

intra-mart im-J2EE Framework

Struts 連携ガイド 版

ver 0.1

2003 年 3 月 31 日

<< 変更履歴 >>

変更年月日	変更内容
2003/03/31	ver 0.1 (版)

<< 目次 >>

1	はじめに.....	1
1.1	目的.....	1
1.2	概要.....	1
1.3	注意.....	1
2	インストール.....	2
2.1	Struts	2
2.1.1	struts.jar	2
2.1.2	タグライブラリ.....	2
2.2	im-J2EE Framework Extension for Struts	2
3	アプリケーションの作成	3
3.1	前提.....	3
3.2	Struts の設定.....	3
3.2.1	web.xml (Struts)	3
3.2.2	タグライブラリの追加	4
3.2.3	struts-config.xml の設定	4
3.2.4	Struts 用に作成したクラス.....	4
3.3	intra-mart のメニューから Struts への連携.....	4
3.3.1	仕組.....	4
3.3.2	設定方法.....	5
3.4	フィルタ.....	6
3.4.1	エンコーディング	6
3.4.2	ログインセッション.....	7
3.5	Struts から im-J2EE Framework のイベントフレームワークへの連携.....	7
3.5.1	仕組.....	7
3.5.2	呼び出し方法.....	9
4	サンプル.....	14
4.1	前提.....	14
4.2	画面.....	14
4.2.1	フレーム分割 (stf_ndp.jsp)	14
4.2.2	ヘッダ (stf_head.jsp)	14
4.2.3	ブランク (blank.jsp)	15
4.2.4	詳細 (stf_body.jsp)	15
4.2.5	成功 (success.jsp)	17
4.3	ActionForm	18
4.3.1	社員選択 (StaffSelectForm.java)	18
4.3.2	更新データ (UpdateForm.java)	18
4.3.3	削除データ (DeleteForm.java)	20
4.4	Action.....	22
4.4.1	一覧表示 (StaffSelectAction.java)	22
4.4.2	更新 (UpdateAction.java)	22
4.4.3	削除 (DeleteAction.java)	24
4.5	データベース.....	25
4.6	Struts の設定.....	25
4.7	Web アプリケーションの設定.....	26
4.8	メニュー	27
4.9	その他の im-J2EE Framework.....	27

4.9.1	イベント	27
4.9.2	データ	28
4.10	実行	28
付録 A	im-J2EE Framework と Struts	30
付録 B	Struts のライセンス.....	31

1 はじめに

1.1 目的

im-J2EE Framework は複数のサブフレームワーク(サービフレームワーク、イベントフレームワーク等)を含むが、それぞれは疎結合であるため他のフレームワークとも大きな変更を加えずに連携できる。ここでは Web 層のフレームワークとして Jakarta プロジェクト¹で公開されている Struts を選択し、Struts と intra-mart を連携する方法について述べる。

1.2 概要

「2 インストール」では Struts や拡張モジュールを intra-mart へインストールする方法について述べる。

「3 アプリケーションの作成」では Struts と im-J2EE Framework を連携する場合のアプリケーションの作成方法について述べる。

「4 サンプル」では Struts と im-J2EE Framework を連携するサンプルを載せる。

1.3 注意

- 本書で扱うソフトウェアのバージョンは以下のとおりである。
 - ◆ Struts: 1.0.2
 - ◆ intra-mart: BaseModule 4.0
 - ◆ 上記以外のソフトウェアについてはそれぞれのソフトウェアで対象となるものに準ずる。
- 本書の内容は Struts と im-J2EE Framework を接続するための参考資料である。Struts や im-J2EE Framework の詳細については紹介していない。
- Struts の上で Servlet 2.3 の API を使いたい場合、Struts のソースに修正を加えて再コンパイルする必要があるがここではその方法については述べない。Servlet 2.3 の API を使わない場合は再コンパイルの必要はない。
- 本書で紹介している拡張モジュールは 版であり、サポート対象外である。正式版では API 等が変更される場合もある。
- 本書の内容は 版であり、サポート対象外である。正式版では内容が修正される場合もある。

