
- サーバーサイド
 - Controller
 - Form
 - Logic
 - Manager
 - Converter
 - シーケンス

クライアントサイド

検索機能の使用方法

検索機能は、検索機能用のタグライブラリからを使用します。
検索機能用のタグライブラリの詳細は次の通りです。

- タグライブラリの設定
検索機能を使用するためには、次のタグライブラリの設定を読み込んでください。

```
<%@ taglib prefix="k_generic" uri="http://kaiden.slcs.co.jp/taglib/v2/generic" %>
```

- タグライブラリの実装サンプル
勘定科目マスタの検索機能を使用する場合のサンプルです。

```
<k_generic:master
masterId="account"
extensionId="{extensionId}"
searchType="search"
mode="0"
name="accountCd"
viewName="accountName"
surfaceKey=""
criteria="{ 'searchCriteriaCompany':'searchCriteriaCompany', 'searchCriteriaDate':'searchCriteriaDate', 'searchCriteriaLocale':'searchCriteriaLocale',
'accountCd':'accountCd', 'includeDisabled':'true', 'includeBlank':'true' }"
resultChain="{ 'accountCd':'accountCd', 'accountName':'accountName' }"
resultValue="accountName"
primaryTarget="accountCd"

refreshTarget="searchCriteriaCompany,searchCriteriaDate,searchCriteriaLocale"
gadgetId="{gadgetId}"
gadgetBlockId="{blockId}"
style="width:200px;"
/>
```

 **注意**

検索機能用のタグライブラリは、必ずガジェット内に実装してください。

- タグライブラリの属性値
検索機能用のタグライブラリでは、次の属性を設定します。

属性名	説明	必須
masterId	対象マスタのマスタIDを設定します。	○
extensionId	マスタ拡張IDを設定します。 本属性は任意設定となりますが、カスタマイズ性を考慮して任意の値設定を推奨します。 属性値は、使用箇所のガジェットIDなどを設定し、他の使用箇所と重複しないように設定してください。	
searchType	検索タイプを設定します。 未設定の場合は対象マスタのMasterBuilder定義から検索タイプを取得します。 ■ search （検索部品） ■ select （セレクトボックス） ■ radio （ラジオボタン）	
mode	表示モードを設定します。 ■ 0 （「input」エレメントを生成します。） ■ 1 （「label」エレメントを生成します。）	○

属性名	説明	必須
name	コード項目の「name」属性を設定します。 検索機能の検索タイプにより使用方法が異なります。 検索部品 検索タイプが「SEARCH（検索部品）」の場合は、コード設定用と名称設定用の2つのテキストボックスを生成します。 本属性に設定した値は、コード設定用テキストボックスの「name」属性となります。 コード設定用テキストボックスは、隠し項目（Hidden）となるため、後述の「resultChain」を使用し値を設定してください セレクトボックス 検索タイプが「SELECT（セレクトボックス）」の場合は、コード設定用のセレクトボックスと名称設定用のテキストボックスを生成します。 本属性に設定した値は、コード設定用テキストボックスの「name」属性となります。 ラジオボタン 検索タイプが「RADIO（ラジオボタン）」の場合は、ラジオボタンとラベルを検索結果分生成します。 本属性に設定した値に「_radio」を付与した値が、ラジオボタンの「name」属性となります。	○
viewName	名称項目の「name」属性を設定します。 検索機能の検索タイプにより使用方法が異なります。 検索部品 検索タイプが「SEARCH（検索部品）」の場合は、コード設定用と名称設定用の2つのテキストボックスを生成します。 本属性に設定した値は、名称設定用テキストボックスの「name」属性となります。 セレクトボックス 検索タイプが「SELECT（セレクトボックス）」の場合は、コード設定用のセレクトボックスと名称設定用のテキストボックスを生成します。 本属性に設定した値は、名称設定用テキストボックスの「name」属性となります。 名称設定用テキストボックスは、隠し項目（Hidden）となるため、後述の「resultChain」を使用し値を設定してください ラジオボタン 使用されません。	○
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。	
criteria	対象マスタへの検索条件を設定します。 形式：<code>criteria="{ '%キー%' : '%値%', '%キー%' : '%値%', ... }"</code> 「%キー%」には、次の様な内容を設定します。 検索基準会社コード（searchCriteriaCompany） 会社化特性のマスタの場合は、必須となります。 検索基準日（searchCriteriaDate） 未設定、または値が空の場合はシステム日付が補完されます。 検索基準ロケール（searchCriteriaLocale） 未設定、または値が空の場合はログインユーザのロケールが補完されます。 無効データ対象有無（includeDisabled） 未設定、または値が空の場合は無効なマスタデータを検索対象としません。 空白行設定有無（includeBlank） 検索タイプが「SELECT（セレクトボックス）」の場合のみ有効となります。 未設定、または値が空の場合はセレクトボックスに空白行を作成しません。 その他 上記の他に対象マスタの主キーや任意のデータが設定可能です。 検索機能では、設定された検索条件を、すべてマスタ検索用SQLに引き渡します。 「%値%」には、次の様な内容を設定します。 エレメントの「name」属性 「%値%」と同じ「name」属性が設定されたエレメントの「value」属性に設定された値に補完します。 固定値 「true/false」や文字列などが設定可能です。	○
resultChain	マスタデータ選択時に、複数のエレメントの「value」に設定するデータのキーを設定します。 形式：<code>resultChain="{ '%設定項目の「name」属性値%' : '%データの項目名%', '%設定項目の「name」属性値%' : '%データの項目名%', ... }"</code> 「%データの項目名%」に設定可能な値は、各タグライブラリの説明を参照してください。	

属性名	説明	必須
resultValue	検索機能の検索タイプにより使用方法が異なります。 検索部品 対象選択時に名称項目の「value」属性に設定するデータの項目名を設定します。 「%データの項目名%」に設定可能な値は、各タグライブラリの説明を参照してください。 セレクトボックス 使用されません。 ラジオボタン 画面表示時の初期値のコードを設定します。 ただし、明細行でラジオボタンを使用する場合は初期値の設定がされません。 初期値の設定が不要の場合、当項目の設定は不要です。	
primaryTarget	検索タイプがSEARCH（検索部品）の場合のみ使用されます。 検索結果の検索有無を判定するエレメントの「name」属性値を設定します。	○
refreshTarget	値が変更された場合に再検索を行うエレメントの「name」属性値を設定します。 形式：refreshTarget="%エレメントの「name」属性値%, %エレメントの「name」属性値%, ..." 設定するエレメントは、本タグライブラリを内包するガジェットと同じガジェットに内包されるエレメントのみとなります。 後述のガジェットID（gadgetId）ブロックID（gadgetBlockId）が設定されていない場合は、再検索は行われません。	
gadgetId	本タグライブラリを内包するガジェットのガジェットIDを設定します。 「refreshTarget」に設定されたエレメントの値が更新された場合の再検索用スクリプトで使します。	
gadgetBlockId	本タグライブラリを内包するガジェットのブロックIDを設定します。 「refreshTarget」に設定されたエレメントの値が更新された場合の再検索用スクリプトで使します。	

**コラム**

本項に記載していない属性（style,classなど）も、通常のエレメントと同様に設定可能です。

**コラム**

ラジオボタンの検索タイプは、検索結果に紐づくデータがラジオボタンですべて表示されるため、検索結果が少数で固定のマスタに対して使用することを推奨します。

**注意**

検索タイプ（searchType）が未設定の場合、検索タイプを正常に取得できない場合があります。
検索タイプ（searchType）を必ず設定してください。

初期値の指定方法

検索タイプ（searchType）を設定しない場合、対象マスタのMasterBuilder定義から検索タイプを取得します。
タグライブラリによって生成されるエレメントは動的に変化しますので、初期値設定を行う場合は注意が必要です。

i コラム

初期値設定方法の違い

- **検索部品**
コード項目に値（初期値や再検索時の値）を設定後に再検索を実行します。

再検索処理は以下のJavaScriptファンクションを利用してください。
KAIDEN.MasterSearch.refresh(accessor(/**HtmlElement*/target, /**boolean*/isDistrustful);

引数	説明
target	再検索を行うエレメントを設定します。
isDistrustful	データベースから値を取得するか否かを設定します。 「true」が指定されている場合は、マスタデータをデータベースから取得します。 「true」以外が指定されている場合は、マスタデータをデータベースから取得しません。 (タグライブラリの「primaryTarget」属性で選択状態を設定します。)

- **セレクトボックス**
再検索後に、コード項目に値（初期値や再検索時の値）を設定します。

再検索処理は以下のJavaScriptファンクションを利用してください。
KAIDEN.SelectListData.refresh(/**HtmlElement*/target, /**Object*/data, /**boolean*/isSingle);

引数	説明
target	再検索を行うエレメントを設定します。
data	再検索時に使用する検索条件（Map形式）を設定します。
isSingle	再検索を指定エレメントのみにするか否かを設定します。 「true」が指定されている場合は、指定エレメントの「Option」タグ（選択肢）に再検索結果を設定します。 「true」以外が指定されている場合は、指定エレメントと同じリストデータを使用しているエレメントの「Option」タグ（選択肢）に再検索結果を設定します。

- **ラジオボタン**
タグライブラリの属性値「resultValue」に値を設定すると初期値が設定されます。
ラジオボタンを明細行で設定する場合、「resultValue」に値を設定しても初期値が設定されません。
コード項目(「name」属性に「_radio」を付与した値)に値（初期値）を設定します。

タグライブラリによって生成されたエレメントの検索タイプを判定するためには、タグライブラリによって生成されたエレメント（コード項目）の「data-kaiden-tag-type」属性の値を使用します。
タグライブラリによって生成されたエレメントがラジオボタンの場合、「data-kaiden-tag-type」属性の値は設定されません。

属性値	検索タイプ
search	タグライブラリによって生成されたエレメントが検索部品の場合に設定されます。
select	タグライブラリによって生成されたエレメントがセレクトボックスの場合に設定されます。

実装サンプル
上記を踏まえ、タグライブラリによって生成されたエレメントの初期値を設定する場合は、次の様に実装します。

```
var accessor = KAIDEN.getGadget("${gadget_id}").blockAccessors["${block_id}"];

// 対象のコード項目から検索タイプ（属性「data-kaiden-tag-type」の値に設定される）を取得
var tagType = accessor("${sample_code}").elem().attr("data-kaiden-tag-type");

// 検索タイプ別の処理を実装（search：検索部品、select：セレクトボックス）
if(tagType == "search"){
  // 対象のコード項目に初期値を設定
  accessor("${sample_code}").setValue("${xxxxx}");
  // 検索部品のリフレッシュ（再検索）を実行
  KAIDEN.MasterSearch.refresh(accessor("${sample_name}").elem(), true);
} else if(tagType == "select"){
  // セレクトボックスのリフレッシュ（再検索）を実行
  KAIDEN.SelectListData.refresh(accessor("${sample_code}").elem());
  // 対象のコード項目に初期値を設定
  accessor("${sample_code}").setValue("${xxxxx}");
} else if(accessor("${sample_code}" + "_radio").elem(true).length > 0){
  // 対象のコード項目に初期値を設定(ラジオボタン)
  accessor("${sample_code}" + "_radio").setValue("${xxxxx}");
}
```

値	説明
\${gadget_id}	対象のタグライブラリを内包するガジェットID

値	説明
<code>\${block_id}</code>	対象のタグライブラリを内包するブロックID
<code>\${sample_code}</code>	コード項目の「name」属性（name）
<code>\${sample_name}</code>	名称項目項目の「name」属性（viewName）
<code>\${xxxxx}</code>	初期値（コード）

関連エレメントの値変更時の再検索方法

検索基準日や検索基準会社の変更に伴い、検索部品やセレクトボックスの検索結果を変更したい場合は、タグライブラリの「refreshTarget」「gadgetId」「gadgetBlockId」を設定することで、再検索を行うJavaScriptを生成します。

生成されるJavaScriptは次を参照してください。

■ 各値解説

値	説明
<code>\${gadget_id}</code>	対象のタグライブラリを内包するガジェットID
<code>\${block_id}</code>	対象のタグライブラリを内包するブロックID
<code>\${sample_code}</code>	コード項目の「name」属性（name）
<code>\${sample_name}</code>	名称項目項目の「name」属性（viewName）
<code>\${xxxxx}</code> 、 <code>\${yyyyy}</code>	値が変更された場合に、再検索を行うエレメントの「name」属性

■ タグライブラリ

```
<k_generic:master
masterId="sample"
mode="0"
name="${sample_code}"
viewName="${sample_name}"
surfaceKey=""
criteria="{ 'sampleCd': '${sample_code}' }"
resultChain="{ ' ${sample_code} ': 'sampleCd', ' ${sample_name} ': 'sampleName' }"
resultValue="sampleName"
primaryTarget="${sample_code}"

refreshTarget="${xxxxx}.${yyyyy}"
gadgetId="${gadget_id}"
gadgetBlockId="${block_id}"
/>
```

■ 生成されるJavaScript

検索部品の場合

```
jQuery(function($){
KAIDEN.gadgetMan.pushPostInitialize("${gadget_id}", function() {
  var _accessor = KAIDEN.getGadget("${gadget_id}").blockAccessors["${block_id}"];

  _accessor("${xxxxx}").elem(true).on("change", function(e) {
    KAIDEN.MasterSearch.refresh(_accessor("${sample_name}").elem(true), true);
  });

  _accessor("${yyyyy}").elem(true).on("change", function(e) {
    KAIDEN.MasterSearch.refresh(_accessor("${sample_name}").elem(true), true);
  });
});
});
```

セレクトボックスの場合

```
jQuery(function($){
KAIDEN.gadgetMan.pushPostInitialize("${gadget_id}", function() {
  var _accessor = KAIDEN.getGadget("${gadget_id}").blockAccessors["${block_id}"];

  _accessor("${yyyyy}").elem(true).on("change", function(e) {
    KAIDEN.SelectListData.refresh(_accessor("${sample_code}").elem(true));
  });

  _accessor("${xxxxx}").elem(true).on("change", function(e) {
    KAIDEN.SelectListData.refresh(_accessor("${sample_code}").elem(true));
  });
});
});
```

