Copyright © 2012 NTT DATA INTRAMART CORPORATION

目次

- 改訂情報
- はじめに
  - 対象読者
  - 本書の構成
  - 本書内の説明について

索引

- Resin
  - resin-web設定
  - ヘルス機能
  - ログ設定
- コアモジュール
  - ネットワーク設定
  - サーバコンテキスト設定
  - ストレージ設定ファイル
  - サービスセレクタ設定
  - Identifier 設定
  - initializer 設定
  - IPアドレス取得元設定
- マルチデバイス
  - クライアントタイプマスタ
- 国際化機能
  - 日付と時刻の形式マスタ
  - ロケールマスタ
  - タイムゾーンマスタ
  - 数値形式マスタ
- スクリプト開発モデル
  - source-config.xml
- データベースアクセス機能
  - データソースマッピング設定
- メールモジュール
  - メール設定
- Webモジュール
  - リクエスト制御設定
  - レスポンスヘッダ設定
- IM-Propagation
  - IM-Propagation 設定
  - IM-Propagation 受信側設定
  - IM-Propagation 送信側設定
- テナント管理機能
  - システム管理者用スクリプト開発モデルルーティング設定
  - システム管理者用IM-JavaEE Frameworkルーティング設定
  - システム管理者用サーブレットルーティング設定
  - 認可判断モジュール設定
  - 認可設定画面設定
  - ポリシー部分編集定義設定
  - 認可設定画面 保護リソース設定
  - ルーティングテーブル用 認可リソースマッパー定義設定
  - リソースタイプ拡張設定
  - サブジェクトタイプ拡張設定
  - 暗号化設定
  - サブジェクトリゾルバ(DeclaredSubjectResolver)拡張設定
  - サブジェクトリゾルバ(OnDemandSubjectResolver)拡張設定
  - 認可ポリシーキャッシュ対象設定
  - パスワード履歴管理設定
  - スクリプト開発モデルルーティング設定
  - IM-JavaEE Frameworkルーティング設定
  - サーブレットルーティング設定
  - ショートカットアクセス設定
  - システム期間情報の管理
  - セキュア・トークンフィルタ設定
- サービス機構モジュール
  - キャッシュ設定
  - キャッシュデフォルト設定
- UI
  - セッション管理設定
  - UI タグコンポーネント設定
  - テーマの適用方法設定 HeadWithFooterThemeBuilder
  - テーマの適用方法設定 HeadWithContainerThemeBuilder
  - テーマの適用方法設定 HeadOnlyThemeBuilder
  - テーマの適用方法設定 BodyOnlyThemeBuilder
  - テーマの適用方法設定 NoThemeBuilder
  - テーマの適用方法設定 FullThemeBuilder
  - ライブラリ群設定
  - SP版ユーティリティメニュー設定
- Webサービス 認証・認可クライアント
  - SOAPClientオブジェクトの設定
- 認証機能
  - 認証設定（一般ユーザ用）
  - 認証確認設定（一般ユーザ用）
  - 認証確認対象ページ設定（一般ユーザ用）
  - 認証セッション設定（一般ユーザ用）
  - 認証外部ページURL許可リスト設定
  - リクエストパラメータ設定
  - セッション情報チェック設定
- 統合Windows認証機能
  - 統合Windows認証設定
  - 統合Windows認証パス設定
- システム管理機能
  - 認証設定(システム管理者用)
  - 認証IPアドレス制限設定(システム管理者用)
  - 認証確認設定(システム管理者用)
  - 認証確認対象ページ設定(システム管理者用)
  - 認証セッション設定(システム管理者用)
- LDAP認証モジュール
  - LDAP認証設定ファイル
- ジョブスケジューラ
  - ジョブスケジューラの設定
- パスワードリマインダ機能
  - パスワードリマインダ設定
- IM-共通マスタ
  - IM-共通マスタ設定
- ViewCreator
  - ViewCreatorの設定
  - ViewCreatorの関数設定
  - ViewCreatorファイルリソースの設定
  - グラフ描画ライブラリ情報の設定
- IMBox
  - Cassandra設定
  - IMBox設定
  - チャット表示設定
- IM-ContentsSearch
  - Solrサーバ接続設定
  - 検索画面設定
  - 検索権限生成設定

- システム管理者用グローバルナビ設定
- システム管理者用ホームウィジェット設定
- システム管理者用ユーティリティメニュー設定
- IM-Notice
  - IM-Notice 設定
  - IM-Notice Mobile設定
  - IM-Notice MQ設定
  - IM-Notice MQブローカー設定
- 招待機能と外部ユーザ
  - 招待権限リスト設定
  - 招待メールデフォルト設定
- リバースプロキシ
  - リバースプロキシの設定
- Salesforce連携
  - Salesforceクライアントの設定
- Kibanaポートレット
  - Kibana ポートレットの設定
- セッション管理モジュール
  - セッションストア設定
  - セッションリスナ設定
- セッション管理 組込Hazelcast連携
  - Hazelcast設定
- 外部メニュー連携
  - 外部メニュー連携接続先設定
- iAP-iWP間SSO連携モジュール(IM-HybridSSO)
  - SSO連携用マッピング設定
- 検索フィールド設定
- テキスト抽出設定
- 検索結果テンプレート設定
- IM-FileExchange
  - IM-FileExchangeの設定
- OAuth認証機能
  - クライアント詳細設定
  - クライアントリソース設定
  - クライアントのアクセス範囲設定
- OAuthクライアント
  - OAuthプロバイダ設定
- ログ
- 互換モジュール
  - 汎用設定ファイル
- IBM Watson 設定
  - IBM Watson クライアント設定

索引

Resin

resin-web設定

項目
- 概要
- リファレンス
  - ログハンドラ設定
  - リソース設定
  - JSP設定
  - タグ再利用設定
  - データベース設定
  - ドライバ設定
  - ドライバタイプ設定
  - ドライバURL設定
  - ユーザ設定
  - パスワード設定
  - 暗号化パスワード設定
  - 最大コネクション数設定
  - プリペアドステートメントキャッシュ設定
  - セッション設定
  - セッションIDの再利用設定
  - セッションタイムアウト設定
  - JSPプリコンパイル設定

概要

Resinに関する設定です。

モジュール	Resin
フォーマットファイル(xsd)	なし
設定場所	WEB-INF/resin-web.xml

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:j
 <log-handler name="" class="jp.co.intra_mart.common.platform
 <!-- im_service(im_asynchronous) -->
 <resource jndi-name="jca/work" type="jp.co.intra_mart.system.
 <jsp>
  <recycle-tags>false</recycle-tags>
 </jsp>
 <database jndi-name="jdbc/default">
  <driver>
   <type>org.postgresql.Driver</type>
   <url>jdbc:postgresql://localhost:5432/iap_db</url>
   <user>imart</user>
   <password>imart</password>
  </driver>
  <max-connections>20</max-connections>
  <prepared-statement-cache-size>8</prepared-statement
 </database>
 <session-config>
  <reuse-session-id>false</reuse-session-id>
  <session-timeout>30</session-timeout>
 </session-config>
</web-app>
```

リファレンス

ログハンドラ設定

タグ名	log-handler

改訂情報

変更年月日	変更内容
2012-12-21	初版
2013-04-01	第2版 下記を追加・変更しました ■ 「*IPアドレス取得元設定* 」を追加 ■ 「*認証IPアドレス制限設定(システム管理者用)* 」を追加 ■ 「*IM-FileExchangeの設定* 」を追加 ■ 「*認可設定画面設定* 」を追加 ■ 「*リソースタイプ拡張設定* 」を追加 ■ 「*LDAP認証設定ファイル* 」を追加 ■ 「*外部メニュー連携接続先設定* 」を追加 ■ 「*ポリシー部分編集定義設定* 」に「*コールバック設定* 」を追加 ■ 「*Cassandra設定* 」に「*キースペース* 」を追加 ■ 「*IMBox設定* 」に「*Noticeタイムラインのメッセージ表示件数* 」、「*1メッセージの通知先に指定できるユーザ数の上限値* 」、「*1メッセージに添付可能なファイル数の上限値* 」を追加
2013-07-01	第3版 下記を追加・変更しました ■ 「*本書内の説明について* 」を追加 ■ 「*UI タグコンポーネント設定* 」を追加 ■ 「*テーマの適用方法設定 HeadWithFooterThemeBuilder* 」を追加 ■ 「*テーマの適用方法設定 HeadWithContainerThemeBuilder* 」を追加 ■ 「*テーマの適用方法設定 HeadOnlyThemeBuilder* 」を追加 ■ 「*テーマの適用方法設定 BodyOnlyThemeBuilder* 」を追加 ■ 「*テーマの適用方法設定 NoThemeBuilder* 」を追加 ■ 「*ViewCreatorの関数設定* 」を追加 ■ 「*認可設定画面設定* 」に「*リソースグループバックアップ設定* 」「*バックアップ先パス設定* 」「*バックアップ最大件数設定* 」を追加 ■ 「*Cassandra設定*」に「*接続認証設定*」「*接続ユーザ名*」「*パスワード*」を追加
2013-10-01	第4版 下記を追加・変更しました ■ 「*外部メニュー連携接続先設定* 」に「メニュー設定画面での表示順序設定」に関する説明を追加 ■ 「システムログ 」に「ログメッセージコード」に関する説明を追加 ■ 「*キャッシュ設定* 」に「属性」に関する説明を追加、更新 ■ 「*キャッシュデフォルト設定* 」に「属性」に関する説明を追加、更新 ■ 「*resin-web設定* 」の「プリペアドステートメントキャッシュ設定」に関する説明を更新 ■ 「*汎用設定ファイル* 」の「*VirtualFile72, NetworkFile72 APIにおけるエンコーディングの設定*」に関する説明を追加 ■ 「*サービスセレクタ設定* 」を追加 ■ 「*暗号化設定* 」の設定ファイルの設置場所を更新 ■ 「*IMBox設定* 」に「非表示のグループ一覧のグループ表示件数」に関する説明を追加 ■ 「*サーブレットルーティング設定* 」に認可を使用する際の注意点を追加 ■ 「*メール設定* 」に「SMTPS(SMTP over SSL)設定」に関する説明を追加

ログハンドラに関する設定です。

【設定項目】

```
<web-app>
  <log-handler name="" class="jp.co.intra_mart.common.platform
</web-app>
```

必須項目	×
複数設定	○
設定値・設定する内容	ログハンドラに関する設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

【属性】

属性名	説明	必須	デフォルト値
name	ログハンドラの名前を指定します。	○	なし
class	ログハンドラのクラス名を指定します。	○	なし

> **注意**
>
> この設定は変更しないでください。

リソース設定

タグ名	resource

リソースに関する設定です。

【設定項目】

```
<web-app>
  <resource jndi-name="jca/work" type="jp.co.intra_mart.system.a
</web-app>
```

必須項目	×
複数設定	○
設定値・設定する内容	リソースに関する設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

【属性】

属性名	説明	必須	デフォルト値
jndi-name	リソースのJNDI名を指定します。	○	なし
type	リソースのタイプを指定します。	○	なし

変更年月日	変更内容
2014-01-01	第5版 下記を追記・変更しました ■ 「ログ」の項目を「ログ仕様書」に移動 ■ 「*認可判断モジュール設定*」の「decision-config」タグ「combinator」属性の説明を修正 ■ 「*キャッシュ設定*」にサイズ計算に関する警告を追加 ■ 「*キャッシュデフォルト設定*」にサイズ計算に関する警告を追加 ■ 「*IM-ContentsSearch*」に「*検索画面設定*」 を追加 ■ 「*IM-ContentsSearch*」に「*検索権限生成設定*」 を追加 ■ 「*IM-ContentsSearch*」に「*テキスト抽出設定*」 を追加 ■ 「*IM-ContentsSearch*」に「*検索フィールド設定*」 を追加 ■ 属性 path に関するワイルドカードの説明を追加 　■ 「*システム管理者用スクリプト開発モデルルーティング設定*」 　■ 「*システム管理者用IM-JavaEE Frameworkルーティング設定*」 　■ 「*システム管理者用サーブレットルーティング設定*」 　■ 「*スクリプト開発モデルルーティング設定*」 　■ 「*IM-JavaEE Frameworkルーティング設定*」 　■ 「*サーブレットルーティング設定*」 ■ 「*IMBox設定*」に「*表示に関する設定*」を追加 ■ 「*IMBox設定*」に「*利用不可ユーザを表示するか否かを判定する値*」を追加 ■ 「*IMBox設定*」に「*アンケートに投票をしたユーザ一覧のユーザ表示件数*」を追加 ■ 「*IMBox設定*」に「*利用不可ユーザのユーザ一覧のユーザ表示件数*」を追加 ■ 「*パスワード履歴管理設定*」の「*パスワード履歴管理対象外クライアントタイプ*」に関する省略時のデフォルト値を更新 ■ 「*レスポンスヘッダ設定*」を追加 ■ 「*テーマの適用方法設定 HeadWithFooterThemeBuilder*」のテーマの適用方法の参照先を変更しました。 ■ 「*テーマの適用方法設定 HeadWithContainerThemeBuilder*」のテーマの適用方法の参照先を変更しました。 ■ 「*テーマの適用方法設定 HeadOnlyThemeBuilder*」のテーマの適用方法の参照先を変更しました。 ■ 「*テーマの適用方法設定 BodyOnlyThemeBuilder*」のテーマの適用方法の参照先を変更しました。 ■ 「*テーマの適用方法設定 NoThemeBuilder*」のテーマの適用方法の参照先を変更しました。 ■ 「*IM-FileExchangeの設定*」の「listener」タグ「show-guest-address」属性を追加 ■ 「*メール設定*」の「smtps」タグ「enable」属性の説明を修正 ■ 「*メール設定*」の「smtps」タグ「starttls」属性の説明を修正 ■ 「*メール設定*」の「encode」タグ「locale」属性の説明を修正 ■ 「*キャッシュ設定*」の「cache」タグ「enable」属性の説明を修正 ■ 「*キャッシュデフォルト設定*」の「cache」タグを「default-cache」に修正 ■ 「*キャッシュデフォルト設定*」の「mbeans」タグを追加

> **注意**
> この設定は変更しないでください。

JSP設定

タグ名	jsp

JSPに関する設定です。

【設定項目】

```
<web-app>
 <jsp>
  ...
 </jsp>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	JSPの設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

タグ再利用設定

タグ名	recycle-tags

JSPに関する設定です。
タグクラスのインスタンスを再利用します。

【設定項目】

```
<web-app>
 <jsp>
  <recycle-tags>false</recycle-tags>
 </jsp>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	true　タグのインスタンスを再利用します。 false　タグのインスタンスを再利用しません。
単位・型	真偽値 (true/false)
省略時のデフォルト値	なし
親タグ	jsp

変更年月日	変更内容
2014-04-01	第6版 下記を追記・変更しました ■ 「*外部メニュー連携接続先設定*」の「menu-provider」タグに「target-tenant」属性を追加 ■ 「*外部メニュー連携接続先設定*」の「メニュープロバイダの接続先ログイングループ設定」に関する説明を更新 ■ 「*統合Windows認証機能*」を追加 ■ 「*IMBox設定*」の「tns:comet_flag」タグの説明にアプリケーション通知を追加 ■ 「*IMBox設定*」の「tns:mail」タグと「tns:from」タグに注意事項を追加 ■ 「*パスワード履歴管理設定*」に「group」タグの説明を追加 ■ 「*パスワード履歴管理設定*」の各タグ・属性に関してデフォルト値の説明を更新 ■ 「*メール設定*」の「smtp-server」タグに関する説明を更新 ■ 「*UI*」から「GoogleChromeFrame設定」を削除 ■ 「*パスワードリマインダ設定*」に「urlLimitDateFormat」タグの説明を追加 ■ 「*パスワードリマインダ設定*」に「mailCc」タグの説明を追加 ■ 「*パスワードリマインダ設定*」に「mailBcc」タグの説明を追加 ■ 「*パスワードリマインダ設定*」に「address」タグの説明を追加 ■ 「*ストレージ設定ファイル*」の「パブリックストレージディレクトリ名設定」に関数説明を更新 ■ 「*システム管理者用グローバルナビ設定*」を追加 ■ 「*システム管理者用ホームウィジェット設定*」を追加 ■ 「*システム管理者用ユーティリティメニュー設定*」を追加 ■ 「*認証設定(システム管理者用)*」に「強制ログイン時のセッション再作成の有無」パラメータの説明を追加 ■ 「*ショートカットアクセス設定*」の概要を更新しました。 ■ 「*パスワード履歴管理設定*」の概要を更新しました。 ■ 「*認証設定（一般ユーザ用）*」に「自動ログインについて」と、「強制ログイン時のセッション再作成の有無」パラメータの説明を追加 ■ 「*リクエストパラメータ設定*」に「im_tenant_id」パラメータの説明を追加
2014-05-30	第7版 下記を追記・変更しました ■ 「*ヘルス機能*」を追加 ■ 「*initializer 設定*」を追加
2014-08-01	第8版 下記を追記・変更しました ■ 「*ログ設定*」を追加 ■ 「*検索結果テンプレート設定*」を追加 ■ 「*resin-web設定*」の「max-connections」タグのデフォルト値を修正 ■ 「*テキスト抽出設定*」の概要に記載しているサンプルを更新 ■ 「*検索画面設定*」に「sort」タグの説明を追加 ■ 「*検索画面設定*」に「default-order」タグの説明を追加 ■ 「*LDAP認証設定ファイル*」の「connect-timeout」タグの単位・型を修正 ■ 「*SSO連携用マッピング設定*」を追加 ■ 「*認可設定画面設定*」に「*Excelインポート設定*」タグの説明を追加 ■ 「*認可設定画面設定*」に「*Excelエクスポート設定*」タグの説明を追加 ■ 「*リクエストパラメータ設定*」で「ログイン・ログアウト時に利用する遷移先URL」設定時の注意を追記

> **注意**
> この設定は変更しないでください。

データベース設定

タグ名	database

データベース接続に関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  ...
 </database>
</web-app>
```

必須項目	×
複数設定	○
設定値・設定する内容	データベース接続を行うための設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

【属性】

属性名	説明	必須
jndi-name	このデータベース設定のJNDI名を設定します。	○

ドライバ設定

タグ名	driver

JDBCドライバに関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  <driver>
   ...
  </driver>
 </database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	JDBCドライバの設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	database

変更年月日	変更内容
2014-12-01	第9版 下記を追記・変更しました ■ 「*OAuth認証機能* 」を追加 ■ 「*IMBox設定* 」に「*スレッド内の返信メッセージ取得件数* 」を追加 ■ 「*スクリプト開発モデル* 」を追加 ■ 「*IM-Notice* 」を追加 ■ 「*LDAP認証設定ファイル* 」は、ひな形として利用されることを追記 ■ 「*セッション管理設定* 」の注意の内容を修正
2015-04-01	第10版 下記を追記・変更しました ■ 「*サービスセレクタ設定* 」 リスナ設定のコラムからQueueServiceのIDを削除しました。 ■ 「*リクエスト制御設定* 」処理中のリクエストの有効期間単位についてを追記 ■ 「*メール設定*」に「mail-headers」タグの説明を追加 ■ 「*メール設定*」に「header」タグの説明を追加 ■ 「*IMBox設定* 」に「*Unread Messages（PC版）のメッセージ表示件数*」を追加 ■ 「*IMBox設定* 」に「*Unread Messages（SP版）のメッセージ表示件数*」を追加 ■ 「*ロケールマスタ* 」 の システムロケール変更に関する説明を修正

ドライバタイプ設定

タグ名	type

JDBCドライバのタイプに関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  <driver>
   <type>org.postgresql.Driver</type>
   ...
  </driver>
 </database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	JDBCドライバのタイプを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

ドライバURL設定

タグ名	url

JDBCドライバのURLに関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  <driver>
   ...
   <url>jdbc:postgresql://localhost:5432/iap_db</url>
   ...
  </driver>
 </database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	JDBCドライバのURLを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

ユーザ設定

タグ名	user

変更年月日	変更内容
2015-08-01	第11版 下記を追記・変更しました ■ 「*resin-web設定* 」に「*JSPプリコンパイル設定*」を追加。 ■ 「*source-config.xml* 」に「*source-config.xmlの有効範囲* 」を追加 ■ 「*IMBox設定* 」に「*非表示メッセージタイムラインのスレッド表示件数*」を追加 ■ 「*IMBox設定* 」に「*入力チェック設定*」を追加 ■ 「*IMBox設定* 」に「*グループ名の重複チェック設定*」を追加 ■ 「*テーマの適用方法設定 HeadWithFooterThemeBuilder* 」に属性client-typeとlibraries-pathを追加 ■ 「*テーマの適用方法設定 HeadWithContainerThemeBuilder* 」に属性client-typeとlibraries-pathを追加 ■ 「*テーマの適用方法設定 HeadOnlyThemeBuilder* 」に属性client-typeとlibraries-pathを追加 ■ 「*テーマの適用方法設定 BodyOnlyThemeBuilder* 」に属性client-typeとlibraries-pathを追加 ■ 「*テーマの適用方法設定 NoThemeBuilder* 」に属性client-typeとlibraries-pathを追加 ■ 「*テーマの適用方法設定 FullThemeBuilder* 」を追加 ■ 「*ライブラリ群設定* 」を追加 ■ 「*認証設定（一般ユーザ用）* 」の以下に関する警告を修正 　■ ログインページURL 　■ ログイン実行ページURL 　■ ログアウト実行ページURL ■ 「*認証設定(システム管理者用)* 」の以下に関する警告を修正 　■ ログインページURL 　■ ログイン実行ページURL 　■ テナント初期設定ページURL ■ 「*認証確認設定（一般ユーザ用）* 」の以下に関する警告を修正 　■ 認証確認ページURL 　■ 認証確認実行ページURL ■ 「*認証確認設定(システム管理者用)* 」の以下に関する警告を修正 　■ 認証確認ページURL 　■ 認証確認実行ページURL ■ 「*IM-Notice* 」に「*Baidu APIキーの設定*」を追加

接続するデータベースのユーザに関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  <driver>
   ...
   <user>imart</user>
   ...
  </driver>
 </database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	データベースのユーザを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

パスワード設定

タグ名	password

接続するデータベースのユーザのパスワードに関する設定です。

【設定項目】

```
<web-app>
 <database jndi-name="jdbc/default">
  <driver>
   ...
   <password>imart</password>
   ...
  </driver>
 </database>
<web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	データベースのユーザのパスワードを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

暗号化パスワード設定

タグ名	resin:Password

接続するデータベースのユーザのパスワードに関する設定です。
暗号化されたパスワードを使用する場合に使用します。

このタグを使用する場合は、予めパスワードを暗号化してください。パ

変更年月日	変更内容
2015-12-01	第12版 下記を追記・変更しました ■ 「source-config.xml 」に「ファイル単位での設定方法 」を追加 ■ 「source-config.xml 」に「HTMLコメントを削除する設定 」を追加 ■ 「JSPプリコンパイル設定」の内容を修正 ■ 「パスワード履歴管理設定」の「パスワードの長さチェック」に 属性 min の最小値、および、属性 max の 最大値に関する説明を追記 ■ 「LDAP認証設定ファイル 」にバージョンによってファイルの扱いが異なることを追記 ■ 「OAuthクライアント 」に「OAuthプロバイダ設定 」を追加 ■ 「ジョブスケジューラ 」に「ジョブスケジューラサービスの設定 」を追記 ■ 「ジョブスケジューラ 」に「遅延起動の設定 」を追記 ■ 「ジョブスケジューラ 」に「起動判定処理の設定 」を追記 ■ 「ジョブスケジューラ 」に「最大遅延時間（秒） 」を追記 ■ 「ジョブスケジューラ 」に「起動判定チェック間隔（ミリ秒） 」を追記 ■ 「ライブラリ群の指定 」にclient-type に pc を指定した定義を追記 ■ 「テーマの適用方法設定 HeadWithFooterThemeBuilder 」の属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「テーマの適用方法設定 HeadWithContainerThemeBuilder 」に属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「テーマの適用方法設定 HeadOnlyThemeBuilder 」に属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「テーマの適用方法設定 BodyOnlyThemeBuilder 」に属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「テーマの適用方法設定 NoThemeBuilder 」に属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「テーマの適用方法設定 FullThemeBuilder 」に属性libraries-versionのclient-typeにpcを指定した場合の記述を追記 ■ 「resin-web設定 」の「プリペアドステートメントキャッシュ設定 」にPostgreSQL JDBC を使用する際の説明を追記 ■ 「ストレージ設定ファイル」に「シンボリックリンク設定」を追記 ■ 「ポリシー部分編集定義設定 」に、「resource-group-authorizer要素」に指定するクラスの型を追記 ■ 「LDAP認証設定ファイル」に「SSL接続(LDAPS)の設定」を追記

```
C:\resin-pro-4.x.xx>resin password-encrypt <パスワード>
password: {RESIN}WFpCuWyYSFzuOK1Gda0TxQ==
```

Linux環境の場合はresinctlコマンドを使用してください。
resinctlコマンドを使用するにはResinがコンパイル、インストール済み

```
[resin-pro-4.x.xx]# bin/resinctl password-encrypt <パスワード>
password: {RESIN}WFpCuWyYSFzuOK1Gda0TxQ==
```

このタグを使用するには設定ファイルに以下のネームスペースが記述さ

```
xmlns:resin="urn:java:com.caucho.resin"
```

【設定項目】

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:j
 ...
 <database jndi-name="jdbc/default">
  <driver>
   ...
   <password>
    <resin:Password value="{RESIN}WFpCuWyYSFzuOK1Gda0T
   </password>
   ...
  </driver>
 </database>
<web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	password

【属性】

属性名	説明	必須	デフォルト値
value	暗号化されたパスワードを設定します。	○	なし

最大コネクション数設定

タグ名	max-connections

データベースプールで利用可能なコネクションの最大数を設定します。

【設定項目】

変更年月日	変更内容
2016-04-01	第13版 下記を追記・変更しました ■ 「*日付と時刻の形式マスタ* 」に警告を追加 ■ 「*IM-共通マスタ設定*」に「*API モデルバリデーション*」を追記 ■ 「*パスワード暗号化アルゴリズムクラス*」にパスワードの保存方式が「暗号化」方式である場合に利用可能である旨の説明を追記 ■ 「*暗号化設定*」にパスワードの保存方式が「暗号化」方式である場合に利用可能である旨の説明を追記 ■ 「*暗号化設定*」のカテゴリの説明に役割に関する記述を追記 ■ 「*招待機能と外部ユーザ* 」を追加 ■ 「*リバースプロキシ*」を追加 ■ 「*テキスト抽出設定* 」の概要に記載しているサンプルを更新 ■ DocuWorks に対応するための設定値を追加 ■ 「*Identifier 設定*」を追加
2016-08-01	第14版 下記を追記・変更しました ■ 「*システム管理者用グローバルナビ設定* 」のメニュー設定に「uselframe」属性の説明を追記 ■ 「*システム管理者用ユーティリティメニュー設定* のメニュー設定に「uselframe」属性の説明を追記 ■ 「*パラメータ設定* 」の「name」属性の必須項目を「×」に修正 ■ 「*パラメータ設定* 」の「value」属性の必須項目を「×」に修正 ■ 「*テーマの適用方法設定 HeadWithFooterThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*テーマの適用方法設定 HeadWithContainerThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*テーマの適用方法設定 HeadOnlyThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*テーマの適用方法設定 BodyOnlyThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*テーマの適用方法設定 NoThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*テーマの適用方法設定 FullThemeBuilder* 」の「*パスの設定*」に注意を追記 ■ 「*リバースプロキシ*」の「proxy-pass-reverse」属性の複数設定を「○」に修正 ■ 「*リクエスト制御設定*」の利用されていない要素を削除 ■ 「inspection」要素を削除 ■ 「*ヘルスアクション*」にヘルスアクションの注意を追記 ■ 「*ジョブスケジューラの設定*」の「*ジョブストアの設定*」を追記 ■ 「*ジョブスケジューラの設定*」の「*ロック獲得設定*」を追記 ■ 「*ネットワーク設定*」の「*ホストアドレス設定*」にコラムを追記 ■ 「*ネットワーク設定*」の「*マルチキャストアドレス*」にコラムを追記 ■ 「*グラフ描画ライブラリ情報の設定* 」を追加 ■ 「*ViewCreatorの設定* 」のグラフ描画用プログラムパスに利用不可である旨の警告を追記 ■ 「*SP版ユーティリティメニュー設定* 」を追加 ■ 「*Salesforceクライアントの設定* 」を追加
2016-12-01	第15版 下記を追記・変更しました ■ 「*数値形式マスタ*」を追加 ■ DB2に関する記述を削除

```
<web-app>
 <database jndi-name="jdbc/default">
  ...
  <max-connections>20</max-connections>
 </database>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	コネクションの最大数
単位・型	数値
省略時のデフォルト値	1024
親タグ	database

プリペアドステートメントキャッシュ設定

タグ名	prepared-statement-cache-size **Postgresql Driver, Version 9.4-1202 以降メータに設定**

PreparedStatementを使用してSQLを発行すると、作成されたステート
次回、同じSQLが発行された場合、キャッシュされたステートメントが
prepared-statement-cache-sizeは、キャッシュを行うステートメント
intra-mart Accel Platform では、機能によりSQLの発行や内容が回数
その為、8, 16程度の個数が設定されている場合キャッシュにヒットせず
プリペアドステートメントのキャッシュが適切にヒットするよう設定す

【設定項目】

以下のJDBCドライバを使用する場合はprepared-statement-cache-si

- Oracle
- Microsoft SQLServer
- Postgresql Driver, Version 9.4-1201 以前

```
<web-app
 <database jndi-name="jdbc/default">
  ...
  <prepared-statement-cache-size>8</prepared-statement-cache
 </database>
</web-app>
```

Postgresql Driver, Version 9.4-1202 以降を使用する場合はJDBCドラ
prepared-statement-cache-sizeタグの設定は適用されません。

```
<web-app
 <database jndi-name="jdbc/default">
  <driver>
   ...
   <init-param>
    <param-name>preparedStatementCacheQueries</param-nar
    <param-value>8</param-value>
   </init-param>
  </driver>
 </database>
</web-app>
```

必須項目	×
複数設定	×

変更年月日	変更内容
2017-04-01	第16版 下記を追記しました ■ 「*Kibana ポートレットの設定*」を追加 ■ 「*セッション管理モジュール*」を追加 ■ 「*セッション管理 組込Hazelcast連携*」を追加 ■ 「*IBM Watson 設定*」を追加

設定値・設定する内容	プリペアドステートメントのキャッシュ数
単位・型	数値
省略時のデフォルト値	0
親タグ	database Postgresql Driver, Version 9.4-1202 以降の場合は

セッション設定

タグ名	session-config

セッションに関する設定です。

【設定項目】

```
<web-app>
 <session-config>
  ...
 </session-config>
</web-app>
```

必須項目	×
複数設定	○
設定値・設定する内容	セッションに関する設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

セッションIDの再利用設定

タグ名	reuse-session-id

HTTPセッションのIDを再利用するかどうかを設定します。
trueを設定した場合、過去に利用されたセッションIDを再利用します。
再利用されるのはIDのみで、過去に保管されていたセッションの情報は
ログイン、ログアウトにおいて、セッションを再作成しますが、その時

【設定項目】

```
<web-app>
 <session-config>
  <reuse-session-id>false</reuse-session-id>
 </session-config>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	HTTPセッションのIDを再利用するかどうかを設定し
単位・型	真偽値(true/false)
省略時のデフォルト値	true
親タグ	session-config

はじめに

項目
- 対象読者
- 本書の構成
- 本書内の説明について

対象読者

次の利用を対象としています。

intra-mart Accel Platform を利用したアプリケーション開発
intra-mart Accel Platform 上でのシステム運用

本書の構成

本書は intra-mart Accel Platform で提供されている設定ファイルに関する説明を行っています。

- 機能(モジュール)別の設定ファイルの種類
- 設定ファイル別の仕様説明
- 設定項目に対する設定内容ならびに注意点の説明

本書内の説明について

- 「デフォルト値」
  該当の設定を省略した場合の値となります。製品出荷時の値ではありません。

セッションタイムアウト設定

タグ名	session-timeout

HTTPセッションのタイムアウト時間です。指定した時間セッションを維

【設定項目】

```
<web-app>
 <session-config>
  <session-timeout>30</session-timeout>
 </session-config>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	HTTPセッションがタイムアウトされるまでの時間
単位・型	数値
省略時のデフォルト値	30
親タグ	session-config

> 注意
> セッションタイムアウトは、設定された時間に対して、最大1
> 例えば、セッションタイムアウトを30分に設定した場合、セッ

JSPプリコンパイル設定

Webアプリケーションに含まれるJSPをサーバ起動時にプリコンパイル
通常JSPはページがアクセスされた時にコンパイルされますが、JSPプリ

JSPプリコンパイル設定を行うには、**com.caucho.jsp.JspPrecomp**

```
<web-app>
 <listener>
  <listener-class>com.caucho.jsp.JspPrecompileListener</liste
  <init>
   <extension>jsp</extension>
  </init>
 </listener>
</web-app>
```

<extension> タグにはプリコンパイルを行うファイルの拡張子を指定
上記の例ではWebアプリケーションに含まれるすべてのjspファイルを

さらに詳細にプリコンパイル対象を指定する場合は **<fileset>** タグを

```
<web-app>
 <listener>
  <listener-class>com.caucho.jsp.JspPrecompileListener</liste
  <init>
   <fileset>
    <include>**/*.jsp</include>
    <exclude>/a.jsp</exclude>
    <exclude>/foo/c.jsp</exclude>
   </fileset>
  </init>
 </listener>
</web-app>
```

<include> タグはプリコンパイルの対象とするファイル名、**<exclud**
上記の例でWebアプリケーション内に含まれるすべてのパスのjspファイ

ヘルス機能

概要

Resinにはサーバの状態を監視するためのヘルス機能が含まれています。

得しPDFレポートを生成することが可能です。

主に、ヘルス機能は「診断」「メーター」「レポート」の機能から成り
的なシステム管理が可能となります。

例えば、ヘルス機能の「診断」部分を設定することにより、ヘルス機能
また、「メーター」とWeb管理コンソールとなる[/resin-admin]を設定

以下にそれぞれの機能の設定・使用方法を記述します。

/resin-admin

項目
- /resin-adminの設定

/resin-admin WebコンソールはResinサーバの管理を提供します。ユー
ることができます。

Webコンソールを有効に利用するためには、スレッドダンプ、プロファ

/resin-adminの設定

/resin-admin を使用にするためには管理者ユーザとパスワードを作成す

1. コマンドラインから「generate-password」を実行します。この
   Linuxの例

   ```
   > resinctl generate-password admin my-password
   admin_user : admin
   admin_password : {SSHA}HTfP60Ceq0K0lAvN31wQtBxtql9D
   ```

   Windowsの例

   ```
   > resin.exe generate-password admin my-password
   admin_user : admin
   admin_password : {SSHA}HTfP60Ceq0K0lAvN31wQtBxtql9D
   ```

2. ユーザとパスワードをresin.propertiesファイルに記述します。

   ```
   # Access to /resin-admin and remote CLI is password restrict
   # Use "resinctl generate-password" and copy/paste here to s
   admin_user : admin
   admin_password : {SSHA}HTfP60Ceq0K0lAvN31wQtBxtql9D
   ```

3. ローカル以外のIPアドレスからアクセスする場合はresin.propert

   ```
   # Permit access to /resin-admin from non-local network ip-ac
   web_admin_external : true
   ```

4. Resinインスタンスを再起動し、ブラウザから /resin-admin に
   http://localhost:8080/resin-admin/

> **コラム**
>
> CPUグラフ、Networkグラフはそれぞれ Linuxの /proc/stats
> Windowsでは使用できません。

診断

設定

ヘルス設定

項目
- 概要
- health.xml
- デフォルトの設定
  - ヘルスシステム
  - ヘルスチェック
  - ヘルスアクションとヘルスコンディション

概要

ヘルス機能の設定はhealth.xmlに記述します。設定には以下のものが含

- **ヘルスシステム** - ヘルスシステム本体の設定

- **ヘルスチェック** - システムのチェックを行う設定
- **ヘルスアクション** - タスクを実行する設定（サーバ再起動、ヒープ
- **ヘルスコンディション** - ヘルスアクションの実行条件を設定

これらの設定を組み合わせて、システムに異常が発生した場合にどのよ

health.xml

health.xmlは以下のパスに配置されています。

- Linuxのパス
  - デフォルトのインストールパスの場合

    ```
    /etc/resin/health.xml
    ```

  - インストールパスを指定した場合

    ```
    %RESIN_HOME%/conf/health.xml
    ```

- Windowsのパス

  ```
  %RESIN_HOME%/conf/health.xml
  ```

デフォルトの設定

ヘルス設定に関する各種設定項目には、デフォルトの値が設定されてい
以下に各設定を解説します。

ヘルスシステム

ヘルスシステム本体の設定はデフォルトで以下のように記述されていま

```
<health:HealthSystem>
 <enabled>true</enabled>
 <startup-delay>15m</startup-delay>
 <period>5m</period>
 <recheck-period>30s</recheck-period>
 <recheck-max>5</recheck-max>
 <check-timeout>5m</check-timeout>
</health:HealthSystem>
```

- **enabled** - ヘルスシステムを有効に設定
- **startup-delay** - Resin起動開始からヘルスシステムが監視を始め
- **period** - ヘルスチェックの監視間隔を5分に設定
- **recheck-period** - 異常を検知した場合に、再確認を行う監視間隔
- **recheck-max** - 再確認を行う回数を5回に設定
- **check-timeout** - ヘルスチェックのタイムアウトを5分に設定

ヘルスチェック

ヘルスチェックはデフォルトで以下のように設定されています。

- **<health:ConnectionPoolHealthCheck>** - データベースコネ
- **<health:CpuHealthCheck>** - CPU使用率が異常に上昇してい
- **<health:HealthSystemHealthCheck>** - ヘルスシステムが正
- **<health:HeartbeatHealthCheck>** - クラスタのハートビート
- **<health:JvmDeadlockHealthCheck>** - デッドロックが発生
- **<health:MemoryPermGenHealthCheck>** - JavaVMのPerr
- **<health:MemoryTenuredHealthCheck>** - JavaVMのTenu
- **<health:TransactionHealthCheck>** - コミットの失敗はない

> **コラム**
>
> <health:CpuHealthCheck> のデフォルト設定はヘルスステ
> 環境に応じて必要な場合に *critical-threshold* 属性を設定して

ヘルスアクションとヘルスコンディション

ヘルスアクションとヘルスコンディションの設定内容はデフォルトで以

```
<health:ActionSequence>
 <health:IfHealthCritical time="2m"/>

 <health:FailSafeRestart timeout="10m"/>
 <health:DumpJmx/>
 <health:DumpThreads/>
 <health:ScoreboardReport/>
 <health:DumpHeap/>
 <health:DumpHeap hprof="true"
                  hprof-path="${resin.logDirectory}/heap.hprof"/>
 <health:StartProfiler active-time="2m" wait="true"/>
 <health:Restart/>
</health:ActionSequence>
```

この設定はヘルスシステムがResinの異常を検知した場合、様々な情報を
以下にその解説を記述します。

1. もしヘルスチェックのどれかひとつでも ヘルスステータス「CRI
2. 情報収集に遅延が発生した場合のために、強制再起動を行うまで
3. JMXダンプを生成
4. スレッドダンプを生成
5. スコアボードレポートを生成
6. ヒープダンプを生成
7. hprofフォーマットのヒープダンプを生成
8. プロファイラを開始し、2分間分のサーバ情報を生成
9. Resinインスタンスの再起動を実行

> **コラム**
>
> <health:IfHealthCritical time=”2m”/>は、ヘルスシステム
> まず5分間隔で監視を行い、ヘルスステータス「CRITICAL」を
> もし、2分間「CRITICAL」のままであった場合Resinの再起動

> **コラム**
>
> プロファイルはResinの制限によりPDFレポートに出力されま
> PDFレポートの「CPU Profile」には「A CPU profile was no

また、デフォルトで以下のヘルスアクションが設定されています。

```
<health:Restart>
 <health:IfHealthFatal/>
</health:Restart>
```

これは、ヘルスチェックのどれかひとつでもヘルスステータス「FATAL

デフォルトの設定では下記のヘルスチェックが「FATAL」を返します。

- <health:HealthSystemHealthCheck> - ヘルスシステム自体が異
- <health:JvmDeadlockHealthCheck> - JavaVMによってデッド

リファレンス

ヘルスシステム

項目
- <health:HealthSystem>
  - 属性

<health:HealthSystem>

ヘルスシステムのチェックや再チェックの頻度を設定します。この要素
ます。

属性

属性名	説明
enabled	ヘルスシステムの有効/無効。
startup-delay	サーバ起動時にチェックの開始を遅延させる時間
period	チェックの間隔。
recheck-period	再チェックの間隔。

属性名	説明
recheck-max	通常のチェック間隔に戻るまでに行う再チェックの回数

コラム

ヘルスチェックの監視間隔を極端に短く設定にすると、環境に

ヘルスチェック

項目
- ヘルスステータス
- システム チェック
  - <health:ConnectionPoolHealthCheck>
  - <health:CpuHealthCheck>
  - <health:HealthSystemHealthCheck>
  - <health:HeartbeatHealthCheck>
  - <health:JvmDeadlockHealthCheck>
  - <health:LicenseHealthCheck>
  - <health:MemoryPermGenHealthCheck>
  - <health:MemoryTenuredHealthCheck>
  - <health:TransactionHealthCheck>
- ユーザ チェック
  - <health:HttpStatusHealthCheck>
  - <health:ExprHealthCheck>

ヘルスステータス

ヘルスチェックは実行されるたびにヘルスステータスとメッセージを生
下記はヘルスステータスと一般的な意味のリストです。

名前	順序	説明
UNKNOWN	0	ヘルスチェックはまだ実行されていない、ま です。
OK	1	ヘルスチェックは正常な状態であることを報
WARNING	2	ヘルスチェックは警告の域に達しているか、
CRITICAL	3	ヘルスチェックは重要の域に達していること
FATAL	4	ヘルスチェックは致命的な状態であることを

システム チェック

<health:ConnectionPoolHealthCheck>

Resinのデータベースコネクションプールの状態を監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
WARNING	*resin-web設定* の **<max-connections>** を超過
CRITICAL	*resin-web設定* の **<max-overflow-connectio**