¹ <http://jakarta.apache.org/>

2 インストール

ここでは Struts と im-J2EE Framework の拡張モジュールのインストールについて説明する。

2.1 Struts

2.1.1 struts.jar

Struts のエンジン部分のインストールは struts.jar を以下のディレクトリにコピーするだけでよい。

<Application Runtime のインストールパス>/doc/imart/WEB-INF/lib

struts.jar がない場合、以下の Web サイトからダウンロードできる。

<http://jakarta.apache.org/builds/jakarta-struts/release/v1.0.2/>

上記のサイトから jakarta-struts-1.0.2.tar.gz または jakarta-struts-1.0.2.zip をダウンロードして展開すると、以下の場所に struts.jar が作成される。

<展開先のディレクトリ>/jakarta-struts-1.0.2/lib

また、上記の jakarta-struts-1.0.2.zip は imart-struts.zip を展開しても入手できる。

<展開先のディレクトリ>/imart-struts/jakarta-struts-1.0.2.zip

2.1.2 タグライブラリ

Struts のタグライブラリを使用するために、タグライブラリ記述ファイルを以下のディレクトリにコピーする。

<Application Runtime のインストールパス>/doc/imart/WEB-INF

コピーするタグライブラリ記述ファイルは以下のものである。

- struts-bean.tld
- struts-html.tld
- struts-logic.tld
- struts-template.tld

これらのタグライブラリ記述ファイルがない場合、「2.1.1 struts.jar」で示した jakarta-struts-1.0.2.tar.gz または jakarta-struts-1.0.2.zip を展開することで入手することができる。展開すると、以下の場所に上記のタグライブラリ記述ファイルが作成される。

<展開先のディレクトリ>/jakarta-struts-1.0.2/lib

2.2 im-J2EE Framework Extension for Struts

im-J2EE Framework の拡張モジュールのインストールは imart-struts.zip を展開したときにできる以下の jar ファイルをコピーするだけでよい。

<展開先のディレクトリ>/imart-struts/imart_struts.jar

展開された imart_struts.jar を以下のディレクトリにコピーする。

<Application Runtime のインストールパス>/doc/imart/WEB-INF/lib

3 アプリケーションの作成

ここでは intra-mart と Struts を連携させたアプリケーションの作成方法を説明する。

3.1 前提

ここでは以下の内容を前提とする。

- intra-mart、im-J2EE Framework 拡張モジュールおよび Struts がインストール済みである。
- Struts の ActionServlet へのマッピングは拡張子でマッピングを行う方式(拡張子マッピング)を採用する。
- 作成するアプリケーションは intra-mart にログインしてから有効であるものとする。

3.2 Struts の設定

3.2.1 web.xml (Struts)

Struts の ActionServlet へアクセスするためにはサーブレットのマッピングが必要である。Struts には標準では以下の 2 つのマッピング方式が用意されている。

- 拡張子マッピング (*.do)
- パスマッピング (/do/*)

ここでは拡張子マッピングを採用する。

また、Struts には JSP で非常に役に立つタグライブラリが存在するのでこれらを使用するように設定する。

web.xml では以下のように設定する。

```
...

<!-- Action Servlet Configuration -->
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
</servlet>

...

<!-- Action Servlet Mapping -->
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

...

<!-- Struts tag liblalies -->
```

```
<taglib>
  <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-template.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-template.tld</taglib-location>
</taglib>
. . .
```

web.xml は以下のディレクトリに存在する。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF
```

3.2.2 タグライブラリの追加

3.2.3 struts-config.xml の設定

struts-config.xml は Struts の動作を決定する設定ファイルである。このファイルを以下のディレクトリに作成する。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF
```

3.2.4 Struts 用に作成したクラス

Struts 上で動作するクラスを作製した場合、以下のディレクトリに置く。この場合、通常の java コマンドのクラスパスと同じルールが適用される。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes
```

作成したクラスを JAR ファイルにまとめた場合、以下のディレクトリに置く。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF/lib
```

3.3 intra-mart のメニューから Struts への連携

3.3.1 仕組

intra-mart のメニューから JSP やサーブレットを直接呼ぶことは可能であるが、ログイン情報(ログインユーザ、ログイングループ)などは開発者が自分で保持する仕組を用意する必要がある(「図 3-1 直接 Struts の画面に遷移」参照)。この手間を省略するために、拡張モジュールの StrutsConnectServlet を利用する(「図 3-2 前処理を行って Struts のページに遷移」参照)。