注意

「primaryTarget」に指定するエレメントはタグライブラリと同じガジェット、およびブロック内に存在する必要があります。
他のガジェットやブロックに存在するエレメントを使用したい場合は、値変更に伴う再検索処理を実装してください。

注意

タグライブラリがTMTable（明細）に含まれる場合は、再検索用のJavaScriptは生成できません。
TMTableに含まれる場合は、値変更に伴う再検索処理を実装してください。

明細でセレクトボックス利用時の注意点

本タグライブラリで生成したセレクトボックスを、明細テーブル（TMTable）内で使用する場合の注意点を説明します。
セレクトボックスのリフレッシュ（再検索）を行う際に、次の2種類の方法があります。

- 同名のリストデータ（セレクトボックスの選択肢）をすべて更新
- 単一のエレメントのリストデータ（セレクトボックスの選択肢）を更新

コラム

前項で解説した「関連エレメントの値変更時の再検索方法」の様に、
関連エレメントの値変更に伴い明細テーブル（TMTable）内のセレクトボックスをリフレッシュする場合、
関連エレメントの実装個所によって再検索方法を切り替える必要があります。

後述の例では、次の様なマスタデータが登録されているものとして説明します。

- サンプル会社

No	▲ サンプルコード	サンプル名
1	sample01	sample01
2	sample02	sample02
3	sample03	sample03
4	sample04	sample04
5	sample05	sample05

- その他会社

No	▲ サンプルコード	サンプル名
1	other01	other01
2	other02	other02
3	other03	other03

例) ヘッダー項目の値変更に合わせて、明細項目をリフレッシュ

ヘッダー項目の値に合わせて明細のセレクトボックス（リストデータ）をリフレッシュする場合は、
全てのリストデータをリフレッシュする必要があります。

コラム

再検索処理のJavaScript関数（第3引数に「true」以外（未設定も可能）を指定することで、
第1引数に指定したエレメントと同じリストデータを利用しているエレメントがリフレッシュされます。
KAIDEN.SelectListData.refresh(/**HtmlElement*/target, /**Object*/data, /**boolean*/isSingle);

1. ヘッダー項目（検索基準会社）を選択します。

2. 明細のセレクトボックスを選択します。

同様に他のセレクトボックスを選択します。

3. ヘッダー項目（検索基準会社）を変更します。

変更に合わせて明細のセレクトボックスすべてがリフレッシュされ、未選択状態に変更されます。

4. 再度、明細のセレクトボックスを選択すると、ヘッダー項目（検索基準会社）に合わせたリストデータが表示されます。

検索基準会社	その他会社
☐ セレクトボックス	
☐ other01	
☐ other02	
☐ other01 other02 other03	

例) 同じ明細項目の値変更に合わせて、明細項目をリフレッシュ

同じ明細項目の値変更に合わせて明細のセレクトボックス（リストデータ）をリフレッシュする場合は、各リストデータ毎にリフレッシュする必要があります。

i コラム

再検索処理のJavaScript関数の第3引数に「true」を指定することで、第1引数に指定したエレメントのみリフレッシュされます。

```
KAIDEN.SelectListData.refresh(/**HTMLElement*/target, /**Object*/data, /**boolean*/isSingle);
```

1. リフレッシュ元の項目（検索基準会社）を選択します。

検索基準会社	セレクトボックス
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	

2. リフレッシュ先のセレクトボックスを選択します。

検索基準会社	セレクトボックス
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	

同様に他のセレクトボックスを選択します。

検索基準会社	セレクトボックス
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	

3. 1行目のリフレッシュ元の項目（検索基準会社）を変更します。

検索基準会社	セレクトボックス
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	

変更に合わせて1行目のセレクトボックスすべてがリフレッシュされ、未選択状態に変更されます。

検索基準会社	セレクトボックス
☐ その他会社	
☐ サンプル会社	
☐ サンプル会社	
☐ サンプル会社	

4. 再度、1行目のリフレッシュ先セレクトボックスを選択すると、リフレッシュ元の項目に合わせたリストデータが表示されます。

他のセレクトボックスはリフレッシュしません。

注意

再検索処理の第3引数に「true」以外を指定した場合は、第1引数に指定したエレメントと同じリストデータを利用しているエレメントがリフレッシュされてしまい、想定外の選択肢が表示される場合がありますのでご注意ください。

サーバーサイド

Controller

検索部品からの検索処理では、「[拡張設定](#)」の「検索用の処理URL」に設定したURLから自動判定されます。
(未設定の場合は「kaiden/v2/generic/masterSearch」を使用します。)

セレクトボックス・ラジオボタンのリフレッシュ処理では、「[拡張設定](#)」の「セレクトボックス用の処理URL」に設定したURLから自動判定されます。
(未設定の場合は「kaiden/v2/generic/masterSelect」を使用します。)

Controllerクラスでは、Logicクラスの対象メソッド起動を行います。
Controllerクラスのメソッドと主な起動処理は次の通りです。

処理名	メソッド	主な起動処理
マスタ検索	masterSearch	検索部品から検索処理実行時
リストデータ取得	masterSelect	セレクトボックス・ラジオボタンのリフレッシュ時

Form

Controllerクラスへアクセスした際のリクエストパラメータは、Formクラス(jp.co.slcs.kaiden.v2.base.feature.form.kaiden.generic.MasterForm)に格納されます。
Formクラスでは、マスタID、マスタ拡張ID、検索キーワードの他に、汎用的にデータを取得するためにJSON化したデータを保持しています。

Controllerクラスの各メソッド呼出し時に格納されるJSONデータの内容は次の通りです。

- マスタ検索 (masterSearch)
タグライブラリの属性値(criteria)に指定した値が設定されます。
- リストデータ取得 (masterSelect)
タグライブラリの属性値(criteria)に指定した値が設定されます。

Logic

Logicクラスは、「[拡張設定](#)」の「Logicクラス」に設定したクラスを使用します。
(未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。)

Logicクラスでは、検索処理を行います。
Logicクラスのメソッドと主な処理内容次の通りです。

処理名	メソッド	主な処理
-----	------	------

マスタ検索	masterSearch(GenericMasterSettings settings, AbstractMasterForm form)	マスタ検索条件の生成と検索結果成型
リストデータ取得	masterSelect(GenericMasterSettings settings, AbstractMasterForm form)	マスタ検索条件の生成と検索結果成型



コラム

上記クラスでは、検索条件の生成と検索結果の成型を行っており、実際の検索処理は、次のメソッドで実施しています。
`search(GenericMasterSettings settings, Map<String, Object> criteria)`



コラム

APIを使用したメンテナンス機能を実現する場合や、独自のロジックを追加する場合はLogicクラスの拡張を行います。
 Controllerクラスからの起動を想定するメソッドを追加する場合はControllerクラスの拡張が必要です。

データの補完や成型を行いたい場合は、後述のConverterクラスの拡張でも対応可能です。
 Logicクラスの各メソッドへのリクエストパラメータやレスポンスパラメータに対してデータ補完や成型が可能です。

Manager

Managerクラスは、「[拡張設定](#)」の「Managerクラス」に設定したコンポーネントを使用します。
 （未設定の場合は「masterManager」を使用します。）

Managerクラスでは、SQLを利用した検索やMirage-SQLを利用した登録・更新・削除処理を行います。
 基本的に拡張の必要はありませんが、標準のManagerクラスでは単一のテーブルに対して処理を実装しているため、複数のテーブルに跨る処理を実装する場合に拡張を行います。
 （検索時のJoinなどは複数テーブルに跨りますが、SQLで実現可能なため、拡張の必要はありません。）

処理名 メソッド

検索	<code>select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria, int limit, int offset)</code>
----	---

Converter

Converterクラスは、「[拡張設定](#)」の「Converterクラス」に設定したクラスを使用します。
 （未設定の場合はConverterクラスを使用しません。）

Converterクラスでは、Logicクラスへのリクエストパラメータ、Logicクラスからのレスポンスパラメータに対し、補完や成型が行えます。
 設定を行わない場合は、補完や成型は行われません。

- 画面表示は%表記だが、データ上は数値として保持（80%→0.8）
- 固定値の補完
- カンマ区切りへの変換



コラム

文字列と日付の変換や、文字列と日付の変換はConverterクラスではなく、
[「項目設定」](#)の「コンバート設定」を利用してください。
 詳細は「[項目値のコンバート](#)」を参照してください。



注意

Converterクラスの処理は各データに対して行われるため、
 データベースの特定データを補完する場合はLogicクラスで実装を行ってください。
 Converterクラスに実装した場合、一覧検索のレスポンスデータの補完などで一覧行の各行単位で処理が行われるため、
 SQLを実行する様な処理では、複数回SQLが実行されてしまい、レスポンス悪化が懸念されます。

シーケンス

サーバーサイドの処理シーケンス（概要）は次の通りです。

コラム

シーケンスに記載の「Controller」「Logic」「Manager」の実際のクラスは次の通りです。
シーケンスの各処理は基底クラスで実装しています。

Controller

基底クラス : `jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.generic.AbstractMasterController`

実装クラス : `jp.co.slcs.kaiden.v2.base.feature.controller.kaiden.generic.MasterController`

Logic

基底クラス : `jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.AbstractMasterLogicBase`

実装クラス : `jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic`

Manager

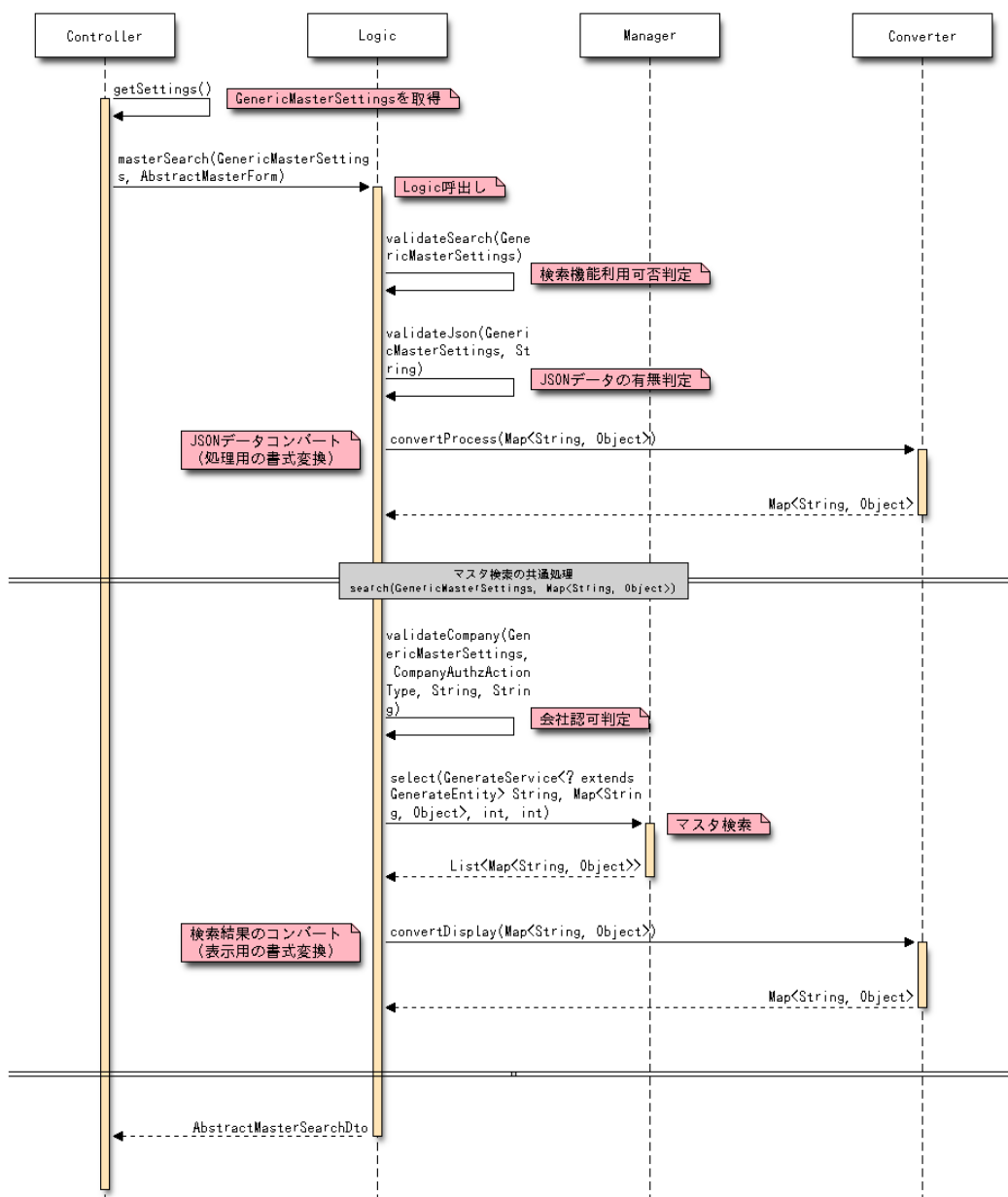
IFクラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.MasterManager`