設定例

```
<health:ConnectionPoolHealthCheck/>
```

<health:CpuHealthCheck>

CPU使用率を監視します。マルチコアマシンでは、各CPUが個別にチェ

属性

属性名	説明
enabled	チェックの有無
warning-threshold	ヘルスステータス「WARNING」のしきい値
critical-threshold	ヘルスステータス「CRITICAL」のしきい値

状態

ヘルスステータス	状態
WARNING	CPUのどれかが *warning-threshold* を超過した場合。
CRITICAL	CPUのどれかが *critical-threshold* を超過した場合。

設定例

```
<health:CpuHealthCheck>
 <warning-threshold>95</warning-threshold>
 <critical-threshold>99</critical-threshold>
</health:CpuHealthCheck>
```

<health:HealthSystemHealthCheck>

ヘルスシステム自体を監視します。
独立したスレッドを使用してヘルスチェックがフリーズしていないか、

属性

属性名	説明
enabled	チェックの有無
thread-check-period	独立したスレッドの確認頻度。
freeze-timeout	ヘルスシステムがフリーズしてヘルスチ 断するまでの最大時間。

状態

ヘルスステータス	状態
FATAL	ヘルスチェックが *freeze-timeout* の
FATAL	ヘルスチェックが許容される期間内に 出されます。

設定例

```
<health:HealthSystemHealthCheck>
 <thread-check-period>1m</thread-check-period>
 <freeze-timeout>15m</freeze-timeout>
</health:HealthSystemHealthCheck>
```

<health:HeartbeatHealthCheck>

クラスタの他のメンバーからのハートビートを監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
WARNING	クラスタの既知のメンバーからのハートビート
WARNING	クラスタの既知のメンバーからのハートビート 合。

設定例

```
<health:HeartbeatHealthCheck/>
```

<health:JvmDeadlockHealthCheck>

JavaVMによってデッドロック状態であると認識されたスレッドを監視し

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルステータス	状態
FATAL	デッドロックのスレッドが検出された場合。

設定例

```
<health:JvmDeadlockHealthCheck/>
```

<health:LicenseHealthCheck>

Resin-Proのライセンスの期限切れをチェックします。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true
warning-period	ライセンス警告の期間	*期間*	30日 (30D)

状態

ヘルステータス	状態
WARNING	ライセンスが *warning-period* の期間内に期限切れに

設定例

```
<health:LicenseHealthCheck>
 <warning-period>30D</warning-period>
</health:LicenseHealthCheck>
```

<health:MemoryPermGenHealthCheck>

JavaVMのPermGenメモリプールの空き領域を監視します。空き領域が

コラム

このヘルスチェックは、JavaVMベンダの実装によっては使用
JavaVMにPermGenメモリプールが存在しない場合はヘルスチ

属性

属性名	説明
enabled	チェックの有無
memory-free-min	クリティカルな状況の空きメモ
free-warning	警告のしきい値
objectName	明示的にメモリの統計を照会す なMBeanを検索します。

状態

ヘルスステータス	状態
UNKNOWN	JavaVMにPermGenメモリプールが存在しない 合。

ヘルスステータス	状態
WARNING	空きメモリが、ガベージコレクション実行後に
CRITICAL	空きメモリが、ガベージコレクション実行後に

設定例

```
<health:MemoryTenuredHealthCheck>
 <memory-free-min>1m</memory-free-min>
 <free-warning>0.01</free-warning>
</health:MemoryTenuredHealthCheck>
```

<health:MemoryTenuredHealthCheck>

JavaVMのTenuredメモリプールの空き領域を監視します。空き領域が

> コラム
>
> JavaVMにTenuredメモリプールが存在しない場合はJavaVM

属性

属性名	説明
enabled	チェックの有無
memory-free-min	クリティカルな状況の空きメモ
free-warning	警告のしきい値
objectName	明示的にメモリの統計を照会す なMBeanを検索します。

状態

ヘルスステータス	状態
UNKNOWN	JavaVMにTenuredメモリプールが存在しない 合。
WARNING	空きメモリが、ガベージコレクション実行後に
CRITICAL	空きメモリが、ガベージコレクション実行後に

設定例

```
<health:MemoryTenuredHealthCheck>
 <memory-free-min>1m</memory-free-min>
 <free-warning>0.01</free-warning>
</health:MemoryTenuredHealthCheck>
```

<health:TransactionHealthCheck>

Resinのトランザクションマネージャのコミット失敗を監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルステータス	状態
WARNING	前回のチェック以降にコミット失敗があった場合。

設定例

```
<health:TransactionHealthCheck/>
```

ユーザ チェック
<health:HttpStatusHealthCheck>

1つ以上のURLを作成し、現在のResinインスタンスにHTTP GETのリク

属性

属性名	説明
enabled	チェックの有無
ping-host	チェック対象のホスト（*url* が URI の場
ping-port	チェック対象のポート（*url* が URI の場
url	チェック対象のURLまたはURI
socket-timeout	ソケットコネクションのタイムアウト時
regexp	HTTPステータスコードの正規表現

状態

ヘルスステータス	状態
CRITICAL	HTTP GET リクエストが接続に失敗した場合、 合。

設定例

```
<health:HttpStatusHealthCheck>
 <ping-host>localhost</ping-host>
 <ping-port>8080</ping-port>
 <url>/custom-test-1.jsp</url>
 <url>/custom-test-2.jsp</url>
 <socket-timeout>2s</socket-timeout>
 <regexp>^2|^3</regexp>
</health:HttpStatusHealthCheck>
```

<health:ExprHealthCheck>

ユーザーが指定したEL式の真偽を評価します。

属性

属性名	説明
enabled	チェックの有無
fatal-test	trueと評価された場合にステータスをFATALするEL式。
critical-test	trueと評価された場合にステータスをCRITICALするEL式。
warning-test	trueと評価された場合にステータスをWARNINGするEL式。

状態

ヘルスステータス	状態
FATAL	*fatal-test* がtrueと評価された場合。
CRITICAL	*critical-test* がtrueと評価された場合。
WARNING	*warning-test* がtrueと評価された場合。

設定例

```
<health:ExprHealthCheck>
 <critical-test>${mbean('java.lang:type=Threading').ThreadCou
</health:ExprHealthCheck>
```

ヘルスアクション

項目

- <health:ActionSequence>
- <health:CallJmxOperation>
- <health:DumpHeap>
- <health:DumpHprofHeap>
- <health:DumpJmx>
- <health:DumpThreads>
- <health:ExecCommand>
- <health:FailSafeRestart>
- <health:PdfReport>
- <health:Restart>
- <health:ScoreboardReport>
- <health:SendMail>
- <health:Snapshot>
- <health:StartProfiler>

ヘルスアクションは特定の条件に応じてタスクを実行、またはヘルスチ
ヘルスシステムによって実行されます。
ヘルスアクションは通常、１つ以上のヘルスコンディションを伴います
される可能性があります。コンディションが無いヘルスアクションは、

> 注意
> ヘルスアクションはダンプ情報を出力するなど実行中にJava V
> ルスコンディションを設定してください。

<health:ActionSequence>

子に設定されているヘルスアクションを順に実行します。

属性

なし

設定例

```
<health:ActionSequence>
  <health:DumpThreads/>
  <health:DumpHeap/>
  <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

<health:CallJmxOperation>

パラメータを指定して、JMX MBeanのオペレーションを実行します。

属性

属性名	説明
objectName	JMX MBeanの名前。
operation	メソッドの名前。
operationIndex	複数のメソッドが一致する場合のユニークなインデ ス。
param	適切な型に変換されるメソッドのパラメータ。

設定例

```
<health:CallJmxOperation>
  <objectName>java.lang:type=Threading</objectName>
  <operation>resetPeakThreadCount</operation>
  <health:IfNotRecent time='5m'/>
</health:CallJmxOperation>
```

<health:DumpHeap>

メモリヒープダンプを生成します。

ヒープダンプは、Resinの内部ログデータベースとログファイルに記録さ

属性

属性名	説明
hprof	ユーザが読めるタイプのダンプではなく、HPROF

属性名	説明
hprof-path	HPROFファイルの出力パス。（*hprof* がtrueの場
hprof-path-format	タイムスタンプのトークンを使用して、動的に出 す。
log	ヒープダンプをResinの内部データベースに格納 ます。

設定例

```
<health:DumpHeap>
 <hprof>true</hprof>
 <hprof-path-format>${resin.home}/log/dump-%H:%M:%S.%s.h
 <health:OnAbnormalStop/>
</health:DumpHeap>
```

<health:DumpHprofHeap>

以下の設定と同様の意味をもつヘルスアクションです。

```
<health:DumpHeap>
 <hprof>true</hprof>
</health:DumpHeap>
```

<health:DumpJmx>

すべてのJMX属性および値のダンプを生成するヘルスアクションです。

JMXダンプは、Resinの内部ログデータベースとログファイルに記録され

属性

属性名	説明
log	JMXダンプをResinの内部データベースに格納される に書き込みます。

設定例

```
<health:DumpJmx/>
```

<health:DumpThreads>

スレッドダンプを生成します。

スレッドダンプは、Resinの内部ログデータベースとログファイルに記録

属性

属性名	説明
only-active	現在アクティブなスレッドのみ出力します。（**RU**
log	スレッドダンプをResinの内部データベースに格納 書き込みます。

設定例

```
<health:DumpThreads>
 <only-active>false</only-active>
</health:DumpThreads>
```

<health:ExecCommand>

オペレーティングシステムのシェルコマンドを実行します。

属性

属性名	説明
command	実行するコマンド。*dir* が設定されている場
dir	実行するディレクトリのパス。
timeout	コマンド実行のタイムアウト。完了してな
env	コマンドに使用する環境変数。

設定例

```
<health:ExecCommand>
 <dir>/tmp</dir>
 <command>remediation.sh</command>
 <timeout>2s</timeout>
 <env>
  <name>resin_home</name>
  <value>${resin.home}</value>
 </env>
 <env>
  <name>password</name>
  <value>foo</value>
 </env>
</health:ExecCommand>
```

<health:FailSafeRestart>

Resinを設定した時間に再起動します。通常、シャットダウン情報を集め

属性

属性名	説明	型	デフォルト
timeout	再起動を強制実行するまでの時間。	*期間*	なし

設定例

```
<health:ActionSequence>
 <health:FailSafeRestart timeout="10m"/>
 <health:DumpThreads/>
 <health:DumpHeap/>
 <health:StartProfiler active-time="5m"/>
 <health:Restart/>

 <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

<health:PdfReport>

PHPスクリプトからPDFレポートを生成するヘルスアクションです。

属性

属性名	説明	型
path	PDFを生成する.phpファイルのパス。	String
report	レポートタイプのキー。	String
period	レポートに出力する過去期間。	*期間*
log-directory	PDFを出力するディレクトリ。	String

設定例

```
<health:PdfReport>
 <path>${resin.home}/doc/admin/pdf-gen.php</path>
 <report>Summary</report>
 <period>7D</report>
 <health:IfCron value="0 0 * * 0"/>
</health:PdfReport>
```

<health:Restart>

Resinを再起動します。

属性

なし

設定例

```
<health:Restart/>
```

<health:ScoreboardReport>

関連したスレッドのグループについて簡潔なスレッドアクティビティレ

属性

属性名	説明
log	PDFレポートに加えて、サーバ·ログに書き込みます。
type	スコアボードレポートのタイプ。
greedy	falseの場合、スレッドが複数のグループに分類されます。

設定例

```
<health:ScoreboardReport>
  <health:OnAbnormalStop/>
</health:ScoreboardReport>
```

<health:SendMail>

現在のResinのヘルスステータスの概要を記載したメールを送信します。

属性

属性名	説明	型	デフォルト
to	メールの送信先アドレス	String	なし
from	メールの送信元アドレス	String	resin@localhost

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <to>another_admin@yourdomain.com</to>
  <from>resin@yourdomain.com</from>
</health:SendMail>
```

<health:Snapshot>

以下に示す一連のアクションを実行します。

- スレッドダンプ
- ヒープダンプ
- JMXダンプ
- PDFレポート

これらはサーバの問題をデバッグするために必要な全ての情報を想定し
このアクションは通常、予期しない問題に対して実行されることを想定

属性

属性名	説明
log	PDFレポートに加えて、サーバ·ログに書き込みます。
path	PDFを生成する.phpファイルのパス。
report	レポートタイプのキー。
period	レポートに出力する過去期間。

設定例

```
<health:Snapshot>
  <health:OnAbnormalStop/>
</health:Snapshot>
```

<health:StartProfiler>

プロファイラ開始します。結果は、Resinの内部ログデータベースとロ

属性

属性名	説明	型	デフォルト
active-time	プロファイラを実行する時間	*期間*	5秒(5

属性名	説明	型	デフォルト
sampling-rate	サンプリングレート	*期間*	10ミリ秒 (10m
depth	スタックトレースの深さ。	int	16

設定例

```
<health:ActionSequence>
 <health:FailSafeRestart timeout="10m"/>
 <health:DumpThreads/>
 <health:DumpHeap/>
 <health:StartProfiler active-time="5m"/>
 <health:Restart/>

 <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

ヘルスコンディション

項目

- 基本条件
  - <health:IfCron>
  - <health:IfExpr>
  - <health:IfNotRecent>
  - <health:IfRechecked>
  - <health:IfUptime>
- 組み合わせ条件
  - <health:And>
  - <health:Nand>
  - <health:Nor>
  - <health:Not>
  - <health:Or>
- ヘルスチェック条件
  - <health:IfHealthOk>
  - <health:IfHealthWarning>
  - <health:IfHealthCritical>
  - <health:IfHealthFatal>
  - <health:IfHealthUnknown>
  - <health:IfMessage>
  - <health:IfRecovered>
- ライフサイクル条件
  - <health:OnStart>
  - <health:OnStop>
  - <health:OnAbnormalStop>
  - <health:OnRestart>

コンディション（条件）またはプレディケート（叙述）は、一連の基準
コンディションは期間ごとに評価されます。すべての条件が真と評価さ
コンディションが無いアクションは期間毎に一度実行されます。複数の

基本条件

基本条件は一般的な基準を評価し条件がマッチした場合にtrueを返しま
基本条件はヘルスチェックの状態を評価しません。代わりに、その日の

<health:IfCron>

現在時刻が **cronスタイル** で設定された範囲だった場合にマッチします
これは、定期的なアクションをスケジュールしたり、重要な時間帯に再

属性

属性名	説明	型	デフォルト
enable-at	有効とするcron時間。	cronスタイル	なし

属性名	説明	型	デフォルト
disable-at	無効とするcron時間。	cronスタイル	なし

設定例

```
<health:Restart>
 <health:IfCron>
  <enable-at>0 0 * * *</enable-at>
  <disable-at>5 0 * * *</disable-at>
 </health:IfCron>
</health:Restart>
```

<health:IfExpr>

JSP EL式の評価に基づいてマッチします。式は、システムプロパティ、

属性

属性名	説明	型	デフォルト
test	JSP EL式。	EL式	なし

設定例

```
<health:Restart>
 <health:IfExpr>
  <test>${mbean('java.lang:type=Threading').ThreadCount > 10
 </health:IfExpr>
</health:Restart>
```

<health:IfNotRecent>

前回の実行後からの時間の長さに応じてマッチします。これは頻繁な実

属性

属性名	説明	型	デフォルト
time	再びアクションを実行できるまでの時間。	*期間*	なし

設定例

```
<health:HttpStatusHealthCheck ee:Named="httpStatusCheck"
 <url>http://localhost:8080/test-ping.jsp</url>
</health:HttpStatusHealthCheck>

<health:DumpHeap>
 <health:IfHealthCritical healthCheck="${httpStatusCheck}"/>
 <health:IfNotRecent time='5m'/>
</health:DumpHeap>
```

<health:IfRechecked>

再チェックの必要回数が実行されている場合にマッチします。
再チェックはヘルスチェック固有の条件ではないので、HealthSystem

属性

なし

設定例

```
<health:Restart>
 <health:IfHealthFatal/>
 <health:IfRechecked/>
</health:Restart>
```

<health:IfUptime>

起動してからの時間の長さに応じてマッチします。

属性

属性名	説明	型	デフォルト
limit	起動後の時間。	*期間*	なし

設定例

```
<health:Restart>
 <health:IfUptime limit="12h"/>
</health:Restart>
```

組み合わせ条件

基本条件やヘルスチェック条件を組み合わせたり、これらの条件を使用

<health:And>

全ての子のプレディケートがマッチしている場合にマッチします。

属性

なし

設定例

```
<health:Restart>
 <health:And>
  <health:IfHealthCritical health-check="${memoryTenuredHe
  <health:IfHealthCritical health-check="${memoryPermGenH
 </health:And>
</health:Restart>
```

<health:Nand>

全ての子のプレディケートが失敗している場合にマッチします。

属性

なし

設定例

```
<health:Restart>
 <health:Nand>
  <health:IfHealthCritical health-check="${memoryTenuredHe
  <health:IfHealthCritical health-check="${memoryPermGenH
 </health:Nand>
</health:Restart>
```

<health:Nor>

全ての子のプレディケートが一致しない場合にマッチします。

属性

なし

設定例

```
<health:Restart>
 <health:Nor>
  <health:IfHealthCritical health-check="${memoryTenuredHe
  <health:IfHealthCritical health-check="${memoryPermGenH
 </health:Nor>
</health:Restart>
```

<health:Not>

子のプレディケートがfalseの場合にマッチします。

属性

なし

設定例

```
<health:Restart>
 <health:IfHealthCritical health-check="${memoryTenuredHeal
 <health:Not>
  <health:IfCron>
   <enable-at>0 7 * * *</enable-at>
   <disable-at>0 11 * * *</disable-at>
  </health:IfCron>
 </health:Not>
</health:Restart>
```

<health:Or>

子のプレディケートのいずれかがマッチしている場合にマッチします。

属性

なし

設定例

```
<health:Restart>
 <health:Or>
  <health:IfHealthCritical health-check="${memoryTenuredHe
  <health:IfHealthCritical health-check="${memoryPermGenH
 </health:Or>
</health:Restart>
```

ヘルスチェック条件

ヘルスチェック条件は、ヘルスチェックの結果をあらゆる側面から評価
パラメータ「health-check」で特定の名前が付けられたヘルスチェック

<health:IfHealthOk>

ヘルスステータスが「OK」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。

time	開始状態からの最少時間。

設定例

```
<health:Restart>
 <health:Not>
  <health:IfHealthOk health-check="${memoryTenuredHealthC
 </health:Not>
</health:Restart>
```

<health:IfHealthWarning>

ヘルスステータスが「WARNING」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。

time	開始状態からの最少時間。

設定例

```
<health:Restart>
 <health:IfHealthWarning health-check="${memoryTenuredHe
</health:Restart>
```

<health:IfHealthCritical>

ヘルスステータスが「CRITICAL」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
 <health:IfHealthCritical health-check="${memoryTenuredHeal
</health:Restart>
```

<health:IfHealthFatal>

ヘルスステータスが「FATAL」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
 <health:IfHealthFatal health-check="${memoryTenuredHealth
</health:Restart>
```

<health:IfHealthUnknown>

ヘルスステータスが「UNKNOWN」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
 <health:IfHealthUnknown health-check="${memoryTenuredH
</health:Restart>
```

<health:IfMessage>

ヘルスチェック結果のメッセージが正規表現に一致する場合にマッチし

属性

属性名	説明

属性名	説明
health-check	参照するヘルスチェック。

time	開始状態からの最少時間。
regexp	ヘルスメッセージにマッチする正規表現。

設定例

```
<health:Restart>
  <health:IfHealthCritical/>
  <health:IfMessage health-check="${httpStatusCheck}" regexp
</health:Restart>
```

<health:IfRecovered>

リカバリされた際にマッチします。リカバリとは「FATAL」、「CRITIC

属性

属性名	説明
health-check	参照するヘルスチェック。

設定例

```
<health:SendMail>
  <to>admin@yourdomain</to>
  <health:IfRecovered health-check="${cpuHealthCheck}"/>
</health:SendMail>
```

ライフサイクル条件

ライフサイクル条件はResinの状態変化を評価します。Resinのライフサ

<health:OnStart>

Resinが開始しようとしている場合にマッチします。

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnStart/>
</health:SendMail>
```

<health:OnStop>

Resinが停止された場合にマッチします。

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnStop/>
</health:SendMail>
```

<health:OnAbnormalStop>

Resinが正常ではない終了コードを返して停止している場合にマッチしま

属性

なし

設定例

```
<health:PdfReport snapshot='true'>
  <health:OnAbnormalStop/>
</health:PdfReport>
```

<health:OnRestart>

Resinがwatchdogによって再起動された場合にマッチします。これは−

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnRestart/>
</health:SendMail>
```

設定値

期間

単位	意味
ms	ミリ秒
s	秒
m	分
h	時
D	日
W	週
M	月
Y	年

メーター

項目
- 設定
  - health.xml
  - ヘルスメーター名
- JMXメーター
  - <health:JmxMeter>
  - <health:JmxDeltaMeter>
- 統計分析
  - <health:AnomalyAnalyzer>
  - <health:IfHealthEvent>

ヘルスメーターはResinの管理コンソール(/resin-admin)に視覚的にわ

設定

health.xml

health.xmlにはあらかじめJMXメーターが設定されています。health.x

ヘルスメーター名

ヘルスメーター名はキーをパイプ文字(|)で連結して名前が付けられてい
ヘルスメーターの統計はResinクラスタの各メンバー間で共有されてい
す。

名前に含まれるパイプ文字は、ヘルスメーターをドリルダウンに分類し

```
<health:JmxDeltaMeter>
  <name>JVM|Compilation|Compilation Time</name>
  <object-name>java.lang:type=Compilation</object-name>
  <attribute>TotalCompilationTime</attribute>
</health:JmxDeltaMeter>
```

この例では **JVM|Compilation|Compilation Time** をベースにヘル
管理コンソールではクラスタメンバーのインデックスと最初の2つのキー

```
00|JVM|Compilation|Compilation Time
```

JMXメーター

JMX MBean属性の数値をヘルスメーターにグラフ表示することができま

<health:JmxMeter>

JMX MBeanの数値属性の現在値をグラフに表示します。

属性

属性名	説明	型
name	管理コンソールに表示するヘルスメーターの名前	String
objectName	JMX MBean の名前	String
attribute	サンプリングするMBeanの属性	String

設定例

```
<health:JmxMeter>
 <name>OS|Memory|Physical Memory Free</name>
 <object-name>java.lang:type=OperatingSystem</object-name
 <attribute>FreePhysicalMemorySize</attribute>
</health:JmxMeter>
```

<health:JmxDeltaMeter>

JMX MBeanの数値属性の現在値と以前の値の差をグラフに表示します。

属性

属性名	説明	型
name	管理コンソールに表示するヘルスメーターの名前	String
objectName	JMX MBean の名前	String
attribute	サンプリングするMBeanの属性	String

設定例

```
<health:JmxDeltaMeter>
 <name>JVM|Compilation|Compilation Time</name>
 <object-name>java.lang:type=Compilation</object-name>
 <attribute>TotalCompilationTime</attribute>
</health:JmxDeltaMeter>
```

統計分析

<health:AnomalyAnalyzer>

AnomalyAnalyzerは平均値からの偏差をチェックし現在の値を監視しま
突然スレッドがブロックされた場合など異常な変化を検出することがで

属性

属性名	説明
meter	分析するヘルスメーターの名前（<he前）
health-event	<health:IfHealthEvent>の条件に一
min-samples	平均値を計算するために必要なサンプ
sigma-threshold	標準偏差から異常とみなされるサンプ

設定例

```
<health:JmxMeter>
 <name>JVM|Thread|JVM Blocked Count</name>
 <objectName>resin:type=JvmThreads</objectName>
 <attribute>BlockedCount</attribute>
</health:JmxMeter>

<health:AnomalyAnalyzer>
 <meter>JVM|Thread|JVM Blocked Count</meter>
 <health-event>caucho.thread.anomaly.jvm-blocked</health-ev
</health:AnomalyAnalyzer>

<health:DumpThreads>
 <health:IfHealthEvent regexp="caucho.thread"/>
 <health:IfNotRecent time="15m"/>
</health:DumpThreads>
```

<health:IfHealthEvent>

一致するヘルスイベントに応じてアクションを起こします。これは、通

属性

属性名	説明	型
regexp	*<health-event>* が一致する正規表現。	java.util.r

設定例

```
<health:JmxMeter>
 <name>JVM|Thread|JVM Blocked Count</name>
 <objectName>resin:type=JvmThreads</objectName>
 <attribute>BlockedCount</attribute>
</health:JmxMeter>

<health:AnomalyAnalyzer>
 <meter>JVM|Thread|JVM Blocked Count</meter>
 <health-event>caucho.thread.anomaly.jvm-blocked</health-ev
</health:AnomalyAnalyzer>

<health:DumpThreads>
 <health:IfHealthEvent regexp="caucho.thread"/>
 <health:IfNotRecent time="15m"/>
</health:DumpThreads>
```

レポート

項目

- レポートの取得
  - 自動的なPDFレポートの生成
- レポートの概要
- ヒープダンプ
  - クラスローダーのヒープダンプ
- スレッドダンプ
- CPUプロファイル
- ロギング
- JMXダンプ

Resin にはシステムの状態をPDFレポートに出力する機能が含まれてい

レポートの取得

レポートを取得する簡単な方法は、コマンド「pdf-report」を使用する

Linuxの例

```
> resinctl pdf-report
```

Windowsの例

```
> resin.exe pdf-report
```

自動的なPDFレポートの生成

PDFレポートは毎週生成されるようにhealth.xmlに設定することができ

```
<health:PdfReport>
 <path>${resin.root}/doc/admin/pdf-gen.php</path>
 <report>Summary</report>
 <period>7D</period>
 <snapshot/>
 <mail-to>${email}</mail-to>
 <mail-from>${email_from}</mail-from>
 <!-- <profile-time>60s</profile-time> -->

 <health:IfCron value="0 0 * * 0"/>
</health:PdfReport>
```

上の例では、スナップショットを作成して、PDFレポートを生成し指定

送信先のメールアドレスはresin.propertiesに記述します。

```
# Set the email address to receive weekly and restart PDF reports
email : admin@example.com
```

次の例では、watchdogよる再起動時にPDFファイルを生成します。

```
<health:PdfReport>
 <path>${resin.root}/doc/admin/pdf-gen.php</path>
 <title>Restart</title>
 <watchdog/>
 <mailto>user@example.com</mailto>

 <health:OnRestart/>
</health:PdfReport>
```

レポートの概要

以下の内容がレポートに出力されます。

- **サーバ環境の概要** - JavaVM、OS、Resinのバージョンなど
- **ヘルスメーターのグラフ** - レポートが出力されるまでの統計グラフ
- **ヒープダンプ** - ヒープメモリ使用量の上位
- **CPUプロファイル** - CPUプロファイルの最も活動的な位置
- **スレッドダンプ** - スナップショット取得時のResin内の全てのスレ
- **ログレポート** - java.util.loggingから出力された最新の「warning
- **JMXダンプ** - スナップショット取得時の全てのJMX Mbeanデータ

レポートはいくつかの状況で使用するように設計されています。

- **Resinのバグレポート** - Resin自体のサポートのために、スナップ
- **再起動やクラッシュの分析** - JavaVMが何らかの理由でクラッシュし
- **CPUとパフォーマンス** - JavaVMが予想外にCPUを使用した場合、
- **スレッドのロック** - スレッドダンプおよびCPUは、ロックされたコ

ヒープダンプ

ヒープダンプは、システムのメモリ割り当ての概要を示します。異常な

ヒープダンプはオブジェクト自身のサイズとその子孫のサイズで「self+

以下の例は、一般的なアイドル状態のResinのヒープダンプです。

```
Class Name                        self+desc  self  count

byte[]                          22.23M  22.23M   14741
com.caucho.db.block.Block                20.28M  134.9K   2410
char[]                          13.89M  13.89M  122606
com.caucho.util.LruCache                  7.52M   30.4K    317
java.lang.String                   7.15M   1.97M   61426
..
```

最初にあるbyte[]とBlockは、主にResinの内部プロキシーキャッシュと
「self+desc」が、大きなbyte[]バッファを持っているためです。

同様に、char[]バッファを含んでいるため、java.lang.Stringの「self+

クラスローダーのヒープダンプ

ヒープダンプには、クラスローダのメモリ使用量を個別に表示する項目
クラスに基づいたメモリーリークをチェックするためにこのレポートを

スレッドダンプ

CPUに関する問題やスタックスレッドの場合、スレッドダンプを確認す
いくつものスレッドが異常な個所で連なりロック状態になっているかを

スレッドダンプのレポートは、同様のスタックトレースを有しているス

ブロックされたスレッドとロックを保持しているスレッドがグループ化

以下の例は正常なブロック状態を示しています。JDKのSSL実装は1つの
この場合「**http://*:8444-17**」という名前のスレッドがSocksSocket
「**http://*:8444-10**」,「**http://*:8444-11**」）

```
http://*:8444-17
   java.net.PlainSocketImpl.socketAccept
   -- locked java.net.SocksSocketImpl@1199747469
   java.net.PlainSocketImpl.accept
   java.net.ServerSocket.implAccept
   ...
   com.caucho.env.thread.ResinThread.runTasks
   com.caucho.env.thread.ResinThread.run

http://*:8444-1
  waiting on java.net.SocksSocketImpl@4782b18d owned by [126] h
http://*:8444-10
  waiting on java.net.SocksSocketImpl@4782b18d owned by [126] h
http://*:8444-11
  waiting on java.net.SocksSocketImpl@4782b18d owned by [126] h
   java.net.PlainSocketImpl.accept
   java.net.ServerSocket.implAccept
   com.sun.net.ssl.internal.ssl.SSLServerSocketImpl.accept
   ...
   com.caucho.env.thread.ResinThread.runTasks
   com.caucho.env.thread.ResinThread.run

...
```

CPUプロファイル

**Resinの制限により、CPUプロファイルをPDFレポートに出力すること
PDFレポートの「CPU Profile」には「A CPU profile was not ge**

ロギング

最新の警告ログが出力されます。

```
Log(Warning)

2011-09-21 11:06:07 warning  WarningService: Resin restarting du
                             configuration change
```

JMXダンプ

JMXダンプはシステム内のすべてのJMX MBeanとその値を出力します。

```
JMX Dump

JMImplementation:type=MBeanServerDelegate
  ImplementationName     Resin-JMX
  ImplementationVendor   Caucho Technology
  ImplementationVersion  Resin-4.0.s110921
  MBeanServerId          Resin-JMX
  SpecificationName      Java Management Extensions
  SpecificationVendor    Sun Microsystems
  SpecificationVersion   1.4

com.sun.management:type=HotSpotDiagnostic
  ...
```

watchdog

項目
- 概要
- コマンドライン
  - console
  - start
  - stop
  - status

信頼性とセキュリティを高めるために、独立したResin watchdogプロ
なっている場合、Resinのインスタンスを再起動します。

概要

watchdogは独立したサービスとして密かに実行されるため、ほとんど
ン上の全てのResin JavaVMを監視します。

ユーザがwatchdogに注意を払う必要がある場合は、環境に深刻な問題

Resinがクラッシュしたり予期せずプロセスが終了した場合、watchdo
指示する必要があります。
コマンドラインから起動、停止、再起動を行ってください。

コマンドライン

console

「console」コマンドは開発時のために使用します。コンソールウィン

Linuxの例

```
> resinctl console -server app-0
```

Windowsの例

```
> resin.exe console -server app-0
```

start

「start」コマンドは指定されたIDの新しいResinインスタンスを起動し

Linuxの例

```
> resinctl start -server app-0
Resin/4.0.xx started -server 'app-0' with watchdog at 127.0.0.1:660
```

Windowsの例

```
> resin.exe start -server app-0
Resin/4.0.xx started -server 'app-0' with watchdog at 127.0.0.1:660
```

stop

「stop」コマンドは指定されたIDのResinインスタンスを停止します。

Linuxの例

```
> resinctl stop -server app-0
Resin/4.0.xx stopped for watchdog at 127.0.0.1:6600
```

Windowsの例

```
> resin.exe stop -server app-0
Resin/4.0.xx stopped for watchdog at 127.0.0.1:6600
```

status

「status」コマンドはwatchdogサービスによって管理されている現在

Linuxの例

```
> resinctl status
Resin/4.0.xx status for watchdog at 127.0.0.1:6600

watchdog:
  watchdog-pid: 25088

server 'app-0' : ACTIVE
  password: missing
  watchdog-user: imart
  user: imart
  root: /home/imart/resin-pro-4.0.xx
  conf: /home/imart/resin-pro-4.0.xx/conf/resin.xml
..
```

Windowsの例

```
> resin.exe status
Resin/4.0.xx status for watchdog at 127.0.0.1:6600

watchdog:
  watchdog-pid: 192

server 'app-0' : ACTIVE
  password: missing
  watchdog-user: imart
  user: imart
  root: /C:/resin-pro-4.0.xx
  conf: /C:/resin-pro-4.0.xx/conf/resin.xml
..
```

ログ設定

項目

- java.util.logging
  - 概要
  - ログ名
  - ログレベル
  - <log-handler>
  - ログハンドラ タイムスタンプ
  - ログハンドラ アーカイブ
  - ログハンドラ ELフォーマット
  - Logger: アプリケーションのロギング
  - カスタムログハンドラとライブラリログハンドラ
  - カスタムログフォーマット
  - Resinビルトイン ログハンドラ
    - BamLogHandler
    - EventLogHandler
    - JmsLogHandler
    - MailLogHandler
    - SyslogHandler
- ログローテーションとアーカイブ
  - サイズによるロールオーバー
  - 時間によるロールオーバー
  - アーカイブファイル
  - ロール オーバーのを無効
  - 圧縮
- 標準出力のリダイレクト
  - stdoutログ
    - 属性
  - stderrログ
    - 属性
- <access-log>
  - フォーマットパターン
  - 属性
- ログのパス

Resinは JDK logging インタフェース の出力先を指定し、Webアプリケ

java.util.logging

概要

Resinは すべての内部ログにJDK 標準の java.util.logging を使用し、ロ
ログ設定は、ログハンドラとロガーレベルの2つの部分があります。

ログハンドラはResinのログをどこに出力するかを指示します。Resinに
ハンドラが含まれています。

ファイルベースログハンドラの例

```
<log-handler name="com.foo" level="all"
             path="${resin.root}/log/foo.log"
             timestamp="[%y-%m-%d %H:%M:%S.%s] {%{thread}} "/>
```

<logger> は名前付きロガーのログレベルを設定します。<logger> は
なりません。

ロガーおよびログハンドラ名は階層的であるため、「com.foo」<logg

「fine」レベルのログ出力設定の例

```
<logger name="com.foo" level="fine"/>
<logger name="com.foo.bar" level="finest"/>
```

ログ名

JDK logging API は階層型の命名方式を使用しています。通常、名前はJ
されます。
例えば、 <logger name=”example.hogwarts” ...> は “example.h

Resin自身のロギングは、Resinのクラス名に基づいています。以下はR

名前	意味
“”	すべてをデバッグします。
com.caucho.ejb	EJBの処理。
com.caucho.jsp	JSPのデバッグ。
com.caucho.java	Java コンパイル。
com.caucho.server.port	TCPポートのデバッグおよびス
ド。
com.caucho.server.http	HTTP関連のデバッグ。
com.caucho.server.webapp	Webアプリケーション関連のデ
バッグ。
com.caucho.server.cache	キャッシュ関連のデバッグ。
com.caucho.sql	データベースプール。
com.caucho.transaction	トランザクションの処理。

ログレベル

ロガーレベルは、与えられたデバッグ粒度のログを有効にします。「se
向けのデバッグ情報を表示します。

ログレベルは JDK java.util.logging.Level に定義される値と一致してい

名前	API	用途
off		ログをオフにする。
severe	log.severe(”...”)	Webアプリケーショ
行を妨げる障害。
warning	log.warning(”...”)	ブラウザにレスポン
info	log.info(”...”)	Webアプリケーショ
config	log.config(”...”)	設定に関する詳細な
fine	log.fine(”...”)	ソースコードに精通し
finer	log.finer(”...”)	開発者が使用する詳
finest	log.finest(”...”)	詳細なトレースを行
all		全てのメッセージが

<log-handler>

JDK java.util.logging.* API のためのログハンドラを設定します。
java.util.logging には 2 つの手順があります。ログハンドラのセットの
グ名にハンドラをアタッチします。

カスタムハンドラを構成する <log-handler> には最も共通的な設定で
ます。

ログハンドラ タイムスタンプ

logタグのタイムスタンプは、日付と時刻の値に置換される「%」のコー

コード	意味
%a	曜日（省略）

コード	意味
%A	曜日（詳細）
%b	月（省略）
%B	月（詳細）
%c	Javaロケールの日付
%d	月（数字2桁）
%H	24時間の時（数字2桁）
%I	12時間の時（数字2桁）
%j	年の日数
%m	月（数字2桁）
%M	分
%p	午前/午後
%S	秒
%s	ミリ秒
%W	年の週数（数字3桁）
%w	週の日数（数字1桁）
%y	年（数字2桁）
%Y	年（数字4桁）
%Z	タイムゾーン（名称）
%z	タイムゾーン（+/-0800）
%{thread}	現在のスレッドの名前
%{level}	現在のログレベル
%{env}	現在のクラスローダ環境

典型的なlogタグのタイムスタンプの例

```
<log-handler name='' path='stderr:' timestamp="[%H:%M:%S.%s
```

```
[22:50:11.648] WebApp[/doc] starting
[22:50:11.698] http listening to *:8080
[22:50:11.828] hmux listening to *:6800
```

ログハンドラ アーカイブ

以下の例はロールオーバー ファイルに書き込む標準的なログハンドラで

```
<log-handler name="" level="all"
    timestamp="[%Y/%m/%d %H:%M:%S.%s] {%{thread}} "/>

<logger name="com.caucho" level="info"/>
```

デフォルトのアーカイブ形式は以下のようになっています。

- rollover-period が 1日（1D）以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日（1D）より小さい場合。

```
path + ".%Y%m%d.%H"
```

たとえば、標準エラー出力を使用するすべてのログに設定する例は以下

```
<log-handler name='' level='all' path='stderr:' timestamp="[%H:
```

役立つテクニックとして、問題を追跡するために全てのデバッグログの

```
<log-handler name='' level='finer' path='log/debug.log'
    timestamp="[%H:%M:%S.%s]"
    rollover-period='1h' rollover-count='1'/>
```

ログハンドラ ELフォーマット

Resinの <log-handler> のformat属性には、各ログメッセージのフォ

フォーマット文字列の例

```
<log-handler name='' level='all' path='stderr:' timestamp="[%H:
    format=" ${log.level} ${log.name} ${log.message}"/>
```

ログ EL 変数

変数	説明
${log.level}	ログレベル。
${log.name}	ロガー名。
${log.shortName}	ロガー名を短 はなく “Foo
${log.message}	ログメッセー
${log.millis}	イベントが発
${log.sourceClassName}	ロギングを要 い場合があり
${log.sourceMethodName}	ロギングを要 ない場合があ
${log.threadID}	ロギングの要 を取得。
${log.thrown}	ロギングの要 java.lang.Tl
${thread}	現在のスレッ
${request}	サーブレット
${session}	HTTPセッシ
${cookie[‘JSESSIONID’]}	リクエスト

また、フォーマット文字列で環境EL変数を使用することもできます。

```
<web-app>
  <log name='' level='all' path='log/debug.log' timestamp="[%H:%
      format=" [${app.contextPath}] ${log.message}"/>

  ...
</web-app>
```

```
[14:55:10.189] [/foo] `null' returning JNDI java:
      model for EnvironmentClassLoader[web-app:http://localhost:80
[14:55:10.189] [/foo] JNDI lookup `java:comp/env/caucho/auth'
      exception javax.naming.NameNotFoundException: java:comp/e
[14:55:10.199] [/foo] Application[http://localhost:8080/foo] starting
```

Logger: アプリケーションのロギング

アプリケーションでロギングを行うために、JDKのロギング機能を利用す
す。多すぎるロギングは少ないロギングと同じくらい開発者の混乱を招

ログ名は実装しているクラスの完全なクラス名にすべきです。別の名前

ログレベルは、アプリケーション全体で一貫しているべきです。Resinで

finerのロギングの例

```
import java.util.logging.Logger;
import java.util.logging.Level;

public class Foo {
 private static final Logger log
   = Logger.getLogger(Foo.class.getName());

 ...
 void doFoo(String bar)
 {
  // check for log level if your logging call does anything more
  // than pass parameters
  if (log.isLoggable(Level.FINER))
     log.finer(this + "doFoo(" + bar + ")");

   ...

   log.info(...);

   try {
     ...
   } catch (ExpectedException ex) {
    log.log(Level.FINEST, "expected exception", ex);
   }
 }
 ...
}
```

カスタムログハンドラとライブラリログハンドラ

カスタムハンドラとライブラリのログハンドラは、CanDI XMLの構文を

JDK FileHandler の例

```
<web-app xmlns="http://caucho.com/ns/resin"
  xmlns:jdk-logging="urn:java.util.logging">

 <log-handler name="com.foo" level="info">
  <jdk-logging:FileHandler>
   <new>
    <value>/tmp/test.out</value>
   </new>
  </jdk-logging:FileHandler>
 </logger>

</web-app>
```

```
package com.foo.demo;

import java.util.logging.*;

public class MyHandler extends Handler
{
 @Override
 public void publish(LogRecord record)
 {
  System.out.println(getFormatter().format(record));
 }

 @Override
 public void flush();
 {
 }

 @Override
 public void close();
 {
 }
}
```

カスタムログフォーマット

ログハンドラと同様に、ログメッセージの書式をカスタマイズできます。
ます。

サイトの情報をより適切に収集するために、ログメッセージの書式を変

カスタムフォーマットの設定例

```
<log-handler name="com.foo" level="warning" path="WEB-INF/lo
 <formatter><mypkg:MyFormatter/></formatter>
</log-handler>
```

MyFormatter.java

```
package com.mycom.mypkg;

import java.util.logging.*;

public class MyFormatter extends Formatter
{
 @Override
 public String format(LogRecord record)
 {
  return "[" + record.getLevel() + "] " + record.getMessage();
 }
}
```

Resinビルトイン ログハンドラ

ResinはJMS、HMTPおよび syslog サービスなどに一般的なログパター
また、カスタム ハンドラを作成することも簡単です。

BamLogHandler

BAMハンドラは、BAMエージェントにログメッセージを発行します。エ
ドレスとしてJID（Jabber id）が必要です。

BAM ハンドラ設定の例

```
<logger name="com.foo">
 <resin:BamLogHandler level="warning">
   <to>test@localhost</to>
 </resin:BamLogHandler>
</logger>
```

