



目次

- 改訂情報
- はじめに
 - 本書の目的
 - 対象読者
 - 本書の構成
- テーマ概要
 - テーマとは
 - 画面レイアウトやスタイルを切り替える仕組み
 - テーマモジュール
 - 用語
- テーマの詳細
 - ThemeFilter
 - ThemeManager
 - PageBuilder
 - テーマモジュール
- テーマモジュールの作成
 - 標準テーマカスタマイズの概要
 - スクラッチ開発
 - サンプル
- 付録
 - テーマモジュールのフォルダ・ファイル構成
 - テーマモジュールのサンプル
 - ユーティリティプラグイン
 - intra-mart が提供するテーマの一覧

改訂情報

変更年月日	変更内容
2013-10-01	初版
2014-04-01	第2版 下記を追加・変更しました <ul style="list-style-type: none"> ▪ 「テーマモジュール」の「JSSP」からGoogleChromeFrameの記述を削除 ▪ 「テーマモジュールのサンプル」の「JSSP」からGoogleChromeFrameの記述を削除
2014-12-01	第3版 下記を変更しました <ul style="list-style-type: none"> ▪ 「テーマモジュール」に「ヘルプドロップダウン」を追加しました。 ▪ 「テーマモジュール」の「header」の画像を変更しました。
2015-08-01	第4版 下記を変更しました <ul style="list-style-type: none"> ▪ 「PageBuilder」の「設定ファイルで指定する」に FullThemeBuilder の設定ファイルを追記しました。
2015-12-01	第5版 下記を変更しました <ul style="list-style-type: none"> ▪ 「PageBuilder」に「ライブラリ群の切り替え」を追加しました。 ▪ 「テーマモジュール」に「ライブラリ群の切り替え」を追加しました。 ▪ 「テーマモジュールのサンプル」の「head」を修正しました。
2016-08-01	第6版 下記を変更しました <ul style="list-style-type: none"> ▪ 「PageBuilder」の「PageBuilder の役割」に設定ファイルリファレンスへのリンクをコラムとして追加しました。 ▪ 「ThemeFilter」の「キャッシュ制御」の説明を更新しました。

はじめに

本書の目的

本書ではテーマの詳細について説明します。

説明範囲は以下のとおりです。

- テーマの持つ機能の全体像
- テーマを実現する構造と動作
- テーマモジュールの作成方法

対象読者

本書では次の利用者を対象としています。

- intra-mart Accel Platform に画面を持つアプリケーションを作成したい開発者の方
- テーマモジュールを作成・カスタマイズしたいデザイナーの方

本書の構成

- [テーマ概要](#)

テーマの持つ機能の全体像について説明します。

- [テーマの詳細](#)

概要で説明したテーマの全体像を実現するための構成要素を説明します。

- [ThemeFilter](#)

テーマの入り口となるサーブレットフィルタについて説明します。

- [ThemeManager](#)

テーマモジュールを管理する部分について説明します。

- [PageBuilder](#)

テーマモジュールとコンテンツを組み合わせたHTMLを生成する部分について説明します。また、テーマモジュールとコンテンツとの組み合わせを決定する方法についても説明します。

- [テーマモジュール](#)

テーマモジュールを説明します。

- [テーマモジュールの作成](#)

テーマモジュールの作成方法について説明します。

テーマ概要

テーマとは

テーマとは

- 画面レイアウトやスタイルを切り替える仕組み
- その構成ファイル群

を指します。

読者の立場によって、見え方が異なるものになります。

アプリケーションの開発者の立場からは、画面レイアウトやスタイルを切り替える仕組みとしての側面が主なものとなるでしょう。

デザイナーの立場からは、構成ファイル群、特に HTML と CSS とで画面のデザインを行う対象となるでしょう。

ユーザの立場からは、画面の見た目や操作感の違い、として見えるでしょう。

画面レイアウトやスタイルを切り替える仕組み

画面レイアウトやスタイルを切り替える仕組みは、主に以下の機能で実現します。

- [ThemeFilter](#)
- [ThemeManager](#)
- [PageBuilder](#)

主にアプリケーションの開発者が関係する部分になります。これらの章を読むことで、設定ファイルの書き方や、プログラムからの指定によってテーマモジュールの組み合わせ方を制御できるようになります。

テーマモジュール

テーマを構成するファイル群を [テーマモジュール](#) と呼びます。

テーマモジュールは、[JSSP](#)、CSS、CSJS、画像、設定ファイル で構成されます。

テーマモジュールには

- 標準
- 標準 (シンプル)
- v5 互換
- v6 互換

の4種類があります。標準 (シンプル) 以外のテーマモジュールには、いくつかのカラーバリエーションがあります。

用語

JSSP

JSSP とは JavaScriptServerPage の略称で、スクリプト開発モデルで実装された HTML と JavaScript の組み合わせを指します。

CSJS

CSJS とは ClientSideJavaScript の略称で、クライアントつまり Web ブラウザ上で動作する JavaScript のことを指します。

SSJS

SSJS とは ServerSideJavaScript の略称で、サーバ上で動作する JavaScript のことを指します。

テーマの詳細

[テーマ概要](#) で述べたとおり、テーマは大きく分けて 2 つの部分でできています。

- 画面レイアウトやスタイルを切り替える仕組み
 - ThemeFilter
 - ThemeManager
 - PageBuilder
- テーマモジュール

以降の説明ではこれらの機能の詳細を解説していきます。

ThemeFilter

ThemeFilter は javax.servlet.Filter を実装した Java のクラスです。*.jsp と JSP に対してマッピングされていません。

ここでは ThemeFilter の役割と制御方法を説明します。

項目

- ThemeFilter の役割
 - テーマモジュールを適用する条件
 - HTML の生成
 - キャッシュ制御
- 制御パラメータ
 - キャッシュ制御
 - テーマの適用制御
 - PageBuilderの制御

ThemeFilter の役割

ThemeFilter は以下の機能を持っています。

- リクエストされたページの URL やパラメータなどをチェックし、テーマモジュールを適用するかどうかを決定します。
- テーマモジュールを適用する場合、コンテンツに対してテーマモジュールを適切に適用し、HTMLを生成します。
- レスポンスに、キャッシュ制御の HTTP ヘッダを付与します。

テーマモジュールを適用する条件

ThemeFilter は、コンテンツに対してテーマモジュールを適用するかどうかを決定します。以下の条件のいずれかを満たす場合、テーマモジュールを適用します。

- Content-Type が指定されていない
- Content-Type が text/html である

なお、後述の [テーマの適用制御](#) で、フラグに false を指定した場合、上記の条件に合致してもテーマモジュールは適用されません。

HTML の生成

ThemeFilter はテーマモジュールを適用する条件に合致した場合、コンテンツをバイト配列としてメモリ内に保存し、後述の *PageBuilder* に渡します。ThemeFilter は PageBuilder が生成した HTML をレスポンスとして Web ブラウザに送信します。

テーマモジュールを適用する条件に合致しない場合、レスポンスの操作は行いません。対象のサーブレット、サーブレットフィルタが生成したレスポンスがそのまま Web ブラウザに戻ることになります。

キャッシュ制御

HTTP ヘッダに、以下のものをセットします。

- Cache-Control: no-store
- Pragma: no-cache

なお、後述の *キャッシュ制御* で、フラグに true をセットした場合、これらの HTTP ヘッダはセットされません。

制御パラメータ

ThemeFilter には、動作を変更するためのパラメータが存在します。

キャッシュ制御

Cache-Control に no-store, Pragma に no-cache を指定するかどうかを制御します。

このパラメータを指定しない場合、または false を指定した場合、キャッシュ制御の HTTP ヘッダがレスポンスに付与されます。true を指定した場合、キャッシュ制御の HTTP ヘッダはレスポンスに付与されません。

このパラメータは、リクエストの属性として指定してください。

JavaScript の例

```
function init(request) {
  //キャッシュ制御の HTTP ヘッダを出力しない
  request.setAttribute('x-jp-co-intra-mart-disable-no-cache', true);
  //キャッシュ制御を自ら行う
  let response = Web.getHTTPResponse();
  response.setHeader("Cache-Control", "private,max-age=3600");
  response.setHeader("Last-Modified", new Date(2013,9,1,13,0,0));
}
```

Java の例

```
request.setAttribute(ThemeFilter.DISABLE_NO_CACHE, true);
response.setHeader("Cache-Control", "private,max-age=7200");
```

テーマの適用制御

テーマモジュールを適用するかどうかを制御します。

このフラグに false を指定した場合、テーマ適用の条件に合致していてもテーマモジュールは適用されません。true を指定した場合、テーマモジュールを適用するかどうかはテーマ適用の条件に従います。

このパラメータは、リクエストのパラメータとして指定してください。

クエリパラメータとして指定することで、一時的にテーマモジュールを適用させないで画面表示を確認する場合に使うことを想定しています。

```
http://xxx.xxx.xxx/imart/test?x-jp-co-intra-mart-apply-theme=false
```

PageBuilderの制御

セッションをスコープとした PageBuilder のビルダーモジュールを指定します。

通常は、リクエストをスコープとしてビルダーモジュールが決まりますが、このパラメータを指定するとセッションをスコープとしてビルダーモジュールを指定することができます。これは、外部メニューに指定された intra-mart Accel Platform の画面を現在のテーマで表示するためのパラメータです。

このパラメータは、リクエストのパラメータとして指定してください。

```
<form action="somewhere">
  ...
  <input type="hidden" name="imui-session-scope-builder-module" value="headwithcontainer"/>
  ...
</form>
```

ThemeManager

ここでは ThemeManager の役割を説明します。

ThemeManager の役割

テーマモジュールの情報を管理します。システムに登録されているすべてのテーマモジュールの情報、ログインしているユーザのテーマの情報などを取得する機能を提供します。

詳細は、[API リスト](#) を参照してください。

PageBuilder

ここでは PageBuilder の役割と、制御方法を説明します。

項目

- PageBuilder の役割
- 組み合わせ方の制御
 - 設定ファイルで指定する
 - リクエストへのパラメータで指定する
 - 適用順位
 - 指定例
 - 設定ファイルで指定する例
 - リクエストへの属性として指定する例
 - リクエストへのパラメータとして指定する例
- ライブラリ群の切り替え
 - 適用順位
 - 指定例
 - 設定ファイルで指定する例
 - リクエストへのパラメータとして指定する例
 - リクエストへの属性として指定する例

PageBuilder の役割

PageBuilder は、テーマモジュールの JSP と、コンテンツを組み合わせた HTML を生成します。

テーマモジュールは、以下の 4 つの JSP で構成されています。

- head
 - HTML の head タグの部分
- header
 - ヘッダ部
- body
 - ボディ部
- footer
 - フッタ部

組み合わせ方は、

- head, header, body, footer
- head, body, footer
- head, body
- body
- テーマ適用無し

の 5 パターンであるものと定義しています。

実装は以下の 6 つがあります。

1. HeadWithFooterThemeBuilder

- head, body, footer を含んだ HTML を生成します。
- header (メニューや、ユーティリティ) を表示したくないが、footer は表示したい場合に使用します。
- body は、<div id="imui-container"> で囲まれて出力されます。

2. HeadWithContainerThemeBuilder

- head, body を含んだ HTML を生成します。
- header (メニューや、ユーティリティ)、footer を表示したくないが、CSS やクライアントサイド JavaScript は使用したい場合に使用します。
- 主に intra-mart Accel Platform 向けに作成した画面を表示するために使用することを想定しています。
- body は、`<div id="imui-container">` で囲まれて出力されます。

3. HeadOnlyThemeBuilder

- head, body を含んだ HTML を生成します。
- header (メニューや、ユーティリティ)、footer を表示したくないが、CSS やクライアントサイド JavaScript は使用したい場合に使用します。
- 主に iWP7.2 以前のシステム向けに作成した画面を表示するために使用することを想定しています。
- body は、指定された URL の HTML そのものが出力されます。

4. BodyOnlyThemeBuilder

- DOCTYPE、htmlタグ、body を含んだ HTML を生成します。
- header (メニューや、ユーティリティ)、footer を表示せず、CSS やクライアントサイド JavaScript も使用しない場合に使用します。
- body は、指定された URL の HTML そのものが出力されます。

5. NoThemeBuilder

- 指定された URL の HTML をそのまま返します。
- テーマを一切使用せず、自分で作成した HTML をそのまま出力したい場合に使用します。
- body は、指定された URL の HTML そのものが出力されます。

6. FullThemeBuilder

- head, header, body, footer のすべてを含んだ HTML を生成します。
- body は、`<div id="imui-container">` で囲まれて出力されます。
- 基本はこれを使用します。

上記の順に処理すべき PageBuilder を検索し、その PageBuilder がリクエストを処理します。リクエストを処理すべきかどうかは、それぞれのモジュールが持つ設定ファイルに記載されたパスがリクエストパスに合致するかどうかや、後述のパラメータなどに合致するかどうかで判断します。

リクエストパスが設定ファイルに合致しない場合や、パラメータで指定されていない場合、FullThemeBuilder がリクエストを処理します。



コラム

設定ファイルについては、「[設定ファイルリファレンス](#)」 - 「UI」より各テーマビルダーのドキュメントを参照してください。



コラム

CSSモジュール一覧内のスタイルの一部は、`<div id="imui-container">` の内部の要素だけに適用されます。この div で内容が囲まれない PageBuilder (HeadOnlyThemeBuilder, BodyOnlyThemeBuilder, NoThemeBuilder) を利用し、かつ CSSモジュール一覧のスタイルを適用したい場合は、`<div id="imui-container">` で内容を囲むように実装してください。

組み合わせ方の制御

head, header, body, footer の組み合わせは上述の PageBuilder の6つの実装の設定で決まります。それぞれの設

定ファイルを記述したり、リクエストへパラメータを指定したりすることで、どの組み合わせ方にするかを指定することができます。

設定ファイルで指定する

どの組み合わせ方にするかが静的に決定する場合、設定ファイルに記述します。

設定ファイルは、WEB-INF/conf 配下の PageBuilder の実装毎のフォルダに配置します。ファイル名は任意です。

- HeadWithFooterThemeBuilder
 - WEB-INF/conf/theme-head-with-footer-path-config
- HeadWithContainerThemeBuilder
 - WEB-INF/conf/theme-head-with-container-path-config
- HeadOnlyThemeBuilder
 - WEB-INF/conf/theme-head-only-path-config
- BodyOnlyThemeBuilder
 - WEB-INF/conf/theme-body-only-path-config
- NoThemeBuilder
 - WEB-INF/conf/theme-no-theme-path-config
- FullThemeBuilder
 - WEB-INF/conf/theme-full-theme-path-config



注意

それぞれの設定ファイルは異なる XML Schema で定義されています。いずれかの設定ファイルを別のフォルダにコピーしても動作しないので注意してください。

リクエストへのパラメータで指定する

どの組み合わせ方にするかが動的に決定する場合や、forward する場合、リクエストへパラメータを指定します。

forward を行うと、PageBuilder が処理対象とする URL は forward 前の URL となります。forward 後のページに対して forward 前の PageBuilder とは別の PageBuilder を指定したい場合、リクエストにパラメータを指定することで PageBuilder を切り替えることができます。

指定するキー imui-theme-builder-module

適用したい PageBuilder	指定する値
HeadWithFooterThemeBuilder	headwithfooter
HeadWithContainerThemeBuilder	headwithcontainer
HeadOnlyThemeBuilder	headonly
BodyOnlyThemeBuilder	bodyonly
NoThemeBuilder	notheme