基底クラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.AbstractMasterManager`

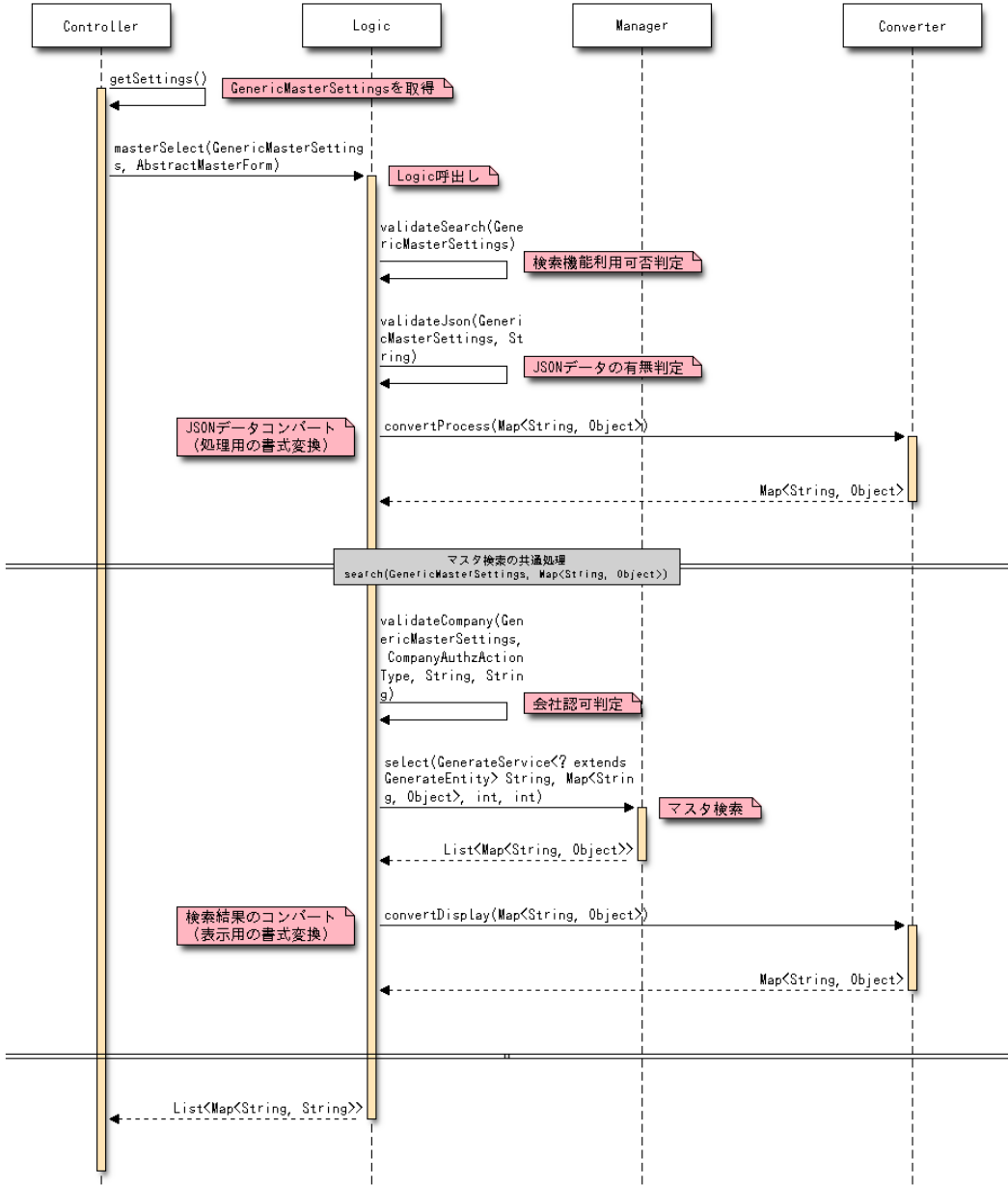
実装クラス : `jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.impl.MasterManagerImpl`

「Converter」は、「[拡張設定](#)」の「Converterクラス」に設定していない場合は実行されません。

■ マスタ検索(masterSearch)



■ リストデータ取得(masterSelect)



インポート

本項では、インポート機能の機能詳細を説明します。

- インポート機能
 - 概要
 - ジョブパラメータ
 - インポートファイル
 - バリデーション
 - 国際化、期間情報
- サーバーサイド
 - Logic
 - Manager
 - Converter
 - シーケンス

インポート機能

概要

MasterBuilderのインポート機能は、次の5つのインポート方法を提供します。
ただし、非期間化マスタのインポートは、「単一期間（差分）」のみ使用可能です。

名称	モード	期間	開始日・終了日	処理方式
----	-----	----	---------	------

単一期間（差分）	1	単一期間	パラメータから取得	差分
単一期間（一新）	2	単一期間	パラメータから取得	一新
単一期間（入替）	3	単一期間	パラメータから取得	入替
複数期間（差分）	4	複数期間	各レコードから取得	差分
複数期間（入替）	5	複数期間	各レコードから取得	入替

- **（インポート）モード**

ジョブパラメータに設定します。（未設定の場合は「1」が設定されたものとします。）

- **処理方式**

差分：インポートファイルのデータと期間重複する対象マスタデータの期間を編集し、インポートファイルのデータを登録します。

一新：インポートデータの開始日以降のデータを論理削除（会社指定時は指定会社のレコードを論理削除）後、差分処理を実行します。

入替：対象マスタの全レコードを物理削除（会社指定時は指定会社の全レコードを物理削除）し、インポートファイルのデータを登録/更新します。

ジョブパラメータ

インポート機能では、MasterBuilder定義やインポートジョブ（ジョブネット）に指定したパラメータによって処理を行います。

詳細は「[MasterBuilder定義](#)」「[インポートジョブで指定可能なパラメータ](#)」を参照してください。

インポートファイル

インポートに使用するデータファイル（以降、インポートファイル）には、CSV（TSV）のフォーマットのみが使用可能です。

インポートファイルの「文字コード」、「区切り文字」、「囲み文字」はどのような設定も可能ですが、動作を保障する設定値は次の値のみです。

設定	設定値
文字コード	UTF-8、SHIFT-JIS
区切り文字	、（カンマ）、\t（タブ）
囲み文字	none（囲み文字なし）、"（ダブルクォーテーション）

バリデーション

MasterBuilderのインポート機能ではインポートデータのバリデーションを実施しています。

インポートファイル中にバリデーションエラーとなるレコードが1件でも存在する場合、処理結果が「異常」となりインポート処理は行われません。

ただし、インポートファイルの途中にバリデーションエラーとなるレコードが存在した場合でも、インポートファイルの全レコードのバリデーションを実行します

次の様なレコードはバリデーションエラーにはなりません。

- 複数期間インポート時に期間重複するレコードがある場合
インポートファイルの出現順で処理を行うため、重複があった場合でもそのまま処理されます。

国際化、期間化情報

インポート機能での国際化情報や期間化情報の処理概要は次の通りです。

- **国際化情報**

インポートファイルには複数ロケールの国際化情報を記述することができます。（1行に1ロケールのデータを記述します。）

複数ロケールをインポートする場合、1つのデータに対して複数行記述することになりますが、非国際化情報はすべて同じ内容を記載してください。

異なる非国際化情報を設定した場合、特定のレコードが一定して選択されることを保証していません。

（異なる非国際化情報を設定した場合でも、特定レコードに合わせる等の処理は実装していません。）

インポートファイルに記載した国際化情報において、システム設定ロケールのレコードが不足している場合は処理結果は「異常」となります。

また、システム設定ロケール外のレコードが存在する場合も処理結果は「異常」となります。

運用開始後にシステム設定ロケールを減らした場合、システム設定ロケール外となったロケールデータの自動削除は実施しません。

インポートモード（入替等）により、削除場合があります。

1. システム設定ロケール「en、ja」に、「ja」ロケールのデータのみをインポートした場合
「en」ロケールが不足しているため、処理結果は「エラー」となります。
2. システム設定ロケール「en、ja」に、「zh_CN」ロケールのデータをインポートした場合
システム設定ロケールに「zh_CN」ロケールが設定されていないため、処理結果は「エラー」となります。
3. システム設定ロケール「en、ja」、各ロケールの非国際化項目が異なる場合
そのまのデータが登録されますが、メンテナンス機能や検索機能ではログインユーザロケールのデータを取得するため、ユーザによって検索結果が異なる様な事象が発生します。

- **期間化情報**

インポートの期間がシステム開始日からシステム終了日までに満たない場合、複数期間を作成する場合があります。

この場合、インポート対象の期間以外のデータは自動で補完されるものであり、任意のデータを指定することはできません。

また、自動補完される期間は論理削除されたデータとなります。

サーバーサイド

Logic

Logicクラスは、「[拡張設定](#)」の「Logicクラス」に設定したクラスを使用します。

（未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。）

Logicクラスでは、インポート処理に必要な各処理を行います。

Logicクラスのメソッドと主な処理内容次の通りです。

処理名	メソッド
インポートモード判定	importModeCheck(GenericMasterSettings settings, String mode)
インポートカラムリスト返却	getImportColumnList(GenericMasterSettings settings, String mode)
プライマリーカラムリスト返却	getImportPrimaryColumnList(GenericMasterSettings settings)
インポートバリデート	validateImport(GenericMasterSettings settings, Map<String, Object> record)
データ更新	importRenewal(GenericMasterSettings settings, Map<String, Object> request)
データ入替	importReplace(GenericMasterSettings settings, Map<String, Object> request)
データインポート	importData(GenericMasterSettings settings, List<Map<String, Object>> recordList)

Manager

Managerクラスは、「[拡張設定](#)」の「Managerクラス」に設定したコンポーネントを使用します。

（未設定の場合は「masterManager」を使用します。）

Managerクラスでは、SQLを利用した検索処理を行います。

基本的に拡張の必要はありませんが、標準のManagerクラスでは単一のテーブルに対して処理を実装しているため、複数のテーブルに跨る処理を実装する場合に拡張を行います。

（検索時のJoinなどは複数テーブルに跨りますが、SQLで実現可能なため、拡張の必要はありません。）

処理名	メソッド
インポートモード判定	
インポートカラムリスト返却	
プライマリーカラムリスト返却	
インポートバリデート	
データ更新	- select(GenerateService<? extends GenerateEntity> service, Class<? extends GenerateEntity> entityClass, List<Where> criteria, List<OrderByItem> order) - insert(GenerateService<? extends GenerateEntity> service, GenerateEntity entity) - update(GenerateService<? extends GenerateEntity> service, GenerateEntity entity, String... includes)
データ入替	- select(GenerateService<? extends GenerateEntity> service, Class<? extends GenerateEntity> entityClass, List<Where> criteria, List<OrderByItem> order) - delete(GenerateService<? extends GenerateEntity> service, GenerateEntity entity)
データインポート	- select(GenerateService<? extends GenerateEntity> service, Class<? extends GenerateEntity> entityClass, List<Where> criteria, List<OrderByItem> order) - insert(GenerateService<? extends GenerateEntity> service, GenerateEntity entity) - update(GenerateService<? extends GenerateEntity> service, GenerateEntity entity, String... includes) - delete(GenerateService<? extends GenerateEntity> service, GenerateEntity entity)

Converter

Converterクラスは、「[拡張設定](#)」の「Converterクラス」に設定したクラスを使用します。

（未設定の場合はConverterクラスを使用しません。）

Converterクラスでは、Logicクラスへのリクエストパラメータ、Logicクラスからのレスポンスパラメータに対し、補完や成型が行えます。

設定を行わない場合は、補完や成型は行われません。

- 画面表示は%表記だが、データ上は数値として保持（80%→0.8）
- 固定値の補完



コラム

文字列と日付の変換や、文字列と日付の変換はConverterクラスではなく、「[項目設定](#)」の「コンバート設定」を利用してください。
詳細は「[項目値のコンバート](#)」を参照してください。