EventLogHandler

イベントハンドラは、CanDIイベントシステムへの LogEvent を発行し
す。ログハンドラのクラス名は com.caucho.log.EventLogHandler で

イベントハンドラ設定の例

```
<logger name="com.foo">
 <resin:EventLogHandler level="warning"/>
</logger>
```

JmsLogHandler

JMS ハンドラは JMS キューにログ メッセージを発行します。

JMS ハンドラ設定の例

```
<web-app xmlns="http://caucho.com/ns/resin"
   xmlns:ee="urn:java:ee"
   xmlns:resin="urn:java:com.caucho.resin">

 <resin:MemoryQueue ee:Named="myQueue"/>

 <logger name="com.foo">
  <resin:JmsLogHandler level="warning">
    <target>${myQueue}</target>
  </resin:JmsLogHandler>
 </logger>

</web-app>
```

MailLogHandler

メールハンドラは email アドレスにログメッセージを送信します。メー

MailLogHandler 属性

属性名	説明

属性名	説明
to	メールアドレス
delay-time	最初のメールを送信する前に待機する時間
mail-interval-min	メールメッセージの最小間隔
properties	JavaMailのプロパティ

メールハンドラ設定の例

```
<logger name="">
 <resin:MailLogHandler level="warning">
   <to>admin@foo.com</to>
   <properties>
    mail.smtp.host=127.0.0.1
    mail.smtp.port=25
   </properties>
 </resin:MailLogHandler>
</logger>
```

SyslogHandler

UNIXシステムでは、SyslogHandlerを使用してsyslogにメッセージを記

SyslogHandler 設定の例

```
<logger name="">
 <resin:SyslogLogHandler level="warning">
   <facility>daemon</facility>
   <severity>notice</severity>
 </resin:SyslogLogHandler>
</logger>
```

facility に使用可能な値は、user, mail, daemon、auth、lpr、news、daemon です。

severityには、emerg、alert、crit、err、warning、notice、info、d

「man 3 syslog」 と 「man syslog.conf」を参照してください。

ログローテーションとアーカイブ

ログローテーションは毎週または毎日ごとにログファイルをアーカイブ

ResinのログローテーションはJDK logging、HTTPアクセスログ、標準

サイズによるロールオーバー

ファイルサイズが一定量に達したときにロールオーバーが行われます。I

「rollover-size」は最大サイズを指定するために使用し、バイト（500(
ます。

時間によるロールオーバー

最後にロールオーバーされてから一定の期間が経過したときにロールオ
れていなければ、時間によるロールオーバーを実行しません。サイズに

「rollover-period」は期間を指定するために使用し、日（15D）、 週

アーカイブファイル

ロールオーバーが行われると、ログファイルの名前が変更され（アーカ

「archive-format」は、アーカイブファイルの名前を指定します。これ
に使用するものと同じです。

デフォルトの動作は「rollover-period」の値に依存します。「rollover
名はオリジナルのパスに「.%Y%m%d」が追加されたものです。
「rollover-period」が一日未満の場合、アーカイブファイル名はオリジ

ロール オーバーのを無効

ロールオーバーを無効にするには、「rollover-size」に起こりえないよ

```
<stdout-log path="log//stdout.log" rollover-size="1024mb"/>
```

圧縮

ログのロールオーバー ファイルは gzip や zip で圧縮することができま

```
<log name="" level="warning" path='log/error.log'
    archive-format="%Y-%m-%d.error.log.gz"
    rollover-period="1D"/>

<access-log path="log/access.log"
        archive-format="access-%Y%m%d.log.gz"
        rollover-period="1D"/>
```

標準出力のリダイレクト

stdoutログ

System.out の送信先を設定します。

stdoutログの設定は、親の設定より優先されます。例えば、<web-app
<host>のstdoutログの設定より優先されます。

> 注意
>
> 「path」はコマンドライン「-stdout」で指定されたパスと同

属性

属性名	説明
archive-format	ロールオーバーされたアーカイブファイル名の
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定しま
rollover-count	ロールオーバーファイルの最大数。
rollover-period	ロールオーバーを行う頻度。日（15D）、週（
rollover-size	ロールオーバーを行うファイルの最大サイズ。 （10mb）。
timestamp	行の先頭に使用する *タイムスタンプ* のフォーマ

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日（1D）以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日（1D）より小さい場合。

```
path + ".%Y%m%d.%H"
```

次の例は <host> に System.out を構成します。<web-app>の出力

```
<host id='foo.com'>
  <stdout-log path='/var/log/foo/stdout.log'
        rollover-period='1W'/>
  ...
</host>
```

stderrログ

System.err の送信先を設定します。

stderrログの設定は、親の設定より優先されます。例えば、<web-app
のstderrログの設定より優先されます。

> 注意
>
> 「path」はコマンドライン「-stderr」で指定されたパスと同

属性

属性名	説明
archive-format	ロールオーバーされたアーカイブファイル名の
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定しま
rollover-count	ロールオーバーファイルの最大数。
rollover-period	ロールオーバーを行う頻度。日（15D）、週（
rollover-size	ロールオーバーを行うファイルの最大サイズ。 （10mb）。
timestamp	行の先頭に使用する *タイムスタンプ* のフォーマ

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日（1D）以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日（1D）より小さい場合。

```
path + ".%Y%m%d.%H"
```

次の例は <host> に System.out を構成します。<web-app>の出力

```
<host id='foo.com'>
 <stderr-log path='/var/log/foo/stderr.log'
        rollover-period='1W'/>
 ...
</host>
```

<access-log>

<access-log> は、アクセス ログ ファイルを構成します。

<web-app> の子に定義されている場合、<host>の定義をオーバーラ

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日（1D）以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日（1D）より小さい場合。

```
path + ".%Y%m%d.%H"
```

アクセス ログのフォーマット変数は Apache 変数に従ってください。

フォーマットパターン

パターン	説明
%b	返却されるコンテンツの長さ。
%D	リクエストの処理が完了するまでにかかった時間。（マ 秒）
%h	リモートIPアドレス。
%{xxx}i	リクエストヘッダ。
%{xxx}o	レスポンスヘッド。
%{xxx}c	Cookieの値。
%n	リクエストの属性値。
%r	リクエストURL。
%s	ステータスコード。
%S	セッションID。
%{xxx}t	日時のフォーマット。
%T	リクエストの処理が完了するまでにかかった時間。（秒

パターン	説明
%u	リモートユーザ。
%U	リクエストURI。
%v	バーチャルホストのサーバ名。

デフォルトのフォーマットは以下のようになっています。

```
"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
```

属性

属性名	説明
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定
archive-format	ロールオーバーされたアーカイブファイル
auto-flush	要求のたびにメモリバッファーをフラッシ
auto-flush-time	メモリ バッファーをフラッシュする時間
exclude	リクエストのURIが一致する場合、アクセ
format	アクセス ログのフォーマット。
hostname-dns-lookup	IPアドレスの代わりにDNS名を記録。（
rollover-period	ロールオーバーを行う頻度。日（15D）、
rollover-size	ロールオーバーを行うログファイルの最大 ます。
rollover-count	ロールオーバーファイルの最大数。

<host> に <access-log> を設定する例。

```
<cluster id="app-tier">

 <host id="">
  <access-log path='log/access.log'>
   <rollover-period>2W</rollover-period>
  </access-log>
 </host>
</cluster>
```

ログのパス

「path」はメッセージの送信先を決定するために使用します。一般的に またはコンソールに表示するために標準出力、標準エラー出力に移行す

パス	説明
ファイルシステムのパス	ファイルに出力します。
stdout:	標準出力に出力します。
stderr:	標準エラー出力に出力します。

標準出力にログメッセージを出力する例

```
<log name="" level="all" path="stdout:"/>
```

コアモジュール

ネットワーク設定

項目

- 概要
- リファレンス
  - クラスタリングID
  - ポート番号
  - ポートレンジ
  - プロトコルバージョン設定
  - ユニキャスト設定
  - マルチキャスト設定
  - 初期起動ホスト設定
  - ホストアドレス設定
  - マルチキャストアドレス
  - マルチキャストポート番号
  - 初期起動ホスト数

概要

intra-mart Accel Platform を分散環境で利用する場合のクラスタリン

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/network-agent-config.xsd
設定場所	WEB-INF/conf/network-agent-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<network-agent-config
   xmlns="http://intra-mart.co.jp/system/service/client/config/netw
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/service/clien
 <id>clusterId</id>
 <bind-port>5200</bind-port>
 <port-range>2</port-range>
 <preferIPv4Stack>true</preferIPv4Stack>

 <TCP>
  <initial-hosts>
   <host address="127.0.0.1"/>
  </initial-hosts>
 </TCP>
</network-agent-config>
```

リファレンス

クラスタリングID

タグ名	id

クラスタリングを構成するグループの一意のIDです。
Webアプリケーション毎に一意となるIDを設定してください。

【設定項目】

```
<network-agent-config>
 <id>clusterId</id>
</network-agent-config>
```

必須項目	○

複数設定	×
設定値・設定する内容	Webアプリケーション毎に一意のIDを設定します。
単位・型	文字列 (xxxxxxx)
省略時のデフォルト値	なし
親タグ	network-agent-config

注意

Webアプリケーション間で同じIDが設定されている場合、予期
必ず、Webアプリケーション毎に一意となるIDを指定してくだ

ポート番号

タグ名	bind-port

Web Application Server 間で通信を行う際に利用するポート番号を指

【設定項目】

```
<network-agent-config>
 <bind-port>5200</bind-port>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	利用するポート番号を設定します。
単位・型	数値 (1024 - 65535)
省略時のデフォルト値	なし
親タグ	network-agent-config

ポートレンジ

タグ名	port-range

bind-portで指定されたポート番号が既に使用されている場合に代替えと
例えば、bind-portに5200、port-rangeに2が設定されており、5200番

【設定項目】

```
<network-agent-config>
 <port-range>2</port-range>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	利用するポート番号のレンジを設定します。
単位・型	数値 (0 -)
省略時のデフォルト値	なし
親タグ	network-agent-config

プロトコルバージョン設定

タグ名	preferIPv4Stack

通信を行う際のプロトコルバージョンを指定します。
trueを設定するとIPv4 ソケットを使用して通信を行います。

【設定項目】

```
<network-agent-config>
 <preferIPv4Stack>true</preferIPv4Stack>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	false　IPv6 ソケットを使用して通信を行います。 true　IPv4 ソケットを使用して通信を行います。
単位・型	真偽値 (true/false)
省略時のデフォルト値	なし
親タグ	network-agent-config

> **注意**
> intra-mart Accel Platform では、IPv6をサポートしていない

ユニキャスト設定

タグ名	TCP

この設定を行うとユニキャストでの通信を行います。

【設定項目】

```
<network-agent-config>
 <TCP>
  ...
 </TCP>
</network-agent-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	ユニキャストでの通信を行う際に必要になる設定をし
単位・型	なし
省略時のデフォルト値	なし
親タグ	network-agent-config

> **注意**
> この設定を行った場合、マルチキャストの設定(UDP)は行えま

マルチキャスト設定

タグ名	UDP

この設定を行うとマルチキャストでの通信を行います。

【設定項目】

```
<network-agent-config>
 <UDP>
  ...
 </UDP>
</network-agent-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	マルチキャストでの通信を行う際に必要になる設定を
単位・型	なし
省略時のデフォルト値	なし
親タグ	network-agent-config

注意

この設定を行った場合、ユニキャストの設定(TCP)は行えませ

初期起動ホスト設定

タグ名	initial-hosts

intra-mart Accel Platform を初期起動時のサーバのホストアドレスを
初期起動ホストはクラスタリングを行う際に通信を行う起点になります。
後から起動したサーバはこの初期起動ホストに指定されているサーバと

【設定項目】

```
<network-agent-config>
 <TCP>
  <initial-hosts>
   ...
  </initial-hosts>
 </TCP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	初期起動時するサーバのホストアドレスを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	TCP

ホストアドレス設定

タグ名	host

Web Application Server のサーバアドレスを指定します。

【設定項目】

```
<network-agent-config>
 <TCP>
  <initial-hosts>
   <host address="127.0.0.1"/>
  </initial-hosts>
 </TCP>
</network-agent-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	サーバアドレスを設定します。
単位・型	文字列 (xxx.xxx.xxx.xxx)
省略時のデフォルト値	なし
親タグ	initial-hosts

【属性】

属性名	説明	必須	デフォルト値
address	サーバアドレスを指定します。	○	なし

コラム

- 「address」属性にはホスト名を設定することも可能です
- ホスト名に紐づくIPアドレスが変わった場合はAPサーバを

```
<TCP>
 <initial-hosts>
  <host address="app0-host"/>
  <host address="app1-host"/>
  <host address="app2-host"/>
 </initial-hosts>
</TCP>
```

マルチキャストアドレス

タグ名	mcast-address

通信時に使用するマルチキャストアドレスを指定します。

【設定項目】

```
<network-agent-config>
 <UDP>
  <mcast-address>228.10.10.10</mcast-address>
 </UDP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	マルチキャストアドレスを設定します。
単位・型	文字列 (xxx.xxx.xxx.xxx)

省略時のデフォルト値	なし
親タグ	UDP

コラム

- <mcast-address>にはホスト名を設定することも可能で
- ホスト名に紐づくIPアドレスが変わった場合はAPサーバを

```
<UDP>
  <mcast-address>app0-host</mcast-address>
</UDP>
```

マルチキャストポート番号

タグ名	mcast-port

通信時に使用するマルチキャストアドレスのポート番号を指定します。

【設定項目】

```
<network-agent-config>
 <UDP>
  <mcast-port>45588</mcast-port>
 </UDP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	マルチキャストアドレスのポート番号を設定します。
単位・型	数値 (1024 - 65535)
省略時のデフォルト値	なし
親タグ	UDP

初期起動ホスト数

タグ名	initial-members

intra-mart Accel Platform を初期起動時のサーバ台数を指定します。

【設定項目】

```
<network-agent-config>
 <UDP>
  <initial-members>1</initial-members>
 </UDP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	初期起動時のサーバ台数を設定します。
単位・型	数値 (0 -)
省略時のデフォルト値	なし

親タグ	UDP

サーバコンテキスト設定

項目
- 概要
- リファレンス
  - ホームディレクトリ
  - 一時ファイルディレクトリ
  - ログファイルディレクトリ
  - サーバ文字コード
  - ベースURL

概要

intra-mart Accel Platform が起動するサーバ情報を設定します。

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/server-context-config.xsd
設定場所	WEB-INF/conf/server-context-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server-context-config
   xmlns="http://intra-mart.co.jp/system/platform/config/server-cor
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/platform/cor
 <home-directory>WEB-INF</home-directory>
 <work-directory>WEB-INF/work</work-directory>
 <log-directory>WEB-INF/log</log-directory>
 <server-charset>UTF-8</server-charset>
 <!--
 <base-url>http://127.0.0.1:8080/imart</base-url>
  -->
</server-context-config>
```

リファレンス

ホームディレクトリ

タグ名	home-directory

アプリケーションのホームディレクトリを指定します。
ホームディレクトリは設定ファイルやソースのパスを解決する際の基底

【設定項目】

```
<server-context-config>
 <home-directory>WEB-INF</home-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	アプリケーションの基底ディレクトリを設定します。
単位・型	文字列 (xxx/xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

> **注意**
> この設定を変更した場合、 intra-mart Accel Platform が起動

一時ファイルディレクトリ

タグ名	work-directory

intra-mart Accel Platform で利用する一時ファイルディレクトリを設定

【設定項目】

```
<server-context-config>
 <work-directory>WEB-INF/work</work-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	一時ファイルディレクトリを設定します。
単位・型	文字列 (xxx/xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

ログファイルディレクトリ

タグ名	log-directory

intra-mart Accel Platform が出力するログファイルの出力先を指定し

【設定項目】

```
<server-context-config>
 <log-directory>WEB-INF/log</log-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ログファイルディレクトリを設定します。
単位・型	文字列 (xxx/xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

サーバ文字コード

タグ名	server-charset

intra-mart Accel Platform の文字コードを設定します。
この設定は、サーバで文字列を扱う時に Unicode との文字コード変換を

【設定項目】

```
<server-context-config>
 <server-charset>UTF-8</server-charset>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	サーバ文字コードを設定します。
単位・型	文字列 (xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

注意

この設定は変更しないでください。

ベースURL

タグ名	base-url

intra-mart Accel Platform のベースURLを設定します。
この設定は、サーバで特定のURLを生成する際等に利用されます。
Web Server と Web Application Server でURLが異なる場合等に、ク

【設定項目】

```
<server-context-config>
 <base-url>http://127.0.0.1:8080/imart</base-url>
</server-context-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	クライアント（ブラウザ等）からアクセスを行
単位・型	文字列
省略時のデフォルト値	リクエストURLから自動的に解決されます。
親タグ	server-context-config

ストレージ設定ファイル

項目

- 概要
- リファレンス
  - 文字コード設定
  - ストレージAPI実装クラス設定
  - 使用禁止文字群
  - 使用禁止文字設定
  - ストレージ設定
  - ストレージルートパス設定
  - システムストレージディレクトリ名設定
  - パブリックストレージディレクトリ名設定
  - グループストレージディレクトリ名設定
  - 設定ファイル配置用ディレクトリ名設定
  - 互換モジュール用ディレクトリ名設定
  - ストレージディレクトリ名設定
  - テンポラリストレージディレクトリ名設定
  - ファイル同期キー設定
  - シンボリックリンク設定

概要

ストレージの利用に関する設定ファイルです。

モジュール	コアモジュール
フォーマットファイル (xsd)	WEB-INF/schema/storage-config.xsd
設定場所	WEB-INF/conf/storage-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<storage-config xmlns="http://jp/co/intra-mart/foundation/servic
mart/foundation/service/client/file/config ../schema/storage-config.x
 <charset>UTF-8</charset>
 <class-name>jp.co.intra_mart.system.service.client.file.StorageI
 <unable-charactors>
  <unable-character>\</unable-character>
  <unable-character>:</unable-character>
  <unable-character>*</unable-character>
  <unable-character>?</unable-character>
  <unable-character>&amp;quot;</unable-character>
  <unable-character>&amp;lt;</unable-character>
  <unable-character>&amp;gt;</unable-character>
  <unable-character>|</unable-character>
 </unable-charactors>
 <storage-info>
  <root-path-name>/tmp/storage</root-path-name>
  <system-directory-name>system</system-directory-name
  <public-directory-name>public</public-directory-name>
  <group-directory-name>groups</group-directory-name>
  <configuration-directory-name>conf</configuration-direc
  <conpatible-directory-name>public</conpatible-directory-
  <storage-directory-name>storage</storage-directory-nam
  <temporary-directory-name>temp</temporary-directory-
 </storage-info>
 <synchronize-info>
  <synchronize-key>im_system_storage_synchronize_key</syn
 </synchronize-info>
</storage-config>
```

リファレンス

文字コード設定

タグ名	charset

テキストファイルの読み込み、書き込み時に利用する文字コードを指定

【設定項目】

```
<storage-config>
 <charset>UTF-8</charset>
 ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	テキストファイルの読み込み、書き込み時に利用
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-config

注意

intra-mart Accel Platform では、利用可能な文字コードをU

ストレージAPI実装クラス設定

タグ名	class-name

Storage APIにおける実装クラスの指定を行います。

【設定項目】

```
<storage-config>
 ...
 <class-name>jp.co.intra_mart.system.service.client.file.StorageI
 ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ストレージ対する読み込み、書き込 指定されるクラスは、jp.co.intra_ 標準で提供されている実装はロー れています。 代替となる実装は提供されていま
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-config

注意

この設定はストレージに対する操作を行うための実装を定義し

使用禁止文字群

タグ名	unable-charactors

ファイル名、ディレクトリ名等における使用禁止文字群です。

【設定項目】

```
<storage-config>
 ...
 <unable-charactors>
  <unable-character>\</unable-character>
  <unable-character>:</unable-character>
  <unable-character>*</unable-character>
  <unable-character>?</unable-character>
  <unable-character>&quot;</unable-character>
  <unable-character>&lt;</unable-character>
  <unable-character>&gt;</unable-character>
  <unable-character>|</unable-character>
 </unable-charactors>
 ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ファイル名、ディレクトリ名等で使用禁止となる文
単位・型	なし
省略時のデフォルト値	なし
親タグ	storage-config

使用禁止文字設定

タグ名	unable-charactor

ファイル名、ディレクトリ名等における使用禁止文字の設定を行います。

【設定項目】

```
<storage-config>
 ...
 <unable-charactors>
  <unable-character>\</unable-character>
  <unable-character>:</unable-character>
  <unable-character>*</unable-character>
  <unable-character>?</unable-character>
  <unable-character>&quot;</unable-character>
  <unable-character>&lt;</unable-character>
  <unable-character>&gt;</unable-character>
  <unable-character>|</unable-character>
 </unable-charactors>
 ...
</storage-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	ファイル名、ディレクトリ名等で使用禁止となる文 標準で禁止している文字は：＊？”＜＞｜が設定され
単位・型	文字列
省略時のデフォルト値	なし
親タグ	unable-charactors

注意

この設定はWindows, Linux等で利用されるファイルシステム

ストレージ設定

タグ名	storage-info

ストレージのルートパス等を指定します。

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
 </storage-info>
 ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ストレージのパス等の情報を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	storage-config

ストレージルートパス設定

タグ名	root-path-name

ストレージルートとして利用するファイルシステム上のパスを指定しま
この設定では、システムプロパティの利用が可能です。
${システムプロパティのキー} 形式で記述を行うことによりシステムプ
例えば、Resinを利用した場合は、Resinがインストール（展開）された
その為 ${resin.home}/storage と記述した場合はResinがインストー
この値は通常、ストレージとして利用するNFS等の共有ディスクのパス
/tmp等、OS起動時に削除される可能性のあるパス等の指定は行わない

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  <root-path-name>/tmp/storage</root-path-name>
  ...
 </storage-info>
 ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ストレージルートとして利用するファイルシステム
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

システムストレージディレクトリ名設定

タグ名	system-directory-name

システムストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <system-directory-name>system</system-directory-name
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	システムストレージとして利用されるディレクトリ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

> 注意
>
> この設定は、他のストレージディレクトリ名設定（public-dir
> この設定は、 intra-mart Accel Platform において利用される

パブリックストレージディレクトリ名設定

タグ名	public-directory-name

パブリックストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ

intra-mart Accel Platform 2013 Winter以前
指定したディレクトリがパブリックストレージのルートディレクトリに

intra-mart Accel Platform 2014 Spring以降
指定したディレクトリの直下に作成されるテナントIDをディレクトリ名
この設定はテナント作成時にストレージルートパスが指定されていない
テナント作成時にストレージルートパスが指定されている場合は、そち

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <public-directory-name>public</public-directory-name>
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	パブリックストレージとして利用されるディレクト
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

> **注意**
>
> この設定は、他のストレージディレクトリ名設定（system-di
> この設定は、 intra-mart Accel Platform において利用される

グループストレージディレクトリ名設定

タグ名	group-directory-name

グループストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <group-directory-name>groups</group-directory-name>
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	グループストレージとして利用されるディレクトリ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

> **注意**
>
> この設定は、他のストレージディレクトリ名設定（system-di
> グループストレージは過去の互換性および、将来的な利用の為
> この設定は、 intra-mart Accel Platform において利用される

設定ファイル配置用ディレクトリ名設定

タグ名	configuration-directory-name

設定ファイル配置用として利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <configuration-directory-name>conf</configuration-direc
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	設定ファイル配置用として利用されるディレクトリ
単位・型	文字列
省略時のデフォルト値	なし

親タグ	storage-info

> **注意**
>
> この設定は、他のストレージディレクトリ名設定（system-di
> 設定ファイル配置用ディレクトリ名設定は、将来的な利用の為
> この設定は、 intra-mart Accel Platform において利用される

互換モジュール用ディレクトリ名設定

タグ名	compatible-directory-name

互換モジュール用として利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <compatible-directory-name>public</compatible-director
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	互換モジュール用として利用されるディレクトリ名
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

> **注意**
>
> 互換モジュールを利用した場合のストレージディレクトリはパ
> 互換モジュール用ディレクトリ名設定は、将来的な利用の為の
> この設定は、 intra-mart Accel Platform において利用される

ストレージディレクトリ名設定

タグ名	storage-directory-name

パブリックストレージ、システムストレージ、テンポラリストレージそ
指定されたディレクトリ名は、パブリックストレージ、システムストレ

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <storage-directory-name>storage</storage-directory-nam
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	ストレージディレクトリとして利用されるディレク
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

注意

この設定は、 intra-mart Accel Platform において利用される

テンポラリストレージディレクトリ名設定

タグ名	temporary-directory-name

テンポラリストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパ
このディレクトリは、セッションストレージ等で利用される一時領域と

【設定項目】

```
<storage-config>
 ...
 <storage-info>
  ...
  <temporary-directory-name>temp</temporary-directory-
  ...
 </storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	テンポラリストレージとして利用されるディレクト
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

注意

この設定は、他のストレージディレクトリ名設定（system-di
この設定は、 intra-mart Accel Platform において利用される

ファイル同期キー設定

タグ名	synchronize-key

分散環境において、ストレージを操作する際の同期化用ロックキーの設
この項目は将来的な利用の為の予約項目です。

【設定項目】

```
<storage-config>
 <synchronize-info>
  <synchronize-key>im_system_storage_synchronize_key</syn
 </synchronize-info>
</storage-config>
```

必須項目	○

複数設定	×
設定値・設定する内容	同期化用のロックに利用するキー値を設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	synchronize-info

シンボリックリンク設定

タグ名	permit-symlink

ストレージのルートパスの配下外に実体をもつシンボリックリンクやジ
ストレージのルートパスとは、ストレージ設定で設定される各ストレー
本設定は、 intra-mart Accel Platform 2015 Winter(Lydia) 以降で利

【設定項目】

```
<storage-config>
 ...
 <permit-symlink>false</permit-symlink>
 ...
</storage-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	true　ストレージのルートパス配下外に false　ストレージのルートパス配下外に ん。
単位・型	真偽値 (true/false)
省略時のデフォルト値	false
親タグ	storage-config

注意

シンボリックリンクやジャンクションを利用した場合、ストレ

サービスセレクタ設定

項目

- 概要
- リファレンス
  - サービス設定
  - リスナの設定
  - サービスの起動するサーバの指定
  - サービスの起動させないサーバの指定
  - サーバアドレス

概要

intra-mart Accel Platform の各サービスが起動するサーバの選出方法

モジュール	コアモジュール
フォーマットファイル (xsd)	WEB-INF/schema/service-selecter-config
設定場所	WEB-INF/conf/service-selecter-config/se

```
<?xml version="1.0" encoding="UTF-8"?>
<service-selecter-config
   xmlns="http://intra-mart.co.jp/system/service/selecter/config/ser
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/service/selec
 <services>
   <service service-id="server.service.controller"/>
 </services>
</service-selecter-config>
```

リファレンス

サービス設定

タグ名　services

各サービスの選出方法を設定します。

【設定項目】

```
<server-selecter-config>
 <services>
   <service service-id="XXXXX" />
 </services>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	service　各サービスの選出方法の設定
単位・型	なし
省略時のデフォルト値	なし
親タグ	service-selecter-config

リスナの設定

タグ名　service

サービスが起動するサーバの選出方法を設定します。

【設定項目】

```
<services>
 <service service-id="XXXXX" />
</services>
```

必須項目	○
複数設定	○
設定値・設定する内容	サービスが起動するサーバの選出方法を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	services

【属性】

属性名	説明
service-id	選出方法を設定するサービスのID。
select-num	サービスが起動するサーバの台数を指定します。 ます。）

コラム

intra-mart Accel Platform で提供されている各サービス

- ServerManager ： 「server.service.controller」
- TaskService ： 「server.service.task.managemen
- JobSchedulerService ： 「server.service.job_sche

サービスの起動するサーバの指定

タグ名	assign-hosts

サービスが起動するサーバを指定する場合に設定します。
単一のサーバで起動するサービスの場合、指定されているサーバを優先
指定されているサーバが起動していない場合は、動的にサービスが起動

複数のサーバで起動するサービスの場合、指定されているサーバでサー
指定されているサーバが起動していない場合は、動的にサービスが起動

【設定項目】

```
<service service-id="XXXXX">
 <assign-hosts>
  <host>10.0.0.1</host>
 </assign-hosts>
</service>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	host	サービスが起動するサーバを指定します。
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	service	

サービスの起動させないサーバの指定

タグ名	invalid-hosts

サービスを起動させないサーバを指定する場合に設定します。
この設定に指定されているサーバのみ起動している場合はサービスは起

【設定項目】

```
<service service-id="XXXXX">
 <invalid-hosts>
  <host>10.0.0.1</host>
 </invalid-hosts>
</service>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	host	サービスを起動させないサーバのアドレス。
単位・型	なし	

省略時のデフォルト値	なし
親タグ	service

サーバアドレス

タグ名	host

サーバアドレスを設定します。
サービスを起動するサーバ、または、サービスを起動しないサーバのア

【設定項目】

```
<assign-hosts>
 <host>10.0.0.1</host>
</assign-hosts>
```

必須項目	○
複数設定	○
設定値・設定する内容	サーバアドレスを指定します。
単位・型	文字列 (xxx.xxx.xxx.xxx)
省略時のデフォルト値	なし
親タグ	assign-hosts, invalid-hosts

サーバアドレスには以下のワイルドカードを指定することができます。

- アスタリスク （＊）

  同じネットワークセグメントの全てのサーバを指定する場合に使用
  例えば以下のような指定が可能です。

  ```
  ＊： 全てのサーバ
  10.*： 「10」のセグメント内のサーバ
  10.0.*： 「10.0」のセグメント内のサーバ
  10.0.0.*： 「10.0.0」のセグメント内のサーバ
  ```

  注意
  - アスタリスクは「10.0.0.1*」のようにドット（.）単位
  - アスタリスクは末尾のみ指定可能です。「10.0.*.0」よ

- レンジ （[XX-XX]）

  サーバアドレスの範囲指定を行う場合に使用します。
  例えば以下のような指定が可能です。

  ```
  10.0.[1-100].1 ： 「10.0.1」から「10.0.100」のセグメントのアド
  10.0.0.[1-100] ： 「10.0.0.1」から「10.0.0.100」までのアドレス
  ```

  コラム
  - レンジはセグメント毎に指定してください。
  - レンジは「10.0.[1-10].[1.-100]」のように複数指定す

Identifier 設定

項目

- 概要
- リファレンス
  - サービスプロバイダ設定

概要

ユニークIDを生成する以下のAPIに関する設定です。

- スクリプト開発
  - Identifier オブジェクト
- Java 開発
  - jp.co.intra_mart.foundation.service.client.information.Id

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/identifier-config.xsd
設定場所	WEB-INF/conf/identifier-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<identifier-config xmlns:xsi="http://www.w3.org/2001/XMLSchen
   xmlns="http://intra-mart.co.jp/foundation/identifier/identifier-con
   xsi:schemaLocation="http://intra-mart.co.jp/foundation/identifier

  <generator-class>jp.co.intra_mart.foundation.service.client.info

</identifier-config>
```

リファレンス

サービスプロバイダ設定

タグ名	generator-class

ユニークIDを生成するサービスプロバイダの設定を行います。
jp.co.intra_mart.foundation.service.client.information.Ide
未設定の場合は intra-mart Accel Platform 2015 Winter(Lydia) 以前

intra-mart Accel Platform 2016 Spring(Maxima) から変更された新
jp.co.intra_mart.foundation.service.client.information.Ne

【設定項目】

```
<identifier-config>
  <generator-class>jp.co.intra_mart.foundation.service.client.info
</identifier-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	identifier-config

注意

intra-mart Accel Platform 2015 Winter(Lydia) 以前の Ide
ます。
- IDに含まれるシーケンス番号が 1,296（36の二乗） で最

intra-mart Accel Platform 2016 Spring(Maxima) 以降では
- 同一ミリ秒で発行可能なシーケンス番号の数を、 1,296
- 同一ミリ秒内に 1,679,616回以上 呼び出された場合には
- 必ず従来の方式で発行されたIDより大きな値が発行されま

その他、以下の仕様については従来と同様です。
- 発行されるIDの桁数（15桁）
- IDに利用される文字の種類（0~9 および a~z）

新しいID発行方式を有効にしてシステムを稼働させた後に、従
従来の方式に戻すとIDをソートした際の順序に不整合が発生し

initializer 設定

項目

- 概要
- リファレンス
  - サーバサイドJavaScript グローバル関数、または拡張APIの設
  - サーバサイドJavaScript グローバル関数：Javaクラス
  - サーバサイドJavaScript グローバル関数：JavaScript関数
  - 拡張API：Javaクラス
  - 拡張API：JavaScript関数
  - IMARTタグの設定
  - 拡張IMARTタグ：Javaクラス
  - 拡張IMARTタグ：JavaScript関数
  - ApplicationInitializer設定
  - 起動時初期化Javaクラス
  - 起動時初期化 js ファイルのパス

概要

intra-mart Accel Platform 起動時に、実行する初期化処理を設定しま
拡張API、グローバル関数、拡張imartタグの設定を含みます。
また、初期化クラス、および、初期化スクリプトの設定を含みます。
既存のファイルは編集しないでください。

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/initializer-config.xsd
設定場所	WEB-INF/conf/products/initializer/initializer-

注意

ファイル名の initializer-XXX.xml のXXXはショートモジュー
ショートモジュールIDとはモジュールIDをピリオド（.）で分
例として、モジュールIDが「org.example.foo」場合は「init

```
<?xml version="1.0" encoding="UTF-8"?>
<initializer-config
   xmlns="http://intra-mart.co.jp/system/secure/product/initializer/c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/secure/produ

 <java-script-api>
  <global-function-class>sample.common_libs.global_fnction.Gl
  <global-function-script>sample/common_libs/global_fnction#
  <api-class>sample.common_libs.api.InstanceApi</api-class>
  <api-script>sample/common_libs/api#staticApi</api-script>
 </java-script-api>

 <jssp-tag>
  <tag-class>sample.common_libs.imart_tag.ImSampleTag#sam
  <tag-script>sample/common_libs/imart_tag#imSampleTag</ta
 </jssp-tag>

 <initializer>
  <class-name>sample.common_libs.Init</class-name>
  <script-name>sample/common_libs/init</script-name>
 </initializer>
</initializer-config>
```

リファレンス

サーバサイドJavaScript グローバル関数、または拡張APIの設定

タグ名	java-script-api

サーバサイドJavaScript グローバル関数、拡張APIの設定を行います。

【設定項目】

```
<initializer-config
   xmlns="http://intra-mart.co.jp/system/secure/product/initializer/c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/secure/produ

 <java-script-api>
  <global-function-class>sample.common_libs.global_fnction.Gl
  <global-function-script>sample/common_libs/global_fnction#
  <api-class>sample.common_libs.api.InstanceApi</api-class>
  <api-script>sample/common_libs/api#staticApi</api-script>
 </java-script-api>

</initializer-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	global-function-class	グローバル関数 関数
	global-function-script	グローバル関数 関数
	api-class	オブジェクトを
	api-script	オブジェクトを クト
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	initializer-config	

サーバサイドJavaScript グローバル関数：Javaクラス

タグ名	global-function-class

指定したJavaクラスで サーバサイドJavaScript グローバル関数の登録を

【設定項目】

```
<java-script-api>
 <global-function-class>sample.common_libs.global_fnction.Glo
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	グローバル関数を実装したJavaクラスのパス#実行関数
単位・型	文字列 (xx.xxx.xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

サーバサイドJavaScript グローバル関数：JavaScript関数

タグ名	global-function-script

指定したJavaScript関数で サーバサイドJavaScript グローバル関数の登

【設定項目】

```
<java-script-api>
 <global-function-script>sample/common_libs/global_fnction#g
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	グローバル関数を定義する js ファイルのパス#実行関
単位・型	文字列 (xx/xxx/xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

拡張API：Javaクラス

タグ名	api-class

指定したJavaクラスで拡張APIの登録を行います。

【設定項目】

```
<java-script-api>
 <api-class>sample.common_libs.api.InstanceApi</api-class>
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	オブジェクトを定義した Javaクラスパス
単位・型	文字列 (xx.xxx.xx)
省略時のデフォルト値	なし

親タグ	java-script-api

拡張API：JavaScript関数

タグ名	api-script

指定したJavaScript関数で拡張APIの登録を行います。

【設定項目】

```
<java-script-api>
 <api-script>sample/common_libs/api#staticApi</api-script>
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	オブジェクトを定義した js ファイルパス#オブジェク
単位・型	文字列 (xx/xxx/xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

IMARTタグの設定

タグ名	jssp-tag

IMARTタグの設定を行います。

【設定項目】

```
<initializer-config
   xmlns="http://intra-mart.co.jp/system/secure/product/initializer/
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/secure/produ

 <jssp-tag>
   <tag-class>sample.common_libs.imart_tag.ImSampleTag</tag
   <tag-script>sample/common_libs/imart_tag#imSampleTag</ta
 </jssp-tag>

</initializer-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	tag-class	IMARTタグを定義したJava
	tag-script	IMARTタグを定義した js フ 実行関数
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	initializer-config	

拡張IMARTタグ：Javaクラス

タグ名	tag-class

指定したJavaクラスでIMARTタグの登録を行います。

【設定項目】

```
<jssp-tag>
 <tag-class>sample.common_libs.imart_tag.ImSampleTag</tag-
</jssp-tag>
```

必須項目	×
複数設定	○
設定値・設定する内容	IMARTタグを定義したJavaクラスのパス
単位・型	文字列 (xx.xxx.xx)
省略時のデフォルト値	なし
親タグ	jssp-tag

拡張IMARTタグ：JavaScript関数

タグ名	tag-script

指定したJavaScript関数でIMARTタグの登録を行います。

【設定項目】

```
<jssp-tag>
 <tag-script>sample/common_libs/imart_tag#imSampleTag</tag
</jssp-tag>
```

必須項目	×
複数設定	○
設定値・設定する内容	IMARTタグを定義した js ファイルのパス#実行関数
単位・型	文字列 (xx.xxx.xx#xxxx)
省略時のデフォルト値	なし
親タグ	jssp-tag

ApplicationInitializer設定

タグ名	initializer

初期化クラス、および、初期化スクリプトの設定を行います。

【設定項目】

```
<initializer-config
   xmlns="http://intra-mart.co.jp/system/secure/product/initializer/c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/secure/produ

 <initializer>
  <class-name>sample.common_libs.Init</class-name>
  <script-name>sample/common_libs/init</script-name>
 </initializer>

</initializer-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	class-name	初期化Javaクラスのパス
	script-name	初期化 js ファイルのパス
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	initializer-config	

起動時初期化Javaクラス

タグ名	class-name

指定したJavaクラスが起動時に初期化対象となる設定を行います。

【設定項目】

```
<initializer>
 <class-name  MBean="false">sample.common_libs.imart_tag.Ir
</initializer>
```

必須項目	×
複数設定	○
設定値・設定する内容	初期化Javaクラスのパス
単位・型	文字列 (xx.xxx.xx)
省略時のデフォルト値	なし
親タグ	initializer

【属性】

属性名	説明	必須
MBean	MBeanServerに登録するかどうかを指定します。	×

起動時初期化 js ファイルのパス

タグ名	script-name

指定したJavaクラスが起動時に初期化対象となる設定を行います。

【設定項目】

```
<initializer>
 <script-name>sample/common_libs/init</script-name>
</initializer>
```

必須項目	×

複数設定	○
設定値・設定する内容	初期化 js ファイルのパス
単位・型	文字列 (xx/xxx/xx)
省略時のデフォルト値	なし
親タグ	initializer

IPアドレス取得元設定

項目
- 概要
- リファレンス
  - リモートIPアドレス設定
  - リモートIPアドレス取得元HTTPヘッダ設定

概要

リモートホストのIPアドレスを取得する元となるHTTPヘッダを設定しま

リモートホストのIPアドレスを使用して認可でアクセス制御を行ったり、

デフォルト（未設定）の状態では javax.servlet.ServletRequest#get
しかし、ロードバランサやプロキシサーバを使用している環境では、 ja
この場合、代替となるHTTPヘッダ（X-Forwarded-For など）を設定す

ここで設定された順にヘッダ値を取得していき、IPアドレスに変換でき
最後まで取得できなかった場合は、 javax.servlet.ServletRequest#g
javax.servlet.ServletRequest#getRemoteAddr() の動作について詳

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/ip-address-detector-config
設定場所	WEB-INF/conf/ip-address-detector-config.xm

```
<?xml version="1.0" encoding="UTF-8"?>
<ip-address-detector-config xmlns="http://www.intra-mart.jp/ht
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.intra-mart.jp/http/utility/ip-addr
  <remote-address>
    <http-header delimiter="," position="first">X-Forwarded-For</
    <http-header>X-LB-Remote-Addr</http-header>
  </remote-address>
</ip-address-detector-config>
```

リファレンス

リモートIPアドレス設定

タグ名	remote-address

アクセス元のリモートホストのIPアドレスを取得する元の場所を定義し

【設定項目】

```
<ip-address-detector-config>
 <remote-address>
  ・・・・・
 </remote-address>
</ip-address-detector-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	ip-address-detector-config

リモートIPアドレス取得元HTTPヘッダ設定

タグ名	http-header

アクセス元のリモートホストのIPアドレスを取得するHTTPヘッダ名を定

ここで設定された順にヘッダ値を取得していき、IPアドレスに変換でき
最後まで取得できなかった場合は、 javax.servlet.ServletRequest#g

【設定項目】

```
<ip-address-detector-config>
 <remote-address>
  <http-header delimiter="," position="first">X-Forwarded-For</
  <http-header>X-LB-Remote-Addr</http-header>
 </remote-address>
</ip-address-detector-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	HTTPヘッダ名
単位・型	文字列
省略時のデフォルト値	なし
親タグ	remote-address

【属性】

属性名	説明
delimiter	ヘッダに設定されている値を区切る文字列です。 省略した場合は区切らず、取得できたそのままの値
position	delimiter で区切られた複数の値から評価に使用す delimiter を省略した場合は使用されません。 first　最初の値を評価します。 last　最後の値を評価します。 数値　指定された位置の値を評価します。範囲外 ん。 正の数を指定した場合は、最初から数えて 定します。 負の数を指定した場合は、最後から数えて 定します。