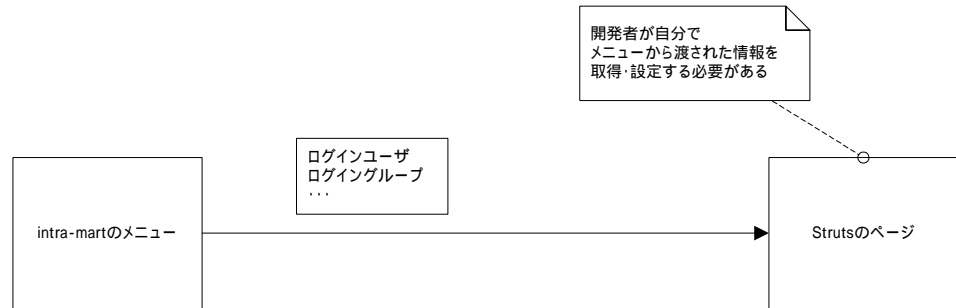


図 3-1 直接 Struts の画面に遷移

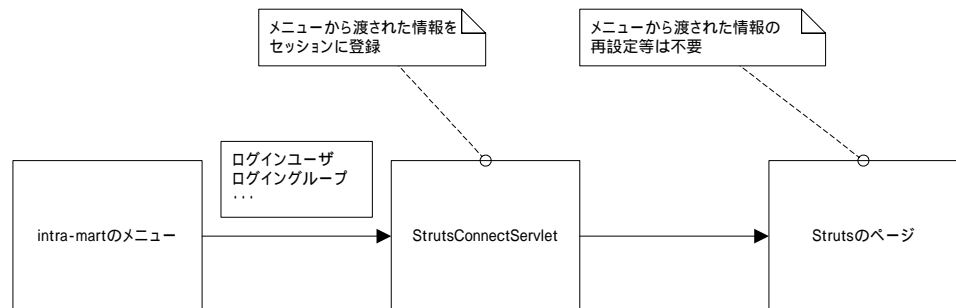


図 3-2 前処理を行って Struts のページに遷移

3.3.2 設定方法

3.3.2.1 web.xml(メニューとの接続)

メニューから Struts への接続を行う Servlet の設定を行う。設定は web.xml に以下の内容を追記することで行う。

```

. . .

<!-- Action Servlet Configuration -->
<servlet>
  <servlet-name>intramart-struts connector</servlet-name>
  <servlet-class>
    jp.co.intra_mart.framework.extension.struts.action.StrutsConnectServlet
  </servlet-class>
</servlet>

. . .

<!-- Action Servlet Mapping -->
<servlet-mapping>
  <servlet-name>intramart-struts connector</servlet-name>
  <url-pattern>/StrutsConnectServlet</url-pattern>
</servlet-mapping>

```

...

3.3.2.2 メニュー

intra-mart のメニューから Struts へは直接遷移せず、一度 StrutsConnectServlet を経由するように設定する。システム管理者でログインした後、メニューの[システム設定]-[メニュー]で以下の内容のメニューを新規に追加する。

項目	内容
フォルダ・ページ名	任意の名前
説明	任意の説明
ソート番号	任意の番号
PC/mobile	" PC 用ページとしてのみ使用" または "PC/mobile 用ページ両方で使用"
形式	ページ [Servlet or JSP]
ページ URL	<コンテキストパス>/StrutsConnectServlet
ページ引数	遷移先の Struts のパス
ロール	任意のロール

3.4 フィルタ

3.4.1 エンコーディング

Servlet 2.3 と現在の多くの Web ブラウザの仕様上、Request の getParameter メソッドで取得できるパラメータは何もしないと ISO-8859-1 になることが多い。そのため、日本語等の文字が正しく取得できない(いわゆる文字化けが起こる)場合が多い。Struts は文字エンコーディングについては特に何もしていないため、この問題はそのまま引き継がれる。

これを解消するために、im-J2EE Framework で用意されているエンコーディングのフィルタを利用する。

intra-mart をインストールした直後の web.xml には以下のようなエンコーディング用のフィルタの定義がある。

```
<filter>
  <filter-name>EncodingFilter</filter-name>
  <filter-class>jp.co.intra_mart.framework.base.service.EncodingFilter</filter-class>
</filter>
```

このフィルタを利用するためには web.xml に以下の内容を追記する。

```
<filter-mapping>
  <filter-name>EncodingFilter</filter-name>
  <servlet-name>action</servlet-name>
</filter-mapping>
<filter-mapping>
  <filter-name>EncodingFilter</filter-name>
  <servlet-name>intramart-struts connector</servlet-name>
</filter-mapping>
```