 注意

Converterクラスの処理は各データに対して行われるため、データベースの特定データを補完する場合はLogicクラスで実装を行ってください。

シーケンス

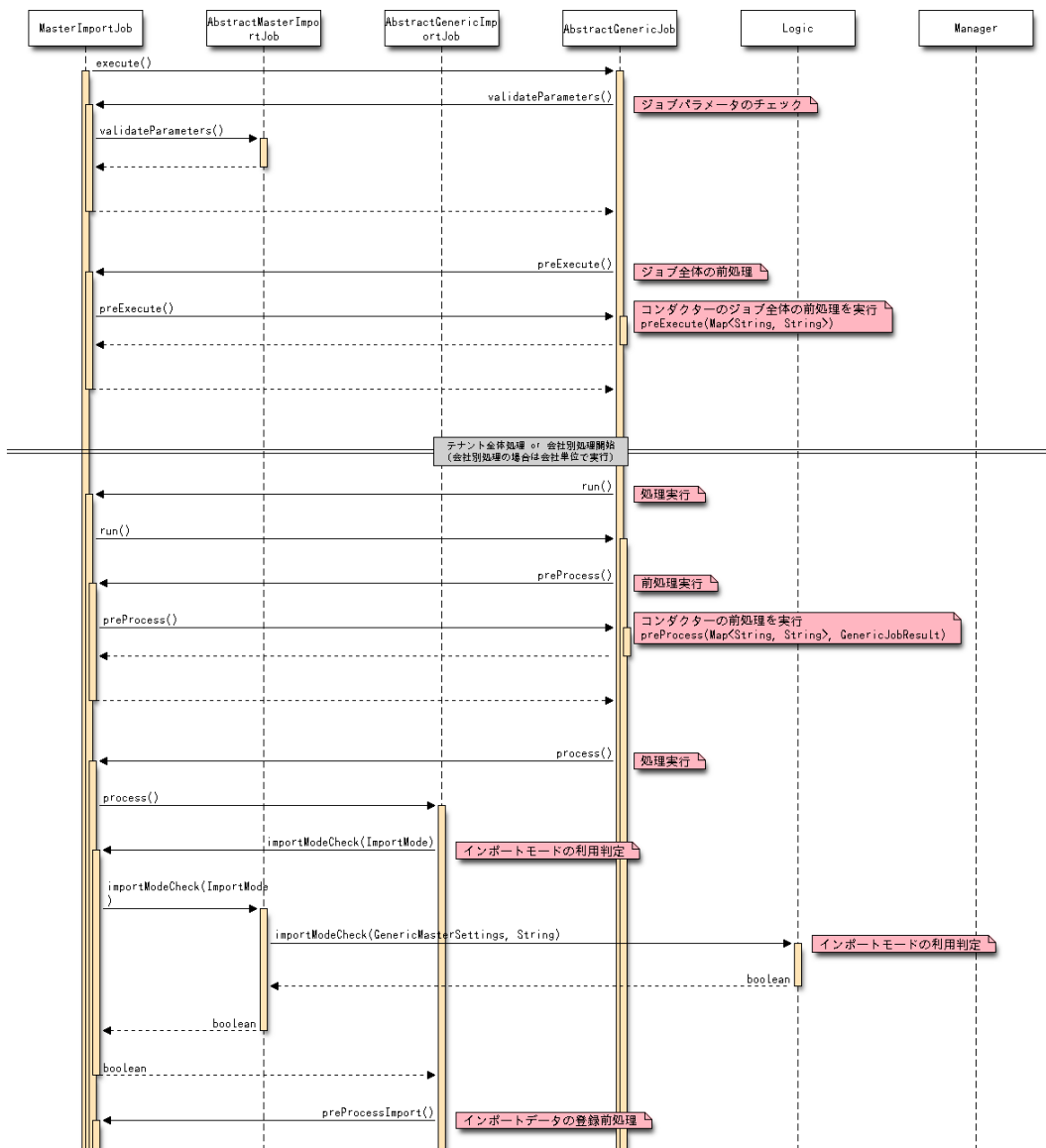
サーバーサイドの処理シーケンス（概要）は次の通りです。

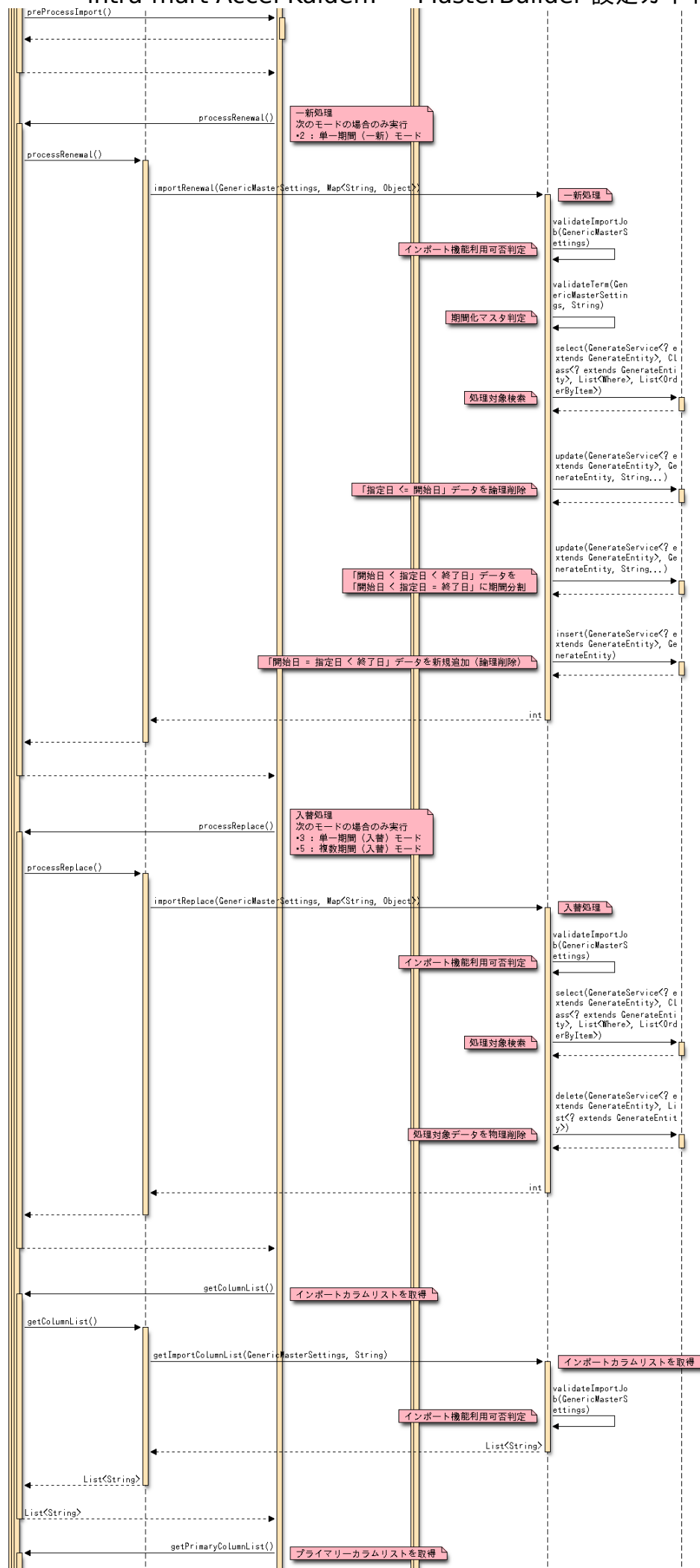
 コラム

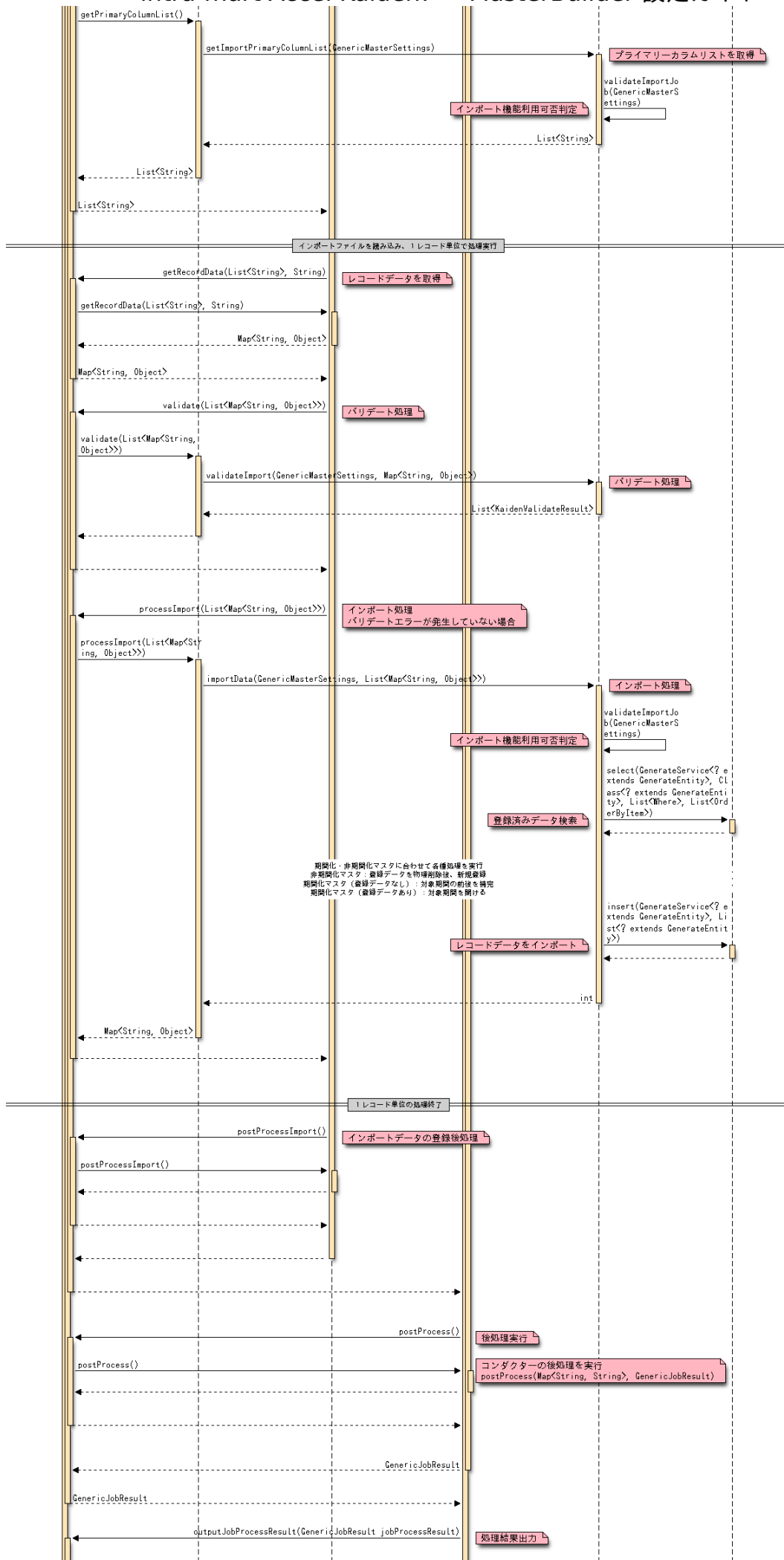
シーケンスに記載の「各Jobクラス」「Logic」「Manager」の実際のクラスは次の通りです。
シーケンスの各処理は基底クラスで実装しています。

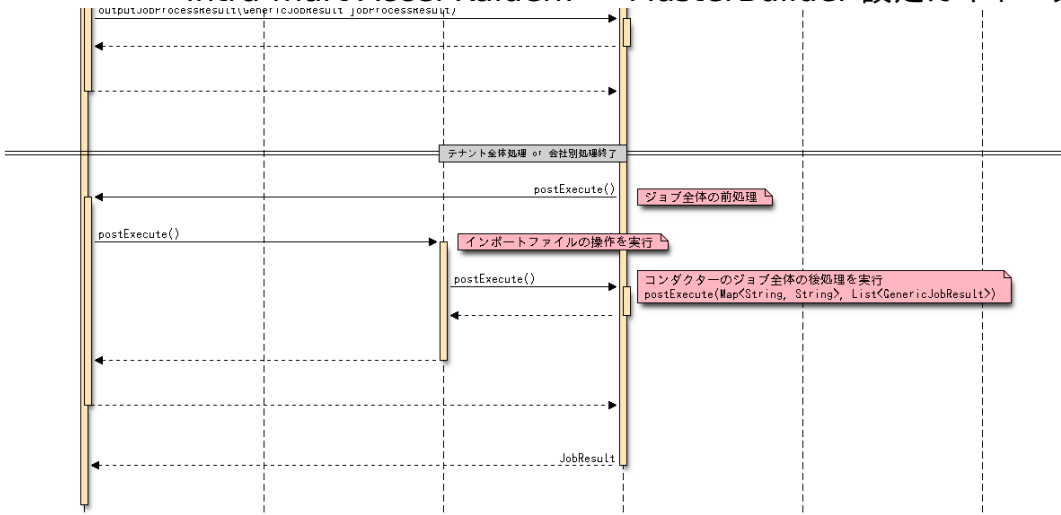
- **MasterImportJob**
jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterImportJob
- **AbstractMasterImportJob**
jp.co.slcs.kaiden.v2.base.feature.job.generic.AbstractMasterImportJob
- **AbstractGenericImportJob**
jp.co.slcs.kaiden.v2.base.foundation.job.AbstractGenericImportJob
- **AbstractGenericJob**
jp.co.slcs.kaiden.v2.base.foundation.job.AbstractGenericJob
- **Logic**
基底クラス : jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.AbstractMasterLogicBase
実装クラス : jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic
- **Manager**
I F クラス : jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.MasterManager
基底クラス : jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.AbstractMasterManager
実装クラス : jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.impl.MasterManagerImpl

「Converter」は、「[拡張設定](#)」の「Converterクラス」に設定していない場合は実行されません。
シーケンス上では省略しています。









エクスポート

本項では、エクスポート機能の機能詳細を説明します。

- エクスポート機能
 - 概要
 - ジョブパラメータ
 - エクスポートファイル
 - 国際化、期間化情報
- サーバサイド
 - Logic
 - Manager
 - Converter
 - シーケンス

エクスポート機能

概要

MasterBuilderのエクスポート機能は、次の3つのエクスポート方法を提供します。
ただし、非期間化マスタの場合は、「期間モード」は使用できません。

名称	モード	処理方式
全件	1	対象マスタデータ全データをエクスポートします。
期間	2	対象マスタデータの特定期間のデータをエクスポートします。
更新日	3	対象マスタデータの特定期日以降に更新されたデータをエクスポートします。

- (エクスポート) モード
ジョブパラメータに設定します。(未設定の場合は「1」が設定されたものとします。)

ジョブパラメータ

エクスポート機能では、MasterBuilder定義やエクスポートジョブ（ジョブネット）に指定したパラメータによって処理を行います。
詳細は「[MasterBuilder定義](#)」「[エクスポートジョブで指定可能なパラメータ](#)」を参照してください。

エクスポートファイル

エクスポートに使用するデータファイル（以降、エクスポートファイル）には、CSV（TSV）のフォーマットのみが使用可能です。
エクスポートファイルの「文字コード」、「区切り文字」、「囲み文字」はどのような設定も可能ですが、動作を保障する設定値は次の値のみです。

設定	設定値
文字コード	UTF-8、SHIFT-JIS
区切り文字	,（カンマ）、\t（タブ）
囲み文字	none（囲み文字なし）,"（ダブルクォーテーション）

国際化、期間化情報

エクスポート機能での国際化情報や期間化情報の処理概要は次の通りです。

- **国際化情報**
エクスポートファイルには複数ロケールの国際化情報を出力します。（1行に1ロケールのデータを記述します。）
また、エクスポートファイル中で国際化情報の並び順をソートする様な機能はありません。
（エクスポート対象データ抽出用SQLで取得された順でエクスポートします。）
- **期間化情報**
エクスポートファイル中で期間化情報の並び順をソートする様な機能はありません。
（エクスポート対象データ抽出用SQLで取得された順でエクスポートします。）

サーバーサイド

Logic

Logicクラスは、「[拡張設定](#)」の「Logicクラス」に設定したクラスを使用します。
（未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。）