注意

X-Forwarded-For ヘッダは非標準のため、ロードバランサや
使用するロードバランサ、プロキシサーバの実装に応じて、d

マルチデバイス

クライアントタイプマスタ

項目
- 概要
- リファレンス
  - クライアントタイプの設定

概要

クライアントタイプを使用するための設定を行います。

モジュール	マルチデバイス
フォーマットファイル(xsd)	WEB-INF/schema/client-type-config.xsd
設定場所	WEB-INF/conf/client-type-config/{任意のフ

```
<?xml version="1.0" encoding="UTF-8"?>
<client-type-config
   xmlns="http://intra-mart.co.jp/system/multi_device/client_type/c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra_mart.co.jp/system/multi_device
 <client-type-info class="jp.co.intra_mart.system.multi_device.cl
</client-type-config>
```

リファレンス

クライアントタイプの設定

タグ名	client-type-info

intra-mart Accel Platformで使用できるクライアントタイプを定義しま

【設定項目】

```
<client-type-config>
 <client-type-info class="jp.co.intra_mart.system.multi_device.cl
</client-type-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし

親タグ	client-type-config

【属性】

属性名	説明
class	Javaインタフェース jp.co.intra_mart.found クラスのFQDNを指定します。
default	このクライアントタイプをシステム・デフォ

注意

default=”true” のクライアントタイプ設定が複数ある場合、
なります。
default=”true” のクライアントタイプが見つからない場合、

国際化機能

日付と時刻の形式マスタ

項目

- 概要
- リファレンス
  - 「日付と時刻の形式」の設定
  - 「日付と時刻の形式」識別子の設定
  - 「日付と時刻の形式」ロケールの設定
  - 「日付と時刻の形式」フォーマットの設定
  - 「日付と時刻の形式」フォーマット識別子の設定
  - 「日付と時刻の形式」フォーマットパターンの設定

概要

intra-mart Accel Platformで利用する日付と時刻の形式のマスタ情報を
ここで設定した内容は、以下の画面から選択して利用することができま

- テナント管理 > テナント情報 > 日付と時刻の形式
- 個人設定 > 日付と時刻の形式

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/date-time-format-confi
設定場所	WEB-INF/conf/date-time-format-config/{

```
<?xml version="1.0" encoding="UTF-8"?>
<date-time-format-config
 xmlns="http://www.intra-mart.jp/system/i18n/datetime/date-time-
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra-mart.jp/system/i18n/dateti
 <format-set default="true">
  <format-set-id>IM_DATETIME_FORMAT_SET_EN_BASE</forma
  <locale>en</locale>
  <format type="date">
   <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</forma
   <pattern default="true">MMM d, yyyy</pattern>
   <pattern>MMM dd, yyyy</pattern>
   <pattern>d/M/yyyy</pattern>
   <pattern>d/MM/yyyy</pattern>
   <pattern>dd/MM/yyyy</pattern>
   <pattern>d-MMM-yyyy</pattern>
   <pattern>dd-MMM-yyyy</pattern>
   <pattern>d MMM, yyyy</pattern>
   <pattern>dd MMM, yyyy</pattern>
   <pattern>d MMM yyyy</pattern>
   <pattern>dd MMM yyyy</pattern>
   <pattern>yyyy-MM-dd</pattern>
   <pattern>yyyy/MM/dd</pattern>
   <pattern>yyyy/M/d</pattern>
   <pattern>MM/dd/yyyy</pattern>
   <pattern>M/d/yyyy</pattern>
  </format>
  <format type="date">
   <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id
   <pattern default="true">MMM d</pattern>
   <pattern>MMM dd</pattern>
   <pattern>d/M</pattern>
   <pattern>d/MM</pattern>
   <pattern>dd/MM</pattern>
   <pattern>d-MMM</pattern>
   <pattern>dd-MMM</pattern>
   <pattern>d MMM</pattern>
   <pattern>dd MMM</pattern>
   <pattern>MM-dd</pattern>
  </format>
  <format type="date">
   <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
   <pattern default="true">yyyy/MM/dd</pattern>
   <pattern>yyyy/M/d</pattern>
   <pattern>MM/dd/yyyy</pattern>
   <pattern>M/d/yyyy</pattern>
```

```
    <pattern>dd/MM/yyyy</pattern>
    <pattern>d/M/yyyy</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</forma
    <pattern default="true">h:mm a</pattern>
    <pattern>hh:mm a</pattern>
    <pattern>H:mm</pattern>
    <pattern>HH:mm</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</forma
    <pattern default="true">h:mm:ss a</pattern>
    <pattern>hh:mm:ss a</pattern>
    <pattern>H:mm:ss</pattern>
    <pattern>HH:mm:ss</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
    <pattern default="true">HH:mm</pattern>
    <pattern>H:m</pattern>
  </format>
</format-set>
<format-set>
  <format-set-id>IM_DATETIME_FORMAT_SET_JA_BASE</format
  <locale>ja</locale>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</forma
    <pattern>yyyy'年'M'月'd'日'</pattern>
    <pattern>yyyy'年'MM'月'dd'日'</pattern>
    <pattern>yyyy/M/d</pattern>
    <pattern default="true">yyyy/MM/dd</pattern>
    <pattern>yyyy-MM-dd</pattern>
  </format>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id
    <pattern>M'月'd'日'</pattern>
    <pattern>MM'月'dd'日'</pattern>
    <pattern>M/d</pattern>
    <pattern default="true">MM/dd</pattern>
    <pattern>MM-dd</pattern>
  </format>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
    <pattern default="true">yyyy/MM/dd</pattern>
    <pattern>yyyy/M/d</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</forma
    <pattern>ah:mm</pattern>
    <pattern>ahh:mm</pattern>
    <pattern default="true">H:mm</pattern>
    <pattern>HH:mm</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</forma
    <pattern>ah:mm:ss</pattern>
    <pattern>ahh:mm:ss</pattern>
    <pattern default="true">H:mm:ss</pattern>
    <pattern>HH:mm:ss</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
    <pattern default="true">HH:mm</pattern>
    <pattern>H:m</pattern>
  </format>
</format-set>
<format-set>
  <format-set-id>IM_DATETIME_FORMAT_SET_ZHCN_BASE</for
  <locale>zh_CN</locale>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</forma
    <pattern>yyyy'年'M'月'd'日'</pattern>
    <pattern>yyyy'年'MM'月'dd'日'</pattern>
    <pattern>yyyy/M/d</pattern>
    <pattern>yyyy/MM/dd</pattern>
    <pattern default="true">yyyy-M-d</pattern>
    <pattern>yyyy-MM-dd</pattern>
    <pattern>d MMM yyyy</pattern>
    <pattern>dd MMM yyyy</pattern>
  </format>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id
    <pattern>M'月'd'日'</pattern>
    <pattern>MM'月'dd'日'</pattern>
    <pattern>M/d</pattern>
```

```
    <pattern>MM/dd</pattern>
    <pattern default="true">M-d</pattern>
    <pattern>MM-dd</pattern>
    <pattern>d MMM</pattern>
    <pattern>dd MMM</pattern>
   </format>
   <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
    <pattern default="true">yyyy/MM/dd</pattern>
    <pattern>yyyy/M/d</pattern>
    <pattern>yyyy-MM-dd</pattern>
    <pattern>yyyy-M-d</pattern>
   </format>
   <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</forma
    <pattern>ah:mm</pattern>
    <pattern>ahh:mm</pattern>
    <pattern default="true">H:mm</pattern>
    <pattern>HH:mm</pattern>
   </format>
   <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</forma
    <pattern>ah:mm:ss</pattern>
    <pattern>ahh:mm:ss</pattern>
    <pattern default="true">H:mm:ss</pattern>
    <pattern>HH:mm:ss</pattern>
   </format>
   <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
    <pattern default="true">HH:mm</pattern>
    <pattern>H:m</pattern>
   </format>
  </format-set>
</date-time-format-config>
```

リファレンス

「日付と時刻の形式」の設定

タグ名	format-set

intra-mart Accel Platformで利用する「日付と時刻の形式」を定義しま
詳細な設定は配下のタグで行います。
このタグでは、システム・デフォルトの設定を行います。

【設定項目】

```
<date-time-format-config>
 <format-set default="true">
  ・・・・・
 </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	date-time-format-config

【属性】

属性名	説明
default	true の場合、システム・デフォルトの「日付と時刻の います。

「日付と時刻の形式」識別子の設定

タグ名	format-set-id

「日付と時刻の形式」を識別するための識別子を設定します。

【設定項目】

```
<date-time-format-config>
 <format-set>
  <format-set-id>IM_DATETIME_FORMAT_SET_EN_BASE</forma
  ・・・・・
 </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	「日付と時刻の形式」の識別子
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format-set

「日付と時刻の形式」ロケールの設定

タグ名	locale

「日付と時刻の形式」のロケールを設定します。
設定するロケールは、ロケールマスタファイルに定義されている必要が

【設定項目】

```
<date-time-format-config>
 <format-set>
  ・・・・・
  <locale>en</locale>
  ・・・・・
 </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	ロケールID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format-set

「日付と時刻の形式」フォーマットの設定

タグ名	format

フォーマットの種類（「日付」「時刻」）を設定します。

【設定項目】

```
<date-time-format-config>
 <format-set>
  ・・・・・
  <format type="date">
   ・・・・・
  </format>
 </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	format-set

【属性】

属性名	説明		必
type	date	日付用のフォーマットとして扱われます。	○
	time	時刻用のフォーマットとして扱われます。	

「日付と時刻の形式」フォーマット識別子の設定

タグ名	format-id

フォーマットを識別するための識別子を設定します。

【設定項目】

```
<date-time-format-config>
 <format-set>
  ・・・・・
  <format type="date">
   <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</forma
   ・・・・・
  </format>
 </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format

「日付と時刻の形式」フォーマットパターンの設定

タグ名	pattern

ユーザに提供するフォーマットパターン文字列を設定します。

【設定項目】

```
<date-time-format-config>
 <format-set>
  ・・・・・
  <format type="date">
   ・・・・・
   <pattern default="true">MMM d, yyyy</pattern>
   <pattern>MMM dd, yyyy</pattern>
   ・・・・・
  </format>
 </format-set>
</date-time-format-config>
```

必須項目	○

複数設定	○
設定値・設定する内容	フォーマットパターン文字列
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format

【属性】

属性名	説明
default	true の場合、初期状態におけるデフォルト値として扱い す。

> **注意**
>
> 2016 Spring(Maxima)より、「 一般ユーザ操作ガイド 」-「
> 入力形式のフォーマットパターン文字列を編集・追加を行った
> 入力形式のフォーマットパターン文字列のデフォルト値は変更
> テナントとユーザで設定した日付・時刻の「入力形式」をシス
> 照してください。

ロケールマスタ

項目

- 概要
- リファレンス
  - ロケールの設定
  - 文字エンコーディングの設定

概要

intra-mart Accel Platformで利用するロケールの設定を行います。

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/locale-config.xsd
設定場所	WEB-INF/conf/locale-config/{任意のファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<locale-config
   xmlns="http://intra_mart.co.jp/system/i18n/locale/config/locale-c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra_mart.co.jp/system/i18n/locale/c
 <locale name="en" default="true">
   <encoding-name>UTF-8</encoding-name>
 </locale>
 <locale name="ja">
   <encoding-name>UTF-8</encoding-name>
 </locale>
 <locale name="zh_CN">
   <encoding-name>UTF-8</encoding-name>
 </locale>
</locale-config>
```

リファレンス

ロケールの設定

タグ名	locale

intra-mart Accel Platformで利用するロケールを定義します。

【設定項目】

```
<locale-config>
 <locale name="en">
  ・・・・・
 </locale>
</locale-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	locale-config

【属性】

属性名	説明
name	ロケール ID を設定します。
default	true の場合、システム・デフォルトのロケールになりま す。

注意

運用開始後にシステムロケールを **追加する** 場合には制約があ

運用開始後にシステムロケールを追加した場合、そのまま運用
加したロケール分のデータを補完してください。

- 言語追加ガイド
- 国際化支援機能仕様書

なお、運用開始後にシステムロケールを **削除する** 事は推奨し

注意

default=”true” のロケールが複数ある場合、設定ファイルの
default=”true” のロケールが見つからない場合、設定ファイ

文字エンコーディングの設定

タグ名	encoding-name

intra-mart Accel Platformで利用する文字エンコーディングを定義しま

【設定項目】

```
<locale-config>
 <locale name="en">
  <encoding-name>UTF-8</encoding-name>
 </locale>
</locale-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	文字エンコーディングを設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	locale

タイムゾーンマスタ

項目
- 概要
- リファレンス
  - タイムゾーンの設定

概要

intra-mart Accel Platformで利用するタイムゾーンの設定を行います。

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/time-zone-config.xsd
設定場所	WEB-INF/conf/time-zone-config/{任意のファ

```
<time-zone-config
 xmlns="http://www.intra_mart.co.jp/system/i18n/timezone/config/
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra_mart.co.jp/system/i18n/tim
 <time-zone-id>Asia/Tokyo</time-zone-id>
 <time-zone-id>Asia/Shanghai</time-zone-id>
 <time-zone-id>UTC</time-zone-id>
 <time-zone-id>America/New_York</time-zone-id>
</time-zone-config>
```

リファレンス

タイムゾーンの設定

タグ名	time-zone-id

intra-mart Accel Platformで利用するタイムゾーンをタイムゾーン ID

【設定項目】

```
<time-zone-config>
  <time-zone-id>Asia/Tokyo</time-zone-id>
</time-zone-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	タイムゾーン ID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	time-zone-config

注意

設定可能なタイムゾーン ID は、JDK でサポートされるタイム

数値形式マスタ

項目
- 概要
- リファレンス
  - 数値形式の設定
  - パラメータの設定

概要

intra-mart Accel Platformで利用する数値形式のマスタ情報を設定しま
ここで設定した内容は、以下の画面から選択して利用することができま

- テナント管理 > テナント情報 > 数値形式
- 個人設定 > 数値形式

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/decimal-format-config.x
設定場所	WEB-INF/conf/decimal-format-config/{任意

```
<?xml version="1.0" encoding="UTF-8"?>
 <decimal-format-config
 xmlns="http://www.intra-mart.jp/system/i18n/number/decimal-for
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra-mart.jp/system/i18n/numbe
 <decimal-format id="comma_period" default="true">
  <parameter param-name="grouping-separator" param-value="
  <parameter param-name="decimal-separator" param-value="
 </decimal-format>
 <decimal-format id="space_comma">
  <parameter param-name="grouping-separator" param-value="
  <parameter param-name="decimal-separator" param-value="
 </decimal-format>
 <decimal-format id="apostrophe_period">
  <parameter param-name="grouping-separator" param-value="
  <parameter param-name="decimal-separator" param-value="
 </decimal-format>
 <decimal-format id="period_comma">
  <parameter param-name="grouping-separator" param-value="
  <parameter param-name="decimal-separator" param-value="
 </decimal-format>
</decimal-format-config>
```

リファレンス

数値形式の設定

タグ名	decimal-format

intra-mart Accel Platformで利用する「数値形式」を定義します。

【設定項目】

```
<decimal-format-config>
  <decimal-format id="comma_period" class="jp.co.intra_mart.fo
      ・・・・・
  </decimal-format>
</decimal-format-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	decimal-format-config

【属性】

属性名	説明
id	数値形式 ID を設定します。
class	フォーマット処理を行うクラスのパスを指定します。
default	true の場合、システム・デフォルトの数値形式になりま す。

> **注意**
>
> classを指定しない場合、フォーマット処理にはjp.co.intra_m

> **注意**
>
> default=”true” の数値形式が複数ある場合、設定ファイルの
> default=”true” の数値形式が見つからない場合、設定ファイ

パラメータの設定

タグ名	parameter

数値形式のパラメータを設定します。

【設定項目】

```
<decimal-format-config>
  <decimal-format id="comma_period" default="true">
    <parameter param-name="grouping-separator" param-value=
    <parameter param-name="decimal-separator" param-value="
  </decimal-format>
</decimal-format-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	decimal-format

【属性】

属性名	説明	必須	デフォルト値
param-name	パラメータ名を設定します。	○	なし
param-value	パラメータ値を設定します。	○	なし

スクリプト開発モデル

source-config.xml

項目

- 概要
- リファレンス
  - 文字エンコーディングの設定
  - JavaScriptの設定
  - JavaScriptコンパイラの設定
  - 最適化レベルの設定
  - Viewの設定
  - Viewコンパイラの設定
  - XMLエスケープの設定
  - JavaScriptエスケープの設定
  - ID属性を自動的に付加する設定
  - HTMLコメントを削除する設定
- source-config.xmlの有効範囲
- ファイル単位での設定方法

概要

source-config.xmlは、スクリプト開発モデルのプログラムソースおよ

この設定ファイルは、ディレクトリに対して有効であり、そのディレク
り、そのディレクトリ以下のすべてのプログラムソースに対して設定内

モジュール	スクリプト開発モデル
フォーマットファイル (xsd)	なし
設定場所	スクリプト開発モデルのプログラムが配置さ

```
<resource-file>
  <charset>UTF-8</charset>
  <javascript>
    <compiler enable="true" />
    <!-- enable:true = Auto compiler to Java class -->
    <!-- enable:false = Interpreter -->

    <optimize level="0" />
    <!-- level:0 to 9 = Optimize level of Compile -->
  </javascript>
  <view>
    <compiler enable="true" />
    <!-- enable:true = Auto compiler -->
    <!-- enable:false = Interpreter -->
  </view>
</resource-file>
```

リファレンス

文字エンコーディングの設定

タグ名	charset

プログラムソースの文字エンコーディングの指定です。intra-mart Acc

【設定項目】

```
<resource-file>
  <charset>UTF-8</charset>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	文字エンコーディング名
単位・型	なし
省略時のデフォルト値	上位ディレクトリに配置された source-config.xml
親タグ	resource-file

> **注意**
>
> intra-mart Accel Platform および関連するアプリケーション
> この設定には必ず「UTF-8」を指定してください。

JavaScriptの設定

タグ名	javascript

スクリプト開発モデルのJavaScriptに関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
  </javascript>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	JavaScriptの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	resource-file

JavaScriptコンパイラの設定

タグ名	compiler

JavaScriptコンパイラに関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
    <compiler enable="true" />
  </javascript>
</resource-file>
```

必須項目	×
複数設定	×

設定値・設定する内容	JavaScriptコンパイラの設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javascript

【属性】

属性名	説明
enable	JavaScriptコンパイラの有効・無効を指定し • *true* を設定した場合、JavaScriptをJav ラスがメモリ上にキャッシュされ、以後 フォーマンスは向上しますが、ソースの サーバを再起動しなければいけません） • *false* を設定した場合、JavaScriptをイ 反映されます。開発をスムーズに進める フォーマンスは期待できません。この設 IM-Jugglingのビルドウィザードから WAR フ は **「結合テスト環境」** を選択していると、 デルのプログラムがインタプリタモードで実

最適化レベルの設定

タグ名	optimize

JavaScriptコンパイラの最適化に関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
    <optimize level="0" />
  </javascript>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	JavaScriptコンパイラの最適化レベルを指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javascript

【属性】

属性名	説明
level	JavaScriptコンパイラのプログラム解析にお 囲が大きくなります。0を設定した場合は、 プログラムを最適化して実行する事は、パフ リケートかつ複雑な機能な為、プログラムの 最適化レベルを適用する場合は、その適用範

> **注意**
>
> intra-mart Accel Platform および関連するアプリケーション
> この設定には必ず「0」を指定してください。

Viewの設定

タグ名	view

スクリプト開発モデルのHTMLに関する設定を行います。

【設定項目】

```
<resource-file>
  <view>
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	HTMLの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	resource-file

Viewコンパイラの設定

タグ名	compiler

Viewコンパイラに関する設定を行います。

【設定項目】

```
<resource-file>
  <view>
    <compiler enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	Viewコンパイラの設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	view

【属性】

属性名	説明

属性名	説明
enable	Viewコンパイラの有効・無効を指定します。 ・*true* を設定した場合、HTMLをコンパイリ）がメモリ上にキャッシュされ、以後フォーマンスは向上しますが、ソースのサーバを再起動しなければいけません） ・*false* を設定した場合、HTMLをインタされます。開発をスムーズに進めること フォーマンスは期待できません。この設 IM-JugglingのビルドウィザードからWARフは **「結合テスト環境」** を選択していると、デルのプログラムがインタプリタモードで実

XMLエスケープの設定

タグ名	escapeXml

XMLエスケープに関する設定を行います。
<imart> タグの属性に指定された文字列に対してXMLエスケープを行う

詳しい説明はAPIリストの エスケープ処理に関して を参照してください

> **コラム**
> 全ての<imart>タグがエスケープ機能に対応しているわけでは
> <imart>タグのAPIドキュメントを参照してください。

【設定項目】

```
<resource-file>
  <view>
    <escapeXml enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	XMLエスケープの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	view

【属性】

属性名	説明
enable	XMLエスケープの有効・無効を指定します
exclusion	XMLエスケープ処理の対象外とする文字列
delimiter4exclusion	XMLエスケープ処理の対象外とする文字列

JavaScriptエスケープの設定

タグ名	escapeJs

JavaScriptエスケープに関する設定を行います。
<imart> タグの属性に指定された文字列に対してJavaScriptエスケー

詳しい説明はAPIリストの エスケープ処理に関して を参照してくださ

> **コラム**
>
> 全ての<imart>タグがエスケープ機能に対応しているわけで
> <imart>タグのAPIドキュメントを参照してください。

【設定項目】

```
<resource-file>
  <view>
    <escapeJs enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	JavaScriptエスケープの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	view

【属性】

属性名	説明
enable	JavaScriptエスケープの有効・無効を指定
exclusion	JavaScriptエスケープ処理の対象外とする
delimiter4exclusion	JavaScriptエスケープ処理の対象外とする

ID属性を自動的に付加する設定

タグ名	replicateNameToId

<imart type="hidden"> タグ使用時に、name属性に指定された値を
例えば、name属性に「foo」と値が指定された場合、プログラム実行時

```
<INPUT type="hidden" name="foo" id="foo">
```

【設定項目】

```
<resource-file>
  <view>
    <replicateNameToId enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	ID属性を自動的に付加する設定を指定します。

単位・型	なし
省略時のデフォルト値	なし
親タグ	view

【属性】

属性名	説明
enable	• *true* を設定した場合、name属性に設定さ • *false* を設定した場合、id属性を付加しませ

> **注意**
>
> intra-mart Accel Platform および関連するアプリケーション
> replicateNameToId は対応しているプログラムソースにのみ

HTMLコメントを削除する設定

タグ名	removeHTMLComments

プログラム実行時に「<!-- -->」で囲まれているHTMLコメント部分を削

【設定項目】

```
<resource-file>
  <view>
    <removeHTMLComments enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	HTMLコメントを削除する設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	view

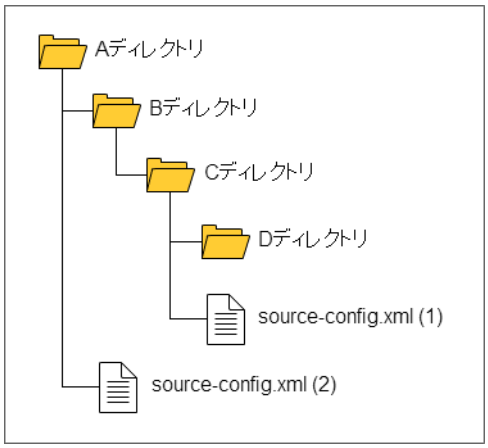
【属性】

属性名	説明
enable	• *true* を設定した場合、HTMLコメントを削除 • *false* を設定した場合、HTMLコメントを削除

source-config.xmlの有効範囲

下図のようにsource-config.xmlを配置した場合、各プログラムが参照

- Aディレクトリ直下のプログラム：(2)の設定内容を参照します。
- Bディレクトリ直下のプログラム：(2)の設定内容を参照します。
- Cディレクトリ直下のプログラム：(1)の設定内容を参照します。
- Dディレクトリ直下のプログラム：(1)の設定内容を参照します。



ファイル単位での設定方法

スクリプト開発モデルのプログラムは、HTMLファイルとJSファイルの

＜対象ファイルラベル名.properties＞ファイルを作成し、以下のように
て＜対象ファイルラベル名.properties＞ファイルの設定を利用します。

```
charset=プログラムの文字エンコーディング名
javascript.compile.enable=JavaScriptコンパイラの設定
javascript.optimize.level=JavaScriptコンパイラの最適化レベル
view.compile.enable=Viewコンパイラの設定
view.escapeXml.enable=XMLエスケープの有効・無効
view.escapeXml.exclusion=XMLエスケープ処理の対象外とする文字列
view.escapeXml.delimiter4exclusion=「view.escapeXml.exclusion」
view.escapeJs.enable=JavaScriptエスケープの有効・無効
view.escapeJs.exclusion=JavaScriptエスケープ処理の対象外とする文字
view.escapeJs.delimiter4exclusion=「view. escapeJs.exclusion」に指
view.replicateNameToId.enable=ID属性を自動的に付加する設定
view.removeHTMLComments.enable=HTMLコメントを削除する設定
```

例えば、ファイルが文字コード「UTF-8」で作成されたsample.html と
sample.properties を作成し、以下の内容を記述します。

```
charset=UTF-8
javascript.compile.enable=true
javascript.optimize.level=0
view.compile.enable=false
```

データベースアクセス機能

データソースマッピング設定

項目
- 概要
- リファレンス
  - システムデータベース設定
  - シェアードデータベース設定
  - テナントデータベース設定
  - リソース参照名
  - データベース種別
  - シェアードデータベース接続ID
  - テナントデータベース テナントID

概要

intra-mart Accel Platform で利用するデータベースと Web Applicati

モジュール	データベースアクセスモジュール
フォーマットファイル(xsd)	WEB-INF/schema/data-source-mapping-con
設定場所	WEB-INF/conf/data-source-mapping-config.x

```
<?xml version="1.0" encoding="UTF-8"?>
<data-source-mapping-config xmlns:xsi="http://www.w3.org/20
  xmlns="http://intra-mart.co.jp/system/database/config/data-sour
  xsi:schemaLocation="http://intra-mart.co.jp/system/database/co
 <system-data-source>
  <resource-ref-name>java:comp/env/jdbc/default</resource-
  <database-name>oracle</database-name>
 </system-data-source>
 <shared-data-source>
  <connect-id>default</connect-id>
  <resource-ref-name>java:comp/env/jdbc/default</resource-
 </shared-data-source>
 <tenant-data-source>
  <tenant-id>default</tenant-id>
  <resource-ref-name>java:comp/env/jdbc/default</resource-
 </tenant-data-source>
</data-source-mapping-config>
```

リファレンス

システムデータベース設定

タグ名	system-data-source

システムデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
 <system-data-source>
  <resource-ref-name>xxxxxxxxxxxxx</resource-ref-name>
  <database-name>xxxxxxxxxxxxx</database-name>
 </system-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	resource-ref-name	システムデータベース
	database-name	接続データベースの種
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	data-source-mapping-config	

シェアードデータベース設定

タグ名	shared-data-source

シェアードデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
 <shared-data-source>
  <connect-id>xxxxxxxxxxxxx</connect-id>
  <resource-ref-name>xxxxxxxxxxxxx</resource-ref-name>
  <database-name>xxxxxxxxxxxxx</database-name>
 </shared-data-source>
</data-source-mapping-config>
```

必須項目	×	
複数設定	○	
設定値・設定する内容	connect-id	シェアードデータベー
	resource-ref-name	シェアードデータベー
	database-name	接続データベースの種
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	data-source-mapping-config	

テナントデータベース設定

タグ名	tenant-data-source

テナントデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
 <tenant-data-source>
  <tenant-id>xxxxxxxxxxxxx</tenant-id>
  <resource-ref-name>xxxxxxxxxxxxx</resource-ref-name>
  <database-name>xxxxxxxxxxxxx</database-name>
 </tenant-data-source>
</data-source-mapping-config>
```

必須項目	×	
複数設定	○	
設定値・設定する内容	tenant-id	このデータベース設定
	resource-ref-name	テナントデータベース
	database-name	接続データベースの種

単位・型	なし
省略時のデフォルト値	なし
親タグ	data-source-mapping-config

リソース参照名

タグ名	resource-ref-name

Web Application Server に設定されているリソース参照名を指定する

【設定項目】

```
<data-source-mapping-config>
 <system-data-source>
  <resource-ref-name>xxxxxxxxxxxxx</resource-ref-name>
 </system-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	intra-mart Accel Platform で利用するデータソー
単位・型	文字列 (java:comp/env/xxxxx)
省略時のデフォルト値	なし
親タグ	system-data-source, shared-data-source, tena

データベース種別

タグ名	database-name

接続先データベースの種別を指定する項目です。
この項目が指定されていない場合は、データベースの種別は接続先デー

【設定項目】

```
<data-source-mapping-config>
 <system-data-source>
  <database-name>xxxxxxxxxxxxx</database-name>
 </system-data-source>
</data-source-mapping-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	接続先データベースの種類
単位・型	文字列 (oracle, sql server, postgresql)
省略時のデフォルト値	接続先データベースのデータベース種別にあわせて
親タグ	system-data-source, shared-data-source, ten

> **注意**
> 指定されたデータベースの種別と接続先データベースの種別が

シェアードデータベース接続ID

タグ名	connect-id

シェアードデータベースの接続IDを指定する項目です。
接続IDには各シェアードデータベース毎の一意のIDを設定してください

【設定項目】

```
<data-source-mapping-config>
 <shared-data-source>
  <connect-id>xxxxxxxxxxxxxx</connect-id>
 </shared-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	シェアードデータベースの接続ID
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	shared-data-source

テナントデータベース テナントID

タグ名	tenant-id

このデータベース設定を利用するテナントのテナントIDを指定する項目

【設定項目】

```
<data-source-mapping-config>
 <tenant-data-source>
  <tenant-id>xxxxxxxxxxxxxx</tenant-id>
 </tenant-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	テナントID
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	tenant-data-source

メールモジュール

メール設定

項目

- 概要
- リファレンス
  - SMTPの設定
  - SMTPサーバ設定
  - SMTPS(SMTP over SSL)設定
  - SMTP Authentication設定
  - SMTP Authenticationユーザ設定
  - SMTP Authenticationパスワード設定
  - X-Mailerヘッダ設定
  - デバッグ設定
  - コネクションタイムアウト設定
  - タイムアウト設定
  - コンテンツタイプ群
  - コンテンツタイプ設定
  - リスナ群設定
  - リスナ設定
  - メールヘッダ群設定
  - メールヘッダ設定
  - メールエンコード設定
  - 文字コード設定
  - MIMEエンコード設定
  - ボディエンコード設定

概要

メール送信に関する設定です。
デフォルトの設定としてjavamail-config.xmlが用意されています。
多言語対応として、javamail-config_{ロケールID}.xmlが存在します。

モジュール	メールモジュール
フォーマットファイル(xsd)	WEB-INF/schema/javamail-config.xsd
設定場所	WEB-INF/conf/javamail-config/javamail-con WEB-INF/conf/javamail-config/{任意のファイ

```
<?xml version="1.0" encoding="UTF-8"?>
<javamail-config xmlns:xsi="http://www.w3.org/2001/XMLSchem
   xmlns="http://intra-mart.co.jp/system/mail/javamail/config/javan
   xsi:schemaLocation="http://intra-mart.co.jp/system/mail/javama

 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   <smtps enable="false" starttls="false"/>
   <auth enable="false">
    <user/>
    <password/>
   </auth>
   <x-mailer>intra-mart MailSender ver 8.0</x-mailer>
   <debug>false</debug>
   <connection-timeout>-1</connection-timeout>
   <timeout>-1</timeout>
  </smtp-server>
 </smtp>

 <content-type>
  <file extension=".txt" content-type="text/plain"/>
  <file extension=".gif" content-type="image/gif"/>
  <file extension=".jpg" content-type="image/jpeg"/>
  <file extension=".png" content-type="image/png"/>
 </content-type>

 <listener>
  <listener-class>jp.co.intra_mart.foundation.mail.javamail.liste
 </listener>

 <encode>
  <charset>UTF-8</charset>
  <mime-encoding>B</mime-encoding>
  <content-transfer-encoding>7bit</content-transfer-encod
 </encode>

</javamail-config>
```

リファレンス

SMTPの設定

タグ名	smtp-server

SMTPの設定を行います。
メールの送信時に使用するSMTPの設定を行います。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ....
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SMTPの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

SMTPサーバ設定

タグ名	smtp-server

SMTPサーバの設定を行います。

複数のSMTPサーバの定義を行うことが可能です。
APIからのメール送信時にSMTPサーバが指定されていない場合、テナン
テナントIDと同じIDのSMTPサーバの設定が存在しない場合に利用され

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ....
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	SMTPサーバの設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp

【属性】

属性名	説明
id	このSMTPサーバの設定を特定する一意の APIからのメール送信時にこのIDを指定す 送信時にIDを指定しなかった場合、テナン しない場合は、標準設定（default）を利
host	SMTPサーバのホスト名またはIPアドレス
port	SMTPサーバのポート名を指定します。

SMTPS(SMTP over SSL)設定

タグ名	smtps

SMTPS(SMTP over SSL)の設定を行います。
SMTPSを利用するにはJavaランタイムがSMTPサーバのSSL証明書を信
SSL証明書が信頼されていない場合、javax.net.ssl.SSLHandshakeEx
JavaランタイムがSMTPサーバのSSL証明書を信頼済みであることを確

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   <smtps enable="true" starttls="false"/>
   ....
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	SMTPS(SMTP over SSL)の設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp-server

【属性】

属性名	説明
enable	SMTPサーバとの通信をSSLによる暗号化通信 合はfalseを設定してください。
starttls	TLS/STARTTLSによる暗号化通信を行います を設定してください。 この設定を有効にする場合はenable属性も有

SMTP Authentication設定

タグ名	auth

SMTP Authenticationを利用する場合に指定する要素です。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   <auth enable="false">
    ....
   </auth>
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際の設定を指定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp-server

【属性】

属性名	説明
enabled	SMTP Authenticationの利用設定を行います 合はfalseを設定してください。

SMTP Authenticationユーザ設定

タグ名	user

SMTP Authenticationを利用する際のユーザを設定します。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   <auth enable="true">
    <user>USER</user>
    ...
   </auth>
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際のユーザを設定し
単位・型	文字列
省略時のデフォルト値	なし
親タグ	auth

SMTP Authenticationパスワード設定

タグ名	password

SMTP Authenticationを利用する際のパスワードを設定します。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   <auth enable="true">
    ....
    <password>PASSWORD</password>
   </auth>
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際のパスワードを
単位・型	文字列
省略時のデフォルト値	なし
親タグ	auth

X-Mailerヘッダ設定

タグ名	x-mailer

X-Mailerメールヘッダに指定する値の設定を行います。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ...
   <x-mailer>intra-mart MailSender ver 8.0</x-mailer>
   ...
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メール送信時にX-Mailerヘッダとして送信する値を設
単位・型	文字列
省略時のデフォルト値	なし
親タグ	smtp-server

デバッグ設定

タグ名	debug

JavaMailにデバッグオプションを設定します。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ...
   <debug>false</debug>
   ...
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	false　デバッグ設定を行いません。 true　デバッグ設定を行います。
単位・型	真偽値 （true/false）
省略時のデフォルト値	なし
親タグ	smtp-server

コネクションタイムアウト設定

タグ名	connection-timeout

SMTPサーバ接続時のコネクションタイムアウト値を設定します。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ...
   <connection-timeout>-1</connection-timeout>
   ...
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SMTPサーバ接続時のコネクションタイムアウト値を -1が設定された場合はコネクションタイムアウトは 設定可能な数値の単位はミリ秒です。
単位・型	数値（1以上 または -1）
省略時のデフォルト値	なし
親タグ	smtp-server

タイムアウト設定

タグ名	timeout

メール送信時におけるタイムアウト値を設定します。

【設定項目】

```
<javamail-config>
 <smtp>
  <smtp-server id="default" host="localhost" port="25">
   ...
   <timeout>-1</timeout>
   ...
  </smtp-server>
 </smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メール送信時におけるタイムアウト値を設定します。 -1が設定された場合はタイムアウトは発生しません。 設定可能な数値の単位はミリ秒です。
単位・型	数値（1以上 または -1）
省略時のデフォルト値	なし
親タグ	smtp-server

コンテンツタイプ群

タグ名	content-type

メール送信時に添付するファイルのコンテンツタイプ群の設定を行いま

【設定項目】

```
<javamail-config>
 <content-type>
  ...
 </content-type>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	メール送信時に添付するファイルのコンテンツタイ
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

コンテンツタイプ設定

タグ名	file

メール送信時に添付するファイルのコンテンツタイプの設定を行います。
コンテンツタイプの設定は添付ファイルの拡張子に対しての設定です。

【設定項目】

```
<javamail-config>
 <content-type>
  <file extension=".txt" content-type="text/plain"/>
  <file extension=".gif" content-type="image/gif"/>
  <file extension=".jpg" content-type="image/jpeg"/>
  <file extension=".png" content-type="image/png"/>
 </content-type>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に添付するファイルのコンテンツタイ
単位・型	なし
省略時のデフォルト値	なし
親タグ	content-type

【属性】

属性名	説明
extension	拡張子を指定します。必ずピリオド（.）で始まる形い。
content-type	コンテンツタイプ属性です。拡張子に対応したコンます。

リスナ群設定

タグ名	listener

メール送信時に動作するリスナ群を指定します。

【設定項目】

```
<javamail-config>
 <listener>
  <listener-class></listener-class>
 </listener>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	メール送信時に動作するリスナ群を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

リスナ設定

タグ名	listener-class

メール送信時に動作するリスナを指定します。
指定するリスナは、jp.co.intra_mart.foundation.mail.javamail.liste
デフォルトでは、半角カナを全角カナに変換する為のリスナ（jp.co.int

【設定項目】

```
<javamail-config>
 <listener>
  <listener-class></listener-class>
 </listener>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に動作するリスナを指定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	listener

メールヘッダ群設定

タグ名	mail-headers

メールヘッダ群を指定します。

【設定項目】

```
<javamail-config>
 <mail-headers>
  ...
 </mail-headers>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	メール送信時に付与されるメールヘッダ群を指定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

メールヘッダ設定

タグ名	header

メール送信時に付与する任意のメールヘッダを指定します。
同じヘッダが複数指定された場合は、最下行に指定されたものがヘッダ
API（jp.co.intra_mart.foundation.mail.javamail.StandardMail#ad

【設定項目】

```
<javamail-config>
 <mail-headers>
  <header name="Error-To" value="xxxx@intra-mart.jp" />
 </mail-headers>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に付与する任意のメールヘッダを指定し
単位・型	文字列
省略時のデフォルト値	なし
親タグ	mail-headers

メールエンコード設定

タグ名	encode

ロケールに対応したメールエンコードに関する設定を行います。
locale属性が設定されていない要素は、ロケールに対応したメールエン

【設定項目】

```
<javamail-config>
 <encode>
  <charset>UTF-8</charset>
  <mime-encoding>B</mime-encoding>
  <content-transfer-encoding>7bit</content-transfer-encod
 </encode>
</javamail-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	ロケールに対応したメールエンコードに関する設定を
親タグ	javamail-config

【属性】

属性名	説明	必須	デフォルト値
locale	ロケールIDを指定します。	×	なし

文字コード設定

タグ名	charset

文字コードの指定を行います。

【設定項目】

```
<javamail-config>
 <encode>
  <charset>UTF-8</charset>
  ...
 </encode>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	文字コードの指定を行います。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	encode

MIMEエンコード設定

タグ名	mime-encoding

MIMEエンコードの指定を行います。
指定可能な値は B または Q です。

【設定項目】

```
<javamail-config>
 <encode>
  ...
  <mime-encoding>B</mime-encoding>
  ...
 </encode>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	MIMEエンコードの指定を行います。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	encode

ボディエンコード設定

タグ名	content-transfer-encoding

ボディのエンコードの指定を行います。
7bit または 8bit 、 Quoted Printable等の値を指定します。

【設定項目】

```
<javamail-config>
 <encode>
  ...
  <content-transfer-encoding>7bit</content-transfer-encod
 </encode>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ボディのエンコードの指定を行います。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	encode

Webモジュール

リクエスト制御設定

項目

- 概要
- リファレンス
  - サーバの同時処理の制限
  - リクエストの処理制限
  - リクエストクエリの制限
  - リクエストクエリの長さ制限
  - リクエスト処理の直列化
  - 直列処理を行うリクエストクエリの制限
  - 直列処理を行うリクエストクエリの長さ制限

概要

リクエスト制御の設定です。
リクエストの同時処理制限やクエリの長さによるページ処理を制御する

モジュール	Webモジュール
フォーマットファイル(xsd)	WEB-INF/schema/request-control-config.xsd
設定場所	WEB-INF/conf/request-control-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<request-control-config
   xmlns="http://intra-mart.co.jp/system/servlet/filter/xml/request_
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/servlet/filter/
 <thread max="32"/>
 <accept queue="100">
  <query>
   <length max="0"/>
  </query>
 </accept>
 <synchronized queue="8">
  <query>
   <length min="1048576"/>
  </query>
 </synchronized>
</request-control-config>
```

リファレンス

サーバの同時処理の制限

タグ名	thread

サーバの同時処理制限を指定します。
制限数を超えるリクエストを受け付けた場合、制限数を超えたリクエス
maxに 0 を指定するとリクエストの同時処理数を制限しません。

【設定項目】

```
<request-control-config>
 <thread max="32"/>
</request-control-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	サーバの同時処理制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-control-config

【属性】

属性名	説明	必須	デフォルト値
max	サーバの最大同時処理数を設定します。	○	なし

リクエストの処理制限

タグ名	accept

サーバが受け付けたリクエストの処理制限を指定します。
同時処理数を超えた場合の処理待ちを行うリクエストの最大数やリクエ

【設定項目】

```
<request-control-config>
 <accept queue="100">
  ...
 </accept>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストの処理制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-control-config

【属性】

属性名	説明	必須
queue	処理待ちを行うリクエストの最大数を設定します。	○

リクエストクエリの制限

タグ名	query

リクエストクエリの制限を指定します。
リクエストクエリに対する制限を行います。

【設定項目】

```
<request-control-config>
 <accept queue="100">
  <query>
   ...
  </query>
 </accept>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストクエリの制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	accept

リクエストクエリの長さ制限

タグ名	length

リクエストクエリの長さ制限を指定します。
リクエストクエリの長さがこの設定値よりも大きい場合は、HTTPレスポ
maxに 0 を指定するとリクエストクエリの長さによる制限を行いません

【設定項目】

```
<request-control-config>
 <accept>
  <query>
   <length max="0"/>
    ...
   </length>
  </query>
 </accept>
<request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストクエリの長さ制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	query