ここで定義した action は「3.2.1 web.xml(Struts)」で、intramart-struts connector は「3.3.2.1 web.xml(メニューとの接続)」で定義したサーブレットの名前である。

3.4.2 ログインセッション

im-J2EE Framework ではログインセッションを以下のフィルタで管理している。

```
<filter>
  <filter-name>SessionFilter</filter-name>
  <filter-class>jp.co.intra_mart.framework.base.session.SessionFilter</filter-class>
</filter>
```

Struts 上でも intra-mart と同様のセッション管理を行いたい場合、web.xml に以下の内容を追記する。(注意:この内容は「3.4.1 エンコーディング」で追記したフィルタよりも後ろに追記する)

```
<filter-mapping>
  <filter-name>SessionFilter</filter-name>
  <servlet-name>action</servlet-name>
</filter-mapping>
```

3.5 Struts から im-J2EE Framework のイベントフレームワークへの連携

Struts は J2EE BluePrints と照らし合わせてみると Web 層に特化したフレームワークである。これは im-J2EE Framework のサービスフレームワークとほぼ一致する。そのため、ここでは Web 層の制御を Struts で行い、バックエンドのビジネスロジックを im-J2EE Framework のイベントフレームワークで行う方法について述べる。

3.5.1 仕組

Struts と im-J2EE Framework は Struts の Action クラスを通じて連携する。im-J2EE Framework のイベントフレームワークを使わずに Action クラス内でビジネスロジックを書くことも可能であるが、ビジネスロジックは外部に出すことを推奨する。実際、Struts のドキュメント「The Struts User's Guide」の「1.4 The Model: System State and Business Logic JavaBeans」²にも同様なことが書かれている。

そこで Struts と im-J2EE Framework を連携する場合、Action クラスはビジネスロジックの Facade(窓口)として実装する。「図 3-3 im-J2EE Framework のイベントフレームワーク」と「図 3-4 Struts と im-J2EE Framework の連携」を参照。

² <http://jakarta.apache.org/struts/doc-1.0.2/userGuide/introduction.html#modelConcepts>