Logicクラスでは、エクスポート処理に必要な各処理を行います。
Logicクラスのメソッドと主な処理内容次の通りです。

処理名	メソッド
エクスポートモード判定	exportModeCheck(GenericMasterSettings settings, String mode)
エクスポートカラムリスト返却	getExportColumnList(GenericMasterSettings settings)
ヘッダーカラムリスト返却	getExportHeaderColumn(GenericMasterSettings settings)
検索条件返却	getExportSearchCriteria(GenericMasterSettings settings, Map<String, Object> criteria)
ソート条件返却	getExportOrderCriteria(GenericMasterSettings settings)
データエクスポート	exportData(GenericMasterSettings settings, Map<String, Object> request, IterationCallback<BeanMap, Void> callback)
エクスポートデータコンバート	convertExportData(GenericMasterSettings settings, Map<String, Object> exportData)

Manager

Managerクラスは、「[拡張設定](#)」の「Managerクラス」に設定したコンポーネントを使用します。
（未設定の場合は「masterManager」を使用します。）

Managerクラスでは、SQLを利用した検索処理を行います。
基本的に拡張の必要はありませんが、標準のManagerクラスでは単一のテーブルに対して処理を実装しているため、複数のテーブルに跨る処理を実装する場合に拡張を行います。
（検索時のJoinなどは複数テーブルに跨りますが、SQLで実現可能なため、拡張の必要はありません。）

処理名	メソッド
エクスポートモード判定	
エクスポートカラムリスト返却	
ヘッダーカラムリスト返却	
検索条件返却	
ソート条件返却	
データエクスポート	■ select(GenerateService<? extends GenerateEntity> service, String sqlPath, Map<String, Object> criteria, IterationCallback<BeanMap, Void> callback)
エクスポートデータコンバート	

Converter

Converterクラスは、「[拡張設定](#)」の「Converterクラス」に設定したクラスを使用します。
（未設定の場合はConverterクラスを使用しません。）

Converterクラスでは、Logicクラスへのリクエストパラメータ、Logicクラスからのレスポンスパラメータに対し、補完や成型が行えます。
設定を行わない場合は、補完や成型は行われません。

- 画面表示は%表記だが、データ上は数値として保持（80%→0.8）
- 固定値の補完

i コラム

文字列と日付の変換や、文字列と日付の変換はConverterクラスではなく、
「[項目設定](#)」の「コンバート設定」を利用してください。
詳細は「[項目値のコンバート](#)」を参照してください。

! 注意

Converterクラスの処理は各データに対して行われるため、
データベースの特定データを補完する場合はLogicクラスで実装を行ってください。

シーケンス

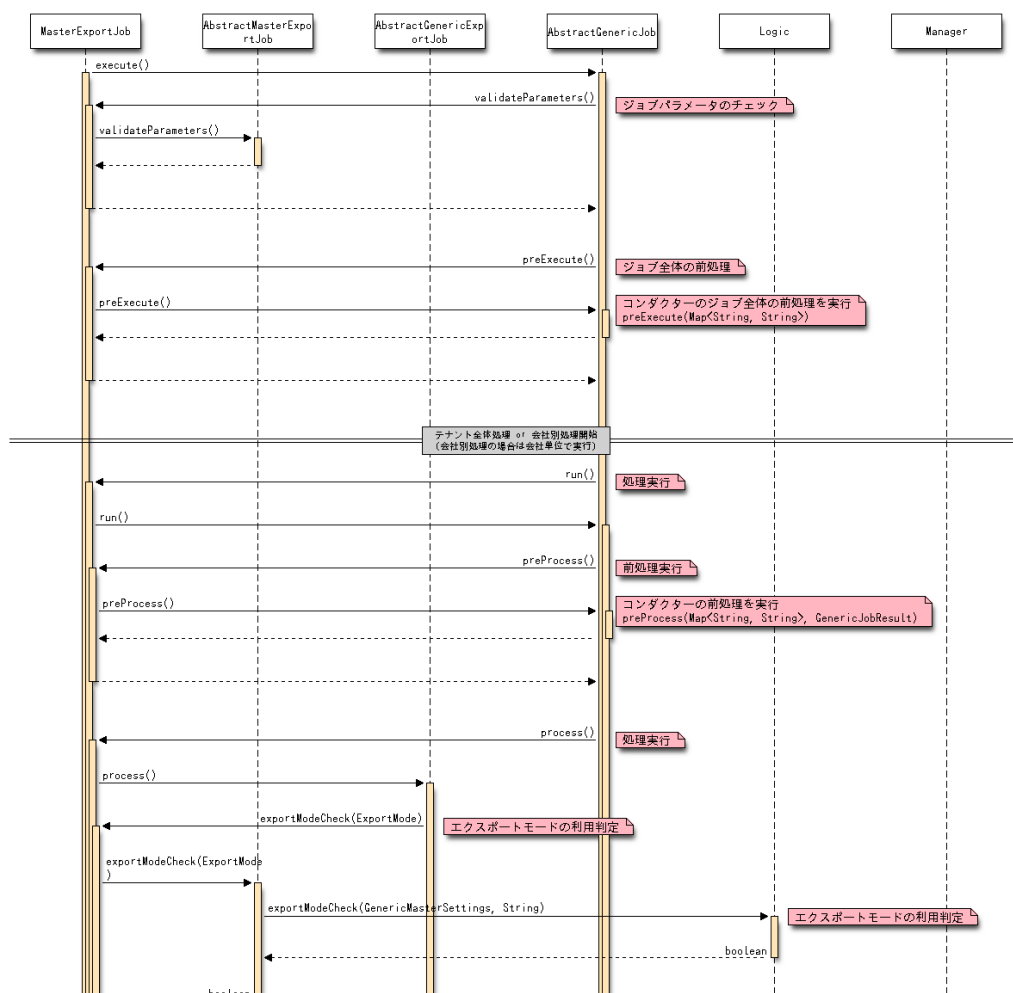
サーバーサイドの処理シーケンス（概要）は次の通りです。

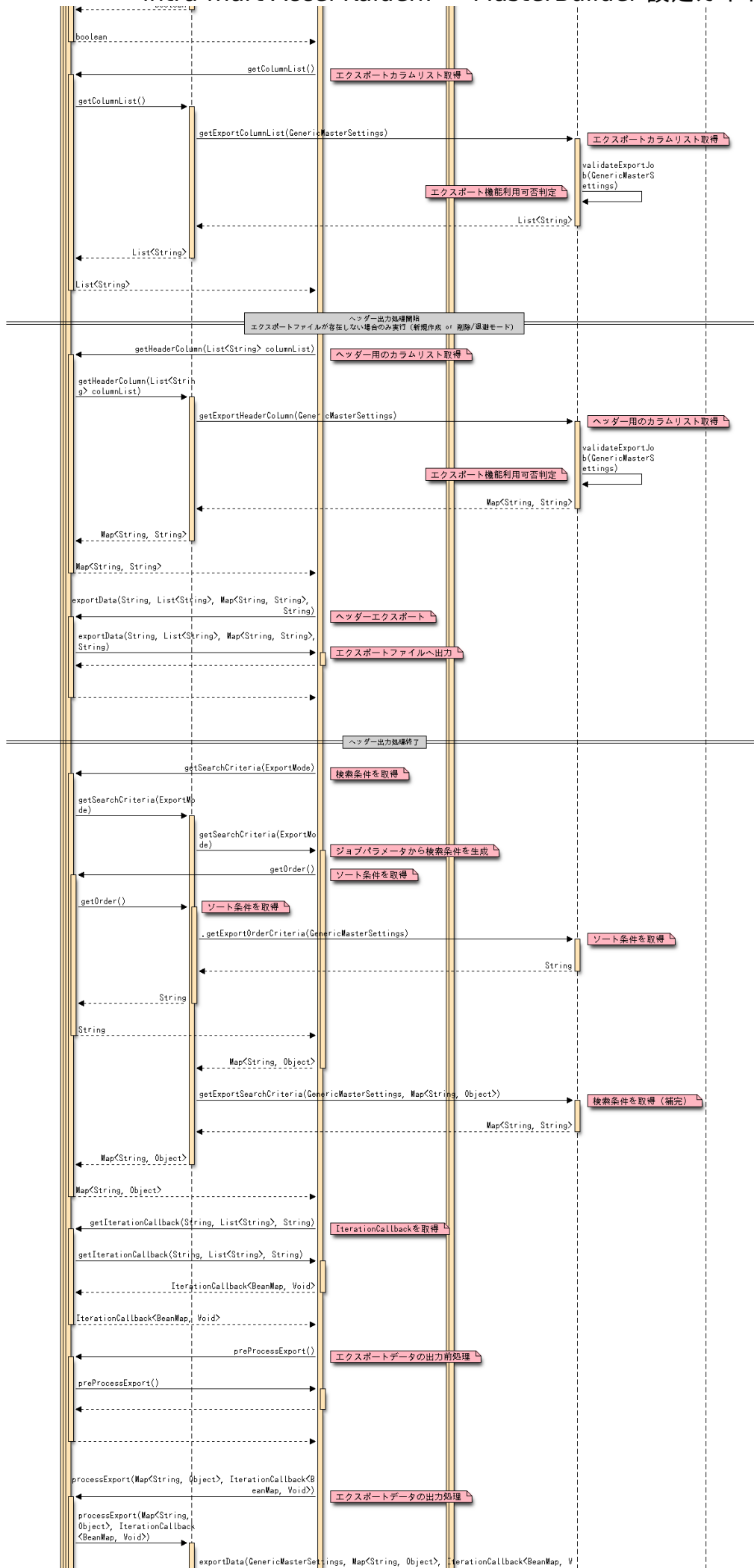
i コラム

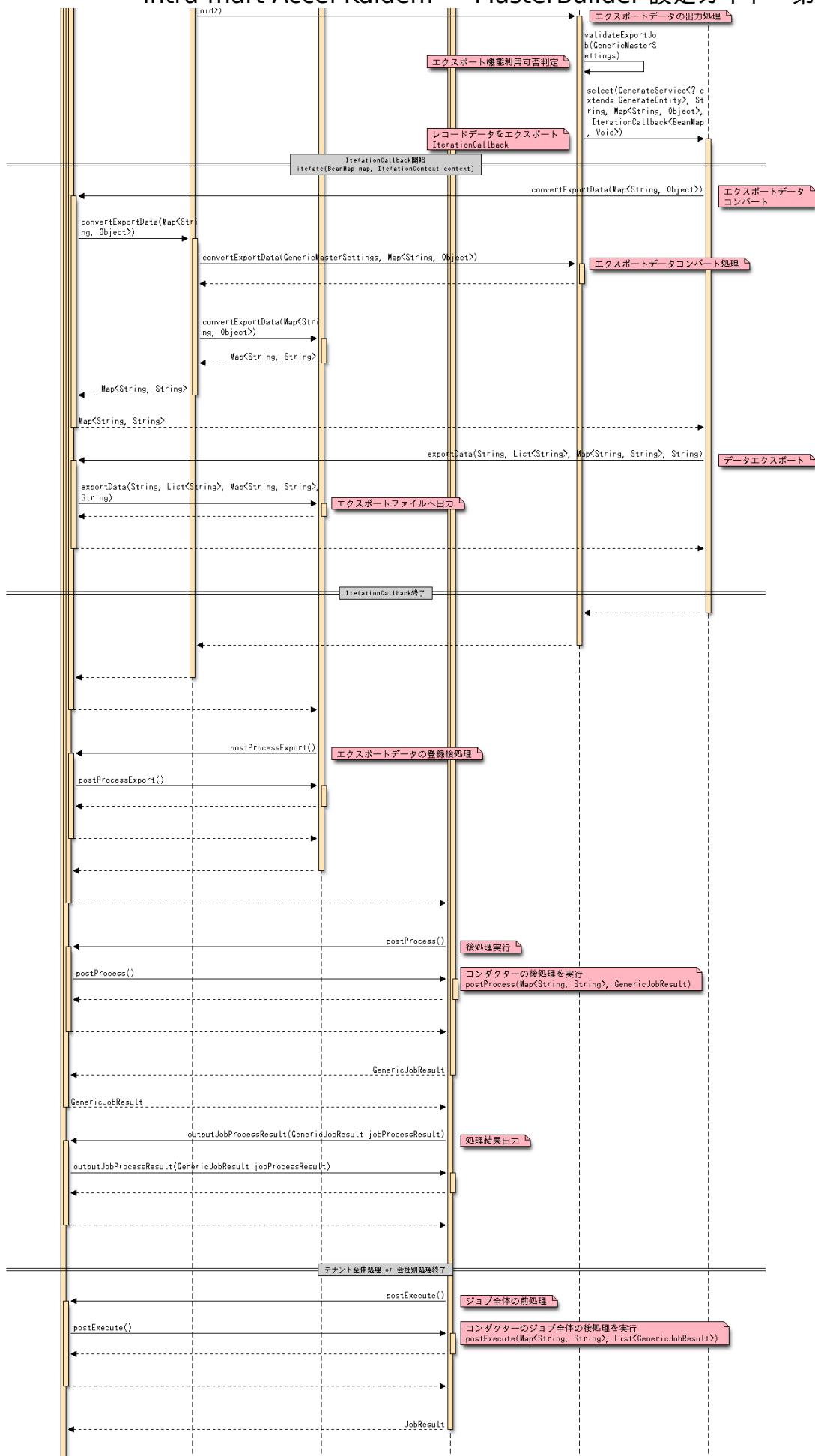
シーケンスに記載の「各Jobクラス」「Logic」「Manager」の実際のクラスは次の通りです。
シーケンスの各処理は基底クラスで実装しています。

- **MasterExportJob**
jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterExportJob
- **AbstractMasterExportJob**
jp.co.slcs.kaiden.v2.base.feature.job.generic.AbstractMasterExportJob
- **AbstractGenericExportJob**
jp.co.slcs.kaiden.v2.base.foundation.job.AbstractGenericExportJob
- **AbstractGenericJob**
jp.co.slcs.kaiden.v2.base.foundation.job.AbstractGenericJob
- **Logic**
基底クラス：jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.AbstractMasterLogicBase
実装クラス：jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic
- **Manager**
IFクラス：jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.MasterManager
基底クラス：jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.AbstractMasterManager
実装クラス：jp.co.slcs.kaiden.v2.base.foundation.manager.generic_master.impl.MasterManagerImpl