【属性】

属性名	説明
max	この設定値に指定された長さよりリクエストクエリ み処理を行います。

リクエスト処理の直列化

タグ名	synchronized

クエリサイズの大きいリクエストの処理に関する制限を指定します。
サイズの大きいリクエストを処理する場合に、リソース不足によるエラー
この設定を行うとサイズの大きいリクエストは受付順に直列処理されま

【設定項目】

```
<request-control-config>
 <synchronized queue="8">
  ...
 </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストの処理制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-control-config

【属性】

属性名	説明	必須
queue	処理待ちを行うリクエストの最大数を設定します。	○

直列処理を行うリクエストクエリの制限

タグ名	query

直列処理を行うリクエストのクエリの制限を指定します。
この設定値を基準に並列処理するリクエストと直列処理するリクエスト

【設定項目】

```
<request-control-config>
 <synchronized queue="8">
  <query>
   ...
  </query>
 </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	直列処理を行うリクエストのクエリの制限を設定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	synchronized

直列処理を行うリクエストクエリの長さ制限

タグ名	length

直列処理を行うリクエストのクエリの長さを指定します。
minに指定されたサイズよりリクエストクエリのサイズが小さいリクエ

【設定項目】

```
<request-control-config>
 <synchronized>
  <query>
   <length min="1048576"/>
  </query>
 </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	直列処理を行うリクエストのクエリの長さを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	query

【属性】

属性名	説明
min	直列処理を行うリクエストのクエリの長さを設定します。

レスポンスヘッダ設定

項目

- 概要
- リファレンス
  - 静的なヘッダーの指定
  - 動的なヘッダーの指定

概要

レスポンスヘッダの設定です。
レスポンスに任意のヘッダを追加する機能を提供します。

モジュール	Webモジュール
フォーマットファイル(xsd)	WEB-INF/schema/response-header-config.xs
設定場所	WEB-INF/conf/response-header-config/defau

```
<?xml version="1.0" encoding="UTF-8"?>
<response-header-config
 xmlns="http://intra-mart.co.jp/system/servlet/filter/xml/response_h
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/servlet/filter/x
   <!-- Avoid IE Content Sniffing XSS Attacks -->
   <!--
   <static-response-header name="X-Content-Type-Options" value=
   -->
</response-header-config>
```

リファレンス

静的なヘッダーの指定

タグ名	static-response-header

静的なレスポンスヘッダを指定します。

【設定項目】

```
<response-header-config>

  <static-response-header name="X-Content-Type-Options" valu

</response-header-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	静的なレスポンスヘッダを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-header-config

【属性】

属性名	説明
name	ヘッダー名です。
value	ヘッダー値です。
type	ヘッダー値の型です。text, int, date のい
format	type 属性に date を指定した際、value を date の value は long を指定します。
useAddHeader	ヘッダーを追加するかどうかのフラグです

動的なヘッダーの指定

タグ名	dynamic-response-header

静的なレスポンスヘッダを指定します。

【設定項目】

```
<response-header-config>
  <dynamic-response-header name="X-Dynamic-Sample" class=
</response-header-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	動的なレスポンスヘッダを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-header-config

【属性】

属性名	説明
name	ヘッダー名です。
class	ヘッダー値を返すJavaのクラス名です。jp したクラスを指定してください。
type	ヘッダー値の型です。text, int, date のい
useAddHeader	ヘッダーを追加するかどうかのフラグです

IM-Propagation

IM-Propagation 設定

項目

- 概要
- リファレンス
  - ロック時待機動作の設定
  - デッドロック検知時間の設定

概要

IM-Propagation 基本機能の設定です。

モジュール	IM-Propagation
フォーマットファイル(xsd)	WEB-INF/schema/propagation-config.xsd
設定場所	WEB-INF/conf/propagation-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-config xmlns="http://www.intra-mart.jp/propagatio
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/propagation/confi
 <each-thread-sleep>100</each-thread-sleep>
 <max-thread-sleep>1000</max-thread-sleep>
</propagation-config>
```

リファレンス

ロック時待機動作の設定

タグ名	each-thread-sleep

データの受信側で IM-Propagation を使用して次のデータ送信を行う場

【設定項目】

```
<propagation-config>
 <each-thread-sleep>100</each-thread-sleep>
</propagation-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	待機する時間
単位・型	数値[ミリ秒] (0 - 1000)
省略時のデフォルト値	100
親タグ	propagation-config

> 注意
> 通常はデフォルト値 (設定なし) のまま使用して、IM-Propaga
>
> 数値を小さく設定すると、ロック確認の間隔が短くなり処理が
> 数値を大きく設定すると、ロック確認の間隔が長くなりサーバ

デッドロック検知時間の設定

タグ名	max-thread-sleep

次のデータ受信先がロック状態の場合に最大で待機する時間を設定しま
この時間を超えた場合、デッドロックが発生したとみなされ、IM-Propa

【設定項目】

```
<propagation-config>
 <max-thread-sleep>1000</max-thread-sleep>
</propagation-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	デッドロック判定までの時間
単位・型	数値[ミリ秒] (0 - 60000)
省略時のデフォルト値	1000
親タグ	propagation-config

> 注意
> 通常はデフォルト値 (設定なし) のまま使用して、IM-Propaga
>
> 数値を小さく設定すると、デッドロック検知が顕著になりデー
> 数値を大きく設定すると、デッドロック検知までの時間が長く

IM-Propagation 受信側設定

項目

- 概要
- リファレンス
  - 受信データ定義の設定
  - 受信側データ変換クラスの設定
  - データ変換クラスパラメータの設定
  - データ変換クラスのパラメータ値の設定
  - 受信側データ処理クラスの設定
  - データ処理クラスパラメータの設定
  - データ処理クラスのパラメータ値の設定

概要

IM-Propagation を利用してデータを伝搬するための受信側を設定しま
受信するデータを格納するクラスと、データ変換クラスを紐付けるため

IM-Propagation を利用してデータを他モジュールから伝搬するために

データを伝搬する際に他モジュールとの依存関係を切り離す必要がある

データの受信側は IM-Propagation 用のモデルから自モジュールで定義
そこで、データをコピーする作業を行うデータ変換クラスの設定を行い

また、受信したデータをもとに処理を行うデータ処理クラスの設定も同

モジュール	IM-Propagation
フォーマットファイル (xsd)	WEB-INF/schema/propagation-receivers
設定場所	WEB-INF/conf/propagation-receivers-co

注意

ファイル名は、他のモジュールが提供しているものと重複しな

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-receivers-config xmlns="http://www.intra-mart.jp
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/propagation/recei
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <decoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </decoder>
  <procedure class="jp.co.intra_mart.system.sample.propagatio
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </procedure>
 </receiver>
</propagation-receivers-config>
```

リファレンス

受信データ定義の設定

タグ名	receiver

IM-Propagation を使用してデータを受信する側が受信するデータの型

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  ・・・・・
 </receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	propagation-receivers-config

【属性】

属性名	説明
source	データ送信側が IM-Propagation に渡すクラスの す。
operationType	データ送信側が IM-Propagation に渡すデータの

コラム

source 属性には IM-Propagation 用のデータが格納されたク
これにより、データの送信側を意識せず、常に共通的なデータ

注意

データ送信側設定の sender タグに設定された source と ope
存在しないクラスや operationType が指定されている場合は

受信側データ変換クラスの設定

タグ名	decoder

IM-Propagation 用のデータを受信側のデータに変換するための、データ

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <decoder class="jp.co.intra_mart.system.sample.propagation.S
 </receiver>
</propagation-receivers-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	receiver

【属性】

属性名	説明	必須	デフォル
class	データ変換クラスの完全修飾クラス名です。	○	な

注意

class 属性に指定したクラスが見つからない場合や適切なイン
この場合、データを受信することができませんので注意してく

データ変換クラスパラメータの設定

タグ名	params

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <decoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    ・・・・・
   </params>
  </decoder>
 </receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	decoder

データ変換クラスのパラメータ値の設定

タグ名	param

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <decoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </decoder>
 </receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	○

設定値・設定する内容	任意のパラメータ値
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	params

【属性】

属性名	説明	必須	デフォルト値
key	パラメータ値を取得するためのキー。	○	なし

受信側データ処理クラスの設定

タグ名	procedure

データ変換クラスによって変換されたデータを処理するための、データ

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <procedure class="jp.co.intra_mart.system.sample.propagatio
 </receiver>
</propagation-receivers-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	receiver

【属性】

属性名	説明	必須	デフォル
class	データ処理クラスの完全修飾クラス名です。	○	な

注意

class 属性に指定したクラスが見つからない場合や適切なイン
この場合、データを受信することができませんので注意してく

データ処理クラスパラメータの設定

タグ名	params

データ処理クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <procedure class="jp.co.intra_mart.system.sample.propagatio
   <params>
    ・・・・・
   </params>
  </procedure>
 </receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	procedure

データ処理クラスのパラメータ値の設定

タグ名	param

データ処理クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.Sa
  <procedure class="jp.co.intra_mart.system.sample.propagatio
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </procedure>
 </receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	○

設定値・設定する内容	任意のパラメータ値
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	params

【属性】

属性名	説明	必須	デフォルト値
key	パラメータ値を取得するためのキー。	○	なし

IM-Propagation 送信側設定

項目

- 概要
- リファレンス
  - 送信データ定義の設定
  - 送信側データ変換クラスの設定
  - データ変換クラスパラメータの設定
  - データ変換クラスのパラメータ値の設定
  - 必須呼出データ処理クラスの設定

概要

IM-Propagation を利用してデータを伝搬するための送信側を設定しま
送信するデータを格納するクラスと、データ変換クラスを紐付けるため

IM-Propagation を利用してデータを他モジュールへ伝搬するためには、

データを伝搬する際に他モジュールとの依存関係を切り離す必要がある
データの送信側は自モジュールで定義されている独自のモデルから、IM
そこで、データをコピーする作業を行うデータ変換クラスの設定を行い

モジュール	IM-Propagation
フォーマットファイル (xsd)	WEB-INF/schema/propagation-senders-c
設定場所	WEB-INF/conf/propagation-senders-conf

注意

ファイル名は、他のモジュールが提供しているものと重複しな

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-senders-config xmlns="http://www.intra-mart.jp/
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/propagation/send
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  <encoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </encoder>
  <required-procedure class="jp.co.intra_mart.system.sample.p
 </sender>
</propagation-senders-config>
```

リファレンス

送信データ定義の設定

タグ名	sender

IM-Propagation を使用してデータを送信する側が送信するデータの型

【設定項目】

```
<propagation-senders-config>
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  ・・・・・
 </sender>
</propagation-senders-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	propagation-senders-config

【属性】

属性名	説明
source	データ送信側が IM-Propagation に渡すクラスの す。
operationType	データ送信側が IM-Propagation に渡すデータの

コラム

operationType 属性には IM-Propagation によりいくつかの
自モジュール内で完結する伝搬処理の場合は、独自の operati

注意

source 属性に定義されていないクラスが IM-Propagation に
送られるデータの定義は必ず設定ファイルに記述してくださ

送信側データ変換クラスの設定

タグ名	encoder

送信側のデータを IM-Propagation 用のデータに変換するための、デー

【設定項目】

```
<propagation-senders-config>
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  <encoder class="jp.co.intra_mart.system.sample.propagation.S
 </sender>
</propagation-senders-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	sender

【属性】

属性名	説明	必須	デフォルト
class	データ変換クラスの完全修飾クラス名です。	○	なし

注意

class 属性に指定したクラスが見つからない場合や適切なイン

データ変換クラスパラメータの設定

タグ名	params

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-senders-config>
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  <encoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    ・・・・・
   </params>
  </encoder>
 </sender>
</propagation-senders-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	encoder

データ変換クラスのパラメータ値の設定

タグ名	param

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-senders-config>
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  <encoder class="jp.co.intra_mart.system.sample.propagation.S
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </encoder>
 </sender>
</propagation-senders-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	任意のパラメータ値
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	params

【属性】

属性名	説明	必須	デフォルト値
key	パラメータ値を取得するためのキー。	○	なし

必須呼出データ処理クラスの設定

タグ名	required-procedure

IM-Propagation でデータを必ず受け取って欲しい受信側のデータ処理

【設定項目】

```
<propagation-senders-config>
 <sender source="jp.co.intra_mart.foundation.sample.model.Sam
  <required-procedure class="jp.co.intra_mart.system.sample.p
 </sender>
</propagation-senders-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	sender

【属性】

属性名	説明
class	データ受信側のデータ処理クラスの完全修飾クラス名です。

注意

class 属性に指定したクラスに対してデータが送信されなかっ

テナント管理機能

システム管理者用スクリプト開発モデルルーテ

項目
- 概要
- リファレンス
  - ファイルマッピング設定
  - フォルダマッピング設定

概要

システム管理者用のURLに対して、スクリプト開発モデルの処理のマッ

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/admin-routing-jssp-con
設定場所	WEB-INF/conf/admin-routing-jssp-config/

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-jssp-config
   xmlns="http://www.intra-mart.jp/router/admin/admin-routing-jss
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/admin/adm
 <file-mapping path="/system/application/main" page="system/a
 <file-mapping path="/system/application/list" page="system/ap
 <file-mapping path="/system/application/process/create" page=
 <file-mapping path="/system/application/process/update" page=
 <file-mapping path="/system/application/process/delete" page=
</admin-routing-jssp-config>
```

リファレンス

ファイルマッピング設定

タグ名	file-mapping

URLとスクリプト開発モデルのプログラムのマッピングを行います。

【設定項目】

```
<admin-routing-jssp-config>
 <file-mapping path="/system/application/main" page="system/a
 <file-mapping path="/system/application/list" page="system/ap
</admin-routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	admin-routing-jssp-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定する 値に {＜識別子＞} を記述することでURLの のプログラム中で使用可能です。 例： /sample/view/{dataId}
page	マッピングを行うスクリプト開発モデルのプ
action	page属性に指定されたプログラムの実行前に from属性を指定した場合はfrom属性に指定 from属性が未指定の場合はpage属性に指定 す。
from	action属性で指定した関数を呼び出すプロク

> **コラム**
>
> page, from 属性で指定するスクリプト開発モデルのプログラ
> スクリプト開発モデルプログラムのソースディレクトリからの
> デフォルトではWEB-INF/jssp/srcからの相対パス形式になりま

フォルダマッピング設定

タグ名	folder-mapping

URLのパターンとスクリプト開発モデルのプログラムのマッピングを行
特定のスクリプト開発モデルのプログラムのディレクトリ配下すべての

【設定項目】

```
<admin-routing-jssp-config>
  <folder-mapping path-prefix="/system/application" folder="sys
</admin-routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-jssp-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
folder	マッピングを行うスクリプト開発モデルのプログラ します。

> **注意**
>
> この設定を使用するとルーティングに時間が掛かる、スクリプ
> 奨されません。
> 使用するスクリプト開発モデルのプログラムに対してfile-map

システム管理者用IM-JavaEE Frameworkルー

項目

- 概要
- リファレンス
  - サービスマッピング設定
  - アプリケーションマッピング設定

概要

システム管理者用のURLに対して、IM-JavaEE Frameworkの処理のマ

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/admin-routing-service
設定場所	WEB-INF/conf/admin-routing-service-co

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-service-config
   xmlns="http://www.intra-mart.jp/router/admin-routing-service-c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/admin-rout
 <service-mapping path="/system/application/main" application=
</admin-routing-service-config>
```

リファレンス

サービスマッピング設定

タグ名	service-mapping

URLとIM-JavaEE Frameworkのプログラムのマッピングを行います。

【設定項目】

```
<admin-routing-service-config>
 <service-mapping path="/system/application/main" application=
</admin-routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-service-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定するこ 値に {<識別子>} を記述することでURLの途中 で取得可能です。 例： /sample/view/{dataId}
application	マッピングを行うIM-JavaEE Frameworkのア
service	マッピングを行うIM-JavaEE Frameworkのサー

アプリケーションマッピング設定

タグ名	application-mapping; admin-routing-service-config

URLのパターンとIM-JavaEE Frameworkのアプリケーションのマッピ
特定のIM-JavaEE Frameworkのアプリケーションが持つすべてのサー

【設定項目】

```
<admin-routing-service-config>
  <application-mapping path-prefix="/system/application" applic
</admin-routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-service-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
application	マッピングを行うIM-JavaEE Frameworkのプロク 定します。

> **注意**
>
> この設定を使用するとルーティングに時間が掛かる、IM-Javal
> されません。
> 使用するIM-JavaEE Frameworkのサービスに対してservice-

システム管理者用サーブレットルーティング設

項目
- 概要
- リファレンス
  - サーブレットマッピング設定

概要

システム管理者用のURLに対して、フォワード先のサーブレットパスへ

モジュール	テナント管理機能

フォーマットファイル(xsd)	WEB-INF/schema/admin-routing-servlet-
設定場所	WEB-INF/conf/admin-routing-servlet-cor

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-servlet-config
   xmlns="http://www.intra-mart.jp/router/admin-routing-servlet-co
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/admin-rout
 <servlet-mapping path="/system/application/main" servlet="ap
</admin-routing-servlet-config>
```

リファレンス

サーブレットマッピング設定

タグ名	servlet-mapping

URLとサーブレットパスのマッピングを行います。

【設定項目】

```
<admin-routing-servlet-config>
 <servlet-mapping path="/user/application/main" servlet="appli
</admin-routing-servlet-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-servlet-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード `「*」` を指定するこ 値に `{＜識別子＞}` を記述することでURLの途中 HttpServletRequest#getAttribute()で取得可 例： `/sample/view/{dataId}`
servlet	マッピングを行うサーブレットパスを指定します

認可判断モジュール設定

項目
- 概要
- リファレンス
  - 認可判断に使用する認可判断モジュールの構成
  - 使用する実装クラスの設定

概要

認可判断に使用する認可判断モジュールの構成と、結果を取りまとめる
この設定による動作の詳細については認可仕様書も合わせて参照してく

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-decision-config.xsd
設定場所	WEB-INF/conf/authz-decision-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decisi
  <decision-config name="default" combinator="permit-override
    <module class="jp.co.intra_mart.foundation.authz.services.de
    <module class="jp.co.intra_mart.foundation.authz.services.de
    <module class="jp.co.intra_mart.foundation.authz.services.de
  </decision-config>
</authz-decision-config>
```

リファレンス

認可判断に使用する認可判断モジュールの構成

タグ名	decision-config

子要素に認可判断に使用するモジュールを <module> タグとして列挙
認可判断に使用するモジュールの列挙と、複数のモジュールの下した判

【設定項目】

```
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decisi
  <decision-config name="default" combinator="permit-override
    ....
  </decision-config>
</authz-decision-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-decision-config

【属性】

属性名	説明
name	この設定の名前です。将来の拡張のために

属性名	説明
combinator	子要素に指定する情報 <module> タグに NotApplicable)。その結果をどのように設定できる値は以下の3種類です • permit-overrides ◦ 上から順に評価して最初に Perm • deny-overrides ◦ 上から順に評価して最初に Deny • first-applicable ◦ 上から順に評価して最初に Perm

使用する実装クラスの設定

タグ名	module

使用する認可判断モジュールのクラス名を設定します。認可判断モジュ

> **注意**
> 認可判断機能は<module>タグの記述された順番にモジュール

【設定項目】

```
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decisi
  <decision-config name="default" combinator="permit-override
    <module class="jp.co.intra_mart.foundation.authz.services.de
  </decision-config>
</authz-decision-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	decision-config

【属性】

属性名	説明
class	認可判断モジュールとして使用するクラスの完全修飾定します。

認可設定画面設定

項目

- 概要
- リファレンス
  - 認可ポリシー更新処理設定
  - キャッシュ更新エラー設定（認可ポリシー）
  - 認可リソースグループ更新処理設定
  - キャッシュ更新エラー設定（認可リソースグループ）
  - リソースグループバックアップ設定
  - バックアップ先パス設定
  - バックアップ最大件数設定
  - Excelインポート設定
  - Excelインポート処理結果配置パス設定
  - Excelインポートコミット間隔設定
  - 未設定認可ポリシー検証設定
  - リソースグループ・リソース存在検証設定
  - サブジェクトグループ存在検証設定
  - Excelインポート処理の処理結果ファイル数設定
  - Excelエクスポート設定
  - テンプレートファイル設定
  - Excelエクスポートファイル配置パス設定
  - Excelエクスポートファイル名設定
  - Excelエクスポートファイル拡張子設定
  - Excelエクスポート保存ファイル数設定
  - 保護パスワード設定

概要

認可設定画面の設定を行います。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-editor-config.xsd
設定場所	WEB-INF/conf/authz-editor-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-editor-config
   xmlns="http://www.intra-mart.jp/authz/authz-editor-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-edito
 <policy-update>
   <cache-update-error rollback="true" />
 </policy-update>
 <resource-group-update>
   <cache-update-error rollback="true" />
 </resource-group-update>
 <resource-group-backup>
   <path>im_authz/backup</path>
   <max-size>10</max-size>
 </resource-group-backup>
 <xlsx-import>
   <path>im_authz/xlsx_import</path>
   <commit-count>0</commit-count>
   <validate-inherits>false</validate-inherits>
   <validate-resource-exists>true</validate-resource-exists>
   <validate-subject-exists>false</validate-subject-exists>
   <max-size>10</max-size>
 </xlsx-import>
 <xlsx-export>
   <template-file>products/im_authz_xlsx_import_export/authz_t
   <path>im_authz/xlsx_export</path>
   <file-name>yyyyMMdd_HHmmss</file-name>
   <file-ext>.xlsx</file-ext>
   <max-size>10</max-size>
   <xlsx-password>intramart</xlsx-password>
 </xlsx-export>
</authz-editor-config>
```

リファレンス

認可ポリシー更新処理設定

タグ名	policy-update

認可ポリシー更新時の処理の設定を行います。

【設定項目】

```
<authz-editor-config>
 <policy-update>
  ...
 </policy-update>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

キャッシュ更新エラー設定（認可ポリシー）

タグ名	cache-update-error

認可ポリシーのキャッシュ更新によるエラー時の挙動について設定しま

【設定項目】

```
<authz-editor-config>
 <policy-update>
  <cache-update-error rollback="true" />
 </policy-update>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし

親タグ	policy-update

【属性】

属性名	説明
rollback	キャッシュ更新失敗し認可ポリシーの更新をロール 定します。

認可リソースグループ更新処理設定

タグ名	resource-group-update

認可リソースグループ更新時の処理の設定を行います。

【設定項目】

```
<authz-editor-config>
 <resource-group-update>
  ...
 </resource-group-update>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

キャッシュ更新エラー設定（認可リソースグループ）

タグ名	cache-update-error

認可リソースグループのキャッシュ更新によるエラー時の挙動について

【設定項目】

```
<authz-editor-config>
 <resource-group-update>
  <cache-update-error rollback="true" />
 </resource-group-update>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	resource-group-update

【属性】

属性名	説明
rollback	キャッシュ更新失敗し認可リソースグループの更新を設定します。

リソースグループバックアップ設定

タグ名	resource-group-backup

リソースグループと、紐づくリソースを削除した場合に、関係する認可

【設定項目】

```
<authz-editor-config>
 <resource-group-backup>
  ...
 </resource-group-backup>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

バックアップ先パス設定

タグ名	path

バックアップした認可設定のファイルを保存する場所について設定しま
パブリックストレージのルートからの相対パスです。

【設定項目】

```
<authz-editor-config>
 <resource-group-backup>
  <path>im_authz/backup</path>
 </resource-group-backup>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	パブリックストレージのルートからの相対パスを指定
単位・型	文字列

省略時のデフォルト値	im_authz/backup
親タグ	resource-group-backup

バックアップ最大件数設定

タグ名	max-size

バックアップとして認可設定のファイルを保存する最大件数について設
0 を指定した場合、保存件数は無制限になります。この場合、パブリッ

【設定項目】

```
<authz-editor-config>
 <resource-group-backup>
  <max-size>30</max-size>
 </resource-group-backup>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	バックアップを残す最大件数を指定します。
単位・型	数値
省略時のデフォルト値	30
親タグ	resource-group-backup

Excelインポート設定

タグ名	xlsx-import

認可設定画面上からExcelファイルを用いて認可設定のインポートを行う

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  ...
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

Excelインポート処理結果配置パス設定

タグ名	path

Excelインポート処理の処理結果が保存されているJSON形式のファイル
ファイルはパブリックストレージに配置されるため、パブリックストレ
認可設定画面にてExcelインポートを行う際にアップロードを行ったExc

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <path>im_authz/xlsx_import</path>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Excelインポート処理結果ファイルを格納す す。
単位・型	文字列
省略時のデフォルト値	im_authz/xlsx_import
親タグ	xlsx-import

Excelインポートコミット間隔設定

タグ名	commit-count

インポート処理で、認可ポリシー変更（追加・更新・削除を指します）
例えば、10 を指定した場合は 10 件の認可ポリシーを変更する度にデー
0 以下を指定した場合、インポート処理が完了するまでコミットが行わ
インポート処理にてコミットを行う順序については、「IM-Authz（認可

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <commit-count>0</commit-count>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	コミットを行うまでのポリシー変更件数
単位・型	数値
省略時のデフォルト値	0
親タグ	xlsx-import

未設定認可ポリシー検証設定

タグ名	validate-inherits

インポートファイルにて明示的に「許可」／「禁止」を設定されていな
検証は、インポートファイルの編集者のミスなどにより意図しない認可
検証を行う場合、「許可」／「禁止」以外が設定されている認可ポリシ

- セルの内容が「未設定（継承した結果、許可）」であり、セルの上
- セルの内容が「未設定（継承した結果、禁止）」であり、セルの上

未設定認可ポリシー検証設定によるインポート時の挙動の違いについて

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <validate-inherits>false</validate-inherits>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	未設定である認可ポリシーセルの検証を行うかどう
単位・型	真偽値（true/false）
省略時のデフォルト値	false
親タグ	xlsx-import

リソースグループ・リソース存在検証設定

タグ名	validate-resource-exists

インポートするExcel（xlsx）ファイルに定義されたリソースグループ・
検証を行う場合、存在しないリソースグループ・リソースが定義されて

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <validate-resource-exists>true</validate-resource-exists>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	リソースグループ・リソース存在検証を行うかどう
単位・型	真偽値（true/false）
省略時のデフォルト値	true
親タグ	xlsx-import

サブジェクトグループ存在検証設定

タグ名	validate-subject-exists

インポートするExcel（xlsx）ファイルに定義されたサブジェクトグル
存在しないサブジェクトグループが定義されていた場合、エラー扱いと

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <validate-subject-exists>false</validate-subject-exists>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	サブジェクトグループを行うかどうかを設定します。
単位・型	真偽値（true/false）
省略時のデフォルト値	false
親タグ	xlsx-import

Excelインポート処理の処理結果ファイル数設定

タグ名	max-size

Excelインポート処理の処理結果ファイルを保存するファイル数を設定し
Excelインポート処理の処理結果ファイルの数が設定値を超えた場合、古
認可設定画面のインポート結果一覧に表示される結果数はこの設定で指
0 以下を指定した場合、保存件数は無制限になります。この場合、パブ

【設定項目】

```
<authz-editor-config>
 <xlsx-import>
  <max-size>10</max-size>
 </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Excelインポート処理の処理結果のファイル数
単位・型	数値
省略時のデフォルト値	10
親タグ	xlsx-import

Excelエクスポート設定

タグ名	xlsx-export

認可設定画面上からExcelファイルを用いて認可設定のエクスポートを行

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  ...
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	authz-editor-config

テンプレートファイル設定

タグ名	template-file

エクスポートするExcel（xlsx）ファイルを作成するための、テンプレー

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <template-file>products/im_authz_xlsx_import_export/authz_t
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	エクスポートするExcel（xlsx）ファ します。
単位・型	文字列
省略時のデフォルト値	products/im_authz_xlsx_import_
親タグ	xlsx-export

Excelエクスポートファイル配置パス設定

タグ名	path

Excelエクスポートを行い出力されるファイルを配置する場所を設定しま
ファイルはパブリックストレージに配置されるため、パブリックストレ

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <path>im_authz/xlsx_export</path>
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Excelエクスポートされたファイルを格納す す。
単位・型	文字列
省略時のデフォルト値	im_auhtz/xlsx
親タグ	xlsx-export

Excelエクスポートファイル名設定

タグ名	file-name

Excelエクスポートされたファイルのファイル名を設定します。
ファイル名には、日付と時刻のフォーマットパターンを利用することが
SimpleDateFormat のJavaDocを参照してください。

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <file-name>yyyyMMdd_HHmmss</file-name>
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	エクスポートするExcelファイルのファイル名を設定
単位・型	文字列
省略時のデフォルト値	yyyyMMdd_HHmmss
親タグ	xlsx-export

Excelエクスポートファイル拡張子設定

タグ名	file-ext

Excelエクスポートされたファイルの拡張子を設定します。
設定値に置換文字列が利用可能です。利用可能な置換文字列は以下の通

- `{^userCd^}`：出力操作を行ったユーザコード
- `{^partId^}`：出力した part-id

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <file-ext>.xlsx</file-ext>
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	エクスポートするExcelファイルの拡張子を設定しま
単位・型	文字列
省略時のデフォルト値	.xlsx
親タグ	xlsx-export

Excelエクスポート保存ファイル数設定

タグ名	max-size

Excelエクスポートを行い出力されるファイルを保存するファイル数を設
Excelエクスポートを行い出力されるファイルの数が設定値を超えた場合
認可設定画面のエクスポート済みファイル一覧に表示される結果数はこ
0 以下を指定した場合、保存件数は無制限になります。この場合、パブ

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <max-size>10</max-size>
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Excelエクスポートを行い出力されるファイルの数が
単位・型	数値
省略時のデフォルト値	10
親タグ	xlsx-export

保護パスワード設定

タグ名	xlsx-password

エクスポートするExcel（xlsx）ファイルの変更不可部分を保護するパス
指定しない場合、保護パスワードはかからず行・列の追加・更新・削除

【設定項目】

```
<authz-editor-config>
 <xlsx-export>
  <xlsx-password>intramart</xlsx-password>
 </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	保護するパスワードの値を設定します。
単位・型	文字列
省略時のデフォルト値	intramart
親タグ	xlsx-export

ポリシー部分編集定義設定

項目

- 概要
- リファレンス
  - 認可設定画面の部品化設定
  - 設定ID
  - 設定のキャプション
  - 対象リソースグループ
  - リソースグループ
  - 対象サブジェクトタイプ
  - サブジェクトタイプ
  - コールバック設定
  - リソースグループ権限設定可否判断クラス

概要

認可設定画面を部品として使用する際に必要になる設定を記述します。

この設定を使用する前に、この設定に記載するリソースグループセット

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-partial-policy-e
設定場所	WEB-INF/conf/authz-partial-policy-edit-

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <part-id>im_authz_impl_router</part-id>
  <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</c
  <resource-groups>
   <resource-group-id>http-services</resource-group-id>
  </resource-groups>
  <subject-types>
   <subject-type-id>im_authz_meta_subject</subject-type-id
   <subject-type-id>imm_user</subject-type-id>
   <subject-type-id>imm_company_post</subject-type-id>
   <subject-type-id>imm_department</subject-type-id>
   <subject-type-id>imm_public_grp</subject-type-id>
   <subject-type-id>imm_public_grp_role</subject-type-id>
   <subject-type-id>b_m_role</subject-type-id>
  </subject-types>
  <callbacks>
   <resource-group-authorizer>jp.co.intra_mart.system.authz
  </callbacks>
 </part-config>
</authz-partial-policy-edit-config>
```

リファレンス

認可設定画面の部品化設定

タグ名	part-config

認可設定画面を部品として使用するための設定項目のひとまとまりです。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  ...
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-partial-policy-edit-config

設定ID

タグ名	part-id

設定のIDを指定します。この値は部品化された認可設定画面を使用する
先頭にモジュールのIDを使用するなどし、他の設定と重複しないようID

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <part-id>im_authz_impl_router</part-id>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	設定のIDを指定します。他の設定と重複しないよ
単位・型	文字列 (半角英数、ハイフン、アンダースコア)
省略時のデフォルト値	なし
親タグ	part-config

設定のキャプション

タグ名	caption-cd

この設定の表示名を取得するためのメッセージコードを設定します。こ

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</c
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	キャプションのメッセージコード
単位・型	文字列
省略時のデフォルト値	なし
親タグ	part-config

対象リソースグループ

タグ名	resource-groups

この設定で使用するリソースグループを指定します。

認可設定画面ではここで指定されたリソースグループ配下を表示するた
定した場合、それらのグループ配下がマージして表示されます。 複数の

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <resource-groups>
   ...
  </resource-groups>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	part-config

リソースグループ

タグ名	resource-group-id

リソースグループIDを指定します。リソースグループIDは動的に作成さ

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <resource-groups>
   <resource-group-id>http-services</resource-group-id>
  </resource-groups>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	使用するリソースグループID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	resource-groups

対象サブジェクトタイプ

タグ名	subject-types

この設定で使用するサブジェクトタイプを指定します。

ここで指定していないサブジェクトタイプは、この設定を使用して開い
るサブジェクトタイプのIDでなければなりません。

【設定項目】

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <subject-types>
   ...
  </subject-types>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	part-config

サブジェクトタイプ

タグ名	subject-type-id

使用するサブジェクトタイプのIDを設定します。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <subject-types>
   <subject-type-id>b_m_role</subject-type-id>
  </subject-types>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	この設定で使用するサブジェクトタイプID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	subject-types

コールバック設定

タグ名	callbacks

この設定によって認可設定画面を部品として使用した際に利用されるコ

認可設定画面を部品として使用する場合は呼び出し元が認可設定画面に

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <callbacks>
   ...
  </callbacks>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	part-config

リソースグループ権限設定可否判断クラス

タグ名	resource-group-authorizer

この設定によって認可設定画面を部品として使用した際に利用されるリ
jp.co.intra_mart.foundation.authz.partial.AuthzPartialResourceGr

認可設定画面を部品として使用する場合は呼び出し元が認可設定画面に
ループが指定された場合に、認証済みユーザの権限を超えて権限設定が
かどうか判断します。

この設定は省略可能です。 省略した場合はデフォルトの判断クラスが使

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-partial-p
 <part-config>
  <callbacks>
   <resource-group-authorizer>jp.co.intra_mart.system.authz.
  </callbacks>
 </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.authz.p 飾名。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	callbacks

認可設定画面 保護リソース設定

項目

- 概要
- リファレンス
  - 保護するリソースとアクションの指定

概要

この設定は、認可設定画面上での変更を保護するリソースの一覧です。

操作中のユーザが、この設定に記載されているリソース・アクションに
中のユーザに対してこれらの権限すべてを失わせる操作を行った場合、

これは認可設定の変更中に自身で操作の続行ができなくなるような変更

モジュール	テナント管理
フォーマットファイル (xsd)	WEB-INF/schema/authz-protected-reso
設定場所	WEB-INF/conf/authz-protected-resource

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:authz-protected-resource-config
 xmlns="http://www.intra-mart.jp/authz/ui/protected-resources"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra-mart.jp/authz/ui/protected-

 <protected-resource uri="service://authz/settings/basic" action
 <protected-resource uri="service://authz/settings/parts" action
 <protected-resource uri="service://authz/settings/procedure" a

</tns:authz-protected-resource-config>
```

リファレンス

保護するリソースとアクションの指定

タグ名	protected-resource

保護するリソースとアクションを指定します。

【設定項目】

```
<tns:authz-protected-resource-config
 xmlns="http://www.intra-mart.jp/authz/ui/protected-resources"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra-mart.jp/authz/ui/protected-

 <protected-resource uri="service://authz/settings/basic" action

</tns:authz-protected-resource-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-protected-resource-config

【属性】

属性名	説明
uri	保護する対象のリソースのURIを指定します。デー要があります。
action	uriに対して保護するアクションを指定します。

> **注意**
>
> この設定は通常変更の必要はありません。

ルーティングテーブル用 認可リソースマッパー

項目
- 概要
- リファレンス
  - リソースマッパー設定

概要

ルーティングテーブルの認可設定で使用するマッパーの登録を行う設定

ルータにはそれぞれのパスに対してマッピングされるプログラムの実体

認可リソースの紐づけに関して、パスに対して特定のリソースを指定す
ルによって 登録されたものから選択できるようになっています。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-resource-mappe
設定場所	WEB-INF/conf/authz-resource-mappers/{

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-mappers xmlns="http://www.intra-mart.jp/rout
xsi:schemaLocation="http://www.intra-mart.jp/router/authz-resourc
  <mapper name="dev-auto-register" class="jp.co.intra_mart.syst
  <mapper name="welcome-all" class="jp.co.intra_mart.system.ro
</authz-resource-mappers>
```

リファレンス

リソースマッパー設定

タグ名	mapper

マッパーの名前と実際にマッピングを行う実装クラスを設定します。
マッパーの実装は jp.co.intra_mart.system.router.authz.user.Authz

【設定項目】

```
<authz-resource-mappers xmlns="http://www.intra-mart.jp/rout
xsi:schemaLocation="http://www.intra-mart.jp/router/authz-resourc
  <mapper name="welcome-all" class="jp.co.intra_mart.system.ro
</authz-resource-mappers>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし

単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-resource-mappers

【属性】

属性名	説明
name	マッパーの名前です。ルーティングテーブルの のはここで指定した値です。
class	この名前に割り当てるマッパーの実装クラスの

リソースタイプ拡張設定

項目
- 概要
- リファレンス
  - リソースタイプ設定

概要

認可機構にリソースタイプを追加するための設定です。
リソースタイプと、リソースタイプの取り扱うリソースモデルを定義し
リソースタイプの詳細については認可仕様書を参照してください。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/authz-resource-type-c
設定場所	WEB-INF/conf/authz-resource-type-confi

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-type-config xmlns="http://www.intra-mart.jp/a
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-resourc
 <resource-type
   type-class="jp.co.intra_mart.system.router.authz.resourcetype.G
   model-class="jp.co.intra_mart.system.router.authz.resourcetype
   cache-class="jp.co.intra_mart.system.router.authz.resourcetype.
</authz-resource-type-config>
```

リファレンス

リソースタイプ設定

タグ名	resource-type

リソースタイプの実装クラスと、この実装クラスが扱うリソースモデル

【設定項目】

```
<authz-resource-type-config xmlns="http://www.intra-mart.jp/a
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-resourc
 <resource-type
   type-class="jp.co.intra_mart.system.router.authz.resourcetype.G
   model-class="jp.co.intra_mart.system.router.authz.resourcetype
   cache-class="jp.co.intra_mart.system.router.authz.resourcetype.
</authz-resource-type-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-resource-type-config

【属性】

属性名	説明
type-class	リソースタイプの実装クラスの完全修飾クラス名を
model-class	このリソースタイプで扱うリソースモデルクラスの す。
cache-class	このリソースタイプで扱うキャッシュコントロー 定します。

サブジェクトタイプ拡張設定

項目
- 概要
- リファレンス
  - サブジェクトタイプ設定

概要

認可機構にサブジェクトタイプを追加するための設定です。
サブジェクトタイプと、サブジェクトタイプの取り扱うサブジェクトモ
サブジェクトタイプの詳細については認可仕様書を参照してください。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-subject-type-con
設定場所	WEB-INF/conf/authz-subject-type-config/

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-subject-type-config xmlns="http://www.intra-mart.jp/au
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-subject-
  <subject-type type-class="jp.co.intra_mart.foundation.authz.sub
    model-class="jp.co.intra_mart.foundation.admin.role.model.Role
</authz-subject-type-config>
```

リファレンス

サブジェクトタイプ設定

タグ名	subject-type

サブジェクトタイプの実装クラスと、この実装クラスが扱うサブジェク

【設定項目】

```
<authz-subject-type-config xmlns="http://www.intra-mart.jp/au
xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-subject-
  <subject-type type-class="jp.co.intra_mart.foundation.authz.sub
    model-class="jp.co.intra_mart.foundation.admin.role.model.Role
</authz-subject-type-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-subject-type-config

【属性】

属性名	説明
type-class	サブジェクトタイプの実装クラスの完全修飾クラス
model-class	このサブジェクトタイプで扱うサブジェクトモデル 定します。

暗号化設定

項目
- 概要
- リファレンス
  - 暗号化設定
  - 暗号化クラス設定

概要

Java API CryptionUtil を利用して、暗号／復号を行うための暗号化方式

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/cryption-config.xsd
設定場所	intra-mart Accel Platform 2013 Autumn 以降 WEB-INF/conf/cryption-config/cryption-c intra-mart Accel Platform 2013 Summer 以 WEB-INF/conf/cryption-config.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cryption-config xmlns="http://intra-mart.co.jp/system/security/c

  <!-- 該当するカテゴリがない場合使われます -->
  <cryption category="default" default="true">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption
  </cryption>

  <!-- アカウントパスワード用 -->
  <cryption category="account">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption
  </cryption>

  <!-- システム管理者パスワード用 -->
  <cryption category="administrator">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption
  </cryption>

</cryption-config>
```

リファレンス

暗号化設定

タグ名	cryption

暗号化方式の定義を行います。

【設定項目】

```
<cryption-config>
  <cryption category="default" default="true">
    ・・・・・
  </cryption>
</cryption-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	cryption-config

【属性】

属性名	説明

属性名	説明
category	暗号化設定にプログラムから利用するためのカテゴ 各カテゴリの役割は以下の通りです。 account　アカウントのパスワードに対す パスワードの保存方式が「暗号 パスワードの保存方式について てください。 administrator　システム管理者のパスワードに
default	true の場合、システム・デフォルトの暗号化設定に

> **注意**
> default=”true” の暗号化設定が複数ある場合、設定ファイル
> default=”true” の暗号化設定が見つからない場合、設定ファ

> **注意**
> カテゴリは一意な値となるように指定してください。

暗号化クラス設定

タグ名	cryption-class

暗号化方式の実装クラスの設定を行います。

【設定項目】

```
<cryption-config>
  <cryption category="default" default="true">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.
  </cryption>
</cryption-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	Javaインタフェース jp.co.intra_mart.fo す。 テナント環境セットアップを行った環境に 標準で用意されているクラスは以下の通り • jp.co.intra_mart.foundation.secur DESを用いて文字列の暗号化・復号化 • jp.co.intra_mart.foundation.secur トリプルDESを用いて文字列の暗号化 • jp.co.intra_mart.foundation.secur AESを用いて文字列の暗号化・復号化 intra-mart Accel Platform 2016 Sp
単位・型	文字列
省略時のデフォルト値	なし
親タグ	cryption