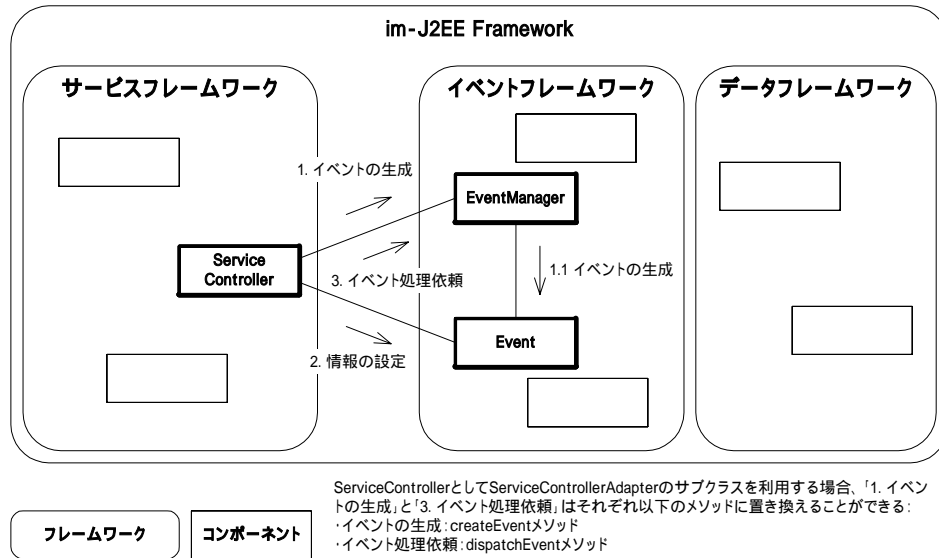


図 3-3 im-J2EE Framework のイベントフレームワーク

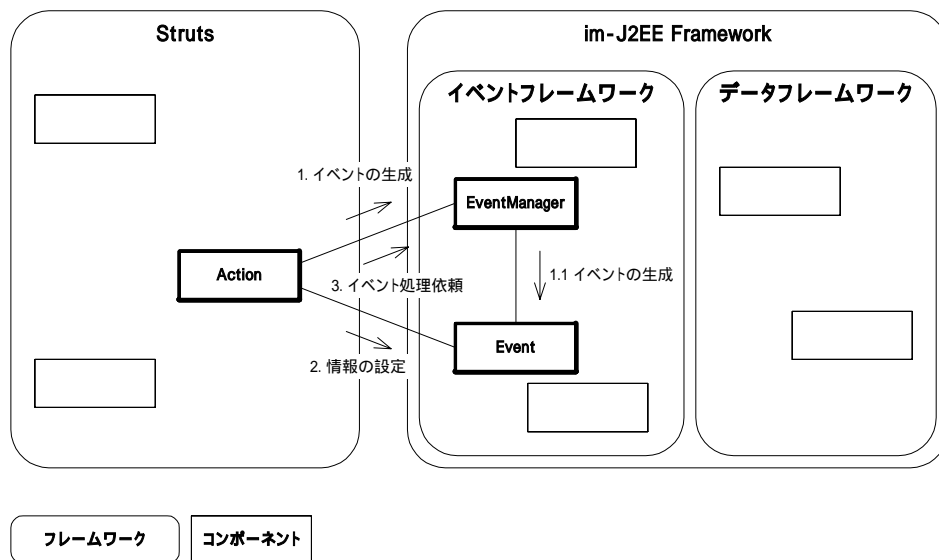


図 3-4 Struts と im-J2EE Framework の連携

3.5.2 呼び出し方法

Struts の Action から im-J2EE Framework のイベントフレームワークを扱う方法として以下のものが考えられる。

- イベントフレームワークを直接利用
- 共通メソッドが含まれるクラスを継承して利用
- 共通メソッドが含まれるクラスに委譲して利用

これらの中では「共通メソッドが含まれるクラスに委譲して利用」が最も推奨される。この方法は拡張しやすく、変更に伴う影響が少ないものと思われる。

3.5.2.1 イベントフレームワークを直接利用

im-J2EE Framework のイベントフレームワークを Action クラスから直接利用する場合、「図 3-5 直接利用」のような構造となる。この場合のコードは「図 3-6 Action から直接利用」に示すようなものになる。この場合、イベントの生成から処理結果の取得までを開発者が自分でコーディングする必要がある。

この方法は im-J2EE Framework のイベントフレームワークに接続するすべての Action クラスに対して必要である。そのため、メンテナンスなどの観点から推奨されない。

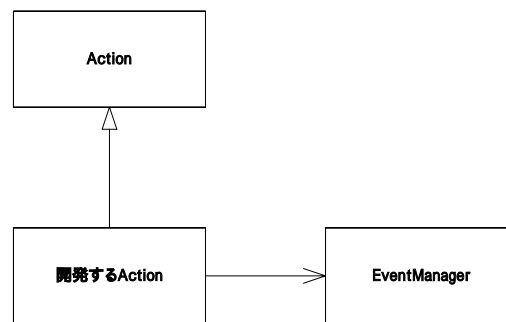


図 3-5 直接利用

```

import java.io.IOException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import jp.co.intra_mart.framework.base.session.SessionManager;
import jp.co.intra_mart.framework.base.session.SessionPropertyHandler;
import jp.co.intra_mart.framework.base.event.Event;
import jp.co.intra_mart.framework.base.event.EventManager;
import jp.co.intra_mart.framework.base.event.EventResult;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

public class TestAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
  
```

```
        HttpServletRequest request,
        HttpServletResponse response)
throws IOException, ServletException {

    // ログイン情報の取得
    HttpSession session = request.getSession();
    SessionManager sm = SessionManager.getSessionManager();
    SessionPropertyHandler sh = sm.getSessionPropertyHandler();
    String userID = session.getAttribute(sh.getLoginUserIDKey());
    String groupID = session.getAttribute(sh.getLoginGroupIDKey());

    // イベントの取得
    EventManager em = EventManager.getEventManager();
    TestEvent event = (TestEvent)em.createEvent(application_id,
                                                key,
                                                userID,
                                                groupID);

    // イベントの設定
    event.setAAA(aaa);
    event.setBBB(bbb);

    // イベントの実行
    TestEventResult result = em.dispatch(event);

    // イベント実行結果による処理・遷移
    ...
}
}
```