「Converter」は、「[拡張設定](#)」の「Converterクラス」に設定していない場合は実行されません。
シーケンス上では省略しています。







バリデート

本項では、バリデート機能の機能詳細を説明します。

- バリデート機能
- サーバーサイド
 - Logic
 - Converter
 - シーケンス

バリデート機能

概要

引数に指定した値が指定した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、指定したマスタに存在する。

チェック**NG**：対象項目の値が、指定したマスタに存在しない。

詳細

バリデーションクラス MasterBuilderExistsValidator

ID (JavaScript)	k_masterBuilderExists
ID (Java)	masterBuilderExists
出力メッセージ	指定されたマスタデータは存在しません。

引数

本バリデーションでは、Map形式で引数を設定します。

どのマスタのバリデーションを実施する場合でも、引数に「masterId」は必須となります。

サーバーサイド

Logic

Logicクラスは、「[拡張設定](#)」の「Logicクラス」に設定したクラスを使用します。

(未設定の場合は「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.impl.MasterLogic」を使用します。)

Logicクラスでは、バリデート処理を行います。

Logicクラスのメソッドと主な処理内容次の通りです。

処理名	メソッド
バリデート処理	validate(GenericMasterSettings settings, Map<String, Object> request)



コラム

APIを使用したメンテナンス機能を実現する場合や、独自のロジックを追加する場合はLogicクラスの拡張を行います。

データの補完や成型を行いたい場合は、後述のConverterクラスの拡張でも対応可能です。

Logicクラスの各メソッドへのリクエストパラメータやレスポンスパラメータに対してデータ補完や成型が可能です。

Converter

Converterクラスは、「[拡張設定](#)」の「Converterクラス」に設定したクラスを使用します。

(未設定の場合はConverterクラスを使用しません。)

Converterクラスでは、Logicクラスへのリクエストパラメータ、Logicクラスからのレスポンスパラメータに対し、補完や成型が行えます。

設定を行わない場合は、補完や成型は行われません。

- 画面表示は%表記だが、データ上は数値として保持 (80%→0.8)
- 固定値の補完
- カンマ区切りへの変換



コラム

文字列と日付の変換や、文字列と日付の変換はConverterクラスではなく、

「[項目設定](#)」の「コンバート設定」を利用してください。

詳細は「[項目値のコンバート](#)」を参照してください。

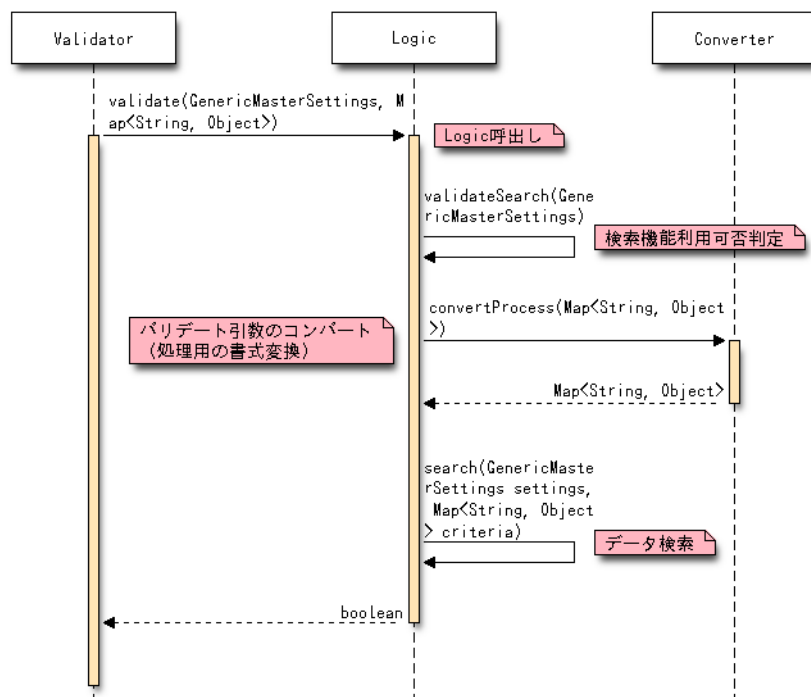
注意

Converterクラスの処理は各データに対して行われるため、データベースの特定データを補完する場合はLogicクラスで実装を行ってください。

Converterクラスに実装した場合、一覧検索のレスポンスデータの補完などで一覧行の各行単位で処理が行われるため、SQLを実行する様な処理では、複数回SQLが実行されてしまい、レスポンス悪化が懸念されます。

シーケンス

サーバーサイドの処理シーケンス（概要）は次の通りです。



API

REST API

REST APIについて説明します。

REST APIについて

MasterBuilderが提供するREST APIは、HTTPプロトコルを使用し、MasterBuilderに関する様々な処理を呼び出すことが可能です。
また、REST APIは全ての呼び出し先に対し認可リソースを持ちます。
REST APIとして利用可能な機能は以下のとおりです。

- 汎用検索
- 汎用検索結果件数取得
- 検索
- 検索結果件数取得
- 新規登録
- 更新
- 削除
- 期間分割
- 期間変更
- 有効化
- 無効化
- インポート
- エクスポート

REST APIで利用可能なマスター一覧

「[REST APIで利用可能なマスター一覧](#)」を参照してください。

エンドポイントプレフィックス

REST APIは以下の通り呼び出してください。
プロトコル、ホスト名、ポート番号、コンテキストパスは環境にあわせて置き換えてください。

http(s)://{HOST}:{PORT}/{CONTEXT_PATH}/kaiden/v2/api/generic/master/....

リクエストパラメータ共通仕様

- 仕様に沿わない構造やプロパティ名の情報が含まれる場合、予期せぬ動作になる可能性があります。
- 各マスタのマスタID、ガジェットバリエーションを確認する場合、「[REST APIで利用可能なマスター一覧](#)」を参照してください。
- ガジェットバリエーションを保持するマスタの場合、ガジェットクラスは「master」を保持しており、マスタの定義ファイル名は「master_%ガジェットバリエーション%」です。

レスポンスデータ共通仕様

- 値が存在しない場合（空白文字列やnull）、プロパティそのものが含まれません。

エンドポイント

- [汎用検索](#)
- [汎用検索結果件数取得](#)
- [検索](#)
- [検索結果件数取得](#)
- [新規登録](#)
- [更新](#)
- [削除](#)
- [期間分割](#)
- [期間変更](#)
- [有効化](#)
- [無効化](#)
- [インポート](#)
- [エクスポート](#)

マスタ拡張IDの利用

- 概要
- カスタマイズ方法

概要

MasterBuilderでは、特定の使用箇所のみをカスタマイズする方法を用意しています。
これにより、次の様な対応が可能となります。

- メンテナンス機能**
特定組織のみ予備項目Aに対して任意の分類を設定する。
- 検索機能**
特定画面のみ予備項目Aに設定された値を検索結果に表示する。
特定画面のみ検索結果の抽出条件を変更する。
- インポート機能**
各外部システムに合わせたレイアウトでインポート処理を行う。
- エクスポート機能**
各外部システムに合わせたレイアウトでエクスポート処理を行う。

カスタマイズ方法

マスタ拡張IDを利用したカスタマイズには、MasterBuilderの使用箇所でもマスタ拡張IDを設定する必要があります。

- メンテナンス機能**
メニューの引数にマスタ拡張IDを追加
- 検索機能**
検索機能のタグライブラリ属性にマスタ拡張IDを追加
- インポート機能**
ジョブ（ネット）パラメータにマスタ拡張IDを追加
- エクスポート機能**
ジョブ（ネット）パラメータにマスタ拡張IDを追加

MasterBuilder定義ファイルの名称を「%マスタID%_%マスタ拡張ID%.xml」となる様に設定することで、
カスタマイズしたMasterBuilder定義が使用されます。

指定可能なエレメントタイプと属性値

MasterBuilder定義の「項目設定」で指定可能なエレメントタイプとエレメント属性は次の通りです。

エレメント名	設定値	属性値
なし	NONE	
テキストボックス	TEXT	surfaceKey,class,style,maxlength,imeMode
テキストボックス（国際化対応）	TEXT_LOCALES	surfaceKey,class,style,maxlength,imeMode
テキストボックス（数値）	TEXT_NUMERIC	surfaceKey,class,style,maxlength
テキストボックス（コード）	TEXT_CODE	surfaceKey,class,style,maxlength
テキストボックス（日付）	TEXT_DATE	surfaceKey,class,style,maxlength
テキストボックス（時間）	TEXT_TIME	surfaceKey,class,style,maxlength,maxHour,timeFormat,minutesUnit
テキストボックス（パスワード）	TEXT_PASSWORD	surfaceKey,class,style,maxlength
テキストエリア	TEXTAREA	surfaceKey,class,style,maxlength,imeMode,cols,rows
テキストエリア（国際化対応）	TEXTAREA_LOCALES	surfaceKey,class,style,maxlength,imeMode,cols,rows
チェックボックス	CHECKBOX	surfaceKey,class,style,label
ラジオボタン	RADIO	surfaceKey,class,style,label,value
テキストボックス（Hidden）	TEXT_HIDDEN	surfaceKey,class,style
マスタ検索	MASTER	surfaceKey,class,style,imeMode,masterId,extensionId,searchType,viewName,criteria,resultChain,resultValue,primaryTarget,refreshTarget

項目値のコンバート

MasterBuilder定義の「項目設定」で指定可能なコンバート設定は次の通りです。

設定値	解説
STRING	文字列へのコンバートを行います。
STRING_DATE	文字列(yyyy/MM/dd 形式)へのコンバートを行います。
STRING_TIMESTAMP	文字列(yyyyMMddHHmmssSSS 形式)へのコンバートを行います。
STRING_SLASH_TIMESTAMP	文字列(yyyy/MM/dd HH:mm:ss.SSS 形式)へのコンバートを行います。
TIMESTAMP	タイムスタンプへのコンバートを行います。
TIMESTAMP_SHIFT	タイムスタンプへのコンバートを行います。(コンバート元の場合は-1日、コンバート先の場合は+1日)
LONG	数値(Long)へのコンバートを行います。
BIG_DECIMAL	数値(BigDecimal)へのコンバートを行います。

HTML、およびJavaのコンバート実行有無は次の通りです。
表に「○」が設定されている場合のみコンバート処理を実行します。

コンバート元	コンバート先							
	STRING	STRING_DATE	STRING_TIMESTAMP	STRING_SLASH_TIMESTAMP	TIMESTAMP	TIMESTAMP_SHIFT	LONG	BIG_DECIMAL
STRING							○	○
STRING_DATE					○	○		
STRING_TIMESTAMP					○	○		
STRING_SLASH_TIMESTAMP					○	○		
TIMESTAMP		○	○		○			
TIMESTAMP_SHIFT		○	○		○			
LONG	○							
BIG_DECIMAL	○							

- 例) 「画面：yyyy/MM/dd形式の文字列、Java：TimesTamp」の場合の設定方法（開始日など）
<convert display="STRING_DATE" process="TIMESTAMP"/>
- 例) 「画面：yyyy/MM/dd形式の文字列、Java：TimesTamp」の場合の設定方法（終了日など）
<convert display="STRING_DATE" process="TIMESTAMP_SHIFT"/>
- 例) 「画面：文字列、Java：BigDecimal」の場合の設定方法
<convert display="STRING" process="BIG_DECIMAL"/>

指定可能なジョブパラメータ

- インポートジョブで指定可能なパラメータ
 - 後続処理スキップ (error_skip)
 - ロック待ち時間 (lock_wait)
 - 会社グループセットコード (company_group_set_cd)
 - 会社グループコード (company_group_cd)
 - 会社コード (company_cd)
 - コンダクター (conductor)
 - インポートモード (import_mode)
 - インポート期間の開始日 (start_date)
 - インポート期間の終了日 (end_date)
 - インポート期間の開始日に対する増減日数 (shift_date)
 - インポートファイルパス (file_path)
 - インポート処理終了後のインポートファイル操作モード (file_mode)
 - インポートファイルのアーカイブ先パス (archive_path)
 - インポートファイルの文字コード (character)
 - インポートファイルの区切り文字 (delimiter)
 - インポートファイルの囲み文字 (enclosing)
 - インポートファイルの読込開始行数 (start_rows)
 - マスタID (masterId)
 - マスタ拡張ID (extensionId)
- エクスポートジョブで指定可能なパラメータ
 - 後続処理スキップ (error_skip)
 - ロック待ち時間 (lock_wait)
 - 会社グループセットコード (company_group_set_cd)
 - 会社グループコード (company_group_cd)
 - 会社コード (company_cd)
 - コンダクター (conductor)
 - エクスポートモード (export_mode)
 - エクスポート対象データの検索対象日 (target_date)
 - エクスポート対象データの検索対象日に対する増減日数 (shift_date)
 - エクスポートファイルパス (file_path)
 - エクスポート時の既存ファイル操作モード (file_mode)
 - エクスポートファイルのアーカイブ先パス (archive_path)
 - エクスポートファイルの文字コード (character)
 - エクスポートファイルの区切り文字 (delimiter)
 - エクスポートファイルの囲み文字 (enclosing)
 - エクスポートファイルのヘッダー出力有無 (header_row)
 - マスタID (masterId)
 - マスタ拡張ID (extensionId)
- パラメータの優先度
- ファイルパスの置換文字列

インポートジョブで指定可能なパラメータ

インポートジョブでは、次のパラメータが指定可能です。

キー	値	設定	省略時
error_skip	会社別処理エラー時の後続処理スキップ	任意	false（スキップしない）
lock_wait	ロック待ち時間	任意	10（秒）
company_group_set_cd	会社グループセットコード	任意	
company_group_cd	会社グループコード	任意	
company_cd	会社コード（カンマ区切りで複数指定可）	任意	
conductor	ジョブの各処理間で実行されるコンダクター	任意	
import_mode	インポートモード	任意	1（単一期間（差分））
start_date	インポート期間の開始日	任意	ジョブの実行日
end_date	インポート期間の終了日	任意	システムの終了日
shift_date	インポート期間の開始日に対する増減日数	任意	0（日）
file_path	インポートファイルのパス	任意	kaiden/generic/master/{MASTER_ID}/import.csv
file_mode	処理後のインポートファイル操作モード	任意	1