サブジェクトリゾルバ(DeclaredSubjectReso

項目

- 概要
- リファレンス
  - DeclaredSubjectResolverの追加設定

概要

認可のサブジェクト解決系を追加する設定です。ここではサブジェクト

サブジェクトリゾルバ、DeclaredSubjectResolverについての詳細は設

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/declared-subject-resol
設定場所	WEB-INF/conf/declared-subject-resolvers

```
<?xml version="1.0" encoding="UTF-8"?>
<declared-subject-resolvers xmlns="http://www.intra-mart.jp/au
xsi:schemaLocation="http://www.intra-mart.jp/authz/declared-subje
  <class-name>jp.co.intra_mart.foundation.authz.subjectresolver.i
</declared-subject-resolvers>
```

リファレンス

DeclaredSubjectResolverの追加設定

タグ名	class-name

認可機構に追加するDeclaredSubjectResolverを定義します。

【設定項目】

```
<declared-subject-resolvers xmlns="http://www.intra-mart.jp/au
xsi:schemaLocation="http://www.intra-mart.jp/authz/declared-subje
  <class-name>jp.co.intra_mart.foundation.authz.subjectresolver.i
</declared-subject-resolvers>
```

必須項目	×
複数設定	○
設定値・設定する内容	認可機構に追加するDeclaredSubjectResolverの
単位・型	文字列
省略時のデフォルト値	なし
親タグ	declared-subject-resolvers

サブジェクトリゾルバ(OnDemandSubjectRe

項目
- 概要
- リファレンス
  - OnDemandSubjectResolverの追加設定

概要

認可のサブジェクト解決系を追加する設定です。ここではサブジェクト

サブジェクトリゾルバ、OnDemandSubjectResolverについての詳細は

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/ondemand-subject-res
設定場所	WEB-INF/conf/ondemand-subject-resolv

```
<?xml version="1.0" encoding="UTF-8"?>
<ondemand-subject-resolvers xmlns="http://www.intra-mart.jp/
xsi:schemaLocation="http://www.intra-mart.jp/authz/ondemand-sul
 <class-name>sample.TimeDependResolver</class-name>
</ondemand-subject-resolvers>
```

リファレンス

OnDemandSubjectResolverの追加設定

タグ名	class-name

認可機構に追加するOnDemandSubjectResolverを定義します。

【設定項目】

```
<ondemand-subject-resolvers xmlns="http://www.intra-mart.jp/
xsi:schemaLocation="http://www.intra-mart.jp/authz/ondemand-sul
 <class-name>sample.TimeDependResolver</class-name>
</ondemand-subject-resolvers>
```

必須項目	×
複数設定	○
設定値・設定する内容	認可機構に追加するOnDemandSubjectResolv
単位・型	文字列
省略時のデフォルト値	なし
親タグ	ondemand-subject-resolvers

認可ポリシーキャッシュ対象設定

項目

- 概要
- リファレンス
  - キャッシュ対象設定
  - キャッシュ対象リソースタイプ設定

概要

認可ポリシー取得処理でのキャッシュ設定を行います。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/policy-cache-config.xsd
設定場所	WEB-INF/conf/policy-cache-config/{任意の

```
<?xml version="1.0" encoding="UTF-8"?>
<policy-cache-config
   xmlns="http://www.intra-mart.jp/authz/policy-cache-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/authz/policy-cach
 <cache-target>
   <resource-type>service</resource-type>
 </cache-target>
</policy-cache-config>
```

リファレンス

キャッシュ対象設定

タグ名	cache-target

キャッシュ対象を設定します。

【設定項目】

```
<policy-cache-config>
 <cache-target>
  ...
 </cache-target>
</policy-cache-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	policy-cache-config

キャッシュ対象リソースタイプ設定

タグ名	resource-type

認可ポリシーのキャッシュの対象となるリソースタイプを設定します。

【設定項目】

```
<policy-cache-config>
 <cache-target>
  <resource-type>service</resource-type>
 </cache-target>
</policy-cache-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	キャッシュ対象とするリソースタイプを です。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	cache-target

パスワード履歴管理設定

項目

- 概要
- リファレンス
  - デフォルト設定
  - テナントごとの設定
  - 初回ログイン時パスワード変更
  - パスワード有効日数
  - 管理するパスワード履歴数
  - パスワード履歴管理対象外クライアントタイプ
  - パスワード変更画面パス
  - パスワード文字列チェック
  - パスワード暗号化アルゴリズムクラス
  - パスワード変更後ログイン画面表示
  - パスワードの長さチェック
  - パスワード利用可能文字(英文字)
  - パスワード利用可能文字(数字)
  - パスワード利用可能文字(その他)
  - パスワード履歴内のパスワードの利用禁止
  - ユーザコードパスワードの利用禁止

概要

パスワードの履歴管理機能の設定です。
パスワード履歴管理機能は、パスワードの履歴管理を行い、パスワード
また、パスワードの様々な制約を設定することが可能です。

モジュール	テナント管理機能モジュール
フォーマットファイル(xsd)	WEB-INF/schema/password-history-config.xs
設定場所	WEB-INF/conf/password-history.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <change-password-first-login>true</change-password-fir
  <password-expire-limit>0</password-expire-limit>
  <password-history-count>0</password-history-count>
  <deny-client-types>sp</deny-client-types>
  <password-expire-page>/user/password/expire</password-
  <check-password enable="true">
   <check-password-length enable="false" min="0" max="50"
   <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQR
   <allow-number required="false">0123456789</allow-numb
   <allow-extra-char required="false">!&amp;quot;#$%&amp;amp;&amp;ap
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.securit
  <return-initial-page>false</return-initial-page>
 </group-default>

 <group name="secondary" accessor-class="jp.co.intra_mart.foun
  <change-password-first-login>true</change-password-fir
  <password-expire-limit>0</password-expire-limit>
  <password-history-count>0</password-history-count>
  <deny-client-types>sp</deny-client-types>
  <password-expire-page>/user/password/expire</password-
  <check-password enable="true">
   <check-password-length enable="false" min="0" max="50"
   <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQR
   <allow-number required="false">0123456789</allow-numb
   <allow-extra-char required="false">!&amp;quot;#$%&amp;amp;&amp;ap
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.securit
  <return-initial-page>false</return-initial-page>
 </group>

 <group name="thirdly" accessor-class="jp.co.intra_mart.foundat
  <change-password-first-login>true</change-password-fir
  <password-expire-limit>0</password-expire-limit>
  <password-history-count>0</password-history-count>
  <deny-client-types>sp</deny-client-types>
  <password-expire-page>/user/password/expire</password-
  <check-password enable="true">
   <check-password-length enable="false" min="0" max="50"
   <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQR
   <allow-number required="false">0123456789</allow-numb
   <allow-extra-char required="false">!&amp;quot;#$%&amp;amp;&amp;ap
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.securit
  <return-initial-page>false</return-initial-page>
 </group>
</password-history>
```

コラム

groupタグは複数設定が行えます。groupタグを複数定義する
詳細はリファレンス内の「テナントごとの設定」を参照してく

コラム

password-history.xmlはpassword-history-config.xsdを参
xmlファイル修正時は上記設定例を参照の上、実施してくださ

リファレンス

デフォルト設定

タグ名	group-default

パスワード履歴管理機能の標準設定を行います。

【設定項目】

```
<password-history>
 <group-default accessor-class="StandardPasswordHistoryAcces
  ...
 </group-default>
</password-history>
```

必須項目	○
複数設定	×
設定値・設定する内容	パスワード履歴管理機能の標準設定を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	password-history

【属性】

属性名	説明
accessor-class	パスワード履歴管理モジュールの実装クラスを設定しま jp.co.intra_mart.foundation.security.password.Pas インタフェースを実装している必要があります。

テナントごとの設定

タグ名	group

テナントごとに、パスワード履歴管理機能の設定を行います。

【設定項目】

```
<password-history>
 <group name="secondary" accessor-class="StandardPasswordH
  ...
 </group>
</password-history>
```

必須項目	×
複数設定	○
設定値・設定する内容	テナントごとの、パスワード履歴管理機能を設定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	password-history

【属性】

属性名	説明
name	テナントID
accessor-class	パスワード履歴管理モジュールの実装クラ jp.co.intra_mart.foundation.security.pa インタフェースを実装している必要があり

初回ログイン時パスワード変更

タグ名	change-password-first-login

初期ログイン時にパスワード変更を要求するかどうかを指定します。
ユーザ属性の初回ログイン設定の値をチェックして、初回ログイン判定

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <change-password-first-login>true</change-password-fir
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	false　初期ログイン時にパスワード変更を要求し true　初期ログイン時にパスワード変更を要求し
単位・型	真偽値 (true/false)
省略時のデフォルト値	親タグが group-default の場合、true 親タグが group の場合、group-default タグ内の設
親タグ	group-default, group

パスワード有効日数

タグ名	password-expire-limit

パスワードの有効日数を設定します。
0 を指定した場合、有効期間は無制限になります。

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <password-expire-limit>0</password-expire-limit>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードの有効日数を設定します。
単位・型	数値 (0 -)
省略時のデフォルト値	親タグが group-default の場合、0 親タグが group の場合、group-default タグ内の設定
親タグ	group-default, group

管理するパスワード履歴数

タグ名	password-history-count

パスワードの履歴を管理する世代。
0 を指定した場合、履歴管理を行いません。

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <password-history-count>0</password-history-count>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	管理するパスワード履歴数を設定します。

単位・型	数値 (0 -)
省略時のデフォルト値	親タグが group-default の場合、0 親タグが group の場合、group-default タグ内の設定
親タグ	group-default, group

パスワード履歴管理対象外クライアントタイプ

タグ名	deny-client-types

パスワード履歴管理対象とならないクライアントタイプを設定します。
複数指定する場合は、カンマ区切りで指定してください。

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <deny-client-types>sp</deny-client-types>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード履歴管理対象とならないクライアントタ
単位・型	文字列 (xxxxxx,xxxxxxxx)
省略時のデフォルト値	親タグが group-default の場合、なし 親タグが group の場合、group-default タグ内の
親タグ	group-default, group

パスワード変更画面パス

タグ名	password-expire-page

パスワード変更画面のパスを設定します。

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <password-expire-page>/user/password/expire</password-
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード変更画面のパスを設定します。
単位・型	文字列 (xxx/xxx/xxx/xxxxx)
省略時のデフォルト値	親タグが group-default の場合、/user/password/e 親タグが group の場合、group-default タグ内の設
親タグ	group-default, group

【属性】

属性名	説明
client-type	クライアントタイプを設定します。 クライアントタイプ別にパスワード変更画面 てください。 クライアントタイプが設定されていない場 変更画面として扱われます。

パスワード文字列チェック

タグ名	check-password

パスワード文字列に対する有効チェックの内容を設定します。

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <check-password enable="true">
   ...
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード文字列に対する有効チェックの内容を設定
単位・型	なし
省略時のデフォルト値	親タグが group-default の場合、なし 親タグが group の場合、group-default タグ内の設
親タグ	group-default, group

【属性】

属性名	説明
enable	パスワード文字列の有効チェックを行うかどうか します。

パスワード暗号化アルゴリズムクラス

タグ名	password-cryption-class

パスワードの履歴を保管する際に用いられる暗号化アルゴリズムクラス。
パスワードの保存方式が「暗号化」方式である場合に利用可能です。
パスワードの保存方式については「 システム管理者操作ガイド 」-「 パ

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <password-cryption-class>jp.co.intra_mart.foundation.securit
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.security.cry
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default の場合、 jp.co.intra_mart.foundation.security.cry 親タグが group の場合、group-default タ
親タグ	group-default, group

パスワード変更後ログイン画面表示

タグ名	return-initial-page

パスワード変更要求画面よりパスワードを変更した後に、ログイン画面

【設定項目】

```
<password-history>
 <group-default accessor-class="jp.co.intra_mart.foundation.sec
  <return-initial-page>false</return-initial-page>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	false　パスワード変更後にユーザのホーム画面を true　パスワード変更後にログイン画面を表示し
単位・型	真偽値 (true/false)
省略時のデフォルト値	親タグが group-default の場合、false 親タグが group の場合、group-default タグ内の設
親タグ	group-default, group

パスワードの長さチェック

タグ名	check-password-length

パスワードの長さ制約に関する設定です。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <check-password-length enable="false" min="0" max="50"
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードの長さ制約を設定します。
単位・型	なし
省略時のデフォルト値	親タグが group-default/check-password の場 親タグが group/check-password の場合、gr
親タグ	check-password

【属性】

属性名	説明
enable	パスワードの長さ制約を行うかどうかを設定
min	パスワードの最小の長さを設定します。最小
max	パスワードの最大の長さを設定します。最大

パスワード利用可能文字(英文字)

タグ名	allow-latin-letters

パスワードに利用できる英文字を設定します。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQ
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードに利用できる英文字を設定します。
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の 親タグが group/check-password の場合、gr
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(英文字)の制約を true の場合は、少なくとも 1 文字は設定し

パスワード利用可能文字(数字)

タグ名	allow-number

パスワードに利用できる数字を設定します。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <allow-number required="false">0123456789</allow-numb
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードに利用できる数字を設定します。
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の 親タグが group/check-password の場合、gr
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(数字)の制約を行 true の場合は、少なくとも 1 文字は設定し

パスワード利用可能文字(その他)

タグ名	allow-extra-char

パスワードに利用できるその他の文字を設定します。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <allow-extra-char required="false">!&amp;quot;#$%&amp;amp;&amp;ap
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードに利用できるその他の文字を設定し
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の 親タグが group/check-password の場合、gr
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(その他)の制約を true の場合は、少なくとも 1 文字は設定し

パスワード履歴内のパスワードの利用禁止

タグ名	deny-old-password

パスワード履歴に存在するパスワードの利用を禁止します。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <deny-old-password>false</deny-old-password>
  </check-password>
 <group-default>
<password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	false　パスワード履歴に存在するパスワード true　パスワード履歴に存在するパスワード
単位・型	真偽値 (true/false)
省略時のデフォルト値	親タグが group-default/check-password の 親タグが group/check-password の場合、gr
親タグ	check-password

ユーザコードパスワードの利用禁止

タグ名	deny-userid

ユーザコードと同じパスワードの利用を禁止します。

【設定項目】

```
<password-history>
 <group-default>
  <check-password enable="true">
   <deny-userid>false</deny-userid>
  </check-password>
 </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	false　ユーザコードと同じパスワードの利用 true　ユーザコードと同じパスワードの利用
単位・型	真偽値 (true/false)
省略時のデフォルト値	親タグが group-default/check-password の場 親タグが group/check-password の場合、gr
親タグ	check-password

スクリプト開発モデルルーティング設定

項目
- 概要
- リファレンス
  - デフォルト認可設定
  - ファイルマッピング設定
  - フォルダマッピング設定
  - 認可設定
  - 認可パラメータ

概要

URLに対して、スクリプト開発モデルの処理のマッピングを行うための

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/routing-jssp-config.xsd
設定場所	WEB-INF/conf/routing-jssp-config/{任意のフ

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-jssp-config
   xmlns="http://www.intra-mart.jp/router/routing-jssp-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/routing-jssp
 <authz-default uri="service://application/default" action="execu
 <file-mapping path="/user/application/main" page="user/applic
 <file-mapping path="/user/application/list" page="user/applicati
  <authz mapper="sample-mapper">
   <param key="foo" value="FOO" />
   <param key="bar" value="BAR" />
  </authz>
 </file-mapping>
 <file-mapping path="/user/application/process/create" page="u
  <authz uri="service://application/update" action="execute" />
 </file-mapping>
 <file-mapping path="/user/application/process/update" page="u
  <authz uri="service://application/update" action="execute" />
 </file-mapping>
 <file-mapping path="/user/application/process/delete" page="u
  <authz uri="service://application/update" action="execute" />
 </file-mapping>
</routing-jssp-config>
```

リファレンス

デフォルト認可設定

タグ名	authz-default

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-jssp-config>
 <authz-default uri="service://application/default" action="execu
</routing-jssp-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はauthz-resource-mapper設定で設定済 です。

> **注意**
> この設定は、以下のいずれかの設定を行う必要があります。
> uri属性とaction属性
> mapper属性

> **注意**
> authz-defaultタグを省略した場合、file-mappingタグ、fold

ファイルマッピング設定

タグ名	file-mapping

URLとスクリプト開発モデルのプログラムのマッピングを行います。

【設定項目】

```
<routing-jssp-config>
 <authz-default uri="service://application/default" action="execu
 <file-mapping path="/user/application/main" page="user/applic
 <file-mapping path="/user/application/list" page="user/applicati
  ...
 </file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード「*」 を指定す 値に {＜識別子＞} を記述することでURLの ログラム中で使用可能です。 例： /sample/view/{dataId}
page	マッピングを行うスクリプト開発モデルの
action	page属性に指定されたプログラムの実行前 from属性を指定した場合はfrom属性に指定 from属性が未指定の場合はpage属性に指定
from	action属性で指定した関数を呼び出すプロ
client-type	マッピングが有効となるクライアントタイ 省略時は、すべてのクライアントタイプにお

コラム

page, from 属性で指定するスクリプト開発モデルのプログラ
スクリプト開発モデルプログラムのソースディレクトリからの
デフォルトではWEB-INF/jssp/srcからの相対パス形式になりま

フォルダマッピング設定

タグ名	folder-mapping

URLのパターンとスクリプト開発モデルのプログラムのマッピングを行
特定のスクリプト開発モデルのプログラムのディレクトリ配下すべての

【設定項目】

```
<routing-jssp-config>
 <folder-mapping path-prefix="/user/application" folder="user/a
  ...
 </folder-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
folder	マッピングを行うスクリプト開発モデルのプログラ します。
client-type	マッピングが有効となるクライアントタイプを指定 省略時は、すべてのクライアントタイプにおいて有 ます。

> 注意
>
> この設定を使用するとルーティングに時間が掛かる、スクリプ
> 奨されません。
> 使用するスクリプト開発モデルのプログラムに対してfile-map

認可設定

タグ名	authz

認可設定

【設定項目】

```
<routing-jssp-config>
  <file-mapping path="/user/application/list" page="user/applicati
    <authz mapper="sample-mapper">
      ...
    </authz>
  </file-mapping>
  <file-mapping path="/user/application/process/create" page="u
    <authz uri="service://application/update" action="execute" />
  </file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	file-mapping folder-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパーの値です。

コラム

authz-defaultタグが指定されていて、かつ、file-mappingタ

注意

この設定は、以下のいずれかの設定を行う必要があります。
uri属性とaction属性
mapper属性

認可パラメータ

タグ名	param

認可リソースマッパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-jssp-config>
 <file-mapping path="/user/application/list" page="user/applicati
  <authz mapper="sample-mapper">
   <param key="foo" value="FOO" />
   <param key="bar" value="BAR" />
  </authz>
 </file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし

親タグ	authz
	authz-default

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

IM-JavaEE Frameworkルーティング設定

項目
- 概要
- リファレンス
  - デフォルト認可設定
  - サービスマッピング設定
  - アプリケーションマッピング設定
  - 認可設定
  - 認可パラメータ

概要

URLに対して、IM-JavaEE Frameworkの処理のマッピングを行うため

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/routing-service-config.x
設定場所	WEB-INF/conf/routing-service-config/{任意

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-service-config
   xmlns="http://www.intra-mart.jp/router/routing-service-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/routing-ser
 <authz-default uri="service://application/default" action="execu
 <service-mapping path="/user/application/main" application="a
 <service-mapping path="/user/application/list" application="ap
   <authz mapper="sample-mapper">
     <param key="foo" value="FOO" />
     <param key="bar" value="BAR" />
   </authz>
 </service-mapping>
 <service-mapping path="/user/application/process/create" appli
   <authz uri="service://application/update" action="execute" />
 </service-mapping>
 <service-mapping path="/user/application/process/update" app
   <authz uri="service://application/update" action="execute" />
 </service-mapping>
 <service-mapping path="/user/application/process/delete" appli
   <authz uri="service://application/update" action="execute" />
 </service-mapping>
</routing-service-config>
```

リファレンス

デフォルト認可設定

タグ名	authz-default

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-service-config>
  <authz-default uri="service://application/default" action="execu
</routing-service-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー の値です。

注意

この設定は、以下のいずれかの設定を行う必要があります。
- uri属性とaction属性
- mapper属性

注意

authz-defaultタグを省略した場合、service-mappingタグ、

サービスマッピング設定

タグ名	service-mapping

URLとIM-JavaEE Frameworkのプログラムのマッピングを行います。

【設定項目】

```
<routing-service-config>
  <authz-default uri="service://application/default" action="execu
  <service-mapping path="/user/application/main" application="a
  <service-mapping path="/user/application/list" application="ap
    ...
  </service-mapping>
  <service-mapping path="/user/application/process/create" appli
    ...
  </service-mapping>
</routing-service-config>
```

必須項目	×
複数設定	○

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定するこ 値に {≪識別子≫} を記述することでURLの途中 で取得可能です。 例： /sample/view/{dataId}
application	マッピングを行うIM-JavaEE Frameworkのア
service	マッピングを行うIM-JavaEE Frameworkのサー
client-type	マッピングが有効となるクライアントタイプを指 省略時は、すべてのクライアントタイプにおい

アプリケーションマッピング設定

タグ名	application-mapping

URLのパターンとIM-JavaEE Frameworkのアプリケーションのマッピン
特定のIM-JavaEE Frameworkのアプリケーションが持つすべてのサー

【設定項目】

```
<routing-service-config>
 <application-mapping path-prefix="/user/application" applicati
  ...
 </application-mapping>
</routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
application	マッピングを行うIM-JavaEE Frameworkのプロク 定します。

属性名	説明
client-type	マッピングが有効となるクライアントタイプを指定 省略時は、すべてのクライアントタイプにおいて有

> **注意**
>
> この設定を使用するとルーティングに時間が掛かる、IM-Javal
> されません。
> 使用するIM-JavaEE Frameworkのサービスに対してservice-

認可設定

タグ名	authz

認可設定

【設定項目】

```
<routing-service-config>
 <service-mapping path="/user/application/list" application="ap
  <authz mapper="sample-mapper">
   ...
  </authz>
 </service-mapping>
 <service-mapping path="/user/application/process/create" appli
  <authz uri="service://application/update" action="execute" />
 </service-mapping>
</routing-service-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	service-mapping application-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー の値です。

> **コラム**
>
> authz-defaultタグが指定されていて、かつ、service-mappi

> **注意**
>
> この設定は、以下のいずれかの設定を行う必要があります。
> uri属性とaction属性
> mapper属性

認可パラメータ

タグ名	param

認可リソースマッパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-service-config>
 <service-mapping path="/user/application/list" application="ap
  <authz mapper="sample-mapper">
   <param key="foo" value="FOO" />
   <param key="bar" value="BAR" />
  </authz>
 </service-mapping>
</routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz authz-default

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

サーブレットルーティング設定

項目

- 概要
- リファレンス
  - デフォルト認可設定
  - サーブレットマッピング設定
  - 認可設定
  - 認可パラメータ

概要

URLに対して、フォワード先のサーブレットパスへのマッピングを行う

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/routing-servlet-config.x

設定場所	WEB-INF/conf/routing-servlet-config/{任意

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-servlet-config
   xmlns="http://www.intra-mart.jp/router/routing-servlet-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/router/routing-ser
 <authz-default uri="service://application/default" action="execu
 <servlet-mapping path="/user/application/main" servlet="appli
 <servlet-mapping path="/user/application/list" servlet="applica
  <authz mapper="sample-mapper">
   <param key="foo" value="FOO" />
   <param key="bar" value="BAR" />
  </authz>
 </servlet-mapping>
 <servlet-mapping path="/user/application/process/create" servl
  <authz uri="service://application/update" action="execute" />
 </servlet-mapping>
 <servlet-mapping path="/user/application/process/update" serv
  <authz uri="service://application/update" action="execute" />
 </servlet-mapping>
 <servlet-mapping path="/user/application/process/delete" servl
  <authz uri="service://application/update" action="execute" />
 </servlet-mapping>
</routing-servlet-config>
```

リファレンス

デフォルト認可設定

タグ名	authz-default

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-servlet-config>
 <authz-default uri="service://application/default" action="execu
</routing-servlet-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-servlet-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。

属性名	説明
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパ- の値です。

> **注意**
>
> この設定は、以下のいずれかの設定を行う必要があります。
> uri属性とaction属性
> mapper属性

> **注意**
>
> authz-defaultタグを省略した場合、servlet-mappingタグに

サーブレットマッピング設定

タグ名	servlet-mapping

URLとサーブレットパスのマッピングを行います。

【設定項目】

```
<routing-servlet-config>
 <authz-default uri="service://application/default" action="execu
 <servlet-mapping path="/user/application/main" servlet="appli
 <servlet-mapping path="/user/application/list" servlet="applica
  ...
 </servlet-mapping>
</routing-servlet-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-servlet-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定するこ 値に {＜識別子＞} を記述することでURLの途中 で取得可能です。 例： /sample/view/{dataId}
servlet	マッピングを行うサーブレットパスを指定しま
client-type	マッピングが有効となるクライアントタイプを指 省略時は、すべてのクライアントタイプにおい

注意

認可によるアクセス制御について

サーブレットに対して認可によるアクセス制御を行う場合、
具体的には、以下の３つで同じパスを指定したマッピング設

- web.xml に定義されたサーブレットに対するURL（se
- <routing-servlet-config> タグ内 <servlet-mapping
- <routing-servlet-config> タグ内 <servlet-mapping

このマッピング設定がない場合、認可を経由せずにサーブレ

例えば web.xml で以下のようにサーブレットが定義されて

```
<servlet-mapping>
  <servlet-name>SampleServlet</servlet-name>
  <url-pattern>/sample_servlet</url-pattern>
</servlet-mapping>
```

この場合 routing-servlet-config には、以下のような <se

```
<servlet-mapping path="/sample_servlet" servlet="sa
  <authz uri="service://sample/servlet" action="execu
</servlet-mapping>
```

ワイルドカード指定について

path属性、servlet属性には、 * （アスタリスク）等のワイ
web.xml で定義されたサーブレットに対するURLがワイル

認可設定

タグ名	authz

認可設定

【設定項目】

```
<routing-servlet-config>
 <servlet-mapping path="/user/application/list" servlet="applica
  <authz mapper="sample-mapper">
   ...
  </authz>
 </servlet-mapping>
 <servlet-mapping path="/user/application/process/create" servl
  <authz uri="service://application/update" action="execute" />
 </servlet-mapping>
</routing-servlet-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	servlet-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパ- の値です。

コラム

authz-defaultタグが指定されていて、かつ、servlet-mappi

注意

この設定は、以下のいずれかの設定を行う必要があります。

uri属性とaction属性

mapper属性

認可パラメータ

タグ名	param

認可リソースマッパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-servlet-config>
 <servlet-mapping path="/user/application/list" servlet="applica
  <authz mapper="sample-mapper">
   <param key="foo" value="FOO" />
   <param key="bar" value="BAR" />
  </authz>
 </servlet-mapping>
</routing-servlet-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz authz-default

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

ショートカットアクセス設定

項目
- 概要
- リファレンス
  - ショートカット操作モジュール
  - ショートカット操作実装クラス
  - エラーページパス
  - メインページパス
  - ショートカット情報検証設定

概要

ショートカットアクセス機能の設定です。
ショートカットアクセス機能はショートカットアクセス用のURLからア

モジュール	テナント管理機能モジュール
フォーマットファイル(xsd)	なし
設定場所	WEB-INF/conf/short-cut-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<short-cut-config>
 <short-cut-accessor>
  <short-cut-accessor-class>jp.co.intra_mart.foundation.securit
  <error-page>/user/shortcut/error</error-page>
  <main-page>/home</main-page>
  <deny-client-types>sp</deny-client-types>
  <validator code="RegExpUser" class="jp.co.intra_mart.foundat
  <validator code="RoleUser" class="jp.co.intra_mart.foundation
  <validator code="Script" class="jp.co.intra_mart.foundation.se
 </short-cut-accessor>
</short-cut-config>
```

> **コラム**
>
> short-cut-config.xmlのXMLスキーマ定義ファイル(xsdファイ
> xmlファイル修正時は上記設定例を参照の上、実施してくださ

リファレンス

ショートカット操作モジュール

タグ名	short-cut-accessor

ショートカット操作モジュールに関する設定を行います。

【設定項目】

```
<short-cut-config>
 <short-cut-accessor>
  ...
 </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカット操作モジュールに関する設定を行いま
単位・型	なし

省略時のデフォルト値	なし
親タグ	short-cut-config

ショートカット操作実装クラス

タグ名	short-cut-accessor-class

ショートカット情報を操作するモジュール(ショートカットアクセッサ)の

【設定項目】

```
<short-cut-config>
 <short-cut-accessor>
  <short-cut-accessor-class>jp.co.intra_mart.foundation.securit
 </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.security.shor
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

エラーページパス

タグ名	error-page

ショートカット情報が存在しない場合に表示するページパスを指定しま

【設定項目】

```
<short-cut-config>
 <short-cut-accessor>
  <error-page>/user/shortcut/error</error-page>
 </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカット情報が存在しない場合に表示するペ
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

メインページパス

タグ名	main-page

ショートカットページを表示するためのメインページパスを設定します。

【設定項目】

```
<short-cut-config>
 <short-cut-accessor>
  <main-page>/home</main-page>
 </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカットページを表示するためのメインペー
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

ショートカット情報検証設定

タグ名	validator

検証コードと検証プログラムの紐付けを行い、ショートカット情報の検

【設定項目】

```
<short-cut-config>
 <short-cut-accessor>
  <validator code="RegExpUser" class="jp.co.intra_mart.foundat
 </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	ショートカット情報の検証プログラムを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	short-cut-accessor

【属性】

属性名	説明
code	検証コードを指定します。
class	検証プログラムのクラスを指定します。 jp.co.intra_mart.foundation.security.shortcu 装クラスを指定します。

システム期間情報の管理

項目

- 概要
- リファレンス
  - システム最小日の設定
  - システム最大日の設定

概要

システム期間情報を管理するための設定を行います。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/system-valid-date-config.x
設定場所	WEB-INF/conf/system-valid-date-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<system-valid-date-config
 xmlns="http://www.intra_mart.co.jp/foundation/admin/validdate/c
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra_mart.co.jp/foundation/adm
 <min-date value="1900-01-01" />
 <max-date value="3000-01-01" />
</system-valid-date-config>
```

リファレンス

システム最小日の設定

タグ名	min-date

システムで取り扱う最小日を設定します。

【設定項目】

```
<system-valid-date-config>
 <min-date value="1900-01-01" />
</system-valid-date-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	system-valid-date-config

【属性】

属性名	説明	必須	デフォル
value	システム最小日の日付 (YYYY-MM-DD) です。	○	な

システム最大日の設定

タグ名	max-date

システムで取り扱う最大日を設定します。

【設定項目】

```
<system-valid-date-config>
  <max-date value="3000-01-01" />
</system-valid-date-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	system-valid-date-config

【属性】

属性名	説明	必須	テル
value	システム最大日の日付 (YYYY-MM-DD) です。	○	な

セキュア・トークンフィルタ設定

項目
- 概要
- リファレンス
  - セキュア・トークン対象URL設定

概要

セキュア・トークンによるリクエスト正当性チェックを実施するための

ここで設定されたURLにアクセスされた場合、特定のリクエストパラメ
内部的には、正当性のチェックには SecureTokenManager を利用して
リクエストパラメータの付与は <imSecureToken> タグを利用します。
詳しくは、APIドキュメントの スクリプト開発モデル・タグライブラリ

正当だと判断された場合は、対象となるURLの内容が表示されます。正

> **コラム**
>
> intra-mart Accel Platform 2014 Summer(Honoka) 以降、
> システムプロパティ jp.co.intra_mart.foundation.secure_to
> （開発用です。運用中は利用しないでください。）

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/token-filtering-target-c
設定場所	WEB-INF/conf/token-filtering-target-conf

```
<?xml version="1.0" encoding="UTF-8"?>
<token-filtering-target-config
  xmlns="http://www.intra-mart.jp/secure-token/target-url-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/secure-token/targe
  <entry url="/application/process/create" />
  <entry url="/application/process/update" />
  <entry url="/application/process/delete" />
</token-filtering-target-config>
```

リファレンス

セキュア・トークン対象URL設定

タグ名	entry

リクエスト時にセキュア・トークンによるチェックを行うパスを指定し

【設定項目】

```
<token-filtering-target-config>
  <entry url="/application/process/create" />
  <entry url="/application/process/update" />
  <entry url="/application/process/delete" />
</token-filtering-target-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	token-filtering-target-config

【属性】

属性名	説明
url	チェック対象となるURLを指定します。 指定された文字列に完全一致するURLへアクセス チェックが実施されます。

システム管理者用グローバルナビ設定

項目
- 概要
- リファレンス
  - メニュー設定
  - メニューの子要素

概要

システム管理者用の画面に表示するグローバルナビの項目を設定します。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/admin-global-navi-conf
設定場所	WEB-INF/conf/admin-global-navi-config/{

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-global-navi-config
   xmlns="http://www.example.org/im-admin-home"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/admin-global-navi
   <menu label="%CAP.Z.IWP.SYSTEM.HOME.CONSTRUCTION.TITL
      <children>
         <menu href="system/settings/license" label="%CAP.Z.IWP
         <menu href="system/settings/tenant" label="%CAP.Z.IWP.
         <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.H
      </children>
   </menu>
   <menu label="%CAP.Z.IWP.SYSTEM.HOME.ADMINISTRATION.TIT
      <children>
         <menu href="master/maintenance/call/user" label="%CAP
         <menu href="system/service/module" label="%CAP.Z.IWP.
         <menu href="system/service/status" label="%CAP.Z.IWP.S
         <menu href="system/service/lock" label="%CAP.Z.IWP.SYS
         <menu href="system/storage/view" label="%CAP.Z.IWP.SY
         <menu href="system/database" label="%CAP.Z.IWP.SYSTE
         <menu href="portal/portlet/app/list" label="%CAP.Z.IWP.S
         <menu href="portal/setting" label="%CAP.Z.IWP.SYSTEM.H
         <menu href="system/async/admin/task_queue_list" label=
         <menu href="system/settings/shared_database" label="%
      </children>
   </menu>
</admin-global-navi-config>
```

リファレンス

メニュー設定

タグ名	menu

メニューツリーの設定を行います。

【設定項目】

```
<menu label="%CAP.Z.IWP.SYSTEM.HOME.CONSTRUCTION.TITLE"
   <children>
      <menu href="system/settings/license" label="%CAP.Z.IWP.SY
      <menu href="system/settings/tenant" label="%CAP.Z.IWP.SY
      <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HO
   </children>
</menu>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-global-navi-config children

【属性】

属性名	説明
label	メニューに表示するラベルを指定します。

属性名	説明
href	メニューをクリックした際に遷移するURL
target	メニューをクリックした際のtargetを指定
id	このメニューのIDを指定します。複数ファ ている場合、そのメニューはマージされて
sort	メニューの要素のルートに同じIDが指定さ
useIframe	IFRAMEを使用した画面遷移を行うかを指

メニューの子要素

タグ名	children

メニューの子要素を指定します。子要素にはmenu要素を指定します。c

【設定項目】

```
<children>
  <menu href="system/settings/license" label="%CAP.Z.IWP.SYST
  <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYST
  <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME
</children>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	menu

コラム
label属性に%から始まる値を指定すると、MessageManage

システム管理者用ホームウィジェット設定

項目
- 概要
- リファレンス
  - ホームウィジェット設定
  - パラメータ設定

概要

システム管理者のホーム画面に表示するウィジェットを設定します。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/admin-home-widget-co
設定場所	WEB-INF/conf/admin-home-widget-confi

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-home-widget-config
   xmlns="http://www.intra-mart.jp/im-admin-home-widget-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/im-admin-home-w
   <widget url="system/home/widgets/status/status" captionUrl="
   <widget url="system/home/widgets/module/module" id="modu
   <widget url="system/home/widgets/license/license" id="license
   <widget url="system/home/widgets/system-info/system-info" id
   <widget url="system/home/widgets/bookmark/bookmark" id="
      <parameter name="bookmark" value="[
         {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/&amp;quot;,
         {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/imart/pa
         {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/support/
         {&amp;quot;href&amp;quot;: &amp;quot;http://imfaq.intra-mart.jp/imfaq/&
         {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/downloa
&amp;quot;external&amp;quot;:true},
         {&amp;quot;href&amp;quot;: &amp;quot;http://demo.intra-mart.jp/&amp;quot;,
         {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/apilist/&
         ]"/>
   </widget>
   <widget url="system/home/widgets/node/node" captionUrl="sy
   <widget url="system/home/widgets/tenant/tenant" id="tenant"
   <widget url="system/home/widgets/memory/memory" id="mer
</admin-home-widget-config>
```

リファレンス

ホームウィジェット設定

タグ名	widget

ホーム画面のウィジェット設定を行います。

【設定項目】

```
<widget url="system/home/widgets/status/status" captionUrl="sy
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-home-widget-config

【属性】

属性名	説明
id	このウィジェットのidを指定します。
url	このウィジェットの内容を返すURLを指定します。
captionUrl	このウィジェットのキャプションを返すURLを指定し
row	このウィジェットが表示される行を指定します。
col	このウィジェットが表示される列を指定します。
sizex	このウィジェットの横幅を指定します。
sizey	このウィジェットの縦幅を指定します。
color	このウィジェットの背景色を指定します。
flip	このウィジェットにキャプションを表示するかどうかす。

パラメータ設定

タグ名	parameter

ウィジェットのパラメータ設定を行います。

【設定項目】

```
<parameter name="bookmark" value="[
  {&amp;quot;href&amp;quot;: &amp;quot;http://www.intra-mart.jp/&amp;quot;, &amp;quo
  ]"/>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	widget

【属性】

属性名	説明	必須	デフォルト値
name	パラメータの名前です。	×	なし
value	パラメータの値です。	×	なし

システム管理者用ユーティリティメニュー設定

項目

- 概要
- リファレンス
  - ユーティリティメニュー設定
  - メニューの子要素

概要

システム管理者のユーティリティメニューの項目を設定します。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/admin-utility-config.xsd
設定場所	WEB-INF/conf/admin-utility-config/{任意の

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-utility-config xmlns="http://www.example.org/im-admin
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.example.org/im-admin-utility-c
   <menu href="javascript:void(0);" label="%CAP.Z.IWP.SYSTEM.H
      <children>
         <menu href="system/administrator/password" label="%CA
         <menu href="system/administrator/locale" label="%CAP.Z
      </children>
   </menu>
   <menu href="logout" label="%CAP.Z.IWP.SYSTEM.COMMON.LO
</admin-utility-config>
```

リファレンス

ユーティリティメニュー設定

タグ名	menu

システム管理者のユーティリティメニュー設定を行います。

【設定項目】

```
<menu href="javascript:void(0);" label="%CAP.Z.IWP.SYSTEM.HOM
   <children>
      <menu href="system/administrator/password" label="%CAP.Z
      <menu href="system/administrator/locale" label="%CAP.Z.IW
   </children>
</menu>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-utility-config children

【属性】

属性名	説明
label	メニューに表示するラベルを指定します。
href	メニューをクリックした際に遷移するURL
target	メニューをクリックした際のtargetを指定
id	このメニューのIDを指定します。複数ファ ている場合、そのメニューはマージされて
sort	メニューの要素のルートに同じIDが指定さ
useIframe	IFRAMEを使用した画面遷移を行うかを指

メニューの子要素

タグ名	children

メニューの子要素を指定します。子要素にはmenu要素を指定します。c

【設定項目】

```
<children>
  <menu href="system/settings/license" label="%CAP.Z.IWP.SYST
  <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYST
  <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME
</children>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	menu

コラム
label属性に%から始まる値を指定すると、MessageManage

IM-Notice

IM-Notice 設定

項目

- 概要
- リファレンス
  - ベースURLの設定
  - ショートカットURLの有効期間の設定

概要

通知機能全般に関する設定です。

モジュール	通知機能全般
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-config.xsd
設定場所	WEB-INF/conf/im-notice-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-config
   xmlns="http://www.intra-mart.jp/notice/config/im-notice-config"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/notice/config/im-n

   <base-url></base-url>
   <short-cut-duration-minutes>43200</short-cut-duration-
</im-notice-config>
```

リファレンス

ベースURLの設定

タグ名	base-url

intra-mart Accel Platform のベースURLを設定します。
この設定は、ショートカットURLの作成に使用されます。
未設定の場合、server-context-config.xml で指定したベースURLが使

IM-Noticeへのメッセージ配信処理をジョブ経由で行う場合は、ベースU
「セットアップガイド」の「ベースURL 」または、「テナント環境情報

> **注意**
> ベースURLを指定しない場合、以下のエラーが発生します。
> jp.co.intra_mart.system.notice.exception.NoticeRuntime

【設定項目】

```
<base-url>http://example.org/imart</base-url>
```

必須項目	×
複数設定	×
設定値・設定する内容	クライアント（ブラウザ等）からアクセスを
単位・型	URI
省略時のデフォルト値	server-context-config.xmlで設定したベー
親タグ	im-notice-config

ショートカットURLの有効期間の設定

タグ名	short-cut-duration-minutes

ショートカットURLの有効期間を設定します。
クライアントがショートカットURLを使用しログインを行った後、有効

【設定項目】

```
<short-cut-duration-minutes>43200</short-cut-duration-mi
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカットURLの有効期間を設定してください。 「0」を設定した場合、ショートカットURLは作成さ
単位・型	数値 (分) (0-)
省略時のデフォルト値	なし
親タグ	im-notice-config

IM-Notice Mobile設定

項目

- 概要
- リファレンス
  - プロキシの設定
  - ホストの設定
  - ポート番号の設定
  - ユーザ名の設定
  - パスワードの設定
  - ワークステーションの設定
  - ドメインの設定
  - Amazon SNSの設定
  - エンドポイントの設定
  - Amazon リソースネームの設定
  - アクセスキーの設定
  - シークレットキーの設定
  - Google Cloud Messaging for Androidのプロジェクト番号の
  - Baidu APIキーの設定
  - Google Cloud Messaging for Androidの設定
  - URLの設定
  - APIキーの設定
  - プロジェクト番号の設定
  - 通知方法の設定
  - 通知タイトルの最長文字数の設定
  - 通知内容の最長文字数の設定

概要

モバイル通知機能に関する設定です。

モジュール	モバイル通知機能
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-mobile-config.xs
設定場所	WEB-INF/conf/im-notice-mobile-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mobile-config
   xmlns="http://www.intra-mart.jp/notice/mobile/config/im-notice-
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/notice/mobile/con

