目次

改訂情報

変更年月日	変更内容
2012-12-21	初版
2014-04-01	第２版　下記を追加・変更しました ▪ 「*概要* 」の説明を追加 ▪ 「*サンプルプログラム* 」にtenantIdの指定を追加
2017-04-01	第３版　下記を変更しました ▪ 「*はじめに* 」の利用に関する注意を修正 ▪ 「*概要* 」の外部ソフトウェア接続モジュールに関するコラムを修正

はじめに

本書の目的

本書では外部ソフトウェア接続モジュールの仕様とそのプログラミング方法や注意点等について説明します。

注意

外部ソフトウェアとの連携は Web API Maker または IM-LogicDesigner を利用してください。

- Web API Maker については「Web API Maker プログラミングガイド」を参照してください。
- IM-LogicDesigner については「IM-LogicDesigner仕様書」を参照してください。

この機能は旧バージョンの互換用として提供されていますが、セキュリティ面での懸念があるため非推奨です。
利用される場合は十分な注意が必要です。

対象読者

次の開発者を対象としてます。

- intra-mart Accel Platform の外部ソフトウェア接続モジュールを利用したい開発者

本書の構成

本書は上記の対象読者に応じて次の２つの構成を取っています。

- *概要*

  外部ソフトウェア接続モジュールの概要について説明します。

- *サンプルプログラム*

  外部ソフトウェア接続モジュールを利用したプログラミング方法について説明します。

概要

項目

外部ソフトウェア接続モジュールとは

市販のアプリケーションパッケージから im-BizAPI の各種 API を呼び出して直接利用するなど、intra-mart と外部ソフトウェアを簡単に連携・接続できるモジュールです。
連携・接続する方法には、次の 2 通りの方法が用意されています。

- この「連携 API コネクタ」を使用して任意のプロセスと im-BizAPI を連携させる方法
- 外部ソフトウェアから Web サービスにより im-BizAPI の各種 API を呼び出す方法

「連携 API コネクタ」は im-BizAPI と連携するための Java ベース API として提供されているので、外部ソフトウェアが Java 実行環境であれば任意のプロセスと im-BizAPI を連携させることができます。
例えば、市販のポータルサーバ製品と組み合わせて、ポータル画面中に intra-mart の画面を表示したり、他のアプリケーションと連携してバッチ動作する独自の Java プロセスからユーザアカウント情報を操作したりすることが可能です。

> **コラム**
>
> ここでは Webサービスにより im-BizAPI の各種 API を呼び出す方法については説明しません。
> 「外部ソフトウェア接続モジュール」は互換用として提供されているものです。セキュリティ面での懸念があり、非推奨となっているため利用される場合は十分な注意が必要です。
> 「Webサービス」については以下のWebサービス各種マニュアルを参照してください。
>
> - Webサービス スクリプト開発プログラミングガイド
> - Webサービス Java開発プログラミングガイド
> - Webサービス 認証・認可 仕様書

仕様

外部ソフトウェア接続モジュールのインタフェースは、Java のクラスです。したがって、連携するアプリケーションは、Java インタフェースを利用できることが前提です。
また、Java インタフェースを利用することから、このモジュールが動作する環境には、Java-VM が必要です。

連携用のコネクタクラスは、ネットワークを介して intra-mart Accel Platform と連動します。したがって、ネットワークが利用可能な環境であることも前提条件です。
ただし、API は URL によってネットワーク通信を制御するため、開発者がネットワークを直接意識することはありません。

コネクタは intra-mart Accel Platform に対して、HTTP 接続を行います。連携用に指定する URL には、専用のサーブレットパス（/imart/HTTPActionEventListener（標準））を指定します。

ネットワークは、API に指定されたURL により解決されますが、この URL は intra-mart Accel Platform の運用形態により以下のような注意が必要です。

- intra-mart Accel Platform がスタンドアロンで運用されている場合は、そのスタンドアロンサーバへの接続 URL を指定します。
- intra-mart Accel Platform が分散環境で運用されている場合は、Web Server への接続 URL を指定します。