上記の値をリクエストのパラメータ、または属性として指定することで PageBuilder が切り替わります。

適用順位

設定ファイル、パラメータ、属性の適用は、以下の順に検索し、最初に合致した PageBuilder を使用します。設定ファイルに記述したものより、属性に指定したものの方が優先されます。

1. 属性
2. パラメータ
3. 設定ファイル

指定例

設定ファイルで指定する例

例として、`http://hostname/iap/sample/page` へのリクエストを head, body, footer を含んだ HTML としたい場合を取り上げます。この場合、使用する PageBuilder は、HeadWithFooterThemeBuilder になります。

HeadWithFooterThemeBuilder の設定ファイルは以下のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-head-with-footer-path-config xmlns="http://www.intra-mart.jp/theme/theme-head-with-footer-path-config">
  <path>/sample/page</path>
</theme-head-with-footer-path-config>
```

path の中に、コンテキストパス以下のパスを、"/" から記述します。

別の例とし

て、`http://hostname/iap/example/{parameter1}`、`http://hostname/iap/example/{parameter1}/{parameter2}` へのリクエストを異なるビルダーモジュールで表示する場合を取り上げます。この場合、正規表現を利用して path を表現します。path 要素に regex 属性を true として追加することで正規表現として扱われます。

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-head-with-container-path-config xmlns="http://www.intra-mart.jp/theme/theme-head-with-container-path-config">
  <path regex="true">/example/[^/]+?</path>
</theme-head-only-path-config>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-head-with-footer-path-config xmlns="http://www.intra-mart.jp/theme/theme-head-with-footer-path-config">
  <path regex="true">/example/[^/]+?/[^/]+?</path>
</theme-head-with-footer-path-config>
```

リクエストへの属性として指定する例

```
function init(request) {
  request.setAttribute("imui-theme-builder-module", "headwithfooter");
  forward("somewhere");
}
```

リクエストへのパラメータとして指定する例

```
<form name="form" action="sample/page">
  <input type="hidden" name="imui-theme-builder-module" value="headwithfooter">
  <input type="submit" value="submit"/>
</form>
```

ライブラリ群の切り替え

intra-mart Accel Platform 2015 Winter(Lydia) からライブラリ群の切り替え機能を追加しました。

この機能は jQuery のバージョンを切り替えることを主な目的としています。指定されなかったり、存在しない組み合わせ名を指定されたりした場合、「[設定ファイルリファレンス](#)」 - 「[ライブラリ群設定](#)」に指定された version を辞書の昇順でソートし、最初のものが利用されます。標準では iap-8.0.0 が指定されたこととなります。

ライブラリの切り替えは上記の設定ファイル、パラメータ、属性でライブラリ群の組み合わせ名を指定します。どのような組み合わせ名が用意されているかは「[設定ファイルリファレンス](#)」 - 「[ライブラリ群設定](#)」を参照してください。

適用順位

設定ファイル、パラメータ、属性の適用は、以下の順に検索し、最初に合致したライブラリ群の組み合わせを使用します。設定ファイルに記述したものより、属性に指定したものの方が優先されます。

1. 属性
2. パラメータ
3. 設定ファイル

指定例

設定ファイルで指定する例

設定ファイルの path 要素に、libraries-version 属性を追加します。ここでは /sample/page に対して iap-8.0.11 を指定します。この指定によって /sample/page は jQuery 2.1.4 や jQueryUI 1.11.4 を使用するようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-head-with-footer-path-config xmlns="http://www.intra-mart.jp/theme/theme-head-with-footer-path-config">
  <path libraries-version="iap-8.0.11">/sample/page</path>
</theme-head-with-footer-path-config>
```

リクエストへのパラメータとして指定する例

リクエストのパラメータとして指定する場合、キーに IMUI_THEME_LIBRARIES_VERSION を、値にライブラリ群の組み合わせ名を指定します。

```
<form name="form" action="sample/page">
  <input type="hidden" name="IMUI_THEME_LIBRARIES_VERSION" value="iap-8.0.11">
  <input type="submit" value="submit"/>
</form>
```

リクエストへの属性として指定する例

リクエストの属性として指定する場合、キーに `IMUI_THEME_LIBRARIES_VERSION` を、値にライブラリ群の組み合わせ名を指定します。

```
function init(request) {
  request.setAttribute("IMUI_THEME_LIBRARIES_VERSION", "iap-8.0.11");
  forward("somewhere");
}
```

テーマモジュール

ここではテーマモジュールの役割と構成を説明します。

項目

- [テーマモジュールの役割](#)
- [テーマモジュールの構成](#)
 - [設定ファイル](#)
 - [theme-config](#)
 - [message](#)
 - [JSSP](#)
 - [head](#)
 - [header](#)
 - [body](#)
 - [footer](#)
 - [画像](#)
 - [CSS](#)
 - [CSS のフォルダ・ファイル構成](#)
 - [CSJS](#)

テーマモジュールの役割

テーマモジュールは、画面レイアウトとスタイルを定義した [JSSP](#)、[CSS](#)、[CSJS](#)、[画像](#) をまとめたものです。一般ユーザから見た場合、テーマモジュールはテーマそのものに見えます。また、「テーマを切り替える」と、現在利用しているテーマモジュールとは別のテーマモジュールを使って画面を表示することになります。

テーマモジュールの構成

テーマモジュールは以下のような要素で構成されます。

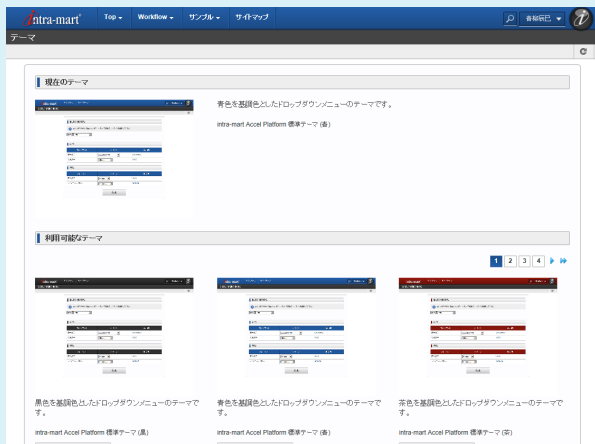
- [設定ファイル](#)
- [JSSP](#)
- [画像](#)
- [CSS](#)
- [CSJS](#)

詳細なフォルダ・ファイル構成は、[テーマモジュールのフォルダ・ファイル構成](#) を参照してください。

コラム

テーマを切り替えるには、下図の「テーマ」画面で利用したいテーマの「このテーマを利用する」ボタンをクリックします。

詳細は [一般ユーザ操作ガイド](#) を参照してください。



「テーマ」画面

設定ファイル

theme-config

%CONTEXT_PATH%/WEB-INF/conf/theme-config 配下に、テーマモジュール毎に設定ファイルが存在します。ファイル名は任意ですが、システム上一意になるようにテーマID と同じ名前をつけることをお勧めします。このファイルには、テーマID や、JSSP のパスなどが記述されています。

theme 要素の属性は以下の通りです。

id

テーマIDを定義します。システム上一意になるような値を指定してください。

imagepath

「テーマ」画面で使用するサムネイル画像のパスを指定します。

theme-folder

このテーマのJSSPのパス。テーマIDと同じ名前をつけることをお勧めします。

sortkey

「テーマ」画面に表示する際のソートキー。昇順でソートされます。ソートキーが同じ場合、テーマID でソートされます。この値には 0 以上の整数を指定してください。

author

作成者（未使用）

version

バージョン（未使用）

theme 要素の子要素として client-type-info 要素を定義します。client-type-info 要素の属性は以下の通りです。

id

クライアントタイプIDを指定します。現在は pc のみ指定できます。

default

このテーマをデフォルトテーマとするかどうかのフラグです。通常は false を指定します。この値が true の設定

ファイルの中で、最初に見つかったテーマモジュールがデフォルトテーマとなります。

例として標準テーマ 青色 の設定ファイルを下に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-config xmlns="http://www.intra-mart.jp/theme/theme"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/theme/theme theme.xsd ">
  <theme
    id="im_theme_dropdown_blue"
    author="intra-mart"
    version="8.0"
    imagepath="ui/theme/im_theme_dropdown_blue/images/thumbnail.png"
    theme-folder="theme/im_theme_dropdown_blue"
    sortkey="10">
    <client-type-info id="pc" default="true"/>
  </theme>
</theme-config>
```

message

「テーマ」画面で使用するメッセージプロパティを定義します。定義するメッセージキーは以下の二つです。

CAP.Z.IWP.THEME.テーマID.NAME

「テーマ」画面で表示されるテーマの名前

CAP.Z.IWP.THEME.テーマID.DESCRPTION

「テーマ」画面で表示されるテーマの説明

%CONTEXT_PATH%/WEB-INF/conf/message/platform/theme/テーマID 配下に以下のファイルを作成します。

- caption.properties
- caption_en.properties
- caption_ja.properties
- caption_zh_CN.properties

詳細は、[多言語化対応](#) を参照してください。

JSSP

head

HTML の head タグを実装します。UIコンポーネントが必要とする [CSJS](#)、[CSS](#) などシステムを動作させるために必要な情報が記述されています。

標準テーマ 青色 を例にすると、head タグは主に以下の要素で構成されています。これらの要素は全テーマモジュールで必須な要素です。

文字コード

文字コードを UTF-8 に指定します。

```
<meta charset="utf-8" />
```

InternetExplorer 向けの設定

対象ブラウザの最新のレンダリングエンジンを使用するように指定します。

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

Google Chrome 向けの設定

Google Chrome が翻訳を行うかどうかを問い合わせるダイアログの表示を抑制します。

```
<meta name="google" content="notranslate">
```

base タグ

base タグを指定します。

```
<base href='<imart type="string" value=base>/imart>' target="_self">
```

テーマのCSS

テーマの CSS を読み込みます。

```
<imart type="imuiLink" href="ui/theme/im_theme_dropdown_blue/css/theme.css"></imart>
```

UIコンポーネントのCSS

UIコンポーネントの CSS を読み込みます。

```
<imart type="imuiLink" href="ui/css/imui.css"></imart>
```

twitter bootstrap のCSS

twitter bootstrap の CSS を読み込みます。

```
<link rel="stylesheet" type="text/css" href="ui/css/bootstrap.css">
```

IE8 向けの設定

IE8 向けに、HTML5 の要素を扱えるようにするライブラリを読み込みます。

```
<!--[if lt IE 9]>
<script type="text/javascript" src="ui/libs/html5.js"></script>
<![endif]-->
```

3rd party ライブラリの読み込み

テーマ、および UIコンポーネントが利用する サードパーティライブラリを読み込みます。

```

<imart type="imuiScript" src="ui/libs/jquery-1.7.2.js"></imart>
<imart type="imuiScript" src="ui/libs/jquery-ui-1.8.21.custom.js"></imart>
<script type="text/javascript" src="ui/libs/jstree_pre1.0_fix/jquery.jstree.js"></script>
<script type="text/javascript" src="csjs/libs/tinymce/jscripts/tiny_mce/jquery.tinymce.js"></script>
<imart type="imuiScript" src="ui/libs/jquery.jqGrid-4.3.3/js/jquery.jqGrid.src.js" suffix="min" regexp="src"
defer="defer"></imart>
<link rel="stylesheet" type="text/css" href="ui/libs/jQuery-File-Upload/css/jquery.fileupload-ui.css">
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/tmpl.min.js" defer="defer"></script>
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/jquery.iframe-transport.js" defer="defer">
</script>
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/jquery.fileupload.js" defer="defer">
</script>
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/jquery.fileupload-ip.js" defer="defer">
</script>
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/jquery.fileupload-ui.js" defer="defer">
</script>
<script type="text/javascript" src="ui/libs/jQuery-File-Upload/js/jquery.fileupload-jui.js" defer="defer">
</script>
<script type="text/javascript" src="ui/js/jquery.imui.fileupload.js" defer="defer"></script>
<script type="text/javascript" src="csjs/libs/lightbox2/js/lightbox.js"></script>
<imart type="imuiScript" src="ui/libs/chardinjs/js/chardinjs.js" suffix="min" regexp="src" defer="defer">
</imart>
<link rel="stylesheet" type="text/css" href="ui/libs/chardinjs/css/chardinjs.css">

```

セッション自動維持機能の読み込み

セッション自動維持機能を実現するタグを記述します。

```
<imart type="imuiSessionKeeper"/>
```

UIコンポーネントのCSJS

UIコンポーネントの ClientSideJavaScript を読み込みます。

```

<imart type="condition" validity=loadSystemLocale><imart type="imuiScript" src=systemLocaleScript />
</imart>
<imart type="condition" validity=loadTenantLocale><imart type="imuiScript" src=tenantLocaleScript />
</imart>
<imart type="imuiScript" src=userLocaleScript></imart>
<imart type="imuiScript" src="ui/js/imui.js"></imart>
<script src="ui/js/imui-form-util.js"></script>

```

テーマのCSJS

テーマの ClientSideJavaScript を読み込みます。読み込む ClientSideJavaScript、ClientSideJavaScript の実装は、各テーマによって異なります。

```

<imart type="imuiScript" src="ui/theme/im_theme_dropdown_blue/js/theme.js"></imart>
<script type="text/javascript" src="csjs/im_json.js" ></script>
<script type="text/javascript" src="csjs/im_window.js" ></script>

```

ライブラリ群の切り替え

intra-mart Accel Platform 2015 Winter(Lydia) からライブラリ群の切り替え機能を追加しました。

標準では上記の [テーマのCSS](#)、[UIコンポーネントのCSS](#)、[twitter bootstrap のCSS](#)、[IE8 向けの設定](#)、[3rd party ライブラリの読み込み](#)、[セッション自動維持機能の読み込み](#)、[UIコンポーネントのCSJS](#)、[テーマのCSJS](#) の

組み合わせを定義しています。

この組み合わせはテーマ共通モジュールとして提供しています。

どのような組み合わせ名が用意されているかの詳細は「[設定ファイルリファレンス](#)」 - 「[ライブラリ群設定](#)」を参照してください。

組み合わせの指定方法は [ライブラリ群の切り替え](#) で説明します。

imart type="head" のプレースホルダー

<imart type="head"> 内に指定された文字列と置き換えられるプレースホルダーです。

```
<imart type="replaceHeadTop"></imart>
```

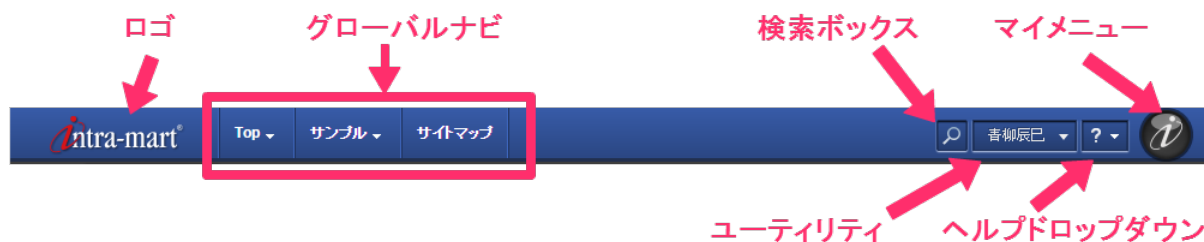
imuiAjaxSubmit のメッセージ表示用関数

imuiAjaxSubmit の結果を表示するための関数です。

```
<script>
(function($) {
  $(document).ready(function() {
    var message = '<imart type="string" value=message />';
    var options = <imart type="string" value=options />;
    var messageType = '<imart type="string" value=messageType />';
    var detail = <imart type="string" value=detail />;
    if(messageType == 'warning') {
      $.imuiFormUtil.showWarningMessage(message, detail, options);
    } else {
      $.imuiFormUtil.showSuccessMessage(message, options);
    }
  });
})(jQuery);
</script>
```

header

ヘッダー部分を実装します。標準テーマでは、グローバルナビゲーション、ユーティリティ、マイメニューなどが実装されています。



ロゴ

標準テーマでは、intra-mart のロゴ画像を配置します。この画像をクリックすると、テナント管理で設定するホーム URL へ遷移します。

グローバルナビ

標準テーマでは、ドロップダウンメニューを配置します。このドロップダウンメニューには、テナント管理のメニュー設定でグローバルナビ（PC用）をメニューグループとして定義したメニューが含まれます。グローバルナビ（PC用）の設定は、[テナント管理者操作ガイド](#) を参照してください。