   <proxy enable="false">
      <host></host>
      <port xsi:nil="true"></port>
      <username></username>
      <password></password>
      <workstation></workstation>
      <domain></domain>
   </proxy>

   <asns name="asns-apns">
      <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
      <platform-application-arn></platform-application-arn>
      <access-key></access-key>
      <secret-key></secret-key>
   </asns>

   <asns name="asns-gcm">
      <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
      <platform-application-arn></platform-application-arn>
      <access-key></access-key>
      <secret-key></secret-key>
      <gcma-project-number></gcma-project-number>
   </asns>

   <asns name="asns-baidu">
      <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
      <platform-application-arn></platform-application-arn>
      <access-key></access-key>
      <secret-key></secret-key>
      <baidu-api-key></baidu-api-key>
   </asns>

   <gcma name="gcma">
      <url>https://android.googleapis.com/gcm/send</url>
      <api-key></api-key>
      <project-number></project-number>
   </gcma>

   <push type="android" engine="gcma" />
   <push type="ios" engine="asns-apns" />

   <max-push-subject-length>30</max-push-subject-length>
   <max-push-body-length>70</max-push-body-length>
</im-notice-mobile-config>
```

リファレンス

プロキシの設定

タグ名	proxy

Amazon SNSやGoogle Cloud Messaging for Androidへ接続するため

【設定項目】

```
<im-notice-mobile-config>
   <proxy enable="false">
      . . . . .
   </proxy>
</im-notice-mobile-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	プロキシを使用する際に必要な設定をしてください。
単位・型	なし

省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
enable	プロキシを使用する場合、trueを設定してください。 プロキシを使用しない場合、falseを設定してください。

ホストの設定

タグ名	host

プロキシサーバのIPアドレスを設定します。

【設定項目】

```
<proxy enable="true">
  <host>XXX.XXX.XXX.XXX</host>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

ポート番号の設定

タグ名	port

プロキシサーバのポート番号を設定します。

【設定項目】

```
<proxy enable="true">
  <port>XXXX</port>
</proxy>
```

必須項目	×

複数設定	×
設定値・設定する内容	なし
単位・型	数値（0-）
省略時のデフォルト値	なし
親タグ	proxy

【属性】

属性名	説明
xsi:nil	ポート番号を指定する場合、この属性を削除してください。 ポート番号を指定しない場合、trueを設定してください。

ユーザ名の設定

タグ名	username

プロキシ使用時にユーザ認証が必要な場合、ユーザ名を設定します。

【設定項目】

```
<proxy enable="true">
  <username>XXXXXXXX</username>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

パスワードの設定

タグ名	password

プロキシ使用時にユーザ認証が必要な場合、パスワードを設定します。

【設定項目】

```
<proxy enable="true">
  <password>XXXXXXXX</password>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

ワークステーションの設定

タグ名	workstation

NTLM認証を行っている場合、ワークステーションを設定します。

【設定項目】

```
<proxy enable="true">
  <workstation>XXXXXXXX</workstation>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列

省略時のデフォルト値	なし
親タグ	proxy

ドメインの設定

タグ名	domain

NTLM認証を行っている場合、ドメインを設定します。

【設定項目】

```
<proxy enable="true">
  <domain>XXXXXXXX</domain>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

Amazon SNSの設定

タグ名	asns

Amazon SNSを使用するための項目値を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    . . . . .
  </asns>
</im-notice-mobile-config>
```

必須項目	○
複数設定	○

設定値・設定する内容	Amazon SNSを使用するために必要な設定をしてくだ
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
name	任意の名前を設定してください。 pushタグのengine要素で通知方法を設定する際に使用

エンドポイントの設定

タグ名	endpoint

エンドポイントを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	使用するリージョンに対応するAmazon SNS 各リージョンのエンドポイントは AWS docu
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	asns

Amazon リソースネームの設定

タグ名	platform-application-arn

Amazonリソースネームを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <platform-application-arn>arn:aws:sns:ap-northeast-1:XX
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Amazon SNSで作成したアプリケーションのAp
単位・型	文字列

省略時のデフォルト値	空文字
親タグ	asns

アクセスキーの設定

タグ名	access-key

アクセスキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <access-key>XXXXXXXXXXXXXXXXXXXX</access-key>
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	AWSで作成したアクセスキーを設定してください。
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	asns

シークレットキーの設定

タグ名	secret-key

シークレットキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <secret-key>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	AWSで作成したシークレットキーを設定してください
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	asns

Google Cloud Messaging for Androidのプロジェクト番号の設

タグ名	gcma-project-number

Google Cloud Messaging for Androidのプロジェクト番号を設定しま

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-gcm">
    <gcma-project-number>XXXXXXXXXXXX</gcma-project-
  </asns>
</im-notice-mobile-config>
```

必須項目	通知手段にAmazon SNS+Google Cloud M
複数設定	×
設定値・設定する内容	Google Cloud Messaging for Androidで作い。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	asns

Baidu APIキーの設定

タグ名	baidu-api-key

Baidu APIキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-baidu">
    <baidu-api-key>XXXXXXXXXXX</baidu-api-key>
  </asns>
</im-notice-mobile-config>
```

必須項目	通知手段にAmazon SNS+Baiduを利用する場合に必
複数設定	×
設定値・設定する内容	Baiduクラウドプロジェクトの APIキーを設定してくだ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	asns

Google Cloud Messaging for Androidの設定

タグ名	gcma

Google Cloud Messaging for Androidを使用するための設定をします

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    . . . . .
  </gcma>
</im-notice-mobile-config>
```

必須項目	○
複数設定	○

設定値・設定する内容	Google Cloud Messaging for Androidを使用す
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
name	任意の名前を設定してください。 pushタグのengine要素で通知方法を設定する際に使用します。

URLの設定

タグ名	url

Google Cloud Messaging for AndroidのURLを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <url>https://android.googleapis.com/gcm/send</url>
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	「https://android.googleapis.com/gcm/send」
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	gcma

APIキーの設定

タグ名	api-key

APIキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <api-key>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Google Cloud Messaging for Androidで作成
単位・型	文字列

省略時のデフォルト値	空文字
親タグ	gcma

プロジェクト番号の設定

タグ名	project-number

プロジェクト番号を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <project-number>XXXXXXXXXXXX</project-number>
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	GCMAで作成したプロジェクトのプロジェクト番号
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	gcma

通知方法の設定

タグ名	push

各通知先へ通知を行う際に使用するサービスを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <push type="android" engine="gcma" />
  <push type="ios" engine="asns-apns" />
</im-notice-mobile-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	識別子
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
type	通知先（android/ios）を設定してください。
engine	各通知先への通知方法（asnsタグまたはgcmaタ してください。

通知タイトルの最長文字数の設定

タグ名	max-push-subject-length

Push通知受信時に表示するタイトルの最大文字数を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <max-push-subject-length>30</max-push-subject-length
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Push通知受信時に表示するタイトルの最大文字数を
単位・型	数値（0-）
省略時のデフォルト値	空文字
親タグ	im-notice-mobile-config

通知内容の最長文字数の設定

タグ名	max-push-body-length

Push通知受信時に表示する本文の最長文字数を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <max-push-body-length>70</max-push-body-length>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Push通知受信時に表示する本文の最長文字数を設定
単位・型	数値（0-）
省略時のデフォルト値	空文字
親タグ	im-notice-mobile-config

IM-Notice MQ設定

項目

- 概要
- リファレンス
  - ポート番号の設定
  - ping機能の設定
  - エンドポイントの設定
  - I/Oスレッドの設定
  - ソケットの最大数の設定
  - リンガーの設定

概要

デスクトップ通知機能を使用するための設定です。

モジュール	デスクトップ通知機能
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-mq-config.xsd
設定場所	WEB-INF/conf/im-notice-mq-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mq-config
   xmlns="http://www.intra-mart.jp/notice/mq/config/im-notice-mq-
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/notice/mq/config/

 <address bind="true">tcp://*:40608</address>
 <ping enable="false" interval-seconds="0" time-to-live-seconds=
 <endpoints>
    <endpoint>tcp://localhost:40608</endpoint>
 </endpoints>
 <zmq-io-threads>1</zmq-io-threads>
 <zmq-max-sockets>1024</zmq-max-sockets>
 <zmq-linger>1000</zmq-linger>

</im-notice-mq-config>
```

リファレンス

ポート番号の設定

タグ名	address

サーバがクライアントとの通信を行う際に使用するポート番号を設定し
分散構成やWeb Serverを利用している構成の場合はブローカーを利用
デフォルトの設定では40608ポートを使用します。

> **コラム**
> ブローカーとは、ネットワーク中継を行うためのプロキシです

【設定項目】

```
<im-notice-mq-config>
   <address bind="true">tcp://*:40608</address>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ポート番号を設定してください。 ブローカーを利用する場合、「*tcp:* 定します。
単位・型	文字列

省略時のデフォルト値	なし
親タグ	im-notice-mq-config

【属性】

属性名	説明
bind	スタンドアローン構成の場合、trueを設定してください ブローカーを利用する構成の場合、falseを設定してくだ さい。

ping機能の設定

タグ名	ping

ping機能に関する設定をします。

【設定項目】

```
<im-notice-mq-config>
  <ping enable="false" interval-seconds="0" time-to-live-seconds
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ping機能の設定をしてください。
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

【属性】

属性名	説明
enable	ping機能の利用設定を行います。 利用する場合はtrue、利用しない場合はfalseを設
interval-seconds	クライアントにpingを流す間隔を数値（秒）で設
time-to live-seconds	クライアントがサーバに再接続を行うまでの時間 い。

エンドポイントの設定

タグ名	endpoint

エンドポイントを設定します。
ブローカーを利用する場合ブローカーのホストまたはIPアドレスとブロ

【設定項目】

```
<im-notice-mq-config>
  <endpoints>
    <endpoint>tcp://localhost:40608</endpoint>
  </endpoints>
</im-notice-mq-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	クライアントの接続先を設定します addressタグで設定したポート番号 ブローカーを利用する場合、「*tcp* 定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	endpoints

I/Oスレッドの設定

タグ名	zmq-io-threads

ZeroMQで使用されるI/Oスレッドの設定をします。
詳細は ZMQ_IO_THREADS: Set number of I/O threads を参照してく

【設定項目】

```
<im-notice-mq-config>
  <zmq-io-threads>1</zmq-io-threads>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	I/O操作を処理するためのZeroMQスレッドプール
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

ソケットの最大数の設定

タグ名	zmq-max-sockets

ZeroMQで使用されるソケットの最大数の設定をします。

【設定項目】

```
<im-notice-mq-config>
  <zmq-max-sockets>1024</zmq-max-sockets>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ZeroMQのコンテキストで許可されているソケッ
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

リンガーの設定

タグ名	zmq-linger

リンガー期間を設定します。
クライアントのソケットが閉じられた場合、未送信のメッセージは設定
詳細は ZMQ_LINGER: Set linger period for socket shutdown を参照

【設定項目】

```
<im-notice-mq-config>
  <zmq-linger>1000</zmq-linger>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リンガー期間を設定してください。
単位・型	数値（秒）（0-）
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

IM-Notice MQブローカー設定

項目

- 概要
- リファレンス
  - ポート番号（フロントエンド）の設定
  - ポート番号（バックエンド）の設定
  - ping機能の設定
  - I/Oスレッドの設定
  - ソケットの最大数の設定
  - リンガー（バックエンド）の設定
  - リンガー（フロントエンド）の設定

概要

ブローカーを含む構成でデスクトップ通知機能を使用するための設定で

モジュール	デスクトップ通知機能
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-mq-broker-confi
設定場所	WEB-INF/conf/im-notice-mq-broker-config.x

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mq-broker-config
   xmlns="http://www.intra-mart.jp/notice/mq/broker/config/im-noti
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/notice/mq/broker/

  <frontend>tcp://*:40608</frontend>
  <backend>tcp://*:40609</backend>
  <ping enable="false" interval-seconds="0" />
  <zmq-io-threads>1</zmq-io-threads>
  <zmq-max-sockets>1024</zmq-max-sockets>
  <zmq-backend-linger>1000</zmq-backend-linger>
  <zmq-frontend-linger>1000</zmq-frontend-linger>

</im-notice-mq-broker-config>
```

リファレンス

ポート番号（フロントエンド）の設定

タグ名	frontend

ブローカーがデスクトップ通知アプリケーションとの通信を行うポート
デフォルトの設定では40608ポートを使用します。

【設定項目】

```
<im-notice-mq-broker-config>
   <frontend>tcp://*:40608</frontend>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ブローカーがクライアントとの通信を行う際に使
単位・型	文字列
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

ポート番号（バックエンド）の設定

タグ名	backend

ブローカーがWeb Application Server との通信を行うポート番号を設
デフォルトの設定では40609ポートを使用します。

【設定項目】

```
<im-notice-mq-broker-config>
   <backend>tcp://*:40609</backend>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ブローカーがサーバとの通信を行う際に使用する
単位・型	なし

省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

ping機能の設定

タグ名	ping

ping機能の設定をします。

【設定項目】

```
<im-notice-mq-broker-config>
    <ping enable="false" interval-seconds="0" />
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ping機能の設定をしてください。
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

【属性】

属性名	説明
enable	ping機能の利用設定を行います。 利用する場合はtrue、利用しない場合はfalseを設定
interval-seconds	クライアントにpingを流す間隔を数値（秒）で設定し

I/Oスレッドの設定

タグ名	zmq-io-threads

ZeroMQで使用されるI/Oスレッドの設定をします。
詳細は ZMQ_IO_THREADS: Set number of I/O threads を参照してく

【設定項目】

```
<im-notice-mq-broker-config>
    <zmq-io-threads>1</zmq-io-threads>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	I/O操作を処理するためのZeroMQスレッドプール
単位・型	数値
省略時のデフォルト値	なし

親タグ	im-notice-mq-broker-config

ソケットの最大数の設定

タグ名	zmq-max-sockets

ZeroMQで使用されるソケットの最大数の設定をします。

【設定項目】

```
<im-notice-mq-broker-config>
  <zmq-max-sockets>1024</zmq-max-sockets>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ZeroMQのコンテキストで許可されているソケッ
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

リンガー（バックエンド）の設定

タグ名	zmq-backend-linger

サーバ側のリンガー期間を設定します。
サーバのソケットが閉じられた場合、未送信のメッセージは設定された
詳細は ZMQ_LINGER: Set linger period for socket shutdown を参照

【設定項目】

```
<im-notice-mq-broker-config>
  <zmq-backend-linger>1000</zmq-backend-linger>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リンガー期間を設定してください。
単位・型	数値（秒）（0-）
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

リンガー（フロントエンド）の設定

タグ名	zmq-frontend-linger

クライアント側のリンガー期間を設定します。
クライアントのソケットが閉じられた場合、未送信のメッセージは設定

詳細は ZMQ_LINGER : Set linger period for socket shutdown を参照

【設定項目】

```
<im-notice-mq-broker-config>
  <zmq-frontend-linger>1000</zmq-frontend-linger>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リンガー期間を設定してください。
単位・型	数値（秒）（0-）
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

招待機能と外部ユーザ

招待権限リスト設定

項目

- 概要
- リファレンス
  - 招待権限デコレータ

概要

招待時に付与される権限デコレータの設定です。
招待情報の権限リストに表示されます。
この招待権限デコレータによって招待されたユーザに対して、様々な権

モジュール	招待機能と外部ユーザ
フォーマットファイル(xsd)	WEB-INF/schema/invitation-decorator-conf
設定場所	WEB-INF/conf/invitation-decorator-config/*

```
<?xml version="1.0" encoding="UTF-8"?>
<invitation-decorator-config
    xmlns="http://intra-mart.co.jp/system/external/user/config/decor
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://intra-mart.co.jp/system/external/use

        <invitation-decorator id="im_role" name="%CAP.Z.IWP.EX
class="jp.co.intra_mart.foundation.external.user.invitation.decorato
        <invitation-decorator id="im_department" name="%CAP.Z
class="jp.co.intra_mart.foundation.external.user.invitation.decorato
path="im_external_user/invitation/admin/decorators/department"/>
        <invitation-decorator id="im_app_license" name="%CAP.Z
class="jp.co.intra_mart.foundation.external.user.invitation.decorato
path="im_external_user/invitation/admin/decorators/app_license"/>
        <invitation-decorator id="im_external_user" name="%CAP.
class="jp.co.intra_mart.foundation.external.user.invitation.decorato

    </invitation-decorator-config>
```

リファレンス

招待権限デコレータ

タグ名	invitation-decorator

招待権限デコレータを設定します。
招待したユーザに対して行う処理（権限の付与など）を定義します。
設定された招待権限デコレータは招待情報の権限リストに表示されます。

【設定項目】

```
<invitation-decorator-config>
        <invitation-decorator id="im_role" name="%CAP.Z.IWP.E
class="jp.co.intra_mart.foundation.external.user.invitation.decorato
</invitation-decorator-config>
```

必須項目	○
複数設定	○

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	invitation-decorator-config

【属性】

属性名	説明
id	権限デコレータを一意に識別するIDです。 利用できる文字は、半角英数および_（アンダース
name	権限デコレータの名称です。 招待情報の権限リストの名称として表示されます。 先頭に%をつけて、メッセージキーを書くことで、 表示ロケールに合わせた文字を取得します。 %をつけない場合は、そのまま設定した値が表示さ 【例】 **%CAP.Z.IWP.EXTERNAL.USER.INVI**
decorator-class	権限デコレータの処理クラスです。 招待されたユーザに対して、実行する処理を記述し jp.co.intra_mart.foundation.external.user.invi したクラスである必要があります。 権限デコレータの処理クラスの開発方法について リストを拡張する 」 を参照してください。
editable	権限デコレータに対してパラメータが設定できるか false ： パラメータを必要としません。 true：パラメータを必要とします。 この値がtrueの場合は、招待情報の権限リストの名 です。
edit-path	権限デコレータのパラメータを編集するための画面 ます。 画面プログラムの開発方法については、「 招待機能 タのパラメータ編集画面の作成 」 を参照してくだ editable属性がtrueの場合で、edit-pathを指定し イアログが表示されます。

招待メールデフォルト設定

項目
- 概要
- リファレンス
  - 招待確認用URL
  - Fromアドレス
  - ReplyToアドレス
  - Ccアドレス
  - Bccアドレス
  - メールアドレス

概要

招待メールのデフォルト設定です。
招待情報を新規作成する場合のメール送信方法に関する設定を行う機能
本設定ファイルでは、招待情報の新規作成時に画面に表示されるデフォ

モジュール	招待機能と外部ユーザ
フォーマットファイル(xsd)	WEB-INF/schema/invitation-mail-config.xsd
設定場所	WEB-INF/conf/invitation-mail-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<invitation-mail-config
   xmlns="http://intra-mart.co.jp/system/external/user/config/mail/i
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/external/use

 <programPath>/invitation/invitation_register</programPath>
 <mailFrom>sender@sample.com</mailFrom>
 <mailReplyTo>reply@sample.com</mailReplyTo>
 <mailCc>
  <address>cc1@sample.com</address>
  <address>cc2@sample.com</address>
 </mailCc>
 <mailBcc>
  <address>bcc1@sample.com</address>
  <address>bcc2@sample.com</address>
 </mailBcc>

</invitation-mail-config>
```

リファレンス

招待確認用URL

タグ名	programPath

招待確認用のURLです。
コンテキストパスからの相対パスを設定します。

【設定項目】

```
<invitation-mail-config>
 <programPath>/invitation/invitation_register</programPath>
</invitation-mail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	/invitation/invitation_registerで固定です。 変更する必要はありません。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	invitation-mail-config

Fromアドレス

タグ名	mailFrom

招待メールのFromアドレスを指定します。

【設定項目】

```
<invitation-mail-config>
 <mailFrom>sender@sample.com</mailFrom>
</invitation-mail-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	招待メールのFromアドレスを設定します。
単位・型	文字列（メールアドレス）
省略時のデフォルト値	なし
親タグ	invitation-mail-config

ReplyToアドレス

タグ名	mailReplyTo

招待メールのReplyToアドレスを指定します。

【設定項目】

```
<invitation-mail-config>
 <mailReplyTo>reply@sample.com</mailReplyTo>
</invitation-mail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	招待メールのReplyToアドレスを設定します。
単位・型	文字列（メールアドレス）
省略時のデフォルト値	なし
親タグ	invitation-mail-config

Ccアドレス

タグ名	mailCc

招待メールのCcアドレスを指定します。

【設定項目】

```
<invitation-mail-config>
 <mailCc>
  <address>cc1@sample.com</address>
  <address>cc2@sample.com</address>
 </mailCc>
</invitation-mail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	招待メールのCcアドレスを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	invitation-mail-config

Bccアドレス

タグ名	mailBcc

招待メールのBccアドレスを指定します。

【設定項目】

```
<invitation-mail-config>
 <mailBcc>
  <address>bcc1@sample.com</address>
  <address>bcc1@sample.com</address>
 </mailBcc>
</invitation-mail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	招待メールのBccアドレスを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	invitation-mail-config

メールアドレス

タグ名	address

メールの送信先メールアドレスを指定します。

【設定項目】

```
<invitation-mail-config>
 <mailCc>
  <address>cc1@sample.com</address>
 </mailCc>
</invitation-mail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メールアドレスを設定します。
単位・型	文字列（メールアドレス）
省略時のデフォルト値	なし
親タグ	mailCc, mailBcc

リバースプロキシ

リバースプロキシの設定

項目
- 概要
- リファレンス
  - リバースプロキシの設定
  - 認可設定
  - リバースプロキシ対象のURLの設定
  - リバースプロキシ対象のリダイレクトURLの設定
  - リクエストヘッダーの設定
  - リクエストヘッダーの変更設定
  - リクエストヘッダーの追加設定
  - リクエストヘッダーの削除設定
  - レスポンスヘッダーの設定
  - レスポンスヘッダーの変更設定
  - レスポンスヘッダーの追加設定
  - レスポンスヘッダーの削除設定
  - レスポンス置換処理一覧設定
  - レスポンス置換処理設定
  - レスポンス置換文字列設定
  - 独自設定

概要

リバースプロキシの動作設定を行います。

モジュール	リバースプロキシ
フォーマットファイル (xsd)	WEB-INF/schema/im-reverse-proxy-path
設定場所	WEB-INF/conf/im-reverse-proxy-path-co

```
<?xml version="1.0" encoding="UTF-8"?>
<im-reverse-proxy-path-config
   xmlns="http://www.intra-mart.jp/reverse-proxy/config/im-reverse
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/reverse-proxy/cor

 <reverse-proxy handler="jp.co.intra_mart.foundation.reverse_pr
  <authz action="execute" uri="service://intra-mart.jp/public-res
  <proxy-pass default-charset="UTF-8" path="/example/" timeou
  <proxy-pass-reverse path="/example/" url="https://www.exar
  <request-headers>
   <set header="x-my-header" value="my-value" />
   <append header="x-my-header" value="my-value" />
   <unset header="x-my-header" where-value="my-value" />
  </request-headers>
  <response-headers>
   <set header="x-my-header" value="my-value" />
   <append header="x-my-header" value="my-value" />
   <unset header="x-my-header" where-value="my-value" />
  </response-headers>
  <substitutes>
   <substitute content-type="text/html" regexp="http://www\.ex
    <replacement>${im-reverse-proxy-base-url}/example/</rep
   </substitute>
  </substitutes>
  <any-config />
 </reverse-proxy>

</im-reverse-proxy-path-config>
```

リファレンス

リバースプロキシの設定

タグ名	reverse-proxy

リバースプロキシを設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy handler="jp.co.intra_mart.foundation.reverse_pr
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	リバースプロキシを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-reverse-proxy-path-config

【属性】

属性名	説明
handler	リバースプロキシを実行するクラス名を設定しま スプロキシハンドラでは実現不可能な処理等を行 じてハンドラを実装し、クラス名を設定してくだ

認可設定

タグ名	authz

リバースプロキシするURLに対する認可を設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <authz action="execute" uri="service://intra-mart.jp/public-resc
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リバースプロキシするURLに対する認 きます。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

【属性】

属性名	説明	必須	デフォルト値
action	認可アクションを設定します。	○	なし
uri	認可リソースURIを設定します。	○	なし

リバースプロキシ対象のURLの設定

タグ名	proxy-pass

リバースプロキシを行う対象のURLを設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <proxy-pass default-charset="UTF-8" path="/example/" timeou
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リバースプロキシを行う対象のURLを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

【属性】

属性名	説明
default-charset	プロキシ対象のURLの文字コードを設定し自動判別できなかった場合、ここで指定さ
path	プロキシ対象のURLの文字コードを設定し「http://<HOST>:<PORT>/<CONTEX す。
timeout-seconds	プロキシを行う際のタイムアウト値を設定
url	プロキシ対象のURLを設定します。

リバースプロキシ対象のリダイレクトURLの設定

タグ名	proxy-pass-reverse

リバースプロキシ対象のリダイレクトレスポンスのURLの設定

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <proxy-pass-reverse path="/example/" url="https://www.exar
 </reverse-proxy>
<im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	リバースプロキシ対象のリダイレクトレスポンスの
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

【属性】

属性名	説明
path	リダイレクトレスポンスの変換対象のURLを、この設換します。
url	リダイレクトレスポンスの変換対象のURLを設定しま

リクエストヘッダーの設定

タグ名	request-headers

リクエストヘッダーに関する設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <request-headers>
   ....
  </request-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	リクエストヘッダーに関する設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

リクエストヘッダーの変更設定

タグ名	set

リクエストヘッダーを変更するための設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <request-headers>
   <set header="x-my-header" value="my-value" />
  </request-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	リクエストヘッダーを変更するための設定を指定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-headers

【属性】

属性名	説明	必須
header	変更するリクエストヘッダー名を設定します。	○
value	新しいリクエストヘッダー値を設定します。	○

リクエストヘッダーの追加設定

タグ名	append

リクエストヘッダーを追加するための設定を行います。
同名のリクエストヘッダー名が存在する場合、リストに追加されます。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <request-headers>
   <append header="x-my-header" value="my-value" />
  </request-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	リクエストヘッダーを追加するための設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-headers

【属性】

属性名	説明	必須
header	追加するリクエストヘッダー名を設定します。	○
value	追加するリクエストヘッダー値を設定します。	○

リクエストヘッダーの削除設定

タグ名	unset

リクエストヘッダーを削除するための設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <request-headers>
   <unset header="x-my-header" where-value="my-value" />
  </request-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○

設定値・設定する内容	リクエストヘッダーを削除するための設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-headers

【属性】

属性名	説明
header	削除するリクエストヘッダー名を設定します。
where-value	本属性に指定された正規表現にマッチするする値を除します。 省略した場合、header属性に設定した値と同名のします。

レスポンスヘッダーの設定

タグ名	response-headers

レスポンスヘッダーに関する設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <response-headers>
   ....
  </response-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	レスポンスヘッダーに関する設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

レスポンスヘッダーの変更設定

タグ名	set

レスポンスヘッダーを変更するための設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <response-headers>
   <set header="x-my-header" value="my-value" />
  </response-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○

設定値・設定する内容	レスポンスヘッダーを変更するための設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-headers

【属性】

属性名	説明	必須
header	変更するレスポンスヘッダー名を設定します。	○
value	新しいレスポンスヘッダー値を設定します。	○

レスポンスヘッダーの追加設定

タグ名	append

レスポンスヘッダーを追加するための設定を行います。
同名のレスポンスヘッダー名が存在する場合、リストに追加されます。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <response-headers>
   <append header="x-my-header" value="my-value" />
  </response-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	レスポンスヘッダーを追加するための設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-headers

【属性】

属性名	説明	必須
header	追加するレスポンスヘッダー名を設定します。	○
value	追加するレスポンスヘッダー値を設定します。	○

レスポンスヘッダーの削除設定

タグ名	unset

レスポンスヘッダーを削除するための設定を行います。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <response-headers>
   <unset header="x-my-header" where-value="my-value" />
  </response-headers>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	レスポンスヘッダーを削除するための設定を指定しま
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-headers

【属性】

属性名	説明
header	削除するレスポンスヘッダー名を設定します。
where-value	本属性に指定された正規表現にマッチするする値を 除します。 省略した場合、header属性に設定した値と同名の します。

レスポンス置換処理一覧設定

タグ名	substitutes

レスポンスの置換処理の一覧を設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <substitutes>
   ....
  </substitutes>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	レスポンスの置換処理の一覧を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

レスポンス置換処理設定

タグ名	substitute

レスポンスの置換処理を設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <substitutes>
   <substitute content-type="text/html" regexp="http://www\.ex
  </substitutes>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	レスポンスの置換処理を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	substitutes

【属性】

属性名	説明
content-type	レスポンスが本属性に設定したコンテントタイプと います。
regexp	レスポンスに対して置換を行う対象を正規表現で設
where-path	URLが本属性に設定した正規表現にマッチする場合

レスポンス置換文字列設定

タグ名	replacement

レスポンスの置換後の文字列を設定します。

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <substitutes>
   <substitute>
    <replacement>${im-reverse-proxy-base-url}/example/</re
   </substitute>
  </substitutes>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	置換後の文字列を設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	substitute

コラム

<replacement> では以下の変数を利用できます。

- ${im-base-url}
  - 「<ベースURL>」に置換されます。
- ${im-reverse-proxy-base-url}
  - 「<ベースURL>/reverse_proxy」に置換されます。

独自設定

タグ名	any-config

独自の設定を行います。
デフォルトハンドラ以外の実装を用いた際に、独自の設定をリバースプ

【設定項目】

```
<im-reverse-proxy-path-config>
 <reverse-proxy>
  <any-config>
   ....
  </any-config>
 </reverse-proxy>
</im-reverse-proxy-path-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	ハンドラー独自の設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	reverse-proxy

Salesforce連携

Salesforceクライアントの設定

項目
- 概要
- リファレンス
  - Salesforceクライアントの設定
  - Salesforceクライアントの詳細設定
  - OAuthプロバイダID
  - トークンエンドポイント
  - コンシューマ鍵
  - コンシューマの秘密
  - Salesforceユーザのユーザ名
  - Salesforceユーザのパスワード
  - UserPasswordResolver の 実装クラス
  - 追加パラメータ

概要

Salesforce連携時の接続先情報を設定します。

モジュール	Salesforce連携
フォーマットファイル(xsd)	WEB-INF/schema/salesforce-client-config
設定場所	WEB-INF/conf/salesforce-client-config/{任

```
<?xml version="1.0" encoding="UTF-8"?>
<salesforce-client-config
   xmlns="http://www.intra-mart.jp/system/salesforce/client/config/
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/system/salesforce

   <salesforce-clients>
      <salesforce-client id="sample_1" type="oauth-client">
         <oauth-provider-id>[Please input OAuth provider ID in oa
      </salesforce-client>

      <salesforce-client id="sample_2" type="fixed-user-passwor
         <token-end-point>https://login.salesforce.com/services/o
         <consumer-key>[Please input your Consumer Key]</con
         <consumer-secret>[Please input your Consumer Secret]<
         <user>[Please input your Salesforce user name]</user>
         <password cryption="sample-cryption-category">[Please
         <extra-parameter name="timeout-minutes">[Please inpu
      </salesforce-client>

      <salesforce-client id="sample_3" type="user-password" ter
         <token-end-point>https://login.salesforce.com/services/o
         <consumer-key>[Please input your Consumer Key]</con
         <consumer-secret>[Please input your Consumer Secret]<
         <user-password-resolver>[Please input fully qualified cla
         <extra-parameter name="timeout-minutes">[Please inpu
      </salesforce-client>
   </salesforce-clients>

</salesforce-client-config>
```

リファレンス

Salesforceクライアントの設定

タグ名	salesforce-clients

Salesforce連携時の接続先情報を設定します。

【設定項目】

```
<salesforce-client-config>
 <salesforce-clients>
  ...
 </salesforce-clients>
</salesforce-client-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	
単位・型	なし
省略時のデフォルト値	なし
親タグ	salesforce-client-config

Salesforceクライアントの詳細設定

タグ名	salesforce-client

Salesforceクライアントの詳細を設定します。

【設定項目】

```
<salesforce-clients>
   <salesforce-client id="sample_2" type="fixed-user-password"
      <token-end-point>https://login.salesforce.com/services/oau
      <consumer-key>[Please input your Consumer Key]</consu
      <consumer-secret>[Please input your Consumer Secret]</c

      <user>foo@your_company.com</user>
      <password>foo's password</password>

      <oauth-provider-id>oauth_provider_for_salesforce_sample<
      <user-password-resolver>sample.salesforce.SampleUserPa

      <extra-parameter name="timeout-minutes">60</extra-pa
   </salesforce-client>
</salesforce-clients>
```

必須項目	×	
複数設定	○	
設定値・設定する内容	token-end-point	トークンエンドポイント
	consumer-key	コンシューマ鍵
	consumer-secret	コンシューマの秘密
	user	Salesforceユーザのユーザ
	password	Salesforceユーザのパスワ
	oauth-provider-id	OAuthプロバイダID
	user-password-resolver	UserPasswordResolver の
	extra-parameter	追加パラメータ
単位・型	なし	

省略時のデフォルト値	なし
親タグ	salesforce-clients

【属性】

属性名	説明
id	Salesforceクライアントを識別する一意のID
type	アクセストークンを取得する際の認証タイプを 認証タイプは以下のいずれかを設定してくださ • **oauth-client**：「*OAuth プロバイダ設定* • **fixed-user-password**：指定されたSa セストークンを取得します。 • **user-password**：UserPasswordResc ストークンを取得します。
tenant	利用できる intra-mart Accel Platform の テ 複数の テナント を指定する場合、テナントID テナントID の指定がない場合、全ての テナン

> **コラム**
>
> アクセストークンを取得する際の認証タイプ毎に、Salesforce
> Salesforceのアプリケーション認証に関する詳細は、以下を参
>
> • **oauth-client** の場合
> ◦ OAuth 2.0 Web サーバ認証フロー (日本語)
> ◦ OAuth 2.0 Web Server Authentication Flow (Englis
> ◦ OAuth 2.0 Web 服□器□□流程 (中文)
> • **fixed-user-password**、および、**user-password** の
> ◦ OAuth 2.0 ユーザ名パスワードフロー (日本語)
> ◦ OAuth 2.0 Username-Password Flow (English)
> ◦ OAuth 2.0 用□名密□流程 (中文)

OAuthプロバイダID

タグ名	oauth-provider-id

利用するOAuthプロバイダのIDを設定します。

【設定項目】

```
<salesforce-client>
 <oauth-provider-id>oauth_provider_for_salesforce_sample</oa
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **oauth-client**
複数設定	×
設定値・設定する内容	「*OAuth プロバイダ設定*」に定義されているOAu
単位・型	文字列
省略時のデフォルト値	なし
親タグ	salesforce-client

トークンエンドポイント

タグ名	token-end-point

アクセストークンを取得するためのトークンエンドポイントを設定しま

【設定項目】

```
<salesforce-client>
  <token-end-point>https://login.salesforce.com/services/oauth2/
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **fixed-user-password** • **user-password**
複数設定	×
設定値・設定する内容	アクセストークンを取得するためのトークンエン
単位・型	文字列
省略時のデフォルト値	なし
親タグ	salesforce-client

コンシューマ鍵

タグ名	consumer-key

Salesforceへ接続する際に利用する「コンシューマ鍵」を設定します。

【設定項目】

```
<salesforce-client>
  <consumer-key>Consumer Key</consumer-key>
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **fixed-user-password** • **user-password**
複数設定	×
設定値・設定する内容	Salesforceへ接続する際に利用する「コンシュー OAuth 2.0 では client_id と呼ばれます。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	salesforce-client

> **コラム**
>
> Salesforceへ接続する際に利用する「コンシューマ鍵」の取得
>
> - 接続アプリケーションの作成 (日本語)
> - Creating a Connected App (English)
> - □建□接的□用程序 (中文)

コンシューマの秘密

タグ名	consumer-secret

Salesforceへ接続する際に利用する「コンシューマの秘密」を設定しま
cryption 属性が設定されている場合、指定の方式で暗号化された文字列
cryption 属性に設定するカテゴリは、「*暗号化設定*」を参照してくださ

【設定項目】

```
<salesforce-client>
 <consumer-secret cryption="category">Consumer Secret</cor
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **fixed-user-password** • **user-password**
複数設定	×
設定値・設定する内容	Salesforceへ接続する際に利用する「コンシュー OAuth 2.0 では client_secret と呼ばれます。 *cryption* 属性が設定されている場合、指定の方 *cryption* 属性が設定されていない場合、パスワ
単位・型	なし
省略時のデフォルト値	なし
親タグ	salesforce-client

【属性】

属性名	説明
cryption	暗号化したパスワードを設定する場合に、暗号化方 します。

注意

パスワード文字列を暗号化するためのツールは提供されません
Java API CryptionUtil を利用して、指定したカテゴリで暗号
Java API CryptionUtil の詳細は「CryptionUtil APIドキュメン

コラム

Salesforceへ接続する際に利用する「コンシューマの秘密」の

- 接続アプリケーションの作成 (日本語)
- Creating a Connected App (English)
- □建□接的□用程序 (中文)

Salesforceユーザのユーザ名

タグ名	user

Salesforceユーザのユーザ名を設定します。

【設定項目】

```
<salesforce-client>
 <user>foo@your_company.com</user>
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **fixed-user-password**
複数設定	×
設定値・設定する内容	Salesforceユーザのユーザ名を設定します。 「OAuth 2.0 ユーザ名パスワードフロー」の エ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	salesforce-client

Salesforceユーザのパスワード

タグ名	password

Salesforceユーザのパスワードを設定します。
cryption 属性が設定されている場合、指定の方式で暗号化された文字列
cryption 属性に設定するカテゴリは、「*暗号化設定*」を参照してくださ

【設定項目】

```
<salesforce-client>
 <password cryption="category">salesforce password[&security t
</salesforce-client>
```

必須項目	アクセストークンを取得する際の認証タイプが以 • **fixed-user-password**
複数設定	×
設定値・設定する内容	Salesforceユーザのパスワードを設定します。 「OAuth 2.0 ユーザ名パスワードフロー」 の ユ *cryption* 属性が設定されている場合、指定の方 *cryption* 属性が設定されていない場合、パスワ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	salesforce-client

【属性】

属性名	説明
cryption	暗号化したパスワードを設定する場合に、暗号化方式 します。

注意

Salesforce の REST API を利用する際にセキュリティトーク
トークンは Salesforceで自動生成されたキーです。

例えば、ユーザのパスワードが mypassword で、セキュリテ

セキュリティトークンを取得するには、Salesforce ユーザイ
るか、セキュリティトークンをリセットすると、Salesforce
ンは、ユーザがセキュリティトークンをリセットするか、パス

セキュリティトークンの詳細は、以下の Salesforce オンライ

- セキュリティトークンのリセット (日本語)
- ログイン履歴にAPI セキュリティトークンが必要ですの表
- Reset Your Security Token (English)
- 重置您的安全□□ (中文)

セキュリティトークンを使用せずに Salesforce の REST API
照してください。

- ユーザが Salesforce にログインできる範囲と時間帯の制
- Restrict Where and When Users Can Log In to Sales
- 限制用□可以 Salesforce 的□□与地点 - □□范□内受信任的

注意

パスワード文字列を暗号化するためのツールは提供されません
Java API CryptionUtil を利用して、指定したカテゴリで暗号
Java API CryptionUtil の詳細は「CryptionUtil APIドキュメン

UserPasswordResolver の 実装クラス

タグ名	user-password-resolver

UserPasswordResolver の 実装クラスを設定します。

【設定項目】

```
<salesforce-client>
 <user-password-resolver>sample.salesforce.SampleUserPassw
</salesforce-client>
```

必須項目	×
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.sal
単位・型	UserPasswordResolver を実装し
省略時のデフォルト値	jp.co.intra_mart.system.salesfo UserPasswordResolverImplForI るように実装されています。
親タグ	salesforce-client

注意

intra-mart Accel Platform のパスワード保存方式に「ハッシ
は利用できません。（理由：平文パスワードを取得することが

以下の代替案を検討してください。

- 認証タイプ **oauth-client**、または、**fixed-user-pass**

注意

以下の全ての条件に該当する場合、省略時のデフォルト値「U
由：UserPasswordResolverImplForPasswordCredentialsC

- Salesforce の REST API を利用する際にセキュリティト
- アクセストークンを取得する際の認証タイプが **user-pas**

以下の代替案を検討してください。

- 認証タイプ **oauth-client**、または、**fixed-user-pass**
- パスワードの末尾に Salesforceのセキュリティトークン
- セキュリティトークンを使用せずに Salesforce の REST
  詳細は、以下の Salesforce オンラインヘルプを参照して
  - ユーザが Salesforce にログインできる範囲と時間帯の
  - Restrict Where and When Users Can Log In to Sal
  - 限制用□可以 Salesforce 的□□与地点 - □□范□内受信任的

追加パラメータ

タグ名	extra-parameter

追加情報のパラメータを設定します。

【設定項目】

```
<salesforce-client>
 <extra-parameter name="timeout-minutes">60</extra-param
</salesforce-client>
```

必須項目	×
複数設定	○
設定値・設定する内容	追加情報のパラメータを設定します。
単位・型	なし

省略時のデフォルト値	なし
親タグ	salesforce-client

【属性】

属性名	説明	必須	デフォルト値
name	パラメータ名を設定します。	○	なし

注意

アクセストークンの有効期間を設定するために、タイムアウト

- 認証タイプが **fixed-user-password**、および、**user-**
  - 「*Salesforceクライアントの設定*」の 「追加パラメー

```
<salesforce-client>
  <extra-parameter name="timeout-minutes">60
</salesforce-client>
```

- 認証タイプが **oauth-client** の場合
  - 「*OAuthプロバイダ設定*」の 「追加設定」

```
<extra-config>
  <parameter name="timeout-minutes">60</par
</extra-config>
```

アクセストークンは、Salesforce のセッションタイムアウト
有効期限の切れたアクセストークンを使用すると、「Session

Salesforce のセッションタイムアウト値は、以下の順番で解

1. 接続アプリケーションのセッションポリシーのタイムア
2. ユーザプロファイル内のタイムアウト値
3. 組織の [セッションの設定] のタイムアウト値

タイムアウト値の設定方法は、以下の Salesforce オンライン

- 接続アプリケーションの編集
  - 接続アプリケーションの編集 (日本語)
  - Edit a Connected App (English)
  - □□□接的□用程序 (中文)
- セッションタイムアウト設定
  - プロファイルでのセッションタイムアウト設定の表示と
  - View and Edit Session Timeout Settings in Profiles
  - □看和□□□档中的会□超□□置 (中文)