外部ソフトウェア接続モジュールの構成

外部ソフトウェア接続モジュールは、サーバ側モジュールとクライアント側モジュールの２つのモジュールで構成されます。

モジュール名	クラスアーカイブ	説明
外部連携クライアント	imaca_client-XXX-main.jar	呼び出し側の外部ソフトウェアに配置するクライアント側ロジックです。
外部連携 認証・認可	imaca_provider-XXX-main.jar	intra-mart Accel Platform に配置するサーバ側ロジックです。

サンプルプログラム

サンプル内容

このサンプルは intra-mart Accel Platform システム外の Java 実行環境から任意のページをリクエストする URL を取得するサンプルソースになります。 スクリプト開発モデルで作成されたページ「sample/chart/default_graph」をリクエストするための URL を作成します。

動作に必要な環境構築

外部ソフトウェア連携を行う場合は、以下の手順で連携に必要な環境を作成してください。

1. 外部連携クライアントモジュールに含まれているクラスアーカイブ・ファイル「imaca_client-XXX-main.jar」を連携させるアプリケーションが動作する環境にコピーしてください。
（jar ファイルは、WARファイル内の「WEB-INF/lib」にあります。）
2. コピーしたアーカイブファイルの「imaca_client-XXX-main.jar」に対してクラスパスを設定してください。

バージョンの異なる jar ファイルの場合、正常に連携動作させることができない可能性がありますので、パッチ等を適用した場合やリビジョンアップをした場合などは、十分注意をして ください。 なお、intra-mart Accel Platform に対してパッチを適用したときなどは、パッチに含まれる imaca_client-XXX-main.jar を上書きコピーしてご利用ください。

プログラムソース

```
package jp.co.intra_mart.sample.service.client.application;

import java.io.IOException;

import jp.co.intra_mart.foundation.service.client.application.HTTPActionEventHandler;
import
jp.co.intra_mart.foundation.service.client.application.HTTPActionEventHandlerException;
import jp.co.intra_mart.foundation.service.client.application.HTTPActionEventURL;
import
jp.co.intra_mart.foundation.service.client.application.PasswordSecurityHTTPActionEventFilterHa

import
jp.co.intra_mart.foundation.service.client.application.WebApplicationHTTPActionEventHandler;
import
jp.co.intra_mart.foundation.service.client.application.content.AccessibleLinkHTTPActionEventFilt
```

```
import jp.co.intra_mart.foundation.service.client.application.content.PresentationPageHTTPActionEvent


public class JSSPConnectURLCreator {

    /**
     * コマンドプロンプトからメインクラスとして指定された場合に実行されるメソッド。
     * @param args コマンドライン引数
     */
    public static void main(final String[] args) {
        try {
            // スクリプト開発モデルの画面をリクエストするためのＵＲＬを取得
            final String jsspURL = createJSSPURL();
            // 結果の表示
            System.out.println("URL: " + jsspURL);
        } catch (final Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * 指定のアカウント・パスワードによるセキュリティセッション環境で、 メニュー画面を作成してソースを返します。
     * @return ページURL
     * @throws IOException 入出力エラーが発生した場合にスローされます。
     * @throws HTTPActionEventHandlerException イベント実行時に例外が発生した場合にスローされます。
     */
    public static String createJSSPURL() throws IOException, HTTPActionEventHandlerException {
        // イベント実行ハンドラの作成
        final String path = "sample/chart/default_graph";
        HTTPActionEventHandler handler = new PresentationPageHTTPActionEventHandler(path);

        // 絶対パスでリンクするための定義
        handler = new AccessibleLinkHTTPActionEventFilterHandler(handler);

        // ログイン・セキュリティ環境の構築
        final String tenantId = "default";
        final String userCd = "user";
        final String password = "password";
        handler = new PasswordSecurityHTTPActionEventFilterHandler(handler, tenantId, userCd, password);

        // URL の取得
        final String url = "http://localhost:8080/imart/HTTPActionEventListener";
        final HTTPActionEventURL result = WebApplicationHTTPActionEventHandler.getURL(handler, url);
        return result.getURL();
    }

}
```