マイメニュー

標準テーマでは、マイメニューを表示するためのアイコンを配置します。

検索ボックス

標準テーマでは、サイト内をキーワード検索するための検索ボックスを表示するためのアイコンを配置します。IM-

ContentsSearch をインストールすると、このアイコンが表示され検索ボックスから全文検索を行うことができます。

ユーティリティ

ユーティリティは、個人設定やログイン/ログアウトなど業務外の操作をまとめたメニュー群です。ユーティリティは UserUtilityTag と、そのタグが呼び出すプラグインで構成されています。

標準で提供しているプラグインは、

- ChangeToSPIItemProvider
 - スマートフォン版へ
- CompanyItemProvider
 - 会社切り替え
- LoginLogoutItemProvider
 - ログインログアウト
- PersonalSettingsItemProvider
 - 個人設定

です。

プラグインは、UtilityItemProvider インタフェースを実装したクラスとして作ります。このプラグインの作成方法は、[ユーティリティプラグイン](#) を参照してください。

ヘルプドロップダウン

ヘルプドロップダウンは、画面に簡易ヘルプを表示する機能とドキュメントライブラリへのリンクをまとめたメニュー群です。メニューはサイトヘルプカテゴリに登録されたメニューアイテムが表示されています。



コラム

ヘルプドロップダウンは 2014 Winter(Iceberg) からの機能になります。

body

ボディ部分を実装します。

標準テーマは、以下のように実装されています。

```
<div id="imui-container">
  <imart type="replaceContents"/>
</div>
```

なお v5,v6 の互換テーマでは、グローバルナビゲーションもこの JSSP に実装されています。

footer

フッター部分を実装します。標準テーマでは、Copyright表記、Powered by intra-mart 画像などが実装されています。

Copyright © 2012 NTT DATA INTRAMART CORPORATION

Powered by  top ↑

↑ Copyright 表示

↑ Powered by intra-mart 画像

**注意**

使用許諾により、Copyright表記、Powered by intra-mart 画像を表示しないことは禁じられています。

画像

ロゴやアイコンなど、テーマモジュール固有の画像ファイルを格納します。

CSS

テーマが使用する CSS を実装します。

CSS の @import を使った際のパフォーマンス劣化を避けつつ、テーマモジュール間で共通な部分を共有するために LESS を用いて実装しています。

テーマモジュールのビルド時に、LESS で書いた .less ファイルをコンパイルし、CSS に変換します。また、CSS の最小化も行います。変換した CSS と 最小化した CSS の両方を Web サーバ、またはアプリケーションサーバにデプロイします。

CSS 上の共通な部分として以下のものがあります。これらはテーマモジュールを作成する際に必須なファイルです。

- intra-mart Accel Platform が提供する [CSS Module List](#)
- intra-mart Accel Platform が提供するコンポーネント
 - [スクリプト開発モデル](#)
 - [JavaEE開発モデル](#)
- jQueryUI が定義するコンポーネント

また、これらのファイルが要求するパラメータも必要になります。

CSS のフォルダ・ファイル構成

CSS をビルドするのに必要なフォルダ・LESS ファイルの構成は以下の通りです。

テーマモジュール固有のファイルは、body.less, footer.less, header.less, parameter.less の 4 つです。

theme.less は、そのほかの LESS ファイルをインポートするように実装されていて、このファイルをコンパイルすることで全ての定義を取り込んだ theme.css を生成することができます。

```

imart/ui/theme/テーマID/css/
├── theme.less ----- 以下の .less ファイルをまとめる less ファイル
├── theme ----- テーマモジュール固有の less ファイルを配置します
│   ├── body.less ----- #imui-container の定義
│   ├── footer.less ----- footer
│   ├── header.less ----- テーマのヘッダ
│   └── parameters.less ----- LESS のパラメータ
└── common ----- テーマモジュール共通の less ファイルを配置します
    ├── components.less ----- intra-mart Accel Platform が提供するコンポーネント
    ├── default.less ----- HTML 要素を定義します。
    ├── icons.less ----- CSS Sprites
    ├── jqueryUI.less ----- jQueryUI が提供する CSS
    ├── mixins.less ----- 共通の関数
    ├── modules.less ----- intra-mart Accel Platform が提供する CSS モジュール
    ├── parameters.less ----- LESS の共通パラメータ
    └── portal.less ----- ポータル
  
```

theme/parameters.less の必須パラメータは以下のものです。

```
/* テーマの基準色のカラーコード*/  
@theme-color:rgb(242,199,98);  
  
/* テーマの暗い基準色のカラーコード*/  
@theme-dark-color:#444444;  
  
/* テーマの基準文字色のカラーコード*/  
@base-text:#333333;  
  
/* テーマの基準色を背景色としたときの文字色のカラーコード*/  
@accent-text:#ffffff;  
  
/* ボタンのハイライトのカラーコード*/  
@button-high-base:#050505;  
  
/* ボタンのボーダーのカラーコード*/  
@button-border-color:#aaaaaa;  
  
/* グローバルナビの左端からの位置*/  
@nav-global-height:42px;  
  
/* ツールバーの高さ*/  
@toolbar-height:27px;
```

CSJS

テーマが使用する [CSJS](#) を実装します。

標準テーマでは、グローバルナビ、検索ボックス、マイメニュー、iframeの大きさを制御する関数を実装しています。

iframe の大きさを制御する関数は、テーマモジュール共通のもので、この関数は以下のように実装されています。

- id が IM_MAIN の iframe が存在する場合、その高さ、幅をウィンドウの高さ、幅からグローバルナビゲーションなどを除いた大きさまで広げる
- id が IM_MAIN の iframe の中の iframe に imui-no-resize-iframe が class 属性にセットされている場合、その iframe は大きさの変更対象外とする

```

(function($) {
  $(document).ready(function() {
    fitlframe();
    $(window).resize(fitlframe);
  });
  function fitlframe() {
    // ヘッダー部分の高さを取得する
    var header = $('#imui-header').height();
    // iframe の高さは window の高さからヘッダーの高さを引いた値
    var height = $(window).height() - header;
    // #IM_MAIN 中の iframe の大きさをセット
    $('#IM_MAIN').find('iframe:not(".imui-no-resize-iframe")').height(height).width($(window).width());
    // #IM_MAIN の大きさをセット
    $('#IM_MAIN').height(height).width($(window).width());

    //iframe のコンテンツ読み込みが終了したら、iframe 内の min-width を window の幅にする
    $('#IM_MAIN').load(function () {
      try {
        if ($('#IM_MAIN').get(0).contentDocument && $("#IM_MAIN").contents().find('body').css('min-width') > $(window).width()) {
          $("#IM_MAIN").contents().find('body').css('min-width', $(window).width());
          $("#IM_MAIN").contents().find('#imui-container').css('min-width', $(window).width());
        }
      } catch(ignore) {
      }
    });
  }
})(jQuery);

```


テーマモジュールの作成

ここではテーマモジュールの作成方法について説明します。

標準テーマカスタマイズを使って、標準テーマをカスタマイズする方法と、HTML や CSS などをはじめから作り込んでいくスクラッチ開発の方法があります。

項目

- [標準テーマカスタマイズの概要](#)
- [スクラッチ開発](#)
 - [設定ファイル](#)
 - [JSSP](#)
 - [CSS](#)
 - [画像](#)
 - [CSJS](#)
- [サンプル](#)
 - [準備](#)
 - [設定ファイル](#)
 - [JSSP](#)
 - [CSS](#)
 - [画像](#)
 - [CSJS](#)

標準テーマカスタマイズの概要

ここでは標準テーマカスタマイズの概要を説明します。

標準テーマカスタマイズは、標準テーマ、標準テーマ（シンプル）をカスタマイズしたテーマモジュールを生成するツールです。このツールは、テーマの標準的な色とロゴファイルを変更したテーマモジュールを簡単に生成することを目的としています。

カスタマイズ可能な項目は、

- テーマの標準的な色
- ロゴ画像
- グローバルナビの開始位置

です。HTML を修正するようなカスタマイズ、特定の要素だけを変更するような細やかなカスタマイズには対応できません。

カスタマイズできる標準テーマには、v5、v6 の互換テーマは含まれません。

詳細は、標準テーマカスタマイズ 操作ガイド を参照してください。

スクラッチ開発

必要なファイルをすべて作り込んでいく方法です。intra-mart e Builder for Accel Platform でユーザモジュールプロジェクトを作り、テーマモジュールに必要なファイルを作成していきます。

**注意**

スクラッチ開発を行うと、標準テーマやそのカスタマイズでは実現できない見た目や操作性を実装することが可能です。その反面、CSS モジュールや UI コンポーネントを適切にカスタマイズしないと、いわゆる画面のレイアウト崩れ、CSJS の競合などが発生する恐れがあります。

スクラッチ開発を行う場合は、対象となる全画面で全操作を行ってもレイアウトの崩れないこと、操作した際に CSJS のエラーが発生しないことなどを十分に確認してください。

設定ファイル

theme-config

src/main/conf/theme-config 配下にテーマID と同じファイル名を持つ xml ファイルを作成します。内容は [設定ファイル](#) を参考にしてください。

message

src/main/conf/message/platform/theme 配下にテーマID と同じフォルダ名のフォルダを作り、その中にプロパティファイルを作成します。[message](#) を参照し、各言語分プロパティファイルを作成します。

JSSP

src/main/jssp/src/theme 配下にテーマID と同じ名前のフォルダを作成し、その中に PageBuilder が要求する head, header, body, footer の 4 つの [JSSP](#) を作成します。

head

[head](#) の内容を含む JSSP を実装します。これら以外に jQuery のプラグインなど必要なものがあれば追記してください。

既存のテーマからコピーするのが簡単です。この場合、テーマID を修正する必要があります。

```
<imart type="imuiLink" href="ui/theme/テーマID/css/theme.css"></imart>
...
<imart type="imuiScript" src="ui/theme/テーマID/js/theme.js"></imart>
```

header

必要な要素を実装してください。

汎用的なテーマモジュールを作成する場合、[ロゴ](#)、[グローバルナビ](#)、[マイメニュー](#)、[検索ボックス](#)、[ユーティリティ](#)、[ヘルプドロップダウン](#) は必須です。

汎用的ではないテーマモジュールを作成する場合、必須な要素はありません。必要な要素だけを実装してください。

id に imui-header を持つ div や header 要素を含むようにしてください。

```
<header id="imui-header">
  <!-- ロゴ -->
  ...
</header>
```

ロゴ

ロゴは `img` タグとして実装することになると思います。この画像をクリックしたとき、ホームURL へ指定された URL に遷移することが望ましい動きになります。ホームURL を取得するには、`ThemeManager.getEncodedHomeUrl` を呼び出してください。

```
<a href='<imart type="string" value=home></imart>'>
</a>
```

```
let themeManager = new ThemeManager();
home = themeManager.getEncodedHomeUrl();
```

グローバルナビ

`MenuGroupManager` を呼び出すことで、ユーザにひも付いたグローバルナビのメニュー情報を取得することができます。 `imuiDropdown` タグを利用して表示する例を下に挙げます。

```
/* imuiDropdown の data 属性にセットするメニュー情報 */
var menu = [];
function init(request) {
  /*
   * グローバルナビ
   * imuiDropdown の形式にメニュー情報を変換します。
   */
  var menuGroupManager = new MenuGroupManager();
  var resultObject = menuGroupManager.getAvailableMenuTree('im_global_nav_pc');
  if (!resultObject.error) {
    var context = Contexts.getAccountContext();
    var menuTree = resultObject.data;
    walk(menuTree, menu, context.locale);
  }
}
function walk(menuTree, menu, locale) {
  if (menuTree) {
    var menuItem = menuTree.menuItem;
    var item = {};
    menu.push(item);
    item.label = 'Menu';
    if (menuItem.displayName[locale]) {
      item.label = menuItem.displayName[locale].displayName;
    }
    if (menuItem.type !== 'FOLDER') {
      item.href = menuItem.url;
    }
    var children = menuTree.children;
    if (children && children.length > 0) {
      item.children = [];
      for (var i = 0; i < children.length; i++) {
        var child = children[i];
        walk(child, item.children, locale);
      }
    }
  }
}
}
```

```
<imart type="imuiDropdown" data=menu />
```

マイメニュー

マイメニューは CSJS で実装されています。header.js の該当部分（480 - 563行目）を参照してください。

検索ボックス

検索ボックスは、SearchBoxTag を実装してください。

```
<imart type="SearchBoxTag" id="imui-nav-global-search-wrapper"/>
```

このタグは下記のような HTML を生成します。

SearchBoxTag に指定する id、その中の imui-nav-global-search、.imui-form-global-search、.imui-form-global-search-input、.imui-form-global-search-submit に対してスタイルを当ててください。また、表示/非表示を切り替えるように CSJS で関数を作成してください。

```
<ul id="imui-nav-global-search-wrapper">
  <li class="imui-nav-global-search active">
    <a><span class="im-ui-icon-common-16-search-white"></span></a>
    <form style="display: block;" method="GET" action="search" target="__search_window__">
      <fieldset class="imui-form-global-search">
        <input type="search" name="q" class="imui-form-global-search-input">
        <input type="submit" value="検索" class="imui-form-global-search-submit">
      </fieldset>
    </form>
  </li>
</ul>
```

ユーティリティ

ユーティリティは、UserUtilityTag を実装してください。

```
<imart type="UserUtilityTag" id="imui-user-utility" />
```

このタグは下記のような HTML を生成します。

UserUtilityTag に指定する id、その中の li.imui-nav-global-pulldown、さらにその中の ul.imui-nav-global-pulldown-inner に対してスタイルを当ててください。また、表示/非表示を切り替えたり、ドロップダウンするように CSJS で関数を作成してください。

```

<ul id="imui-user-utility">
  <li class="imui-nav-global-pulldown">
    <a href="javascript:void(0);">青柳辰巳</a>
    <ul class="imui-nav-global-pulldown-inner" style="display:none">
      <li>
        <a href="javascript:void(0)">個人設定</a>
        <ul>
          <li>
            <a href="...">パスワード</a>
          </li>
          <li>
            <a href="...">カレンダー</a>
          </li>
          ...
        </ul>
      </li>
      <li>
        <a href="mobile_fw/to_sp">スマートフォン版へ</a>
      </li>
      <li>
        <a href="logout">ログアウト</a>
      </li>
    </ul>
  </li>
</ul>