Kibanaポートレット

Kibana ポートレットの設定

項目

- 概要
- リファレンス
  - Kibana ポートレットの設定
  - Kibana ポートレットの詳細設定
  - カスタムフィールド
  - フィールド
  - 認可の設定
  - ハンドラー実装クラス

概要

Kibana ポートレット連携時の接続先情報を設定します。

モジュール	Kibana ポートレット連携
フォーマットファイル(xsd)	WEB-INF/schema/im-kibana-config.xsd
設定場所	WEB-INF/conf/im-kibana-config/{任意のファ

```
<?xml version="1.0" encoding="UTF-8"?>
<im-kibana-config
   xmlns="http://www.intra-mart.jp/system/im-kibana/config/im-kib
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/system/im-kibana

   <authz action="execute" uri="service://im_kibana_portlet/web_
   <kibana-config-list>
      <kibana-config id="sample_1">
         <root-url>http://localhost:5601</root-url>
         <kibana-version>5.2.0</kibana-version>
      </kibana-config>
      <kibana-config id="sample_2" tenant="default another">
         <root-url>http://localhost:5602</root-url>
         <kibana-version>4.3.1</kibana-version>
         <kibana-meta-index>.kibana</kibana-meta-index>
         <authorizer>sample.Authorizer</authorizer>
         <custom-field>
            <field type="boolean" required="true" key="foo1">bar
            <field required="true" key="foo2">bar2</field>
            <field key="foo3">bar3</field>
         </custom-field>
         <field-mapper>sample.FieldMapper</field-mapper>
         <field-provider>sample.FieldProvider</field-provider>
      </kibana-config>
   </kibana-config-list>
</im-kibana-config>
```

リファレンス

Kibana ポートレットの設定

タグ名	kibana-config-list

Kibana ポートレット連携時の接続先情報を設定します。

【設定項目】

```
<im-kibana-config>
 <kibana-config-list>
  ...
 </kibana-config-list>
</im-kibana-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-kibana-config

Kibana ポートレットの詳細設定

タグ名	kibana-config

Kibana ポートレットの詳細を設定します。

【設定項目】

```
<kibana-config id="sample_2" tenant="default another">
   <root-url>http://localhost:5602</root-url>
   <kibana-version>4.3.1</kibana-version>
   <kibana-meta-index>.kibana</kibana-meta-index>
   <authorizer>sample.Authorizer</authorizer>
   <custom-field>
      <field type="boolean" required="true" key="I18N.MESSAGE.
      <field required="true" key="I18N.MESSAGE.EXAMPLE2">exa
      <field key="I18N.MESSAGE.EXAMPLE3">example3</field>
   </custom-field>
   <field-mapper>sample.FieldMapper</field-mapper>
   <field-provider>sample.FieldProvider</field-provider>
</kibana-config>
```

必須項目	×	
複数設定	○	
設定値・設定する内容	root-url	接続先Kibanaのルート
	kibana-version	接続先Kibanaのバージ
	kibana-meta-index	接続先Kibanaのメタイ
	authorizer	アクセス権のチェック
	custom-field	カスタムフィールド
	field-mapper	フィールド名の物理名
	field-provider	フィールドの一覧を提
単位・型	なし	

省略時のデフォルト値	なし
親タグ	kibana-config-list

【属性】

属性名	説明
id	Kibana ポートレットを識別する一意のIDを設定し
timeout	Kibana ポートレットへ接続する際のコネクション を設定します。
tenant	利用できる intra-mart Accel Platform の テナン 複数の テナント を指定する場合、テナントID は半 てください。 テナントID の指定がない場合、全ての テナント て

カスタムフィールド

タグ名	custom-field

カスタムフィールドを設定します。

【設定項目】

```
<kibana-config>
 <custom-field>
  ...
 </custom-field>
</kibana-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	field　カスタムフィールドの詳細設定
単位・型	なし
省略時のデフォルト値	なし
親タグ	kibana-config

フィールド

タグ名	field

カスタムフィールドの詳細を設定します。

【設定項目】

```
<custom-field>
 <field type="boolean" required="true" key="I18N.MESSAGE.EXA
</custom-field>
```

必須項目	×
複数設定	○
設定値・設定する内容	カスタムフィールドの詳細を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	custom-field

【属性】

属性名	説明
type	カスタムフィールドのタイプを設定します • string, boolean, integer のいずれか
required	このカスタムフィールドが必須かどうかを
key	このカスタムフィールドの名称を取得する ティファイル (conf/message/*.properti

認可の設定

タグ名	authz

認可の設定をします。
通常は変更しません。

【設定項目】

```
<im-kibana-config>
  <authz action-"" uri=""/>
</im-kibana-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Kibana ポートレット への接続を許可する認可の設
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-kibana-config

【属性】

属性名	説明	必須	デフォルト値
action	認可のアクションを指定します。	×	execute
uri	認可のリソースURIを指定します。	×	service://im_kit

ハンドラー実装クラス

タグ名	handler

Kibana ポートレットへ接続する処理を実装したクラスを設定します。
通常は変更しません。

【設定項目】

```
<im-kibana-config>
  <handler />
</im-kibana-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Kibana ポートレットへ接続する処理を実装したク
単位・型	なし

省略時のデフォルト値	jp.co.intra_mart.foundation.kibana.proxy.Kiba
親タグ	im-kibana-config

【属性】

属性名	説明
className	KibanaDefaultReverseProxyHandler を実装し ます。

セッション管理モジュール

セッションストア設定

項目

- 概要
- リファレンス
  - セッションリーパ実行間隔（秒）
  - 強制削除間隔（秒）
  - セッション設定
  - セッションタイムアウト間隔（分）
  - セッションバージョニング
  - セッションCookie設定
  - Cookie名
  - Cookieのdomain設定
  - Cookieのdomain属性
  - Cookieのdomainの正規表現
  - Cookieのpath属性
  - max-age属性
  - httponly属性
  - secure属性

概要

セッション管理モジュールに関する設定です。

モジュール	セッション管理モジュール
フォーマットファイル(xsd)	WEB-INF/schema/session-store-config.xsd
設定場所	WEB-INF/conf/session-store-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<session-store-config
   xmlns="http://intra-mart.co.jp/session-store/session-store-config
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/session-store/sessio

 <reaper-interval>300</reaper-interval>
 <forced-removal-duration>10800</forced-removal-duratio
 <session-config>
  <session-timeout>30</session-timeout>
  <session-versioning>true</session-versioning>
  <cookie-config>
   <name>JSESSIONID</name>
   <domain>
    <name>localhost</name>
   </domain>
   <path>/imart</path>
   <max-age>-1</max-age>
   <http-only>true</http-only>
   <secure>true</secure>
  </cookie-config>
 </session-config>
</session-store-config>
```

リファレンス

セッションリーパ実行間隔（秒）

タグ名	reaper-interval

セッションリーパの実行間隔を設定します。

【設定項目】

```
<session-store-config>
 <reaper-interval>300</reaper-interval>
</session-store-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッションリーパの実行間隔を指定します。
単位・型	秒
省略時のデフォルト値	300
親タグ	session-store-config

コラム
セッションリーパについて
セッションは、時間の経過によって期限切れが発生します。
セッションリーパは、時間の経過によって期限切れとなった
セッション管理モジュールでは、基本的に管理しているセッ
ます。

注意
セッションタイムアウト間隔を長くした場合、セッションが期
行される必要はありません。 よって、セッションタイムアウト

セッションタイムアウト間隔を短くした場合、セッションが期
ください。 セッションリーパの実行間隔を短くすることで、期

強制削除間隔（秒）

タグ名	forced-removal-duration

不要なセッションとして扱う期限を設定します。

【設定項目】

```
<session-store-config>
 <forced-removal-duration>10800</forced-removal-duration
</session-store-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	不要なセッションとして扱う期限を設定します。
単位・型	秒
省略時のデフォルト値	10800
親タグ	session-store-config

コラム
不要なセッションについて
セッションが最後に使われてからこの期限を超えてセッショ
不要なセッションは、期限切れのセッションとして扱われな

セッション設定

タグ名	session-config

セッションに関する情報を設定します。

【設定項目】

```
<session-store-config>
 <session-config>
  ...
 </session-config>
</session-store-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	session-timeout	セッションタイムアウト間隔
	session-versioning	セッションバージョニングの
	cookie-config	Cookieに関する情報を設定
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	session-store-config	

セッションタイムアウト間隔（分）

タグ名	session-timeout

セッションタイムアウト間隔を設定します。

【設定項目】

```
<session-config>
 <session-timeout>30</session-timeout>
</session-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッションタイムアウト間隔を指定します。
単位・型	分
省略時のデフォルト値	30
親タグ	session-config

セッションバージョニング

タグ名	session-versioning

セッションバージョニングを有効するかどうか設定します。

【設定項目】

```
<session-config>
 <session-versioning>true</session-versioning>
</session-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッションバージョニングを有効するかどうか設定し

単位・型	真偽値 (true/false)
省略時のデフォルト値	true
親タグ	session-config

コラム

セッションバージョニングについて

セッション管理モジュールにおいて、セッションは以下で管

- アプリケーションサーバ内のメモリ
- クラスタ間の分散キャッシュ
- 永続化先

クラスタ間の分散キャッシュからセッションを読み込む場合
が発生します。
セッションバージョニングを有効にすると、それぞれが管理
バージョン情報からクラスタ間の分散キャッシュや永続化先

セッション管理モジュールでは、このセッションバージョニ

セッションCookie設定

タグ名	cookie-config

セッションのCookieに関する情報を設定します。

【設定項目】

```
<session-config>
 <cookie-config>
  ...
 </cookie-config>
</session-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	name	セッションIDのCookie名を設定
	domain	セッション関連のCookieのdor す。
	path	セッション関連のCookieのpat
	max-age	セッション関連のCookieのma ます。
	http-only	セッション関連のCookieのhtt す。
	secure	セッション関連のCookieのsec す。
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	session-config	

Cookie名

タグ名	name

セッションIDのCookie名を設定します。

【設定項目】

```
<cookie-config>
 <name>JSESSIONID</name>
</cookie-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッションIDのCookie名を設定します。
単位・型	なし
省略時のデフォルト値	JSESSIONID
親タグ	cookie-config

Cookieのdomain設定

タグ名	domain

セッション関連のCookieのdomain属性に関する情報を設定します。

【設定項目】

```
<cookie-config>
 <domain>
  ...
 </domain>
</cookie-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	name	セッション関連のCookie
	name-pattern	セッション関連のCookie す。
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	cookie-config	

Cookieのdomain属性

タグ名	domain

セッション関連のCookieのdomain属性を設定します。

【設定項目】

```
<domain>
 <name>localhost</name>
</domain>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのdomainを設定します。
単位・型	なし

省略時のデフォルト値	なし
親タグ	domain

Cookieのdomainの正規表現

タグ名	name-pattern

セッション関連のCookieのdomainを正規表現で設定します。

【設定項目】

```
<domain>
 <name-pattern>^.+?\\.(\\w+\\.[a-z]+)$</name-pattern>
</domain>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのdomainを正規表現で設定
単位・型	なし
省略時のデフォルト値	なし
親タグ	domain

Cookieのpath属性

タグ名	domain

セッション関連のCookieのpath属性を設定します。

【設定項目】

```
<cookie-config>
 <path>/imart</path>
</cookie-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのpath属性を設定します。
単位・型	なし
省略時のデフォルト値	コンテキストパスが設定されます。
親タグ	cookie-config

max-age属性

タグ名	domain

セッション関連のCookieのmax-age属性を設定します。

【設定項目】

```
<cookie-config>
 <max-age>0</max-age>
</cookie-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのmax-age属性を設定しま
単位・型	なし
省略時のデフォルト値	-1
親タグ	cookie-config

httponly属性

タグ名	http-only

セッション関連のCookieのhttponly属性を設定します。

【設定項目】

```
<cookie-config>
  <http-only>true</http-only>
</cookie-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのmax-age属性を設定しま
単位・型	真偽値 (true/false)
省略時のデフォルト値	true
親タグ	cookie-config

secure属性

タグ名	secure

セッション関連のCookieのsecure属性を設定します。

【設定項目】

```
<cookie-config>
  <secure>true</secure>
</cookie-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッション関連のCookieのsecure属性を設定します。
単位・型	真偽値 (true/false)
省略時のデフォルト値	false
親タグ	cookie-config

セッションリスナ設定

項目

- 概要
- リファレンス
  - リスナクラス設定
  - リスナパッケージ設定

概要

セッションリスナに関する情報を設定します。

セッション管理モジュールを利用する場合には、ここで設定したクラス

- セッションのライフサイクルに関する通知
- セッション属性の変更に関する通知

モジュール	セッション管理モジュール
フォーマットファイル(xsd)	WEB-INF/schema/session-listener-config.x
設定場所	WEB-INF/conf/session-listener-config/{任

```
<?xml version="1.0" encoding="UTF-8"?>
<session-listener-config
 xmlns="http://intra-mart.co.jp/session-store/session-listener-confi
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://intra-mart.co.jp/session-store/session-

   <!-- HttpSessionListenerを実装したクラス -->
   <listener-class>jp.co.intra_mart.sample.SampleHttpSessionLis
   <!-- HttpSessionAttributeListenerを実装したクラス -->
   <listener-class>jp.co.intra_mart.sample.SampleHttpSessionAtt

   <!-- WebListenerアノテーションが付与されたセッションリスナを検索
   <listener-package>jp.co.intra_mart.sample</listener-packag

</session-listener-config>
```

リファレンス

リスナクラス設定

タグ名	listener-class

HttpSessionListenerまたはHttpSessionAttributeListenerを実装した

【設定項目】

```
<session-listener-config>
  <listener-class>jp.co.intra_mart.sample.SampleHttpSessionListe
</session-listener-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	HttpSessionListenerまたはHttpSessionAtt
単位・型	なし
省略時のデフォルト値	なし
親タグ	session-listener-config

リスナパッケージ設定

タグ名	listener-package

WebListenerアノテーションを付与したセッションリスナを検索するパ
ここで設定したパッケージ配下のクラスのみが検索されます。

【設定項目】

```
<session-listener-config>
  <listener-package>jp.co.intra_mart.sample</listener-package
</session-listener-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	WebListenerアノテーションを付与したセッシ
単位・型	なし
省略時のデフォルト値	なし
親タグ	session-listener-config

注意

セッションリスナは、 intra-mart Accel Platform の起動時に
指定したパッケージ配下の検索対象が多くなると、 intra-mar

Hazelcast設定

項目

- 概要
- リファレンス
  - クラスタグループ設定
  - ID設定
  - 通信スレッド数設定
  - ネットワーク構成設定
  - ポート設定
  - クラスタグループ構成方法設定
  - マルチキャスト通信設定
  - マルチキャストアドレス設定
  - マルチキャスト通信ポート設定
  - ユニキャスト通信設定
  - メンバ設定
  - メンバアドレス設定
  - セッションストア設定
  - セッションストアメモリサイズ設定
  - セッションストア永続化設定
  - データソース設定
  - 永続化情報記録ユーザ設定

概要

intra-mart Accel Platformに組み込んでセッション管理に利用する Ha

コラム

Hazelcast の仕様の詳細については、以下を参照してくださ

「Hazelcast Reference Manual Version: 3.7.5」

- http://docs.hazelcast.org/docs/3.7/manual/html-sing

モジュール	im_session_store_hazelcast_em
フォーマットファイル(xsd)	WEB-INF/schema/hazelcast-config.xsd
設定場所	WEB-INF/conf/hazelcast-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<hazelcast-config
 xmlns="http://intra-mart.co.jp/session-store/hazelcast-config"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://intra-mart.co.jp/session-store/hazelca

 <base>
  <id>im_session_store</id>
 </base>

 <network>
  <port port-count="2">5701</port>
  <join>
   <TCP>
    <member-list>
     <member>127.0.0.1</member>
    </member-list>
   </TCP>
  </join>
 </network>

 <store>
  <max-memory-size>128</max-memory-size>
  <persistence enabled="false">
   <jndi-name>java:comp/env/jdbc/iap-session</jndi-name>
  </persistence>
 </store>

</hazelcast-config>
```

クラスタグループ設定

タグ名	base

クラスタを構成するグループを設定します。

【設定項目】

```
<hazelcast-config>
 <base>
  ...
 </base>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	クラスタグループを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	hazelcast-config

ID設定

タグ名	id

クラスタを構成するグループの一意のIDです。
Webアプリケーション毎に一意となるIDを設定してください。

【設定項目】

```
<hazelcast-config>
 <base>
  <id>im_session_store</id>
 </base>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	クラスタグループのIDを設定します。
単位・型	文字列
省略時のデフォルト値	im_session_store
親タグ	base

通信スレッド数設定

タグ名	io-thread-count

クラスタグループ内の通信に利用するスレッド数を設定します。
ここで設定した値と同数のスレッドが、読み込み用、および書き込み用

Hazelcast における 設定値 hazelcast.io.thread.count に相当します。

【設定項目】

```
<hazelcast-config>
 <base>
  <io-thread-count>3</io-thread-count>
 </base>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	クラスタグループ内の通信に利用するスレッド数を設
単位・型	数値（正の整数）
省略時のデフォルト値	3
親タグ	base

ネットワーク構成設定

タグ名	network

クラスタグループのネットワーク構成に関する設定を行います。

【設定項目】

```
<hazelcast-config>
 <network>
  ...
 </network>
</hazelcast-config>
```

必須項目	○	
複数設定	×	
設定値・設定する内容	port	クラスタグループ内の通信に利用 します。
	join	クラスタグループメンバの発見方法
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	hazelcast-config	

ポート設定

タグ名	port

クラスタグループ内の通信に利用するポート番号を設定します。

【設定項目】

```
<hazelcast-config>
 <network>
  <port port-count="2">5701</port>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	利用するポート番号の最小値を設定します。 このポート番号から、 port-count 属性の値を
単位・型	数値（ポート番号）
省略時のデフォルト値	5701
親タグ	network

【属性】

属性名	説明	必須
port-count	利用するポート番号の範囲を指定します。	×

クラスタグループ構成方法設定

タグ名	join

クラスタグループを構成する方法を設定します。
子要素に <UDP> または <TCP> のいずれか片方を記述してください。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   ...
  <join>
 </network>
</hazelcast-config>
```

必須項目	○	
複数設定	×	
設定値・設定する内容	UDP	マルチキャスト通信でメンバを発見 ます。
	TCP	ユニキャスト通信でメンバを特定す す。
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	network	

マルチキャスト通信設定

タグ名	UDP

マルチキャスト通信でメンバを発見する設定を行います。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <UDP>
    ...
   </UDP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×

複数設定	×
設定値・設定する内容	multicast-group　マルチキャスト通信に利 multicast-port　マルチキャスト通信を送す。
単位・型	なし
省略時のデフォルト値	なし
親タグ	join

> **コラム**
>
> この設定を行った場合、ユニキャストの設定(TCP)は行えませ

マルチキャストアドレス設定

タグ名	multicast-group

マルチキャスト通信に利用するIPアドレスを設定します。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <UDP>
    <multicast-group>224.2.2.3</multicast-group>
    ...
   </UDP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	マルチキャスト通信に利用するIPアドレスを設定しま 224.0.0.0 から 239.255.255.255 の範囲内で指定
単位・型	IPアドレス
省略時のデフォルト値	224.2.2.3
親タグ	UDP

マルチキャスト通信ポート設定

タグ名	multicast-port

マルチキャスト通信を送受信するポート番号を設定します。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <UDP>
    <multicast-port>54327</multicast-port>
    ...
   </UDP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	マルチキャスト通信を送受信するポート番号を設定し
単位・型	数値（ポート番号）
省略時のデフォルト値	54327
親タグ	UDP

ユニキャスト通信設定

タグ名	TCP

ユニキャスト通信でメンバを特定する設定を行います。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <TCP>
    ...
   </TCP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	member-list　クラスタグループのメンバ情報を す。
単位・型	なし
省略時のデフォルト値	なし
親タグ	join

> **コラム**
>
> この設定を行った場合、マルチキャストの設定(UDP)は行えま

メンバ設定

タグ名	member-list

クラスタグループのメンバ情報を列挙します。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <TCP>
    <member-list>
     ...
    </member-list>
   </TCP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	×

設定値・設定する内容	member	メンバのIPアドレスを設定します。
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	TCP	

メンバアドレス設定

タグ名	member

メンバのIPアドレスを設定します。

【設定項目】

```
<hazelcast-config>
 <network>
  <join>
   <TCP>
    <member-list>
     <member>192.168.0.1</member>
     <member>192.168.0.2</member>
     ...
    </member-list>
   </TCP>
  <join>
 </network>
</hazelcast-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	メンバのIPアドレスを設定します。
単位・型	IPアドレス
省略時のデフォルト値	127.0.0.1
親タグ	member-list

セッションストア設定

タグ名	store

セッションストアに関する設定を行います。

【設定項目】

```
<hazelcast-config>
 <store>
  ...
 </store>
</hazelcast-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	max-memory-size	セッションストアのメモリサイズ
	persistence	セッションの永続化について設定

単位・型	なし
省略時のデフォルト値	なし
親タグ	hazelcast-config

セッションストアメモリサイズ設定

タグ名	max-memory-size

セッションストアのメモリサイズを設定します。

【設定項目】

```
<hazelcast-config>
 <store>
  <max-memory-size>
   ...
  </max-memory-size>
 </store>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	セッションストアのメモリサイズを設定します。
単位・型	MB（メガバイト）
省略時のデフォルト値	128
親タグ	store

セッションストア永続化設定

タグ名	persistence

セッションの永続化について設定します。

【設定項目】

```
<hazelcast-config>
 <store>
  <persistence enabled="true">
   ...
  </persistence>
 </store>
</hazelcast-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	jndi-name	データソースのJNDI名を設定します。
	access-user	永続化情報に記録するユーザ名を設定
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	store	

【属性】

属性名	説明	必
enabled	セッションの永続化を行うかどうかを設定します。	×

データソース設定

タグ名	jndi-name

データソースのJNDI名を設定します。

【設定項目】

```
<hazelcast-config>
 <store>
  <persistence enabled="true">
   <jndi-name>java:comp/env/jdbc/iap-session</jndi-name>
   ...
  </persistence>
 </store>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	データソースのJNDI名を設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	persistence

永続化情報記録ユーザ設定

タグ名	access-user

永続化情報に記録するユーザ名を設定します。

【設定項目】

```
<hazelcast-config>
 <store>
  <persistence enabled="true">
   <access-user>im_http_session</access-user>
   ...
  </persistence>
 </store>
</hazelcast-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	永続化情報に記録するユーザ名を設定します。
単位・型	文字列
省略時のデフォルト値	im_http_session
親タグ	persistence

外部メニュー連携

外部メニュー連携接続先設定

項目

- 概要
- リファレンス
  - 外部メニュー連携接続先の共通設定
  - メニュープロバイダー覧の設定
  - メニュープロバイダの設定
    - 外部メニューを全て表示する
    - 任意の外部メニューを表示する
  - メニュープロバイダの接続先エンドポイント設定
  - メニュープロバイダの接続先ログイングループ設定
  - メニュープロバイダの接続先ユーザ設定
  - メニュープロバイダの接続先ユーザパスワード設定

概要

外部メニュー情報を取得するための接続先となるメニュープロバイダ情

モジュール	外部メニュー連携
フォーマットファイル(xsd)	WEB-INF/schema/menu-provider-config.xsd
設定場所	WEB-INF/conf/menu-provider-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<menu-provider-config
  xmlns="http://www.intra-mart.jp/menu/external/menu-provider-c
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/menu/external/menu
  skip-on-error="true" retry-interval="300">

  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      <end-point>http://127.0.0.1:8080/imart/services/MenuServi
      <login-group>default</login-group>
      <user>aoyagi</user>
      <password cryption="default">0A1msI0DLNo=</password
    </menu-provider>
  </menu-providers>

</menu-provider-config>
```

リファレンス

外部メニュー連携接続先の共通設定

タグ名	menu-provider-config

外部メニュー連携処理に関する共通項目を設定します。

【設定項目】

```
<menu-provider-config skip-on-error="true" retry-interval="300
  ...
</menu-provider-config>
```

必須項目	○
複数設定	×

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	なし

【属性】

属性名	説明
skip-on-error	接続先への接続エラー時に次のメニューフ • *false* を設定した場合、接続エラーが • *true* を設定した場合、接続エラーが す。 その場合、このメニュープロバ
retry-interval	接続先への接続エラー発生によりメニュー す。

コラム

retry-interval について

skip-on-error が *true* の場合、接続エラーが発生したメニュー

エラー発生後、*retry-interval* に設定した時間が経過すると、
その後、メニュー情報取得時に、再び外部メニュー連携により

この値に *0* またはマイナスの値を設定した場合、一度エラーが

グローバルナビのメニュー情報はキャッシュされています。
そのため *retry-interval* が経過しても、メニューキャッシュを

メニュープロバイダー一覧の設定

タグ名	menu-providers

外部メニュー連携の接続先となるメニュープロバイダ情報をまとめる設

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    ...
  </menu-providers>
</menu-provider-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし

単位・型	なし
省略時のデフォルト値	なし
親タグ	menu-provider-config

メニュープロバイダの設定

タグ名	menu-provider

外部メニュー連携の接続先となるメニュープロバイダを設定します。

メニュープロバイダの設定は複数設定することが可能です。

外部メニュー連携機能では、グローバルナビ、サイトマップ表示時に、
取得したメニューツリー情報は、メニュークライアントのメニューツリ

外部メニューを全て表示する

editable 属性を設定しなかった場合、取得したメニューツリー情報はメ
グローバルナビでは、取得した全てのメニューツリー情報の後ろに「サ

表示順序を変更したい場合は、この設定項目の記述順序を変更してくだ

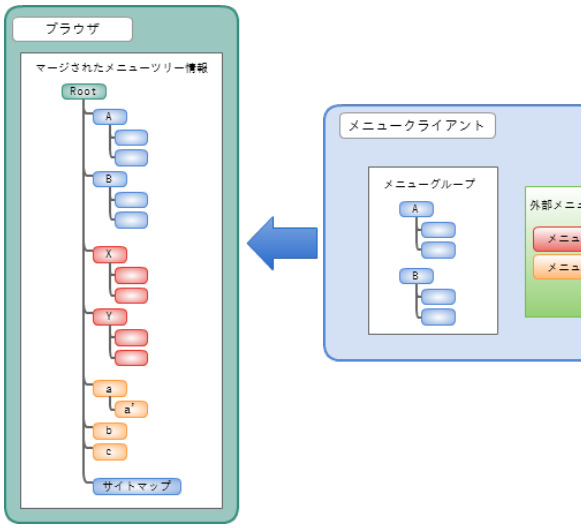


図 外部メニュー連携におけるメニューツリー表示順序

任意の外部メニューを表示する

editable 属性に「true」を設定した場合、取得したメニューツリー情報

メニュー設定画面では、取得したメニューツリー情報のトップ階層をメ
設定した外部メニューの表示位置情報は、「外部メニューリンク」とし

グローバルナビの表示時にメニュープロバイダからメニューツリー情報

メニュー設定画面で設定されなかったメニューツリー情報は、表示され

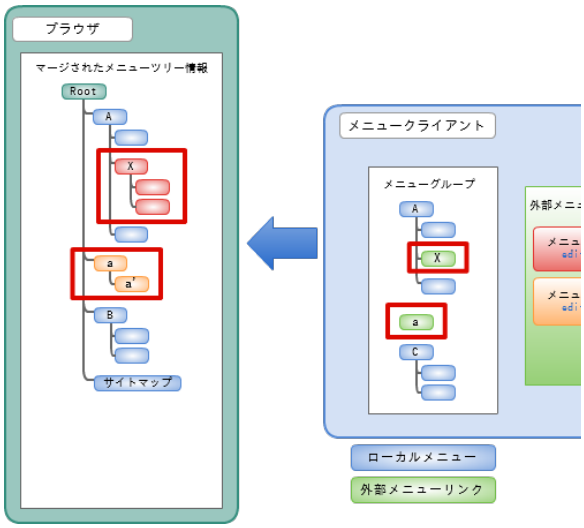


図 任意の外部メニューリンクを表示する場合のメニューツリー表示

> **コラム**
>
> メニュー設定画面での表示順序の設定については「テナント管

> **コラム**
>
> 外部メニューを任意の位置に表示する設定は、グローバルナビ
> サイトマップおよび互換テーマのメニューツリーでは、メニュ

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      ...
    </menu-provider>
  </menu-providers>
</menu-provider-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	menu-providers

【属性】

属性名	説明
id	メニュープロバイダを特定する、システム この値は全てのメニュープロバイダで重複
target-tenant	メニュープロバイダの接続元を指定するテ この設定で指定したテナントにのみ、この 省略した場合、デフォルト値はデフォルト

属性名	説明
enable	メニュープロバイダ有効フラグです。 *true* メニュープロバイダを利用しま *false* メニュープロバイダを利用しま ん。
editable	メニュープロバイダ管理フラグです。 この 置を変更することが可能となります。 *true* メニュー管理画面でこのプロバ *false* メニュー管理画面でこのプロバ
use-popup	ポップアップ表示フラグです。 リンクをクリックした場合に、別ウィンド *true* ポップアップで表示します。 *false* IFrame で表示します。
support-anonymous	未認証ユーザの外部メニュー連携利用フラ メニュープロバイダが intra-mart Accel 「*true*」を設定します。 *true* 未認証の場合でも外部メニュー す。 *false* 未認証の場合は外部メニュー連 ん。

メニュープロバイダの接続先エンドポイント設定

タグ名	end-point

メニュープロバイダの管理者から提供された接続先エンドポイントを設
接続先エンドポイントは、Webサービス「*MenuService*」のエンドポイ

エンドポイントのフォーマットは、以下となります。

http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuServi

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      <end-point>http://127.0.0.1:8080/imart/services/MenuServi
        ...
    </menu-provider>
  </menu-providers>
</menu-provider-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メニュープロバイダの管理者から提供された接続先
単位・型	文字列（URL）
省略時のデフォルト値	なし
親タグ	menu-provider

メニュープロバイダの接続先ログイングループ設定

タグ名	login-group

メニュープロバイダの管理者から提供された接続先ログイングループID,
このタグが必要な場合は、以下の通りです。

- 接続先のメニュープロバイダが intra-mart WebPlatform の場合
- 接続先のメニュープロバイダが intra-mart Accel Platform でかつ

> **コラム**
>
> 接続するメニュープロバイダがintra-mart Accel Platformの
> ログイングループIDの詳細については、「Webサービス 認証
> い。

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      ...
      <login-group>default</login-group>
      ...
    </menu-provider>
  </menu-providers>
</menu-provider-config>
```

必須項目	×（接続先のメニュープロバイダに依存します）
複数設定	×
設定値・設定する内容	メニュープロバイダの管理者から提供されたログイ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	menu-provider

メニュープロバイダの接続先ユーザ設定

タグ名	user

メニュープロバイダの管理者から提供された接続先の接続用ユーザコー

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      ...
      <user>aoyagi</user>
      ...
    </menu-provider>
  </menu-providers>
</menu-provider-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メニュープロバイダの管理者から提供された接続用ユ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	menu-provider

メニュープロバイダの接続先ユーザパスワード設定

タグ名	password

メニュープロバイダの管理者から提供された接続先の接続用ユーザパス

cryption 属性が設定されている場合、指定の方式で暗号化された文字列
cryption 属性に設定するカテゴリは、「*暗号化設定*」を参照してくださ

【設定項目】

```
<menu-provider-config>
  <menu-providers>
    <menu-provider id="sample" target-tenant="default" enable
      ...
      <password cryption="default">0A1msI0DLNo=</password
    </menu-provider>
  </menu-providers>
</menu-provider-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メニュープロバイダの管理者から提供された接続 *cryption* 属性が設定されている場合、指定の方 *cryption* 属性が設定されていない場合、パスワ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	menu-provider

【属性】

属性名	説明
cryption	暗号化したパスワードを設定する場合に、暗号化方式 リです。

注意

パスワード文字列を暗号化するためのツールは提供されません
Java API CryptionUtil を利用して、指定したカテゴリで暗号化

CryptionUtil APIドキュメント

SSO連携用マッピング設定

項目

- 概要
  - IM-HybridSSO の共通設定
  - サイトドメイン設定
  - 再実行時間設定
  - HttpOnly設定
  - ログアウトタイムアウト設定
  - SSOマッピング一覧の設定
  - SSOマッピング設定
  - SSO認証プロバイダ対象テナント設定
  - SSOサービスプロバイダ設定
  - SSOサービスプロバイダID設定
  - 認証情報取得の接続先エンドポイント設定
  - SSOサービスプロバイダ対象ログイングループ設定
  - Webサービス実行ユーザ設定
  - Webサービス実行ユーザパスワード設定
  - ログアウト先URL
  - サイトパス設定

概要

intra-mart Accel Platform（SSO認証プロバイダ） から iWP / iAF（S
IM-HybridSSO の詳細は、「セットアップガイド」-「iAP-iWP間SSO連

モジュール	iAP-iWP間SSO連携モジュール(IM-HybridSSO)
フォーマットファイル(xsd)	WEB-INF/schema/hybrid-sso-mapping-config
設定場所	WEB-INF/conf/hybrid-sso-mapping-config.xm

```
<?xml version="1.0" encoding="UTF-8"?>
<hybrid-sso-mapping-config
 xmlns="http://www.intra-mart.jp/hybrid_sso/hybrid-sso-mapping-c
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intra-mart.jp/hybrid_sso/hybrid-s

 <common-settings>
  <site-domain>example.com</site-domain>
  <retry-interval>300</retry-interval>
  <http-only>true</http-only>
  <logout-timeout>60</logout-timeout>
 </common-settings>

 <mappings>
  <mapping id="sample">
   <tenant-id>tenant-a</tenant-id>
   <provider>
    <provider-id>provider-0</provider-id>
    <end-point>http://intra-mart.example.com:8080/imart/servic
    <login-group>default</login-group>
    <user>aoyagi</user>
    <password cryption="category">0A1msI0DLNo=</passwor
    <logout-url>http://intra-mart.example.com:8080/imart/user.
    <site-path>/imart</site-path>
   </provider>
  </mapping>
 </mappings>

</hybrid-sso-mapping-config>
```

IM-HybridSSO の共通設定

タグ名	common-settings

IM-HybridSSO に関する共通項目を設定します。

【設定項目】

```
<hybrid-sso-mapping-config>
 <common-settings>
  ...
 </common-settings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	hybrid-sso-mapping-config

サイトドメイン設定

タグ名	site-domain

サイトドメインの設定を行います。
IM-HybridSSO では、認証情報の管理に Cookie を利用します。
この設定では認証情報を格納した Cookie を参照可能なサイトドメイン

IM-HybridSSO を構成するサーバは、全て同一のドメイン上に構築する
以下の構成の場合、サイトドメインに設定する値は「intra-mart.jp」と

- SSO認証プロバイダ： iap.intra-mart.jp
- SSOサービスプロバイダ１： iwp1.intra-mart.jp
- SSOサービスプロバイダ２： iwp2.intra-mart.jp

【設定項目】

```
<hybrid-sso-mapping-config>
 <common-settings>
  <site-domain>example.com</site-domain>
 </common-settings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	サイトドメインを指定します。
単位・型	文字列

省略時のデフォルト値	なし
親タグ	common-settings

再実行時間設定

タグ名	retry-interval

SSOサービスプロバイダ へのWebサービスの再実行間隔を設定します。
SSO認証プロバイダ はユーザがログイン時にSSO連携用マッピング設定
このWebサービスで接続エラーが発生した場合、一定の時間が経つまで
再実行時間設定では、接続エラーが発生した際に再度 認証情報の取得を

この値に 0 を設定した場合、一度接続エラーが発生した SSOサービス
また、マイナスの値を設定した場合は、デフォルト値（300）が採用さ

【設定項目】

```
<hybrid-sso-mapping-config>
 <common-settings>
  <retry-interval>300</retry-interval>
 </common-settings>
</hybrid-sso-mapping-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	認証情報取得時の接続エラー発生後にふたた
単位・型	数値（秒）
省略時のデフォルト値	300
親タグ	common-settings

HttpOnly設定

タグ名	http-only

HttpOnly属性を利用するかどうかを設定します。
IM-HybridSSO では、認証情報の管理に Cookie を利用します。
この設定では認証情報を格納する Cookie に HttpOnly属性を指定する

【設定項目】

```
<hybrid-sso-mapping-config>
 <common-settings>
  <http-only>true</http-only>
 </common-settings>
</hybrid-sso-mapping-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	true	CookieにHttpOnly属性を付与します。
	false	CookieにHttpOnly属性を付与しません。
単位・型	真偽値（true/false）	
省略時のデフォルト値	true	
親タグ	common-settings	

ログアウトタイムアウト設定

タグ名	logout-timeout

SSOサービスプロバイダ へのログアウト実行時のタイムアウト時間を設
IM-HybridSSO では、SSO認証プロバイダ にてログアウトを行うと SS
ログアウト時には SSOサービスプロバイダ でのログアウト処理の完了を
ログアウトタイムアウト設定では、SSOサービスプロバイダ でのログア
SSOサービスプロバイダからログアウト時に、ログアウトタイムアウト

この値に 0 を設定した場合、SSOサービスプロバイダ でのログアウトを
また、マイナスの値を設定した場合は、デフォルト値（60）が採用され

【設定項目】

```
<hybrid-sso-mapping-config>
 <common-settings>
  <logout-timeout>60</logout-timeout>
 </common-settings>
</hybrid-sso-mapping-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SSOサービスプロバイダのログアウトの最大応答
単位・型	数値（秒）
省略時のデフォルト値	60
親タグ	common-settings

SSOマッピング一覧の設定

タグ名	mappings

IM-HybridSSO 接続を行う SSO認証プロバイダ と SSOサービスプロバ

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  ...
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	hybrid-sso-mapping-config

SSOマッピング設定

タグ名	mapping

SSO接続を行う SSO認証プロバイダ と SSOサービスプロバイダのSSO

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   ...
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	mappings

【属性】

属性名	説明
id	IM-HybridSSO のマッピングを特定する、システムでIDです。 この値はすべてのマッピングで重複しないようにして
enable	マッピングの有効フラグです。 true　マッピングを有効とします。 false　マッピングを無効とします。

SSO認証プロバイダ対象テナント設定

タグ名	tenant-id

SSOマッピング設定に対して、SSO認証プロバイダとして動作する対象

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <tenand-id>tenant-a</tenant-id>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SSO認証プロバイダとして動作する対象となるテナン
単位・型	文字列
省略時のデフォルト値	なし
親タグ	mapping

SSOサービスプロバイダ設定

タグ名	provider

SSOマッピング設定に対して、SSOサービスプロバイダの情報を設定し

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    ...
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	mapping

SSOサービスプロバイダID設定

タグ名	provider-id

SSOサービスプロバイダのプロバイダIDを設定します。
プロバイダIDは SSOサービスプロバイダ側で設定ファイルで定義したID

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <provider-id>provider-0</provider-id>
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SSOサービスプロバイダのプロバイダID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	provider

認証情報取得の接続先エンドポイント設定

タグ名	end-point

SSOサービスプロバイダから認証情報を取得するための接続先エンドポ
接続先エンドポイントは、Webサービス「*AdmissionService*」のエン

通常、エンドポイントのURLは以下となります。

http://<HOST>:<PORT>/<CONTEXT_PATH>/services/Admission

または

https://<HOST>:<PORT>/<CONTEXT_PATH>/services/Admissio

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <end-point>http://intra-mart.example.com:8080/imart/servic
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SSOサービスプロバイダから認証情報を取得する
単位・型	文字列（URL）
省略時のデフォルト値	なし
親タグ	provider

SSOサービスプロバイダ対象ログイングループ設定

タグ名	login-group

SSOサービスプロバイダとして動作するログイングループを設定します。

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <login-group>default</login-group>
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SSOサービスプロバイダとして動作するログイングル
単位・型	文字列
省略時のデフォルト値	なし
親タグ	provider

Webサービス実行ユーザ設定

タグ名	user

SSOサービスプロバイダ から認証情報を取得するWebサービスを実行す
ここで設定するユーザは SSOサービスプロバイダ にて WebサービスA
Webサービス実行用に新しくユーザを作成してWebサービス実行用のロ

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <user>aoyagi</user>
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	認証情報を取得するWebサービスを実行するユーザ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	provider

Webサービス実行ユーザパスワード設定

タグ名	password

SSOサービスプロバイダから認証情報を取得するWebサービスを実行す

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <password cryption="category">0A1msl0DLNo=</passwor
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	認証情報を取得するWebサービスを実行するユー *cryption* 属性を設定することで、パスワードの値
単位・型	文字列
省略時のデフォルト値	なし
親タグ	provider

【属性】

属性名	説明
cryption	パスワードの暗号化方式のカテゴリを指定します。 暗号化方式のカテゴリは「*暗号化設定*」で設定した値です。 この属性が未指定の場合は、暗号化なしでパスワードす。

> **注意**
>
> パスワード文字列を暗号化するためのツールは提供されません
> *Java API CryptionUtil* を利用して、指定したカテゴリで暗号化
>
> CryptionUtil APIドキュメント

ログアウト先URL

タグ名	logout-url

SSOサービスプロバイダからログアウトを行うためのURLを指定します。
ログアウト先URLは、SSOサービスプロバイダのログアウトサーブレッ

通常、ログアウトサーブレットのURLは以下となります。

http://<HOST>:<PORT>/<CONTEXT_PATH>/user.logout

または

https://<HOST>:<PORT>/<CONTEXT_PATH>/user.logout

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <logout-url>http://intra-mart.example.com:8080/imart/user.
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SSOサービスプロバイダからログアウトを行うための

単位・型	文字列（URL）
省略時のデフォルト値	なし
親タグ	provider

サイトパス設定

タグ名	site-path

サイトパスを設定します。
IM-HybridSSO では、認証情報の管理に Cookie を利用します。
この設定では認証情報を格納した Cookie を参照可能なサイトパス（Co

通常、サイトパスは、/ + <CONTEXT_PATH> の値となります。

【設定項目】

```
<hybrid-sso-mapping-config>
 <mappings>
  <mapping id="sample">
   <provider>
    <site-path>/imart</site-path>
   </provider>
  </mapping>
 </mappings>
</hybrid-sso-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	サイトパス
単位・型	文字列（/xxxxx）
省略時のデフォルト値	なし
親タグ	provider