コラム

このソースをコンパイルするときは、「imaca_client-XXX-main.jar」をクラスパスに設定してください。
実行時は、このソースをコンパイルしてできたクラスファイルを、実行する Java プロセス環境から利用できる場所に保存してください。

> **注意**
>
> バーチャルテナントによる複数テナントの環境で使用する場合は、PasswordSecurityHTTPActionEventFilterHandlerのインスタンス生成時にアクセスするテナントの「tenantId」を指定します。
>
> 「tenantId」が指定されていない場合はデフォルトテナントに指定されているテナントにアクセスします。
>
> intra-mart Accel Platform 2013 Winter以前
> 　テナントIDの指定は必要ありません。

ログイン・セキュリティ環境の構築

ログインセッション環境でイベントを実行するためには、以下のいずれかの ActionEventFilterHandler を利用する必要があります。

- jp.co.intra_mart.foundation.service.client.application.GroupSecurityHTTPActionEventFilterHandler

  Anonymousユーザでイベントを実行します。

- jp.co.intra_mart.foundation.service.client.application.AccountSecurityHTTPActionEventFilterHandler

  指定されたユーザコードのユーザでログイン認証を行って、イベントを実行します。

- jp.co.intra_mart.foundation.service.client.application.PasswordSecurityHTTPActionEventFilterHandler

  指定されたユーザコード・パスワードでログイン認証を行って、イベントを実行します。

> **コラム**
>
> GroupSecurityHTTPActionEventFilterHandlerおよび、AccountSecurityHTTPActionEventFilterHandlerは標準の状態では利用できなくなっています。
> 利用したい場合は、「WEB-INF/web.xml」に記載されている HTTPActionEventListener の init-param 「use.account.security」の値を true に指定ください。

外部ソフトウェア連携時の認可設定

外部ソフトウェア連携モジュールを利用したイベント実行時に認可機能によるアクセス制御を行いたい場合は以下のActionEventFilterHandlerを ログイン・ログインセキュリティ用のActionEventFilterHandlerの前に設定してください。

- jp.co.intra_mart.foundation.service.client.application.AuthorizationHTTPActionEventFilterHandler

  指定された認可URIでアクセス制御を行います。

実装例

認可 URI 「service://sample/chart/default_graph」のリソースのアクセス権限を持つユーザのみ実行できるようにする場合は、以下のように実装します。

```
// イベント実行ハンドラの作成
final String path = "sample/chart/default_graph";
HTTPActionEventHandler handler = new PresentationPageHTTPActionEventHandler(path);

// 絶対パスでリンクするための定義
handler = new AccessibleLinkHTTPActionEventFilterHandler(handler);

// アクセス制御を行うための定義
handler = new AuthorizationHTTPActionEventFilterHandler(handler,
"service://sample/chart/default_graph", "execute");

// ログイン・セキュリティ環境の構築
final String tenantId = "default";
final String userCd = "user";
final String password = "password";
handler = new PasswordSecurityHTTPActionEventFilterHandler(handler, tenantId, userCd, password);

// URL の取得
final String url = "http://localhost:8080/imart/HTTPActionEventListener";
final HTTPActionEventURL result = WebApplicationHTTPActionEventHandler.getURL(handler, url);
return result.getURL();
```

コラム

認可機能によるアクセス制御を利用する場合は、あらかじめ利用するURIのリソースが設定してください。

注意

バーチャルテナントによる複数テナントの環境で使用する場合は、PasswordSecurityHTTPActionEventFilterHandlerのインスタンス生成時にアクセスするテナントの「tenantId」を指定します。

「tenantId」が指定されていない場合はデフォルトテナントに指定されているテナントにアクセスします。

intra-mart Accel Platform 2013 Winter以前
　テナントIDの指定は必要ありません。