```

ヘルプドロップダウン

MenuGroupManager を呼び出すことで、ユーザにひも付いたサイトヘルプのメニュー情報を取得することができます。imuiDropdown タグを利用して表示する例を下に挙げます。

```

/* imuiDropdown の data 属性にセットするメニュー情報 */
var helpMenu = [];
function init(request) {
  /*
  * ヘルプドロップダウン
  * imuiDropdown の形式にメニュー情報を変換します。
  */
  var menuGroupManager = new MenuGroupManager();
  var locale = Contexts.getAccountContext().locale;
  var resultObject = menuGroupManager.getAvailableMenuTree('im_site_help_pc');
  var childrenData = [];
  if (!resultObject.error) {
    for(var index=0; index<resultObject.data.children.length; index++) {
      childrenData.push({
        href: resultObject.data.children[index].menuItem.url,
        label: resultObject.data.children[index].menuItem.displayNames[locale].displayName,
        identity: resultObject.data.children[index].menuItem.id
      })
    }
    helpMenu = [
      {
        iconClass:"im-ui-icon-common-16-help-white",
        children:childrenData
      }
    ]
  }
}

```

```
<imart type="imuiDropdown" data=helpMenu />
```

id 属性「start_help_pc」に対して、簡易ヘルプを呼び出すスクリプトを実装してください。また、href 属性の URL に遷移しないように URL を無効化してください。

body

id が imui-container となる div と、その中に <imart type="xxx"/> を必ず実装してください。

```
<div id="imui-container">
  <imart type="replaceContents"/>
</div>
```

footer

poweredbyim.png を表示してください。

```
<footer id="imui-footer">
  <div class="imui-footer-inner">
    <ul class="imui-footer-utility">
      <li></li>
    </ul>
  </div>
</footer>
```

CSS

src/main/public/ui/theme 配下にテーマID と同じ名前のフォルダを作成し、その中に css フォルダを作成します。このフォルダの中にテーマモジュールで使用するCSSを配置します。CSS ファイルの名前は theme.css, theme.min.css と命名することをお勧めします。head.html に imuiLink タグの href 属性に theme.css と記述されているので、この名前にするだけで、修正を最小限に抑えることができます。

テーマが使用するCSS の通り、CSS は LESS を利用して生成しています。テーマモジュール間で共通な部分を共有するため、スクラッチでテーマモジュールを作成する場合も LESS を使用して CSS を生成してください。

基となる LESS ファイル群は、%CONTEXT_PATH%/ui/theme/元とするテーマID/css に配置されています。このフォルダに含まれるファイルをコピーしてください。

LESS ファイル群をコンパイルし、テーマの CSS を生成します。また、パフォーマンス向上を目的として最小化したテーマの CSS も生成することをお勧めします。

LESS ファイルをコンパイルするには [lessc](#) や、[koala](#) などを利用してください。

LESS ファイルの修正は、テーマ固有の LESS ファイル、theme/parameter.less にとどめることをお勧めします。共通の LESS ファイルに対してカスタマイズを行うことはお勧めしません。

画像

src/main/public/ui/theme 配下にテーマID と同じ名前のフォルダを作成し、その中に images フォルダを作成します。このフォルダの中にテーマモジュールで使用する画像を配置します。

テーマ毎に必要な画像は、以下の4つです。

マイメニューのアイコン

btn_im_01.png

ロゴ画像

logo.png

Powred by intra-mart 画像

poweredbyim.png

「テーマ」画面に表示するサムネイル画像

thumbnail.png

CSJS

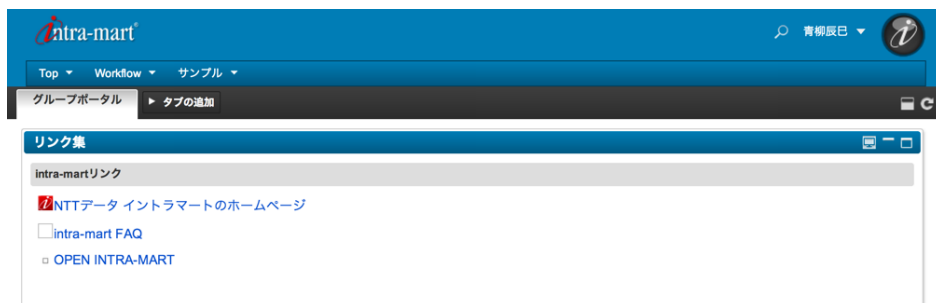
src/main/public/ui/theme 配下にテーマID と同じ名前のフォルダを作成し、その中に js フォルダを作成します。このフォルダの中にテーマモジュールで使用する CSJS を配置します。CSJS ファイルの名前は theme.js, theme.min.js と命名することをお勧めします。head.html に imuiScript タグの src 属性に theme.js と記述されているので、この名前にするすることで、修正を最小限に抑えることができます。

テーマモジュールに必須の iframe の大きさを制御する関数と、作成するテーマモジュールで必要になる CSJS を作成します。

iframe の大きさを制御する関数は、%CONTEXT_PATH%/ui/theme/テーマID/js/im_fit_iframe.js として配置されています。

サンプル

ここでは、以下のようなテーマを作成していきます。



Powered by 

- テーマカラーは、#005678 とします。
- ヘッダーは、ロゴ、グローバルナビ、検索ボックス、ユーティリティ、マイメニューの全要素を配置します。
 - 標準テーマとの違いは、上下の2段とし、丈夫にロゴ、検索ボックス、ユーティリティ、マイメニュー

を、下段にグローバルナビを配置します。

- フッタは、Powred by intra-mart 画像だけを配置します。
- head, body は標準テーマと同じにします。
- テーマID は、sample_theme とします。

準備

intra-mart e Builder for Accel Platform でモジュールプロジェクトを作成しておきます。モジュールプロジェクトの作り方は「intra-mart e Builder for Accel Platform アプリケーション開発ガイド」の「[モジュール・プロジェクト作成](#)」を参照してください。

設定ファイル

theme-config

まず、設定ファイルを作成します。

src/main/conf/theme-config/sample_theme.xml を作成します。内容は以下のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<theme-config
  xmlns="http://www.intra-mart.jp/theme/theme"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/theme/theme ../../schema/theme.xsd ">
  <theme
    id="sample_theme"
    author="intra-mart"
    version="8.0"
    imagepath="ui/theme/sample_theme/images/thumbnail.png"
    theme-folder="theme/sample_theme"
    sortkey="0">
    <client-type-info
      id="pc"
      default="false"/>
  </theme>
</theme-config>
```

message

src/main/conf/message/platform/theme/sample_theme に以下の4つのファイルを作成します。

- caption.properties
- caption_en.properties
- caption_ja.properties
- caption_zh_CN.properties

内容は以下のようになります。

```
CAP.Z.IWP.THEME.SAMPLE_THEME.NAME=Sample Theme
CAP.Z.IWP.THEME.SAMPLE_THEME.DESCRPTION=This is sample theme.
```

CAP.Z.IWP.THEME.テーマID.NAME をキーに持つ値がテーマ名、CAP.Z.IWP.THEME.テーマID.DESCRPTION を

キーに持つ値がテーマの説明になります。また、Java のプロパティファイルとなるため、ASCII 以外の文字列は native2ascii 等でエンコードしてください。

JSSP

次に、JSSP を作成します。

head

既存のテーマの head をコピーし、それを修正します。

ここでは標準テーマ 青色 をコピー対象とします。以下のファイルを src/main/jssp/src/theme/sample_theme にコピーします。

- %CONTEXT_PATH%/WEB-INF/jssp/platform/src/theme/im_theme_dropdown_blue/head.html
- %CONTEXT_PATH%/WEB-INF/jssp/platform/src/theme/im_theme_dropdown_blue/head.js

head.html は、標準テーマ 青色のテーマID である im_theme_dropdown_blue を sample_theme に書き換えるように修正します。

```

--- %CONTEXT_PATH%/imart/WEB-INF/jssp/platform/src/theme/im_theme_dropdown_blue/head.html
+++ src/main/jssp/src/theme/sample_theme/head.html
@@ -10,7 +10,7 @@
     <link rel="icon" href="<imart type="string" value=base></imart>favicon.ico" type="image/x-icon" />
     <link rel="Shortcut Icon" type="img/x-icon" href="<imart type="string" value=base>
</imart>favicon.ico" />

-   <imart type="imuiLink" href="ui/theme/im_theme_dropdown_blue/css/theme.css"></imart>
+   <imart type="imuiLink" href="ui/theme/sample_theme/css/theme.css"></imart>
   <imart type="imuiLink" href="ui/css/imui.css"></imart>
   <link rel="stylesheet" type="text/css" href="ui/css/bootstrap.css">

@@ -39,7 +39,7 @@
   <imart type="imuiScript" src=userLocaleScript></imart>
   <imart type="imuiScript" src="ui/js/imui.js"></imart>
   <script src="ui/js/imui-form-util.js"></script>
-   <imart type="imuiScript" src="ui/theme/im_theme_dropdown_blue/js/theme.js"></imart>
+   <imart type="imuiScript" src="ui/theme/sample_theme/js/theme.js"></imart>
   <script type="text/javascript" src="csjs/im_json.js" ></script>
   <script type="text/javascript" src="csjs/im_window.js" ></script>

```

head.js は修正不要です。

header

header は、以下のように実装します。

src/main/jssp/src/theme/sample_theme/header.html

```

<header id="imui-header" class="container-fluid sample-header">
  <div class="row-fluid">
    <!-- ロゴ -->
    <div class="span6">
      <a id="logo" href='<imart type="string" value=home></imart>'></a>
    </div>
    <div class="span6">
      <!-- マイメニュー -->
      <div style="float:right;">
        <ul id="imui-nav-global-launcher-wrapper" style="display:inline-block;">
          <li class="imui-nav-global-launcher"><a class="action"></a></li>
        </ul>
      </div>
      <!-- ユーティリティ -->
      <div style="float:right;">
        <imart type="UserUtilityTag" id="imui-user-utility"></imart>
      </div>
      <!-- 検索ボックス -->
      <div style="float:right;">
        <imart type="SearchBoxTag" id="imui-nav-global-search-wrapper"></imart>
      </div>
    </div>
  </div>
  <div class="row-fluid">
    <!-- グローバルナビ -->
    <div class="span12">
      <imart type="imuiDropdown" data=menu />
    </div>
  </div>
</header>

```

src/main/jssp/src/theme/sample_theme/header.js

```

var home;
var menu = [];
function init(request) {
  /*
  * ロゴ
  */
  var themeManager = new ThemeManager();
  home = themeManager.getEncodedHomeUrl();
  /*
  * グローバルナビ
  * imuiDropdown の形式にメニュー情報を変換します。
  */
  var menuGroupManager = new MenuGroupManager();
  var resultObject = menuGroupManager.getAvailableMenuTree('im_global_nav_pc');
  if (!resultObject.error) {
    var context = Contexts.getAccountContext();
    var menuTree = resultObject.data;
    var tmpMenu = [];
    walk(menuTree, tmpMenu, context.locale);
    for (var i = 0; i < tmpMenu[0].children.length; i++) {
      // 再帰的にメニューの情報を取得します。
      menu.push(tmpMenu[0].children[i]);
    }
  }
}
function walk(menuTree, menu, locale) {
  if (menuTree) {
    var menuItem = menuTree.menuItem;
    var item = {};
    menu.push(item);
    item.label = 'Menu';
    if (menuItem.displayName[locale]) {
      item.label = menuItem.displayName[locale].displayName;
    }
    if (menuItem.type !== 'FOLDER') {
      item.href = menuItem.url;
    }
    var children = menuTree.children;
    if (children && children.length > 0) {
      item.children = [];
      for (var i = 0; i < children.length; i++) {
        var child = children[i];
        walk(child, item.children, locale);
      }
    }
  }
}
}

```

body

既存のテーマの body をコピーします。

以下のファイルを src/main/jssp/src/theme/sample_theme にコピーします。

- %CONTEXT_PATH%/WEB-INF/jssp/platform/src/theme/im_theme_dropdown_blue/body.html
- %CONTEXT_PATH%/WEB-INF/jssp/platform/src/theme/im_theme_dropdown_blue/body.js

footer

footer は、以下のように実装します。

src/main/jssp/src/theme/sample_theme/footer.html

```
<footer>
  
</footer>
```

src/main/jssp/src/theme/sample_theme/footer.js

```
function init() {}
```

CSS

既存のテーマの LESS ファイルをコピーし、それを修正します。

以下のフォルダを src/main/public/ui/theme/sample_theme にコピーします。

- %CONTEXT_PATH%/ui/theme/im_theme_dropdown_blue/css

修正対象は src/main/public/ui/theme/sample_theme/css/theme 配下にある、以下の3ファイルです。

- parameter.less
- footer.less
- header.less

parameter.less

テーマカラーを #005678 にします。その他の値は修正しません。

```
@theme-color: #005678;
```

footer.less

footer の背景を黒に、Powred by intra-mart の画像を中央に配置します。

```
footer {
  background-color: #000;
  height: 100px;
  margin-top: 20px;
  text-align: center;
  img {
    margin-top: 24px;
  }
}
```

header.less

ヘッダのスタイルを指定します。

```
#imui-nav-global-launcher-wrapper {  
  ...  
}  
#imui-header{  
  color: @accent-text;  
  background-color: @theme-color;  
  /* nav-global */  
  #imui-nav-global{  
    height:@nav-global-height;  
    ...  
    /* nav-global(wrap) */  
    .imui-nav-global-wrap{  
      position:relative;  
      min-width:960px;  
      margin:0 10px;  
      height:@nav-global-height;  
    }  
  }  
}
```

画像

既存のテーマの画像ファイルをコピーし、それを修正します。

以下のフォルダを src/main/public/ui/theme/sample_theme にコピーします。

- %CONTEXT_PATH%/ui/theme/im_theme_dropdown_blue/images

コピーしたファイルの内、thumbnail.png のみこのテーマモジュールと実体が伴いません。あとでスクリーンショットを取得し、その画像と入れ替えてください。

CSJS

このテーマモジュールで使用する関数を実装します。

今回は、iFrame の大きさ制御、マイメニュー、検索ボックスの制御関数を実装します。

詳細は、Appendix を参照してください。

付録

テーマモジュールのフォルダ・ファイル構成

```

テーマモジュール
├── build.xml
├── build_js_css.xml
├── message.properties
├── message_en.properties
├── message_ja.properties
├── message_zh_CN.properties
├── module.xml
├── pom.xml
├── src
│   ├── main
│   │   ├── conf
│   │   │   ├── message
│   │   │   │   └── platform
│   │   │   │       └── theme
│   │   │   │           ├── caption.properties
│   │   │   │           ├── caption_en.properties
│   │   │   │           ├── caption_ja.properties
│   │   │   │           ├── caption_zh_CN.properties
│   │   │   │           └── テーマID
│   │   │   │               ├── caption.properties
│   │   │   │               ├── caption_en.properties
│   │   │   │               ├── caption_ja.properties
│   │   │   │               └── caption_zh_CN.properties
│   │   │   └── theme-config
│   │   │       └── テーマID.xml
│   │   ├── jssp
│   │   │   ├── platform
│   │   │   │   └── src
│   │   │   │       └── theme
│   │   │   │           └── テーマID
│   │   │   │               ├── body.html
│   │   │   │               ├── body.js
│   │   │   │               ├── footer.html
│   │   │   │               ├── footer.js
│   │   │   │               ├── head.html
│   │   │   │               ├── head.js
│   │   │   │               ├── header.html
│   │   │   │               └── header.js
│   │   └── public
│   │       └── ui
│   │           └── theme
│   │               └── テーマID
│   │                   ├── css
│   │                   │   └── CSS を参照してください。
│   │                   ├── images
│   │                   │   ├── logo.png
│   │                   │   ├── poweredbyim.png
│   │                   │   ├── thumbnail.png
│   │                   │   └── ...
│   │                   └── js
│   │                       └── CSJS を参照してください。

```

テーマモジュールのサンプル

項目

- 設定ファイル
 - message
 - sample_theme.xml
- JSSP
 - head
 - header
 - body
 - footer
- CSS
 - parameter.less
 - header.less
 - footer.less
- CSJS
- フォルダ・ファイル構成

設定ファイル

message

src/main/conf/message/platform/theme/sample_theme/caption.properties