図 3-6 Action から直接利用

3.5.2.2 共通メソッドが含まれるクラスを継承して利用

これはイベントフレームワークを利用するメソッドを実装した Action クラスのサブクラスを作成する方法である。この方法では、「図 3-7 継承を利用」のような構造となる。この場合のコードは「図 3-8 イベントフレームワークを使用するメソッドがある Action クラス」に示すようなものになる。

開発者は以下のクラスを継承して新しい Action クラスを作成する。

```
jp.co.intra_mart.framework.extension.struts.action.IntramartAction
```

この方法を採用した場合、jp.co.intra_mart.framework.base.service.ServiceControllerAdapter とほぼ同様の感覚で開発者はコーディングすることが可能である。欠点としては、開発者が他の Action クラスを extends して開発をしたい場合、多重継承の問題が出てくるといった点が挙げられる。

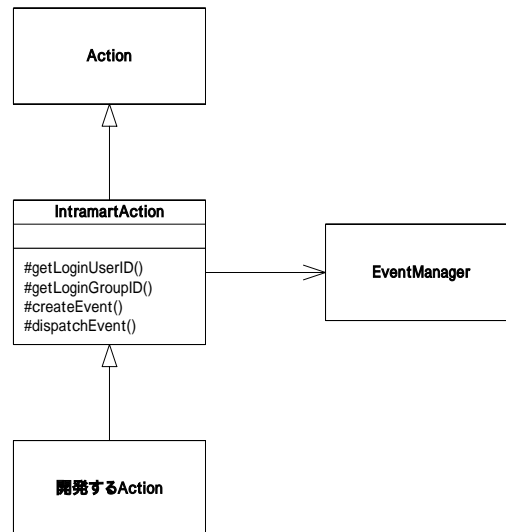


図 3-7 継承を利用

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import jp.co.intra_mart.framework.extension.struts.action.IntramartAction;

import java.io.IOException;
import javax.servlet.ServletException;

import jp.co.intra_mart.framework.system.exception.SystemException;

public class SampleAction extends IntramartAction {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException
    {
        // イベントの取得
        SampleEvent event = null;
        try {
            event = (DeleteStaffEvent)createEvent("sampleApp",
                                                  "sampleKey",
                                                  request,
                                                  response);
        } catch (SystemException e) {
            throw new ServletException(e.getMessage(), e);
        }
    }
}

```

```

    }

    // イベントの設定
    SampleForm sampleForm = (SampleForm)form;
    event.setSampleData(sampleForm.getSampleData());

    // イベントの実行
    try {
        dispatchEvent(event);
    } catch (Exception e) {
        throw new ServletException(e.getMessage(), e);
    }

    // 次の画面の準備
    return mapping.findForward("next");
}
}

```

図 3-8 イベントフレームワークを使用するメソッドがある Action クラス

3.5.2.3 共通メソッドが含まれるクラスに委譲して利用

これはイベントフレームワークを利用するメソッドを実装したユーティリティクラスを Aciton クラスから利用する方法である。この方法では、「図 3-9 委譲を利用」のような構造となる。この場合のコードは「図 3-10 ユーティリティクラスの利用」に示すようなものになる。

開発者は以下のクラスを利用するように Action クラスを作成する。

```

jp.co.intra_mart.framework.extension.common.util.ServiceUtils

```

この場合、「3.5.2.2 共通メソッドが含まれるクラスを継承して利用」のような多重継承の問題は出ないが、ユーティリティを使うという点で開発者は ServiceControllerAdapter とは若干コーディングスタイルを変更する必要がある。

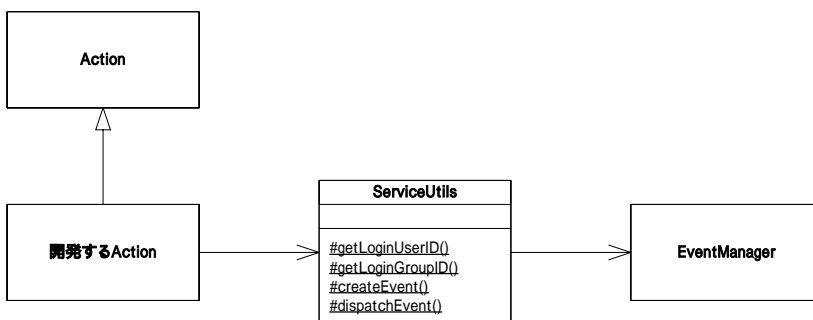


図 3-9 委譲を利用

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import jp.co.intra_mart.framework.extension.common.util.ServiceUtils;

import java.io.IOException;
import javax.servlet.ServletException;

import jp.co.intra_mart.framework.system.exception.SystemException;

public class SampleAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException {
        // イベントの取得
        SampleEvent event = null;
        try {
            event = (SampleEvent)ServiceUtils.createEvent("sampleApp",
                                                         "sampleKey",
                                                         request,
                                                         response);
        } catch (SystemException e) {
            throw new ServletException(e.getMessage(), e);
        }

        // イベントの設定
        SampleForm sampleForm = (SampleForm)form;
        event.setSampleData(sampleForm.getSampleData());

        // イベントの実行
        try {
            ServiceUtils.dispatchEvent(event);
        } catch (Exception e) {
            throw new ServletException(e.getMessage(), e);
        }

        // 次の画面の準備
        return mapping.findForward("next");
    }
}
```