キー	値	設定	省略時
archive_path	インポートファイルのアーカイブ先パス	任意	kaiden/generic/master/{MASTER_ID}/import_{YMDHMSN}.csv
character	インポートファイルの文字コード	任意	UTF-8
delimiter	インポートファイルの区切り文字	任意	, (カンマ)
enclosing	インポートファイルの囲み文字	任意	" (ダブルクォーテーション)
start_rows	インポートファイルの読込開始行数	任意	1 (行目から読込)
masterId	マスタID	必須	
extensionId	マスタ拡張ID	任意	

後続処理スキップ (error_skip)

特定会社の処理でエラーが発生した場合に、後続の会社別処理をスキップするか否かを設定します。

キー	error_skip
設定	任意
設定値 (省略時)	true, false (false)

ロック待ち時間 (lock_wait)

ロックキー「%ジョブID_%マスタID%」または「%ジョブID_%マスタID_%会社コード%」でアプリケーションがロックされていた場合の待ち時間 (秒) を設定します。

キー	lock_wait
設定	任意
設定値 (省略時)	数値 (10)

会社グループセットコード (company_group_set_cd)

対象会社を絞り込む際の検索条件「会社グループセットコード」を設定します。

「会社グループセットコード」と「会社グループコード」の両方が設定されている場合に絞り込みを行います。

会社グループは、IM-共通マスタのマスタです。

詳細は『IM-共通マスタ 管理者操作ガイド』を参照してください。

(指定した会社グループに所属する会社を処理対象とします。)

キー	company_group_set_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

会社グループコード (company_group_cd)

対象会社を絞り込む際の検索条件「会社グループコード」を設定します。

「会社グループセットコード」と「会社グループコード」の両方が設定されている場合に絞り込みを行います。

会社グループは、IM-共通マスタのマスタです。

詳細は『IM-共通マスタ 管理者操作ガイド』を参照してください。

(指定した会社グループに所属する会社を処理対象とします。)

キー	company_group_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

会社コード (company_cd)

対象会社を絞り込む際の検索条件「会社コード」を設定します。

本パラメータは、カンマ区切りで複数指定可能です。

キー	company_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

コラム

会社グループ、会社の両方を設定した場合は、対象の会社グループに所属する設定した会社のみを対象とします。
設定した会社が、対象の会社グループに所属していない場合は、設定した会社を対象にしません。

コンダクター (conductor)

ジョブの各処理間で実行されるコンダクターのコンポーネント名を設定します。

本パラメータは、カンマ区切りで複数指定可能です。

キー	conductor
設定	任意
設定値 (省略時)	文字列 (未指定)



コラム

AbstractGenericJobConductorを継承したクラスのコンポーネント名を指定してください。

インポートモード (import_mode)

インポートの処理方法 (モード) を設定します。

キー	import_mode
設定	任意
設定値 (省略時)	1~5 (1)

設定可能なインポートモードと処理概要

モード	モード名	解説
1	単一期間 (差分)	インポートデータと期間重複する対象マスタデータの期間を編集または削除し、インポートデータを登録します。 インポート期間の開始日、終了日は後述のパラメータから取得します。
2	単一期間 (一新)	インポート開始日以降の全データを論理削除 (会社指定時は指定会社のレコードを論理削除) し、 インポートデータと期間重複する対象マスタデータの期間を編集または削除し、インポートデータを登録します。 論理削除の対象はインポートファイルに無いデータを含みます。 インポート期間の開始日、終了日は後述のパラメータから取得します。
3	単一期間 (入替)	対象マスタの全レコードを物理削除 (会社指定時は指定会社の全レコードを物理削除) し、インポートデータを登録します。 インポート期間の開始日、終了日は後述のパラメータから取得します。 インポート期間がシステム期間よりも短い場合は、インポートデータで補完します。
4	複数期間 (差分)	インポートデータと期間重複する対象マスタデータの期間を編集または削除し、インポートデータを登録します。 インポート期間の開始日、終了日は、インポートファイル中の開始日、終了日を使用します。
5	複数期間 (入替)	対象マスタの全レコードを物理削除 (会社指定時は指定会社の全レコードを物理削除) し、インポートデータを登録します。 インポート期間の開始日、終了日は、インポートファイル中の開始日、終了日を使用します。



コラム

非期間化マスタの場合は、「1:単一期間 (差分) モード」のみ使用可能です。

インポート期間の開始日 (start_date)

インポートモード1,2,3を設定した場合のインポート期間開始日を設定します。

キー	start_date
設定	任意
設定値 (省略時)	yyyy/MM/dd (ジョブの実行日)



コラム

インポートモード4,5を設定した場合は、本パラメータは使用されません。

インポート期間の終了日 (end_date)

インポートモード1,2,3を設定した場合のインポート期間終了日を設定します。

キー	end_date
設定	任意
設定値 (省略時)	yyyy/MM/dd (システムの終了日)



コラム

インポートモード4,5を設定した場合は、本パラメータは使用されません。

インポート期間の開始日に対する増減日数 (shift_date)

インポートモード1,2,3を設定した場合のインポート期間開始日に対する増減日数を設定します。

キー	shift_date
設定	任意
設定値 (省略時)	数値 (0)



コラム

インポートモード4,5を設定した場合は、本パラメータは使用されません。

インポートファイルパス (file_path)

インポートファイルのファイルパスを設定します。

ファイルパスは、パブリックストレージからのパスを指定してください。

キー	file_path
設定	任意
設定値 (省略時)	文字列 (kaiden/generic/master/{MASTER_ID}/import.csv)



コラム

「{MASTER_ID}」などの置換文字列の詳細は「[ファイルパスの置換文字列](#)」を参照してください。

インポート処理終了後のインポートファイル操作モード (file_mode)

インポート処理終了後のインポートファイル操作モードを設定します。

キー	file_mode
設定	任意
設定値 (省略時)	1～8 (1)

設定可能なファイル操作モードと処理概要

モード	正常終了時の処理 (警告終了を含む)	エラー終了時の処理
1	操作なし	操作なし
2	コピー(copy)	コピー(copy)
3	アーカイブ(move)	アーカイブ(move)
4	アーカイブ(move)	操作なし
5	操作なし	アーカイブ(move)
6	削除(delete)	削除(delete)
7	削除(delete)	操作なし
8	操作なし	削除(delete)

インポートファイルのアーカイブ先パス (archive_path)

ファイル操作モードでアーカイブを指定した場合のインポートファイルのアーカイブ先ファイルパスを設定します。

アーカイブ先ファイルパスは、パブリックストレージからのパスを指定してください。

キー	archive_path
設定	任意
設定値 (省略時)	文字列 (kaiden/generic/master/{MASTER_ID}/import_{YMDHMSN}.csv)



コラム

「{MASTER_ID}」などの置換文字列の詳細は「[ファイルパスの置換文字列](#)」を参照してください。

インポートファイルの文字コード (character)

インポートファイルの文字コードを設定します。

キー	character
設定	任意
設定値（省略時）	UTF-8, SHIFT-JISなど（UTF-8）

インポートファイルの区切り文字（delimiter）

インポートファイルの区切り文字を設定します。

キー	delimiter
設定	任意
設定値（省略時）	,[カンマ], \t[タブ] (,)

インポートファイルの囲み文字（enclosing）

インポートファイルの囲み文字を設定します。

キー	enclosing
設定	任意
設定値（省略時）	none,"[ダブルクォーテーション] (")

インポートファイルの読込開始行数（start_rows）

インポートファイルの読込開始行数を設定します。

キー	start_rows
設定	任意
設定値（省略時）	数値（1）

マスタID（masterId）

インポートを行うマスタのマスタIDを設定します。

キー	masterId
設定	必須
設定値（省略時）	文字列

マスタ拡張ID（extensionId）

インポートを行うマスタのマスタ拡張IDを設定します。

キー	extensionId
設定	任意
設定値（省略時）	文字列（未指定）

エクスポートジョブで指定可能なパラメータ

エクスポートジョブでは、次のパラメータが指定可能です。

キー	値	設定	省略時
error_skip	会社別処理エラー時の後続処理スキップ	任意	false（スキップしない）
lock_wait	ロック待ち時間	任意	10（秒）
company_group_set_cd	会社グループセットコード	任意	
company_group_cd	会社グループコード	任意	
company_cd	会社コード（カンマ区切りで複数指定可）	任意	
conductor	ジョブの各処理間で実行されるコンダクター	任意	
export_mode	エクスポートモード	任意	1
target_date	エクスポート対象データの検索対象日	任意	ジョブの実行日
shift_date	エクスポート対象データの検索対象日に対する増減日数	任意	0（日）

キー	値	設定	省略時
file_path	エクスポートファイルのパス	任意	kaiden/generic/master/{MASTER_ID}/export.csv
file_mode	エクスポート時の既存ファイル操作モード	任意	1
archive_path	エクスポートファイルのアーカイブ先パス	任意	kaiden/generic/master/{MASTER_ID}/export_{YMDHMSN}.csv
character	エクスポートファイルの文字コード	任意	UTF-8
delimiter	エクスポートファイルの区切り文字	任意	, (カンマ)
enclosing	エクスポートファイルの囲み文字	任意	" (ダブルクォーテーション)
header_row	エクスポートファイルのヘッダー出力有無	任意	false (出力しない)
masterId	マスタID	必須	
extensionId	マスタ拡張ID	任意	

後続処理スキップ (error_skip)

特定会社の処理でエラーが発生した場合に、後続の会社別処理をスキップするか否かを設定します。

キー	error_skip
設定	任意
設定値 (省略時)	true, false (false)

ロック待ち時間 (lock_wait)

ロックキー「%ジョブID%_%マスタID%」または「%ジョブID%_%マスタID%_%会社コード%」でアプリケーションがロックされていた場合の待ち時間 (秒) を設定します。

キー	lock_wait
設定	任意
設定値 (省略時)	数値 (10)

会社グループセットコード (company_group_set_cd)

対象会社を絞り込む際の検索条件「会社グループセットコード」を設定します。

「会社グループセットコード」と「会社グループコード」の両方が設定されている場合に絞り込みを行います。

会社グループは、IM-共通マスタのマスタです。

詳細は『IM-共通マスタ 管理者操作ガイド』を参照してください。

(指定した会社グループに所属する会社を処理対象とします。)

キー	company_group_set_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

会社グループコード (company_group_cd)

対象会社を絞り込む際の検索条件「会社グループコード」を設定します。

「会社グループセットコード」と「会社グループコード」の両方が設定されている場合に絞り込みを行います。

会社グループは、IM-共通マスタのマスタです。

詳細は『IM-共通マスタ 管理者操作ガイド』を参照してください。

(指定した会社グループに所属する会社を処理対象とします。)

キー	company_group_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

会社コード (company_cd)

対象会社を絞り込む際の検索条件「会社コード」を設定します。

本パラメータは、カンマ区切りで複数指定可能です。

キー	company_cd
設定	任意
設定値 (省略時)	文字列 (未指定)

i コラム

会社グループ、会社の両方を設定した場合は、対象の会社グループに所属する設定した会社のみを対象とします。
設定した会社が、対象の会社グループに所属していない場合は、設定した会社を対象にしません。

コンダクター (conductor)

ジョブの各処理間で実行されるコンダクターのコンポーネント名を設定します。
本パラメータは、カンマ区切りで複数指定可能です。

キー	conductor
設定	任意
設定値 (省略時)	文字列 (未指定)

i コラム

AbstractGenericJobConductorを継承したクラスのコンポーネント名を指定してください。

エクスポートモード (export_mode)

エクスポートの処理方法 (モード) を設定します。

キー	export_mode
設定	任意
設定値 (省略時)	1〜3 (1)

設定可能なエクスポートモードと処理概要

モード	モード名	解説
1	全件	対象マスタデータ全件をエクスポートします。
2	期間	後述のエクスポート対象データの検索対象日が含まれる期間をエクスポートします。
3	更新日	後述のエクスポート対象データの検索対象日以降に更新されたレコードをエクスポートします。

i コラム

非期間化マスタの場合は、「2:期間モード」は使用できません。

エクスポート対象データの検索対象日 (target_date)

エクスポートモード2,3を設定した場合のエクスポート対象データの検索対象日を設定します。

キー	target_date
設定	任意
設定値 (省略時)	yyyy/MM/dd (ジョブの実行日)

i コラム

エクスポートモード1を設定した場合は、本パラメータは使用されません。

エクスポート対象データの検索対象日に対する増減日数 (shift_date)

エクスポートモード2,3を設定した場合のエクスポート対象データの検索対象日に対する増減日数を設定します。

キー	shift_date
設定	任意
設定値 (省略時)	数値 (0)

エクスポートファイルパス (file_path)

エクスポートファイルのファイルパスを設定します。
ファイルパスは、パブリックストレージからのパスを指定してください。

キー	file_path
設定	任意

設定値（省略時） 文字列 （kaiden/generic/master/{MASTER_ID}/export.csv）



コラム

「{MASTER_ID}」などの置換文字列の詳細は「[ファイルパスの置換文字列](#)」を参照してください。

エクスポート時の既存ファイル操作モード（file_mode）

エクスポート時の既存ファイル操作モードを設定します。

キー	file_mode
設定	任意
設定値（省略時）	1～3 （1）

設定可能なエクスポート時の既存ファイル操作モードと処理概要

モード	処理概要
1	既存ファイルを操作しない（追記）
2	既存ファイルを削除し、エクスポート
3	既存ファイルをアーカイブ(移動)し、エクスポート

エクスポートファイルのアーカイブ先パス（archive_path）

ファイル操作モードでアーカイブを指定した場合のエクスポートファイルのアーカイブ先パスを設定します。
アーカイブ先ファイルパスは、パブリックストレージからのパスを指定してください。