```
1 CAP.Z.IWP.THEME.SAMPLE_THEME.NAME=Sample Theme
2 CAP.Z.IWP.THEME.SAMPLE_THEME.DESCRPTION=This is sample theme.
```

caption_en.properties, caption_ja.properties, caption_zh_CN.properties も同じ内容です。

sample_theme.xml

src/main/conf/theme-config/sample_theme.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <theme-config
3   xmlns="http://www.intra-mart.jp/theme/theme"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://www.intra-mart.jp/theme/theme ../../schema/theme.xsd ">
6   <theme
7     id="sample_theme"
8     author="intra-mart"
9     version="8.0"
10    imagepath="ui/theme/sample_theme/images/thumbnail.png"
11    theme-folder="theme/sample_theme"
12    sortkey="0">
13     <client-type-info
14       id="pc"
15       default="false"/>
16   </theme>
17 </theme-config>
```

JSSP

head

src/main/jssp/src/theme/sample_theme/head.html


```
1 <head>
2   <meta charset="utf-8" />
3   <meta name="author" content="NTT DATA INTRAMART CORPORATION" />
4   <meta name="description" content="" />
5   <meta name="keywords" content="" />
6   <meta http-equiv="X-UA-Compatible" content="IE=Edge">
7   <meta name="google" content="notranslate">
8
9   <base href='<imart type="string" value=base></imart>' target="_self">
10  <link rel="icon" href='<imart type="string" value=base></imart>favicon.ico' type="image/x-
11  icon" />
12  <link rel="Shortcut Icon" type="img/x-icon" href='<imart type="string" value=base>
13  </imart>favicon.ico' />
14
15  <imart type="string" value=libraries/>
16
17  <imart type="replaceHeadTop"></imart>
18  <script>
19    (function($) {
20      $(document).ready(function() {
21        var message = '<imart type="string" value=message />';
22        var options = <imart type="string" value=options />;
23        var messageType = '<imart type="string" value=messageType />';
24        var detail = <imart type="string" value=detail />;
25        if(messageType == 'warning') {
26          $.imuiFormUtil.showWarningMessage(message, detail, options);
27        } else {
28          $.imuiFormUtil.showSuccessMessage(message, options);
29        }
30      });
31    })(jQuery);
  </script>
</head>
```

src/main/jssp/src/theme/sample_theme/head.js

```

1  let base;
2  let message;
3  let detail;
4  let locale;
5  let options = {};
6  let messageType = 'success';
7  let libraries;
8
9  function init(request) {
10     //imuiAjaxSubmit で送られたメッセージキーをチェックし、存在すればメッセージを表示
11     if (request['imui-message-key'] == Client.get('x-jp-co-intra-mart-ajaxsubmit-message-key')) {
12         message = Client.get('x-jp-co-intra-mart-ajaxsubmit-message');
13         messageType = Client.get('x-jp-co-intra-mart-ajaxsubmit-message-type');
14         if (messageType == 'warning') {
15             detail = Client.get('x-jp-co-intra-mart-ajaxsubmit-detail-message');
16         }
17         if (request.getParameter("imui-message-closable") != null) {
18             options.closable = request.getParameter("imui-message-closable").getValue() === 'true' ?
19 true : false;
20         }
21         if (request.getParameter("imui-message-duration") != null) {
22             options.duration = request.getParameter("imui-message-duration").getValue() - 0;
23         }
24         if (request.getParameter("imui-message-escape") != null) {
25             options.escape = request.getParameter("imui-message-escape").getValue() === 'true' ?
26 true : false;
27         }
28     }
29     if(detail != null){
30         detail = (ImJjson.parseJSON(detail)[0]).toSource();
31     } else {
32         var a = new Array();
33         detail = a.toSource();
34     }
35     options = (options).toSource();
36     base = Web.base() + '/';
37
38     var themeManager = new ThemeManager();
39     var contents = new Content(themeManager.getLibrariesPath());
40     libraries = contents.execute();
41 }

```

header

src/main/jssp/src/theme/sample_theme/header.html

```

1 <header id="imui-header" class="container-fluid sample-header">
2   <div class="row-fluid">
3     <!-- ロゴ -->
4     <div class="span6">
5       <a id="logo" href='<imart type="string" value=home></imart>'></a>
7     </div>
8     <div class="span6">
9       <!-- マイメニュー -->
10      <div style="float:right;">
11        <ul id="imui-nav-global-laucher-wrapper" style="display:inline-block;">
12          <li class="imui-nav-global-launcher"><a class="action"></a></li>
14        </ul>
15      </div>
16      <!-- ユーティリティ -->
17      <div style="float:right;">
18        <imart type="UserUtilityTag" id="imui-user-utility"></imart>
19      </div>
20      <!-- ヘルプドロップダウン -->
21      <div style="float:right;">
22        <imart type="imuiDropdown" data=helpMenu id="imui-nav-help-dropdown"
23 style="top:10px;position:relative;"/>
24      </div>
25      <!-- 検索ボックス -->
26      <div style="float:right;">
27        <imart type="SearchBoxTag" id="imui-nav-global-search-wrapper"></imart>
28      </div>
29    </div>
30  </div>
31  <div class="row-fluid">
32    <!-- グローバルナビ -->
33    <div class="span12">
34      <imart type="imuiDropdown" data=menu />
35    </div>
36  </div>
37 </header>

```

src/main/jssp/src/theme/sample_theme/header.js


```

1  var home;
2  var menu = [];
3  var helpMenu = [];
4  function init(request) {
5      /*
6       * ロゴ
7       */
8      var themeManager = new ThemeManager();
9      home = themeManager.getEncodedHomeUrl();
10     /*
11     * グローバルナビ
12     * imuiDropdown の形式にメニュー情報を変換します。
13     */
14     var menuGroupManager = new MenuGroupManager();
15     var resultObject = menuGroupManager.getAvailableMenuTree('im_global_nav_pc');
16     if (!resultObject.error) {
17         var context = Contexts.getAccountContext();
18         var menuTree = resultObject.data;
19         var tmpMenu = [];
20         walk(menuTree, tmpMenu, context.locale);
21         for (var i = 0; i < tmpMenu[0].children.length; i++) {
22             // 再帰的にメニューの情報を取得します。
23             menu.push(tmpMenu[0].children[i]);
24         }
25     }
26     getHelpDropdown();
27 }
28 function walk(menuTree, menu, locale) {
29     if (menuTree) {
30         var menuItem = menuTree.menuItem;
31         var item = {};
32         menu.push(item);
33         item.label = 'Menu';
34         if (menuItem.displayName[locale]) {
35             item.label = menuItem.displayName[locale].displayName;
36         }
37         if (menuItem.type !== 'FOLDER') {
38             item.href = menuItem.url;
39         }
40         var children = menuTree.children;
41         if (children && children.length > 0) {
42             item.children = [];
43             for (var i = 0; i < children.length; i++) {
44                 var child = children[i];
45                 walk(child, item.children, locale);
46             }
47         }
48     }
49 }
50
51 function getHelpDropdown(){
52     var menuGroupManager = new MenuGroupManager();
53     var locale = Contexts.getAccountContext().locale;
54     var resultObject = menuGroupManager.getAvailableMenuTree('im_site_help_pc');
55     var childrenData = [];
56     if (!resultObject.error) {
57         for(var index=0; index<resultObject.data.children.length; index++) {
58             childrenData.push({
59                 href: resultObject.data.children[index].menuItem.url,
60                 label: resultObject.data.children[index].menuItem.displayName[locale].displayName,

```

```

61     identity: resultObject.data.children[index].menuItem.id
62     })
63   }
64   helpMenu = [
65     {
66       iconClass:"im-ui-icon-common-16-help-white",
67       children:childrenData
68     }
69   ]
70 }
71 }

```

body

src/main/jssp/src/theme/sample_theme/body.html

```

1   <div id="imui-container">
2
3     <imart type="replaceContents"></imart>
4
5   </div>

```

src/main/jssp/src/theme/sample_theme/body.js

```

1   function init(){ }

```

footer

src/main/jssp/src/theme/sample_theme/footer.html

```

1   <footer>
2     
3   </footer>

```

src/main/jssp/src/theme/sample_theme/footer.js

```

1   function init(){ }

```

CSS

parameter.less

src/main/public/ui/theme/sample_theme/css/theme/parameter.less

```
1 /* テーマの標準的な色のカラーコード*/
2 @theme-color:#005678;
3
4 /* テーマの標準的な暗い色のカラーコード*/
5 @theme-dark-color:#444444;
6
7 /* テーマの標準的な文字色のカラーコード*/
8 @base-text:#333333;
9
10 /* テーマの標準的な色を背景色としたときの文字色のカラーコード*/
11 @accent-text:#ffffff;
12
13 /* グローバルナビの左端からの位置*/
14 @global-navigation-position-left: 158px;
15
16 /* ボタンのハイライトのカラーコード*/
17 @button-high-base:#050505;
18
19 /* ボタンのボーダーのカラーコード*/
20 @button-border-color:#aaaaaa;
21
22 /* ツールバーの高さ*/
23 @toolbar-height:27px;
24
25 /* グローバルナビの高さ*/
26 @nav-global-height:42px;
27
28 @theme-header-color-start:@theme-header-color;
29 @theme-header-color-end:darken(@theme-header-color-start, 10%);
30 @theme-header-colorstop:100%;
31 @theme-header-color-type:linear;
```

header.less

src/main/public/ui/theme/sample_theme/css/theme/header.less


```

1  #logo {
2    position: relative;
3    top: 5px;
4  }
5
6  #imui-user-utility {
7    position: relative;
8    top: 10px;
9    > li.imui-nav-global-pulldown {
10     > a {
11       &.ui-state-active {
12         > span.ui-button-text {
13           color: #333333;
14         }
15       }
16     > span.ui-button-text {
17       color: #ffffff;
18     }
19   }
20 }
21 }
22
23 #imui-nav-global-search-wrapper {
24   position: relative;
25   top: 13px;
26   form {
27     float: left;
28   }
29   > li.imui-nav-global-search {
30     > a {
31       > span {
32         float: right;
33         margin: 3px;
34       }
35     }
36   }
37 }
38
39 /* IM-Launcher area */
40 #imui-nav-global-launcher-wrapper {
41   margin-top: 4px;
42   .imui-nav-global-launcher {
43     margin-left: 8px;
44     .action {
45       z-index: 1000;
46       position: relative;
47     }
48   }
49   .imui-nav-global-launcher.active {
50     position: relative;
51     /* IM-Launcher area box */
52     .imui-box-launcher {
53       position: absolute;
54       top: 5px;
55       right: 20px;
56       width: 830px;
57       z-index: 999;
58       padding: 30px 0 0 30px;
59       background: rgb(0,0,0);
60       background: rgba(0,0,0,0.8);

```

```

61 .theme-border-radius(10px, 10px, 10px, 10px);
62 .theme-box-shadow(0px, 3px, 10px, #333333);
63 a{
64   display:block;
65   overflow:hidden;
66 }
67 .imui-list-launcher{
68   margin-left:5px;
69   margin-bottom:20px;
70   li{
71     float:left;
72     text-align:center;
73     padding:0 10px 0 0;
74     height:80px;
75     width:80px;
76     a{
77       span.imui-list-launcher-label{
78         display:block;
79         font-size:75%;
80         text-decoration:none;
81         padding:2px 10px;
82         background:#000000;
83         width:60px;
84         overflow:hidden;
85         white-space: normal;
86         word-wrap: break-word;
87         word-break: break-all;
88         color:@theme-header-accent-text-color;
89         .theme-border-radius(10px, 10px, 10px, 10px);
90       }
91       span.imui-list-launcher-info{
92         margin-left:140px;
93         width:500px;
94       }
95     }
96   }
97 }
98 .imui-btn-launcher-back{
99   position:absolute;
100  left:10px;
101  top:28px;
102  border:1px solid #888888;
103  .theme-border-radius(3px, 3px, 3px, 3px);
104  a{
105    line-height:50px;
106  }
107 }
108 .imui-btn-launcher-add{
109   position:absolute;
110   right:68px;
111   bottom:4px;
112   .theme-border-radius(3px, 3px, 3px, 3px);
113 }
114 .imui-btn-launcher-settings{
115   position:absolute;
116   right:38px;
117   bottom:4px;
118   .theme-border-radius(3px, 3px, 3px, 3px);
119 }
120 .imui-btn-launcher-close{
121   position:absolute;

```

```

122     right:7px;
123     bottom:7px;
124     padding:0px 1px;
125     border:1px solid #888888;
126     .theme-border-radius(3px, 3px, 3px, 3px);
127 }
128 }
129 }
130 }
131
132 #imui-header{
133     color: @accent-text;
134     background-color: @theme-color;
135     /* nav-global */
136     #imui-nav-global{
137         height:@nav-global-height;
138         border-bottom:3px solid desaturate(darken(@theme-header-color, 22%),10%);
139         .theme-gradient(@theme-header-color-type; @theme-header-color-start;
140 @theme-header-color-end; @theme-header-colorstop);
141         .theme-gradient-filter(@theme-header-color-type; @theme-header-color-start;
142 @theme-header-color-end; @theme-header-colorstop);
143         /* nav-global(wrap) */
144         .imui-nav-global-wrap{
145             position:relative;
146             min-width:960px;
147             margin:0 10px;
148             height:@nav-global-height;
149         }
150     }
151 }
152 .imui-triangle-right {
153     width: 0;
154     height: 0;
155     border-top: 5px solid transparent;
156     border-bottom: 6px solid transparent;
157     border-left: 6px solid #EEE;
158     position: absolute;
159     top: 6px;
160     right: 3px;
161 }
162 .imui-triangle-down {
163     width: 0;
164     height: 0;
165     border-top: 6px solid #EEE;
166     border-left: 5px solid transparent;
167     border-right: 5px solid transparent;
168 }

```

footer.less

src/main/public/ui/theme/sample_theme/css/theme/footer.less

```
1 footer {  
2   background-color: #000;  
3   height: 100px;  
4   margin-top: 20px;  
5   text-align: center;  
6   img {  
7     margin-top: 24px;  
8   }  
9 }
```

CSJS

src/main/public/ui/theme/sample_theme/js/theme.js


```

1  /*
2  * 検索ボックス
3  */
4  (function($) {
5      $.widget("imui.imuiSearchBox", {
6          active: false,
7          _create: function() {
8              var self = this;
9
10             var form = $(this.element).find('form');
11             var trigger = $(this.element).find('li.imui-nav-global-search');
12             self.formArea = form;
13             self._close();
14             trigger.children('a').bind('click', function() {
15                 if (self.active) {
16                     self._close();
17                     this.focus();
18                 } else {
19                     self._open();
20                     self.formArea.find('.imui-form-global-search-input').focus();
21                 }
22                 return false;
23             });
24             form.bind('submit', function() {
25                 if (self.active) {
26                     self._close();
27                 }
28             });
29         },
30         _open: function() {
31             this.active = true;
32             $(this.element.children('li')).addClass('active');
33             this.formArea.show();
34         },
35         _close: function() {
36             this.active = false;
37             $(this.element.children('li')).removeClass('active');
38             this.formArea.hide();
39         }
40     });
41
42     $.widget("imui.imuiMenuItem", {
43         defaultElement: "<ul>",
44         delay: 150,
45         options: {
46             iconClass: null,
47             position: {
48                 my: "left top",
49                 at: "right top",
50                 collision: 'flip none'
51             }
52         },
53         _create: function() {
54             var self = this;
55             this.activeMenu = this.element;
56             this.element
57                 .addClass( "imui-menu ui-widget ui-widget-content" )
58                 .attr({
59                     id: this.menuId,
60                     role: "menu"

```

```

61     })
62     .bind( "mouseover.imuiMenuItem", function( event ) {
63         if ( self.options.disabled ) {
64             return;
65         }
66         clearTimeout($.data(self.options.root.element, 'imui_dropdown_menu.closeAllTimer'));
67         var target = $( event.target ).closest( ".imui-menu-item" );
68         if ( target.length ) {
69             self.focus( event, target );
70         }
71     })
72     .bind("mouseout.imuiMenuItem", function( event ) {
73         if ( self.options.disabled ) {
74             return;
75         }
76         clearTimeout($.data(self.options.root.element, 'imui_dropdown_menu.closeAllTimer'));
77         var timer = setTimeout(function() {
78             self.closeAll();
79             self.options.root._close();
80         }, 500);
81         $.data(self.options.root.element, 'imui_dropdown_menu.closeAllTimer', timer);
82         var target = $( event.target ).closest( ".imui-menu-item" );
83         if ( target.length ) {
84             self.blur( event );
85         }
86     })
87     .find('a')
88     .each(function() {
89         var anchor = $(this);
90         var iconClassName = anchor.data('iconClass') || self.options.iconClass;
91         if (anchor.data('icon')) {
92             anchor.prepend(
93                 $('<span class="imui-menu-item-icon"/>')
94                 .css('background-image', 'url(' + anchor.data('icon') + ')')
95             );
96         } else if (iconClassName) {
97             anchor.prepend(
98                 $('<span class="imui-menu-item-icon"/>')
99                 .addClass(iconClassName)
100            );
101        }
102        if ($(anchor).next('ul').length == 0) {
103            anchor.bind('click', function() {
104                self.closeAll();
105                self.options.root._close();
106            });
107        }
108    })
109    .end()
110    .imuiScrollMenu({
111        position: this.options.position
112    })
113    .find('ul')
114    .imuiScrollMenu({position: this.options.position});
115
116    this.refresh();
117 },
118 refresh: function() {
119     var self = this;
120     // initialize nested menus
121     var submenus = this.element.find("ul:not(.imui-menu)")

```

```

122 .addClass( "imui-menu ui-widget ui-widget-content" )
123 .attr("role", "menu")
124 .hide();
125
126 // don't refresh list items that are already adapted
127 var items = submenus.add(this.element).children( "li:not(.imui-menu-item):has(a)" )
128 .addClass( "imui-menu-item" )
129 .attr( "role", "presentation" );
130
131 items.children( "a" )
132 .addClass( " ui-state-default" )
133 .attr( "tabIndex", -1 )
134 .attr( "role", "menuitem" );
135
136 items.children('a').each(function() {
137     var html = $(this).html();
138     $(this).empty().append($('

```



```

183     this.activeMenu = item.parent();
184
185     this._trigger( "focus", event, { item: item } );
186 },
187 blur: function(event) {
188     if (!this.active) {
189         return;
190     }
191
192     clearTimeout(this.timer);
193
194     this.active.children( "a" ).removeClass( "ui-state-focus" );
195     this.active = null;
196 },
197 _startOpening: function(submenu) {
198     clearTimeout(this.timer);
199     var self = this;
200     self.timer = setTimeout(function() {
201         self._close();
202         self._open(submenu);
203     }, self.delay);
204 },
205
206 _open: function(submenu) {
207     clearTimeout(this.timer);
208     this.element.find(".imui-menu").not(submenu.parents()).hide().attr("aria-hidden", "true");
209     /*
210     var position = $.extend({}, {
211       of: this.active
212     }, $.type(this.options.position) == "function"
213     ? this.options.position(this.active)
214     : this.options.position
215     );
216     submenu.show().removeAttr("aria-hidden").attr("aria-expanded", "true").position(position);
217     */
218     submenu.width(submenu.width());
219     submenu
220     .show()
221     .removeAttr("aria-hidden")
222     .attr("aria-expanded", "true")
223     .css('opacity', 0)
224     .imuiScrollMenu('adjust', $.extend(this.options.position, {of:
225 submenu.parent().parent()}))
226     .css('opacity', 1);
227 },
228
229 closeAll: function() {
230     this.element
231     .find("ul").hide().attr("aria-hidden", "true").attr("aria-expanded", "false").end()
232     .find("a.ui-state-active").removeClass("ui-state-active");
233     this.blur();
234     this.activeMenu = this.element;
235 },
236
237 _close: function() {
238     this.active.parent()
239     .find("ul").hide().attr("aria-hidden", "true").attr("aria-expanded", "false").end()
240     .find("a.ui-state-active").removeClass("ui-state-active");
241 },
242 _hasScroll: function() {
243     // TODO: just use .prop() when we drop support for jQuery older than 1.6

```

```

243 // TODO: just use .prop() which we drop support for jquery order than 1.0
244     return this.element.height() < this.element[ $.fn.prop ? "prop" : "attr" ]( "scrollHeight" );
245 },
246 select: function( event ) {
247     // save active reference before closeAll triggers blur
248     var ui = {
249         item: this.active
250     };
251     this.closeAll();
252     this._trigger( "select", event, ui );
253 },
254 destroy: function() {
255     this.element
256         .removeAttr('aria-activedescendant')
257         .find('.imui-menu')
258         .andSelf()
259         .removeClass('imui-menu ui-widget ui-widget-content ')
260         .removeAttr('role')
261         .removeAttr('tabIndex')
262         .removeAttr('aria-labelledby')
263         .removeAttr('aria-expanded')
264         .removeAttr('aria-hidden')
265         .show();
266
267     this.element.find('.imui-menu-item')
268         .unbind('.imuiMenuItem')
269         .removeClass('imui-menu-item')
270         .removeAttr('role')
271         .children('a')
272         .removeClass('ui-state-hover')
273         .removeAttr('tabIndex')
274         .removeAttr('role')
275         .removeAttr('aria-haspopup')
276         .removeAttr('id')
277         .children('.ui-icon')
278         .remove();
279
280     $.Widget.prototype.destroy.call(this);
281 }
282 });
283 $.widget("imui.imui_dropdown_menu",{
284     options: {
285         iconClass: null,
286         menuBarClass: "imui-nav-global-list imui-menubar ui-widget-header ui-helper-clearfix",
287         menuBarItemClass: "imui-menubar-item",
288         activeForIE8: null
289     },
290     _create: function() {
291         // this will be ul
292         var that = this;
293         // items will be toplevel li
294         var items = this.items = this.element.children('li')
295             .addClass(that.options.menuBarItemClass)
296             .children('a');
297         // let only the first item receive focus
298         items.slice(1).attr( "tabIndex", -1 );
299
300         //this.element.addClass("imui-menubar ui-widget-header ui-helper-clearfix");
301         this.element.addClass(this.options.menuBarClass);
302
303         items.each(function() {
304             // ...

```

```

304     var anchor = $(this);
305     anchor
306         .bind('mouseenter.imui_dropdown_menu', function(event) {
307             anchor.addClass('ui-state-hover');
308         })
309         .bind('mouseleave.imui_dropdown_menu', function(event) {
310             anchor.removeClass('ui-state-hover');
311         })
312         .bind('click.imui_dropdown_menu', function(event) {
313             if (anchor.next('ul').size() == 0) {
314                 that._trigger('select', event, {item: anchor.closest('li')});
315             }
316         });
317 });
318
319     items.next('ul')
320     .imuiMenuItem({
321         select: function( event, ui ) {
322             ui.item.parents( "ul.imui-menu:last" ).hide();
323             that._trigger( "select", event, ui );
324             that._close();
325             // TODO what is this targetting? there's probably a better way to access it
326             $(event.target).prev().focus();
327         },
328         iconClass: this.options.iconClass,
329         root: this
330     })
331     .hide();
332
333     items.each(function() {
334         var input = $(this),
335             menu = input.next('ul');
336
337         input.bind("click.imui_dropdown_menu focus.imui_dropdown_menu
338 mouseenter.imui_dropdown_menu mouseleave.imui_dropdown_menu", function(event) {
339             // ignore triggered focus event
340             if ( event.type == "focus" && !event.originalEvent ) {
341                 return;
342             }
343             //サブメニューがないメニューは、eventをそのまま続行
344             if ($(this).next('ul').length != 0) {
345                 event.preventDefault();
346             }
347             if ( event.type == "click" && menu.is( ":visible" ) && that.activeMenu &&
348 that.activeMenu[0] == menu[0] ) {
349                 that._close();
350                 return;
351             }
352             if ( ( that.open && event.type == "mouseenter" ) || event.type == "click" ) {
353                 that._open( event, menu );
354             }
355             if (event.type == 'mouseenter') {
356                 clearTimeout($.data(that, 'imui_dropdown_menu.closeAllTimer'));
357                 that._open(event, menu);
358                 return;
359             }
360             if (event.type == 'mouseleave' && menu.is(':visible')) {
361                 clearTimeout($.data(that.element, 'imui_dropdown_menu.closeAllTimer'));
362                 var timer = setTimeout(function() {
363                     that._close();
364                 }, 500);

```

```

365     $.data(that.element, 'imui_dropdown_menu.closeAllTimer', timer);
366     return;
367   }
368 })
369 .addClass('ui-button ui-widget ui-button-text-only imui-menubar-link')
370 .wrapInner( "<span class='ui-button-text'></span>" )
371 .removeClass( "ui-button-text-only" ).addClass( "ui-button-text-icon-secondary" );
372
373 //for IE8
374 if (input.width() > input.children().outerWidth(true)) {
375   input.width(input.children().outerWidth(true));
376 }
377
378 //サブメニューがある場合は、下向き三角アイコンを追加
379 if (input.next('ul').length != 0) {
380   input.prepend('<div class="imui-triangle-down" style="position: absolute;top:
381 10px;right: .5em;"></div>');
382   }
383 });
384
385 $(that.element).each(function(event){
386   $(this).bind('focusin.imui_dropdown_menu', function(event) {
387     clearTimeout(that.closeTimer);
388   });
389   $(this).bind('focusout.imui_dropdown_menu', function(event) {
390     that.closeTimer = setTimeout(function() {
391       that._close(event);
392     }, 100);
393   });
394 });
395 },
396 _close: function(event, menu) {
397   if (!this.activeMenu || !this.activeMenu.length) {
398     //for IE8
399     if ($.data(this.element, 'imui_dropdown_menu.active') != null) {
400       this.activeMenu = $.data(this.element, 'imui_dropdown_menu.active');
401     } else {
402       return;
403     }
404   }
405   this.element.children('li').removeClass('active');
406   this.activeMenu
407     .imuiMenuItem("closeAll")
408     .hide();
409   this.activeMenu
410     .prev()
411     .removeClass("ui-state-active")
412     .removeAttr("tabIndex");
413   this.activeMenu = null;
414   this.open = false;
415
416   //for IE8
417   this.options.activeForIE8 = null;
418   $.removeData(this.element, 'imui_dropdown_menu.active');
419 },
420 _open: function(event, menu) {
421   var that = this;
422   if (menu.length == 0) {
423     return;
424   }
425   if ( this.activeMenu && this.activeMenu[0] == menu[0] ) {

```

```

426 //for IE8
427 if (this.options.activeForIE8 != null) {
428     this.activeMenu = this.options.activeForIE8;
429 } else {
430     return;
431 }
432 }
433 this.element.children('li').addClass('active');
434 if (this.activeMenu) {
435     this.activeMenu
436     .imuiMenuItem("closeAll")
437     .hide();
438     this.activeMenu
439     .prev()
440     .removeClass("ui-state-active");
441 }
442 var button = menu.prev().addClass("ui-state-active");
443 menu.width(menu.width());
444 if (menu.width() < menu.parent().width()) {
445     menu.width(menu.parent().width());
446 }
447 this.activeMenu = menu
448     .show()
449     .css('opacity', 0)
450     .imuiScrollMenu('adjust', {
451         collision: 'flip none',
452         my: 'right top',
453         at: 'right bottom',
454         of: button
455     }, true)
456     .css('opacity', 1);
457
458 //for IE8
459 this.options.activeForIE8 = this.activeMenu;
460 $.data(this.element, 'imui_dropdown_menu.active', this.activeMenu);
461
462 this.open = true;
463 this._trigger('select', event, {item: this.activeMenu.closest('li')});
464 },
465 destroy: function() {
466     this.element.find(':imui-imui-menu').imuiMenuItem('destroy');
467
468     this.element
469     .removeClass('imui-menubar ui-widget-header ui-helper-clearfix')
470     .removeAttr('role')
471     .unbind('.imui_dropdown_menu');
472
473     this.element.children('li')
474     .removeClass('imui-menubar-item')
475     .removeAttr('role');
476
477     $('> li > a', this.element)
478     .removeClass('ui-button ui-widget ui-button-text-only ui-menubar-link ui-state-default')
479     .removeAttr('role')
480     .unbind('.imui_dropdown_menu');
481
482     $.Widget.prototype.destroy.call(this);
483 }
484 });
485
486 /*

```

```

487  * マイメニュー
488  */
489  $(document).ready(function() {
490      $('#imui-user-utility').imui_dropdown_menu();
491
492      $('#start_help_pc').children().removeAttr('href');
493      var startHelp = function(){
494          $('body').imuiSiteTour('start');
495      };
496      $('body').imuiSiteTour();
497      $('#start_help_pc').on('click.siteHelp',startHelp);
498
499      var parent = $(this).parent()[0];
500      var removeLauncher = function() {
501          $('.imui-box-launcher').remove();
502          $('div[aria-labelledby=ui-dialog-title-imui-launcher-add-dialog]').remove();
503          $('#imui-launcher-add-dialog').remove();
504          $(parent).removeClass('active');
505      };
506
507      $('a', '.imui-nav-global-launcher').bind('click', function(event) {
508          event.preventDefault();
509          if ($.imui-box-launcher().size() == 0) {
510              var launcher = new Array();
511              launcher.push('<div class="imui-box-launcher">');
512              launcher.push('<div id="imui-list-launcher" class="imui-list-launcher"><ul><li></li>');
513              launcher.push('</ul></div>');
514              launcher.push('<p class="imui-btn-launcher-back"><a href="javascript: void(0);">');
515              launcher.push('<span></span></a></p>');
516              launcher.push('<p class="imui-btn-launcher-add"><a href="javascript: void(0);">');
517              launcher.push('<span></span></a></p>');
518              launcher.push('<p class="imui-btn-launcher-settings"><a');
519              launcher.push('href="user/mymenu/settings"><span class="im-ui-icon-common-24-settings"></span></a>');
520              launcher.push('</p>');
521              launcher.push('<p class="imui-btn-launcher-close"><a href="javascript: void(0);">');
522              launcher.push('<span></span></a></p>');
523              launcher.push('<div id="imui-launcher-add-dialog"></div>');
524              launcher.push('</div>');
525              $(this).after(launcher.join(""));
526
527              $('.imui-box-launcher').hide().fadeOut(200);
528              $('.imui-box-launcher .imui-btn-launcher-back').hide();
529              $('.imui-box-launcher .imui-btn-launcher-add').hide();
530              $('.imui-box-launcher .imui-btn-launcher-settings').hide();
531              var parent = $(this).parent()[0];
532              $(parent).addClass('active');
533
534              $.ajax({
535                  headers: {'x-jp-co-intra-mart-ajax-request-from-imui-form-util':'true'},
536                  dataType:'text', type:'GET', url:'user/mymenu/js_launcher',
537                  data:{},
538                  success:function(data) {
539                      eval(data.replace(/<V?script[ ^>]*>/g, ""));
540                  },
541                  error:function(request, textStatus, errorThrown) {
542                      removeLauncher();
543                      if (jQuery.imuiFormUtil && jQuery.imuiFormUtil.transitionToErrorPage) {
544                          jQuery.imuiFormUtil.transitionToErrorPage(request, textStatus, errorThrown);
545                      } else {
546                          imuiShowErrorMessage(request.statusText + '(' + request.status + ')', "");
547                      }
548                  }
549              });
550          }
551      });
552  }