キー	archive_path
設定	任意
設定値（省略時）	文字列 （kaiden/generic/master/{MASTER_ID}/export_{YMDHMSN}.csv）



コラム

「{MASTER_ID}」などの置換文字列の詳細は「[ファイルパスの置換文字列](#)」を参照してください。

エクスポートファイルの文字コード（character）

エクスポートファイルの文字コードを設定します。

キー	character
設定	任意
設定値（省略時）	UTF-8, SHIFT-JISなど （UTF-8）

エクスポートファイルの区切り文字（delimiter）

エクスポートファイルの区切り文字を設定します。

キー	delimiter
設定	任意
設定値（省略時）	,[カンマ], \t[タブ] （,）

エクスポートファイルの囲み文字（enclosing）

エクスポートファイルの囲み文字を設定します。

キー	enclosing
設定	任意
設定値（省略時）	none, "[ダブルクォーテーション]" （"）

エクスポートファイルのヘッダー出力有無（header_row）

エクスポートファイルの読込開始行数を設定します。

キー	header_row
----	------------

設定	任意
設定値（省略時）	true, false （false）

マスタID（masterId）

エクスポートを行うマスタのマスタIDを設定します。

キー	masterId
設定	必須
設定値（省略時）	文字列

マスタ拡張ID（extensionId）

エクスポートを行うマスタのマスタ拡張IDを設定します。

キー	extensionId
設定	任意
設定値（省略時）	文字列（未指定）

パラメータの優先度

指定したパラメータの優先度は次の通りです。

1. [インポート設定](#)、または [エクスポート設定](#)
2. ジョブ（ネット）パラメータ
3. 初期値（省略時の値）

ファイルパスの置換文字列

各ファイルパスには次の置換文字列が使用可能です。

いずれも置換内容にあたる値が設定されていない場合は置換処理を行いません。

置換文字列	置換内容
{COMPANY_CD}	会社コード（会社別の処理ではない場合は置換されません。）
{YMD}	「yyyyMMdd」形式のジョブ実行日時
{YMDHMSN}	「yyyyMMddHHmmssSSS」形式のジョブ実行日時
{JOB_NET_ID}	ジョブネットID
{JOB_NET_NAME}	ジョブネット名
{JOB_ID}	ジョブID
{JOB_NAME}	ジョブ名
{MASTER_ID}	マスタID
{EXTENSION_ID}	マスタ拡張ID

インポートファイルのカラム判定について

インポートジョブでは、「[インポート設定](#)」の設定に応じてインポートファイルのカラム順が変動します。

本項では、「[インポート設定](#)」の設定に合わせたカラムの変動内容を解説します。

- 「xxxxCd,xxxxName」が設定された場合
- 「companyCd,xxxxCd,xxxxName」が設定された場合
- 「companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName」が設定された場合

「xxxxCd,xxxxName」が設定された場合

```
<importJob>
<items>
  <item name="xxxxCd" />
  <item name="xxxxName" />
</items>
</importJob>
```

1,2,3	x	x	x	xxxxCd,xxxxName
	○	x	x	companyCd,xxxxCd,xxxxName
	x	○	x	deleteFlag,xxxxCd,xxxxName
	x	x	○	localeId,xxxxCd,xxxxName
	○	○	x	companyCd,deleteFlag,xxxxCd,xxxxName
	○	x	○	companyCd,localeId,xxxxCd,xxxxName
	x	○	○	deleteFlag,localeId,xxxxCd,xxxxName
	○	○	○	companyCd,deleteFlag,localeId,xxxxCd,xxxxName
4,5	x	x	x	xxxxCd,xxxxName
	○	x	x	companyCd,xxxxCd,xxxxName
	x	○	x	startDate,endDate,deleteFlag,xxxxCd,xxxxName
	x	x	○	localeId,xxxxCd,xxxxName
	○	○	x	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	x	○	companyCd,localeId,xxxxCd,xxxxName
	x	○	○	startDate,endDate,deleteFlag,localeId,xxxxCd,xxxxName
	○	○	○	companyCd,startDate,endDate,deleteFlag,localeId,xxxxCd,xxxxName

「companyCd,xxxxCd,xxxxName」が設定された場合

```
<importJob>
<items>
  <item name="companyCd" />
  <item name="xxxxCd" />
  <item name="xxxxName" />
</items>
</importJob>
```

インポートモード	会社化	期間化	国際化	カラム
1,2,3	x	x	x	companyCd,xxxxCd,xxxxName
	○	x	x	companyCd,xxxxCd,xxxxName
	x	○	x	deleteFlag,companyCd,xxxxCd,xxxxName
	x	x	○	localeId,companyCd,xxxxCd,xxxxName
	○	○	x	deleteFlag,companyCd,xxxxCd,xxxxName
	○	x	○	localeId,companyCd,xxxxCd,xxxxName
	x	○	○	deleteFlag,localeId,companyCd,xxxxCd,xxxxName
	○	○	○	deleteFlag,localeId,companyCd,xxxxCd,xxxxName
4,5	x	x	x	companyCd,xxxxCd,xxxxName
	○	x	x	companyCd,xxxxCd,xxxxName
	x	○	x	startDate,endDate,deleteFlag,companyCd,xxxxCd,xxxxName
	x	x	○	localeId,companyCd,xxxxCd,xxxxName
	○	○	x	startDate,endDate,deleteFlag,companyCd,xxxxCd,xxxxName
	○	x	○	localeId,companyCd,xxxxCd,xxxxName
	x	○	○	startDate,endDate,deleteFlag,localeId,companyCd,xxxxCd,xxxxName
	○	○	○	startDate,endDate,deleteFlag,localeId,companyCd,xxxxCd,xxxxName

「companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName」が設定された場合

```
<importJob>
<items>
  <item name="companyCd" />
  <item name="startDate" />
  <item name="endDate" />
  <item name="deleteFlag" />
  <item name="xxxxCd" />
  <item name="xxxxName" />
</items>
</importJob>
```

インポートモード	会社化	期間化	国際化	カラム
1,2,3	×	×	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	×	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	○	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	×	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	○	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	×	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	○	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	○	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
4,5	×	×	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	×	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	○	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	×	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	○	×	companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	×	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	×	○	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName
	○	○	○	localeId,companyCd,startDate,endDate,deleteFlag,xxxxCd,xxxxName



コラム

インポートモード1,2,3の場合、「startDate,endDate」に指定されたインポートファイルのデータは、ジョブパラメータで上書きされます。）

マスタ検索、バリデーションのみを使用するマスタの作成

- 概要
- マスタ検索、バリデーションのみを使用する場合
- Enumクラスを使用した検索部品を作成する場合
- XMLの固定値を使用した検索部品を作成する場合

概要

MasterBuilderでは、マスタ検索、バリデーションのみを使用するマスタを作成可能です。
作成する場合、拡張設定に拡張用のLogicクラスを設定します。

マスタ検索、バリデーションのみを使用する場合

拡張設定に「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSearchLogic」を設定します。
Logicをカスタマイズする場合、「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSearchLogicBase」を継承したクラスを作成してください。

Enumクラスを使用した検索部品を作成する場合

「jp.co.slcs.kaiden.v2.base.foundation.taglib.EnumListDataIF」を継承したEnumクラスを作成してください。

Enumクラス

```
package jp.co.slcs.kaiden.v2.extension_imw.foundation.conf;

import java.util.Locale;
import jp.co.slcs.kaiden.v2.base.foundation.exception.KaidenIllegalParametersException;
import jp.co.slcs.kaiden.v2.base.foundation.extension.spring.context.ApplicationContextHolder;
import jp.co.slcs.kaiden.v2.base.foundation.helper.ContextHelper;
import jp.co.slcs.kaiden.v2.base.foundation.helper.PropertyHelper;
import jp.co.slcs.kaiden.v2.base.foundation.taglib.EnumListDataIF;

/**
 * 代理申請状態を表す列挙.
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
public enum AgentType implements EnumListDataIF {

    /** 本人申請. */
    PRINCIPAL("0", "EIMW.AgentType.principal"),
    /** 代理申請. */
    AGENT("1", "EIMW.AgentType.agent");
}
```

```

/** コード */
private final String code;
/** 名称 */
private final String surface;

/**
 * コンストラクタ.
 *
 * @param code コード
 * @param name 名称
 */
private AgentType(String code, String name) {
    this.code = code;
    surface = name;
}

/**
 * {@inheritDoc}
 */
@Override
public String getCode() {
    return code;
}

/**
 * {@inheritDoc}
 */
@Override
public String getName() {
    ContextHelper contextHelper = ApplicationContextHolder.getApplicationContext().getBean(ContextHelper.class);
    return getName(contextHelper.getLoginUserLocale());
}

/**
 * コードから自身を返却.
 *
 * @param code コード値
 * @return 自身
 */
public static AgentType codeOf(String code) {
    for (AgentType that : AgentType.values()) {
        if (that.getCode().equals(code)) {
            return that;
        }
    }
    throw KaidenIllegalParametersException.makeInstance();
}

/**
 * 名称から自身を返却.
 *
 * @param name 名称
 * @return 自身
 */
public static AgentType nameOf(String name) {
    for (AgentType that : AgentType.values()) {
        if (name.equals(that.getName())) {
            return that;
        }
    }
    throw KaidenIllegalParametersException.makeInstance();
}

/**
 * {@inheritDoc}
 */
@Override
public String getName(Locale locale) {
    PropertyHelper propertyHelper = ApplicationContextHolder.getApplicationContext().getBean(PropertyHelper.class);
    return propertyHelper.getSurface(locale, surface);
}
}

```

Enumクラスをリスト化するために「jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSearchEnumLogicBase」を継承したクラスを作成し、拡張設定に設定してください。

LimitationMasterSearchEnumLogicBaseを継承したクラス

```
package jp.co.slcs.kaiden.v2.extension_imw.feature.logic.kaiden.generic;

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.feature.logic.kaiden.generic.LimitationMasterSearchEnumLogicBase;
import jp.co.slcs.kaiden.v2.base.foundation.taglib.EnumListDataIF;
import jp.co.slcs.kaiden.v2.extension_imw.foundation.conf.AgentType;

/**
 * 代理タイプ検索(Logic).
 *
 * @author Sumitomo Life Information Systems Co.,Ltd.
 */
@Component
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public class AgentTypeLogic extends LimitationMasterSearchEnumLogicBase {

    /**
     * {@inheritDoc}
     */
    @Override
    protected EnumListDataIF[] values() {
        return AgentType.values();
    }
}
```

XMLの固定値を使用した検索部品を作成する場合

拡張設定に「jp.co.slcs.kaiden2.base.feature.logic.kaiden.generic.LimitationMasterSelectLogic」を設定します。

XMLに表示したい固定値を定義してください。

設定

```
<?xml version="1.0" encoding="UTF-8"?>
<genericMasterSettings xmlns="http://kaiden.slcs.co.jp/xmlschema/genericMasterSettings">
  <masterId>use_data_status</masterId>
  <surfaceKey>master_name</surfaceKey>
  <extension>
    <logic>jp.co.slcs.kaiden.v2.base.feature.logic.kaiden.generic.LimitationMasterSelectLogic</logic>
  </extension>

  <args>
    <arg key="datas"
      value="[
        {'code':'00', 'name':'untreated'},
        {'code':'10', 'name':'processing'},
        {'code':'20', 'name':'processed'},
        {'code':'90', 'name':'no_process_required'}
      ]"
    />
  </args>

  <!-- 検索設定 -->
  <search type="SELECT">
    <items>
      <item name="code" type="CODE" />
      <item name="name" type="NAME" />
    </items>
  </search>

  <!-- ラベル設定 -->
  <surfaces>
    <!-- 略 -->
    <surface surfaceKey="untreated" surface="未処理">
      <name localeId="ja" surface="未処理" />
      <name localeId="en" surface="Untreated" />
      <name localeId="zh_CN" surface="未理" />
    </surface>
    <!-- 略 -->
  </surfaces>
</genericMasterSettings>
```

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

概要

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。

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「[設定の流れ](#)」 - 「Service、Entity作成」を参照ください。

MasterBuilder定義

カスタマイズしたMasterBuilder定義の修正方法を説明します。

拡張設定

拡張設定に、標準で提供しているJavaのクラス、SQLパス、JSPパスを設定している場合、次の通り修正してください。

- Javaのクラス
「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
「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
「/WEB-INF/view/kaiden2」の部分を「/WEB-INF/views/kaiden/v2」に変更してください。

検索用SQL

検索用SQLをカスタマイズしている場合、次の格納先にSQLを移動してください。
MasterBuilder定義の拡張設定にSQLパスを設定している場合、格納先の移動は不要です。

%ModuleProject%/src/main/resources/META-INF/sql/jp/co/slcs/kaiden/v2/generic/master/%マスタID%.sql
(マスタIDには、MasterBuilder定義で設定したマスタIDを指定してください。)

動作確認

カスタマイズしたMasterBuilderの動作確認方法について説明します。

メンテナンス機能

メニューのURLを「kaiden/generic/master」から「kaiden/v2/generic/master」に変更してください。
MasterBuilder定義の拡張設定にマスタメンテナンス用の処理URLを設定している場合、URLの変更は不要です。

インポートジョブ

ジョブの実行プログラムを「jp.co.slcs.kaiden2.base.feature.job.generic.MasterImportJob」から
「jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterImportJob」に変更してください。

エクスポートジョブ

ジョブの実行プログラムを「jp.co.slcs.kaiden2.base.feature.job.generic.MasterExportJob」から
「jp.co.slcs.kaiden.v2.base.feature.job.generic.MasterExportJob」に変更してください。

Java、JSPソース

Java、JSPソースを拡張している場合、『[intra-mart Accel Kaiden! プログラミングガイド](#)』の「カスタマイズ時の注意事項」を参照して修正してください。