```

```

548     }
549   });
550
551   $('.imui-box-launcher .imui-btn-launcher-back a').mouseover(function() {
552     $('.imui-btn-launcher-back').addClass('ui-state-hover');
553     $(this).find('span:first').removeClass('im-ui-icon-launcher-16-move-up-
554 gray').addClass('im-ui-icon-launcher-16-move-up-black');
555   }).mouseout(function() {
556     $('.imui-btn-launcher-back').removeClass('ui-state-hover');
557     $(this).find('span:first').removeClass('im-ui-icon-launcher-16-move-up-
558 black').addClass('im-ui-icon-launcher-16-move-up-gray');
559   }).mouseout();
560
561   $('.imui-box-launcher .imui-btn-launcher-add a').mouseover(function() {
562     $('.imui-btn-launcher-add').addClass('ui-state-hover');
563   }).mouseout(function() {
564     $('.imui-btn-launcher-add').removeClass('ui-state-hover');
565   }).mouseout();
566
567   $('.imui-box-launcher .imui-btn-launcher-settings a').mouseover(function() {
568     $('.imui-btn-launcher-settings').addClass('ui-state-hover');
569   }).mouseout(function() {
570     $('.imui-btn-launcher-settings').removeClass('ui-state-hover');
571   }).mouseout();
572
573   $('.imui-box-launcher .imui-btn-launcher-close a').click(function() {
574     $('.imui-box-launcher').fadeOut(200, function() {
575       removeLauncher();
576     });
577   }).mouseover(function() {
578     $('.imui-btn-launcher-close').addClass('ui-state-hover');
579     $(this).find('span:first').removeClass('im-ui-icon-launcher-16-close-
580 gray').addClass('im-ui-icon-launcher-16-close-black');
581   }).mouseout(function() {
582     $('.imui-btn-launcher-close').removeClass('ui-state-hover');
583     $(this).find('span:first').removeClass('im-ui-icon-launcher-16-close-
584 black').addClass('im-ui-icon-launcher-16-close-gray');
585   }).mouseout();
586   } else {
587     $('.imui-box-launcher').fadeOut(200, function() {
588       removeLauncher();
589     });
590   }
591   });
592   });
593 })(jQuery);
594
595 /*
596  * iFrame の大きさ制御関数
597  */
598 (function($) {
599   $(document).ready(function() {
600     fitIframe();
601     $(window).resize(fitIframe);
602   });
603   function fitIframe() {
604     var header = $('#imui-header').height();
605     var height = $(window).height() - header;
606     $('#IM_MAIN').find('iframe:not(".imui-no-resize-
607 iframe")').height(height).width($(window).width());

```

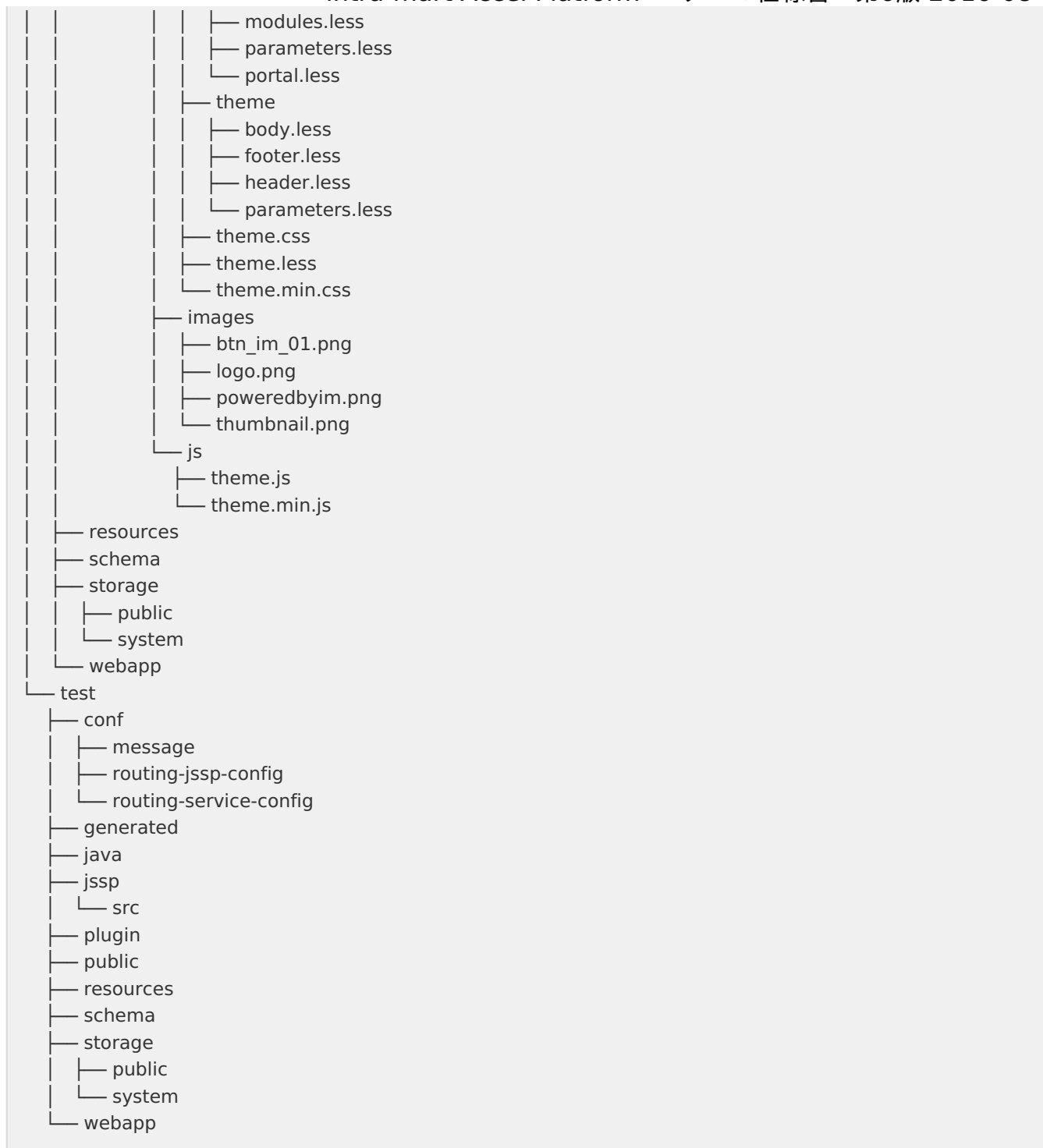
```
$('#IM_MAIN').height(height).width($(window).width());
```

//iframe のコンテンツ読み込みが終了したら、iframe 内の min-width を window の幅にする

```
$('#IM_MAIN').load(function () {
  try {
    if ($('#IM_MAIN').get(0).contentDocument &&
      $("#IM_MAIN").contents().find('body').css('min-width') > $(window).width()) {
      $("#IM_MAIN").contents().find('body').css('min-width', $(window).width());
      $("#IM_MAIN").contents().find('#imui-container').css('min-width', $(window).width());
    }
  } catch(ignore) {
  }
});
}
})(jQuery);
```

フォルダ・ファイル構成

```
src
├── main
│   ├── conf
│   │   ├── message
│   │   │   └── platform
│   │   │       └── theme
│   │   │           └── sample_theme
│   │   │               ├── caption.properties
│   │   │               ├── caption_en.properties
│   │   │               ├── caption_ja.properties
│   │   │               └── caption_zh_CN.properties
│   │   ├── routing-jssp-config
│   │   ├── routing-service-config
│   │   ├── theme-config
│   │   └── sample_theme.xml
│   ├── generated
│   ├── java
│   ├── jssp
│   │   └── src
│   │       └── theme
│   │           └── sample_theme
│   │               ├── body.html
│   │               ├── body.js
│   │               ├── footer.html
│   │               ├── footer.js
│   │               ├── head.html
│   │               ├── head.js
│   │               ├── header.html
│   │               └── header.js
│   ├── plugin
│   ├── public
│   │   └── ui
│   │       └── theme
│   │           └── sample_theme
│   │               └── css
│   │                   ├── common
│   │                   ├── components.less
│   │                   ├── default.less
│   │                   ├── icons.less
│   │                   ├── jqueryUI.less
│   │                   └── mixins.less
```

ユーティリティプラグイン

ここではユーティリティプラグインの作成の概要と、サンプルプログラムについて説明します。



コラム

スマートフォン版のユーティリティメニューの設定に関しては「[設定ファイルリファレンス](#)」 - 「[SP版ユーティリティメニュー設定](#)」を参照してください。

項目

- ユーティリティプラグイン作成の概要
 - プラグイン本体
 - plugin.xml
- サンプルプログラム
 - プラグイン本体
 - plugin.xml
 - 処理を実行するサーバ側のプログラムを呼び出すための CSJS
 - 処理を実行するサーバ側のプログラム
 - 実行例
- サンプルプログラムの実装例
 - プラグイン本体
 - plugin.xml
 - CSJS
 - フォルダ構成

ユーティリティプラグイン作成の概要

ここではユーティリティプラグインの作成の概要を説明します。

ユーティリティプラグインは、PluginManager が管理するプラグインとして実装します。このプラグインは以下の要素で構成されます。

- plugin.xml
- プラグイン本体

この他に必要であれば、以下のような要素を追加します。

- 処理を実行するサーバ側のプログラムを呼び出すための CSJS
- 処理を実行するサーバ側のプログラム
 - Java
 - JSSP
- 設定ファイル
 - ルータの設定ファイル
 - 多言語ファイル

プラグイン本体

プラグイン本体は、UtilityItemProvider インタフェースを実装したクラスとして作ります。

サブメニューを持つプラグインを作成する場合、UtilityItemProvider#getChildrenを実装し、UtilityItemProviderを実装したクラスのインスタンスを返します。

プラグインが呼び出されると、ul > li > a のような DOM が構築されます。対象のリンクをクリックした際の動作は a タグの href で表現したり、a タグに id を割り振り、その id で何らかの処理を行うように実装することになります。

- isVisible
- getLabel
- getUrl

- scriptSource
- getChildren
- toListString

plugin.xml

PluginManager が管理する plugin.xml を作成します。

extension 要素の point 属性には jp.co.intra_mart.foundation.ui.theme.utility.item を指定してください。

その他の要素、属性については [PluginManager の API リファレンス](#) を参照してください。

処理を実行するサーバ側のプログラムを呼び出すための CSJS

ユーティリティメニューをクリックした際、クライアント側で動作するロジックを実装します。 オプションです。

処理を実行するサーバ側のプログラム

ユーティリティメニューをクリックした際、サーバ側で動作するロジックを実装します。 オプションです。

設定ファイル

必要であれば、以下のようなファイルを作成します。

- ルータの設定ファイル
- 多言語ファイル

サンプルプログラム

ここでは、ユーティリティプラグインのサンプルとして、ログインユーザのロケールを切り替えるプラグインを作成します。

このプラグインは、UtilityItemProvider を実装したクラスとして実装します。メニューは、現在のロケールを表示する親メニューと、選択可能なロケールの一覧を表示するサブメニューとで構成します。

ロケール切り替えのサーバ側のプログラムは、既存のロケール切り替えの仕組みを流用します。

- プラグイン本体
- plugin.xml
- 処理を実行するサーバ側のプログラムを呼び出すための CSJS
- 処理を実行するサーバ側のプログラム
 - 既存のロケール切り替えの仕組みを流用します。

プラグイン本体

jp.co.intra_mart.sample.LocaleItemProvider として実装します。以下の必要なメソッドを実装します。

- isVisible
- getLabel
- getUrl
- scriptSource
- getChildren
- toListString

isVisible

ここでは認証済ユーザであればこのプラグインを表示できるようにします。

```

public boolean isVisible() {
    AccountContext context = Contexts.get(AccountContext.class);
    return context.isAuthenticated();
}

```

getLabel

ここでは認証済ユーザのロケールを表示します。

アカウントコンテキストに格納されているロケールを表示します。表示する際にエスケープを行います。

```

public String getLabel() throws JspException {
    AccountContext context = Contexts.get(AccountContext.class);
    Locale locale = context.getLocale();
    String name = locale.getDisplayName(locale);
    return Util.escape(name, Escaping.STRICT_HTML);
}

```

getUrl

ここでは表示だけを行い、処理はしないため null を返します。

```

public String getUrl() throws JspException {
    return null;
}

```

scriptSource

ここでは、CSJS のパスを返します。

UserUtilityTag、プラグインには CSJS の実装を直接 HTML に出力する方法がないため、外部ファイルに実装した関数を呼び出すという方法を採用します。このメソッドが返すパスに、CSJS の実装を記述します。

```

public String scriptSource() {
    return "sample/js/changeLocale.js";
}

```

getChildren

ここでは、システムに登録されているロケールの一覧を返します。また、返すクラスは UtilityItemProvider を実装したクラスとして実装します。

```

public List<UtilityItemProvider> getChildren() throws JspException {
    final List<UtilityItemProvider> children = new ArrayList<UtilityItemProvider>();
    // システムロケールの一覧を作成します。
    LocaleInfo[] localeInfos = SystemLocale.getLocaleInfos();
    for (LocaleInfo localeInfo: localeInfos) {
        children.add(new LocaleListItemProvider(localeInfo));
    }
    return children;
}

```

toListString

このプラグインが返す HTML を生成します。

ul > li > a となるように実装します。また、サブメニューを ul > li の中の ul として返すように実装します。

```

public String toListString() throws JspException {
    final StringBuilder buf = new StringBuilder();
    final String url = getUrl() == null ? StringUtil.EMPTY_STRING : "href=\"" + getUrl() + "\"";
    buf.append("<li><a ").append(url).append(">").append(getLabel()).append("</a>");
    if (getChildren() != null) {
        buf.append("<ul>");
        for (final UtilityItemProvider item : getChildren()) {
            buf.append(item.toListString());
        }
        buf.append("</ul>");
    }
    buf.append("</li>");
    return buf.toString();
}

```

サブメニューの項目の一つ一つはこのクラスで実装します。

```

private static class LocaleListItemProvider implements UtilityItemProvider {
    private LocaleInfo localeInfo;
    public LocaleListItemProvider(final LocaleInfo localeInfo) {
        this.localeInfo = localeInfo;
    }
}

```

ラベルとして、システムロケールの一つを表示します。インスタンス化した際のロケールの値を表示します。

```

public String getLabel() throws JspException {
    return Util.escape(this.localeInfo.getDisplayName(), Escaping.STRICT_HTML);
}

```

クリックした際の URL を返します。ここで、CSJS の関数を呼び出すことになります。

```

public String getUrl() throws JspException {
    return String.format("javascript:changeLocale('%s');", this.localeInfo.getLocale());
}

```

HTML として、li タグを返します。

```

public String toListString() throws JspException {
    final StringBuilder buf = new StringBuilder();
    final String url = "href=\" + getUrl() + "\"";
    buf.append("<li><a ").append(url).append(">");
    buf.append(getLabel()).append("</a>");
    buf.append("</li>");
    return buf.toString();
}

```

plugin.xml

プラグインの設定を記述します。

extension 要素の point 属性に jp.co.intra_mart.foundation.ui.theme.utility.item を指定します。また、item 要素の classname 属性に、プラグイン本体のクラス名を指定します。

```

<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.ui.theme.utility.item">
    <item
      id="sample_localeitemprovider"
      name="localeitemprovider"
      classname="jp.co.intra_mart.sample.LocaleItemProvider"
      version="1.0"
      rank="1"
      enable="true"
    />
  </extension>
</plugin>

```

処理を実行するサーバ側のプログラムを呼び出すための CSJS

ここでは、簡単のため既存のロケール切り替えの仕組みを呼び出すような実装を行います。

プラグイン本体の、scriptSource メソッドが返す値である sample/js/changeLocale.js に下記の実装を記述します。アクセス先の user/settings/locale/update_locale は、ルータの設定、サーバ側のロジックの実装などはテナント管理モジュールで実装されています。

```

function changeLocale(locale) {
  (function($, locale) {
    var form = $('<form/>')
      .attr('id', 'sampleform')
      // 既存のロジックを呼び出します。
      .attr('action', 'user/settings/locale/update_locale')
      .attr('method', 'POST');
    var locale = $('<input/>')
      .attr('type', 'hidden')
      .attr('name', 'locale')
      .attr('value', locale);
    form.append(locale);
    form.appendTo('body');
    imuiAjaxSubmit('#sampleform', 'POST', 'json', 'home', []);
  })(jQuery, locale);
}

```

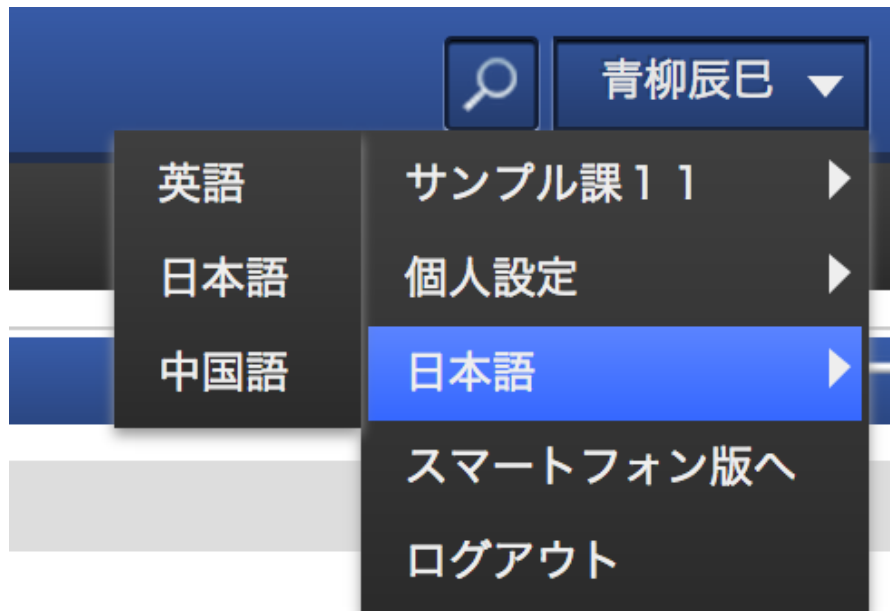
処理を実行するサーバ側のプログラム

このサンプルでは実装を行いません。

サーバ側のロジックはテナント管理モジュールで実装されています。

実行例

このプラグインをデプロイすると以下のような画面が表示され、ロケールを切り替えることができます。



サンプルプログラムの実装例

プラグイン本体

```
src/main/java/jp/co/intra_mart/sample/LocaleItemProvider.java
```



```

1 package jp.co.intra_mart.sample;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.Locale;
6
7 import javax.servlet.jsp.JspException;
8
9 import jp.co.intra_mart.common.aid.jdk.java.lang.StringUtil;
10 import jp.co.intra_mart.foundation.context.Contexts;
11 import jp.co.intra_mart.foundation.context.model.AccountContext;
12 import jp.co.intra_mart.foundation.i18n.locale.LocaleInfo;
13 import jp.co.intra_mart.foundation.i18n.locale.SystemLocale;
14 import jp.co.intra_mart.foundation.ui.tags.theme.UtilityItemProvider;
15 import jp.co.intra_mart.system.ui.util.Util;
16
17 import org.jamon.escaping.Escaping;
18
19 public class LocaleItemProvider implements UtilityItemProvider {
20     private static class LocaleListProvider implements UtilityItemProvider {
21         private LocaleInfo localeInfo;
22
23         /**
24          * コンストラクタ。
25          * @param localeInfo ロケール情報
26          */
27         public LocaleListProvider(final LocaleInfo localeInfo) {
28             this.localeInfo = localeInfo;
29         }
30
31         /**
32          * サブメニューを返します。
33          * @return null: サブメニューはありません。
34          * @throws JspException
35          */
36         @Override
37         public List<UtilityItemProvider> getChildren() throws JspException {
38             return null;
39         }
40
41         /**
42          * このメニューのラベルを返します。
43          * @return ロケール名
44          * @throws JspException エラーが発生したときにスローします。
45          */
46         @Override
47         public String getLabel() throws JspException {
48             return Util.escape(this.localeInfo.getDisplayName(), Escaping.STRICT_HTML);
49         }
50
51         /**
52          * このメニューをクリックしたときの URL を返します。
53          * <br/>
54          * このメニューをクリックするとロケールを選択したことになるので、CSJS のロケール変更リクエスト
55          * を送信するように実装します。
56          * @throws JspException エラーが発生したときにスローします。
57          */
58         @Override
59         public String getUrl() throws JspException {
60             return String.format("javascript:changeLocale('%s');", this.localeInfo.getLocale());

```

```

61     }
62
63     /**
64     * このUtilityItemProviderを表示するかどうかを返します。
65     * @return true:表示する
66     */
67     @Override
68     public boolean isVisible() {
69         return true;
70     }
71
72     /**
73     * このUtilityItemProviderが使用する外部CSJSのパスを返します。
74     * @return null:外部CSJSを使用しません。
75     */
76     @Override
77     public String scriptSource() {
78         return null;
79     }
80
81     /**
82     * このUtilityItemProviderのHTML表現を返します。
83     * <br/>
84     * 親メニューのリストの子要素とします。
85     * @return メニュー項目とするliタグ
86     * @throws JspException
87     */
88     @Override
89     public String toListString() throws JspException {
90         final StringBuilder buf = new StringBuilder();
91
92         final String url = "href=\"" + getUrl() + "\"";
93         buf.append("<li><a ").append(url).append(">");
94         buf.append(getLabel()).append("</a>");
95         buf.append("</li>");
96         return buf.toString();
97     }
98
99     }
100
101     /**
102     * 子メニューのリストを返します。
103     * @return List 子メニューのリスト
104     * @throws JspException エラーが発生したときにスローします。
105     */
106     @Override
107     public List<UtilityItemProvider> getChildren() throws JspException {
108         final List<UtilityItemProvider> children = new ArrayList<UtilityItemProvider>();
109         LocaleInfo[] localeInfos = SystemLocale.getLocaleInfos();
110
111         for (LocaleInfo localeInfo: localeInfos) {
112             children.add(new LocaleListItemProvider(localeInfo));
113         }
114
115         return children;
116     }
117
118     /**
119     * このプラグインを表示したときのラベルを返します。
120     * @return ラベル
121     * @throws JspException エラーが発生したときにスローします。

```

```

122 */
123 @Override
124 public String getLabel() throws JspException {
125     AccountContext context = Contexts.get(AccountContext.class);
126     Locale locale = context.getLocale();
127     String name = locale.getDisplayName(locale);
128
129     return Util.escape(name, Escaping.STRICT_HTML);
130 }
131
132 /**
133  * クリックしたときの遷移先 URL を返します。
134  * @return null
135  * @throws JspException エラーが発生したときにスローします。
136  */
137 @Override
138 public String getUrl() throws JspException {
139     return null;
140 }
141
142 /**
143  * ロケールの切り替え項目を表示するかどうかを決定します。
144  * <br/>
145  * 認証済ユーザであれば切り替え項目を表示します。
146  * @return 切り替え項目を表示する場合は true, しない場合は false
147  */
148 @Override
149 public boolean isVisible() {
150     AccountContext context = Contexts.get(AccountContext.class);
151     return context.isAuthenticated();
152 }
153
154 /**
155  * このプラグインを動作させるのにCSJSが必要な場合そのパスを返します。
156  */
157 @Override
158 public String scriptSource() {
159     return "sample/js/changeLocale.js";
160 }
161
162 /**
163  * このUtilityItemProviderのHTML表現を返します。
164  * @return liタグ
165  */
166 @Override
167 public String toListString() throws JspException {
168     final StringBuilder buf = new StringBuilder();
169
170     final String url = getUrl() == null ? StringUtil.EMPTY_STRING : "href=\"" + getUrl() + "\"";
171     buf.append("<li><a ").append(url).append(">").append(getLabel()).append("</a>");
172     if (getChildren() != null) {
173         buf.append("<ul>");
174         for (final UtilityItemProvider item : getChildren()) {
175             buf.append(item.toListString());
176         }
177         buf.append("</ul>");
178     }
179     buf.append("</li>");
180     return buf.toString();
181 }

```

plugin.xml

src/main/plugin/jp.co.intra_mart.sample.locale_item_provider/plugin.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <plugin>
3  <extension point="jp.co.intra_mart.foundation.ui.theme.utility.item">
4    <item
5      id="sample_localeitemprovider"
6      name="localeitemprovider"
7      classname="jp.co.intra_mart.sample.LocaleItemProvider"
8      version="1.0"
9      rank="1"
10     enable="true"
11    />
12 </extension>
13 </plugin>

```

CSJS

src/main/public/sample/js/changeLocale.js

```

1  function changeLocale(locale) {
2    (function($, locale) {
3      var form = $('<form/>')
4        .attr('id', 'sampleform')
5        .attr('action', 'user/settings/locale/update_locale')
6        .attr('method','POST');
7      var locale = $('<input/>')
8        .attr('type','hidden')
9        .attr('name','locale')
10       .attr('value',locale);
11      form.append(locale);
12
13      form.appendTo('body');
14      imuiAjaxSubmit('#sampleform', 'POST', 'json', 'home', []);
15    })(jQuery, locale);
16  }










```

フォルダ構成

```
├─ message.properties
├─ message_en.properties
├─ message_ja.properties
├─ message_zh_CN.properties
├─ module.xml
├─ src
├─ main
│  ├─ conf
│  │  ├─ message
│  │  ├─ routing-jssp-config
│  │  └─ routing-service-config
│  ├─ generated
│  ├─ java
│  │  └─ jp
│  │     └─ co
│  │        └─ intra_mart
│  │           └─ sample
│  │              └─ LocaleItemProvider.java
│  ├─ jssp
│  │  └─ src
│  ├─ plugin
│  │  └─ jp.co.intra_mart.sample.locale_item_provider
│  │     └─ plugin.xml
│  ├─ public
│  │  └─ sample
│  │     └─ js
│  │        └─ changeLocale.js
│  ├─ resources
│  ├─ schema
│  ├─ storage
│  │  ├─ public
│  │  └─ system
│  └─ webapp
├─ test
│  ├─ conf
│  │  ├─ message
│  │  ├─ routing-jssp-config
│  │  └─ routing-service-config
│  ├─ generated
│  ├─ java
│  ├─ jssp
│  │  └─ src
│  ├─ plugin
│  ├─ public
│  ├─ resources
│  ├─ schema
│  ├─ storage
│  │  ├─ public
│  │  └─ system
│  └─ webapp
```

intra-mart が提供するテーマの一覧

テーマ一覧

テーマサンプル画像	テーマ名	テーマID
	intra-mart Accel Platform 標準テーマ (黒)	im_theme_dropdown_black
	intra-mart Accel Platform 標準テーマ (青)	im_theme_dropdown_blue
	intra-mart Accel Platform 標準テーマ (茶)	im_theme_dropdown_brown
	intra-mart Accel Platform 標準テーマ (セラドングリーン)	im_theme_dropdown_celadon_green
	intra-mart Accel Platform 標準テーマ (ココアブラウン)	im_theme_dropdown_cocoa_brown
	intra-mart Accel Platform 標準テーマ (コッパーラスト)	im_theme_dropdown_copper_rust
	intra-mart Accel Platform 標準テーマ (灰色)	im_theme_dropdown_gray
	intra-mart Accel Platform 標準テーマ (緑)	im_theme_dropdown_green
	intra-mart Accel Platform 標準テーマ (ヒアシンスブルー)	im_theme_dropdown_hyacinth_blue

テーマサンプル画像

テーマ名

テーマID



intra-mart Accel Platform 標準テーマ (アイビーグリーン) im_theme_dropdown_ivy_green



intra-mart Accel Platform 標準テーマ (ラベンダーグレー) im_theme_dropdown_lavender_gray



intra-mart Accel Platform 標準テーマ (ミディアムパープル) im_theme_dropdown_mediumpurple



intra-mart Accel Platform 標準テーマ (ネイビーブルー) im_theme_dropdown_navy_blue



intra-mart Accel Platform 標準テーマ (オレンジ) im_theme_dropdown_orange



intra-mart Accel Platform 標準テーマ (オーキッドピンク) im_theme_dropdown_orchid_pink



intra-mart Accel Platform 標準テーマ (ピンク) im_theme_dropdown_pink



intra-mart Accel Platform 標準テーマ (赤) im_theme_dropdown_red



intra-mart Accel Platform 標準テーマ (レッドオレンジ) im_theme_dropdown_red_orange



intra-mart Accel Platform 標準テーマ (ローズグレー) im_theme_dropdown_rose_gray

テーマサンプル画像

テーマ名

テーマID



intra-mart Accel Platform 標準テーマ (サーモンピンク) im_theme_dropdown_salmon_pink



intra-mart Accel Platform 標準テーマ (紫) im_theme_dropdown_violet



intra-mart Accel Platform 標準テーマ (ヨットブルー) im_theme_dropdown_yacht_blue



intra-mart Accel Platform 標準テーマ (黄) im_theme_dropdown_yellow



intra-mart Accel Platform 標準テーマ (イエローグリーン) im_theme_dropdown_yellow_green



intra-mart Accel Platform 標準テーマ シンプルスタイル im_theme_dropdown_simple



iWP/iAF Version6 互換テーマ (青) im_theme_60_blue



iWP/iAF Version6 互換テーマ (緑) im_theme_60_green



iWP/iAF Version6 互換テーマ (オレンジ) im_theme_60_orange



iWP/iAF Version6 互換テーマ (赤) im_theme_60_red

テーマサンプル画像

テーマ名

テーマID



iWP/iAF Version5 互換テーマ (青)

im_theme_50_blue



iWP/iAF Version5 互換テーマ (緑)

im_theme_50_green



iWP/iAF Version5 互換テーマ (オレンジ)

im_theme_50_orange



iWP/iAF Version5 互換テーマ (赤)

im_theme_50_red