図 3-10 ユーティリティクラスの利用

4 サンプル

ここでは Struts を利用した場合のサンプルを掲載する。intra-mart BaseModule 4.0 の im-J2EE Framework のサンプルであるマスタメンテナンスと同じものを作る。この場合、サービフレームワークの部分 (JSP、ServiceController、Transition など) が Struts に置き換わっている。

このサンプルの内容は imart-struts.zip の中にも含まれている。

<展開先のディレクトリ>/imart-struts/sample

4.1 前提

このサンプルを使うときの前提を以下に示す。

- intra-mart のインストールが終了していること。
- 「2 インストール」で示した Struts と連携モジュールのインストールが両方とも終了していること。
- インストール後にロール・ユーザ等のデータを変更していないこと。
- intra-mart のコンテキストパスが"/imart"であること。Application Runtime をインストールした後に設定ファイル(imart.xml など)を修正していなければ特に気にする必要はない。
- intra-mart のサンプルを登録していること。

4.2 画面

画面は JSP で作成する。ここに載せる JSP は以下の場所に置くものとする。

<Application Runtime のインストールパス>/doc/imart/struts/staff

4.2.1 フレーム分割(stf_ndp.jsp)

```
<%@ page contentType="text/html; charset=Windows-31J" pageEncoding="Windows-31J" %>
<%@ page errorPage="/j2ee/document/error/error.jsp" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=Shift_JIS">
    <TITLE>社員マスタメンテフレーム定義</TITLE>
  </HEAD>

  <FRAMESET ROWS="200, *">
    <FRAME name="mmtMain"
      src="<%= response.encodeURL(request.getContextPath() +
        "/tutorial/index_next_stfhead.do") %>" /><BR>
    <FRAME name="mmtBody"
      src="<%= response.encodeURL(request.getContextPath() +
        "/tutorial/index_next_blank.do") %>" /><BR>
  </FRAMESET>

</HTML>
```

4.2.2 ヘッダ(stf_head.jsp)

```
<%@ page contentType="text/html; charset=Windows-31J" pageEncoding="Windows-31J" %>
<%@ page errorPage="/j2ee/document/error/error.jsp" %>
```

```

<%@ taglib prefix="tutorial" uri="http://www.intra-mart.co.jp/taglib/core/framework" %>
<%@ taglib prefix="html" uri="/WEB-INF/struts-html.tld" %>
<%@ taglib prefix="bean" uri="/WEB-INF/struts-bean.tld" %>

<tutorial:HelperBean id="HeadBean" class="staff.view.bean.StaffHeadHelperBean" />
<bean:define id="staffs" name="HeadBean" property="staff" type="java.util.Collection" />

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html:html>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=Shift_JIS">
<TITLE>社員情報検索画面</TITLE>
</HEAD>

<BODY bgcolor="whitesmoke" onLoad="document.select_form.submit();">
<CENTER>
<B>社員情報検索画面</B>
<BR>

<html:form name="select_form" action="/tutorial/on_change.do" target="mmtBody"
method="post" type="staff.struts.action.StaffSelectForm" scope="request">
<TABLE border="1">
<TR>
<TH bgcolor="lightpink">社員名</TH>
</TR>
<TR>
<TD>
<html:select property="staff_cd" onchange="document.select_form.submit();">
<html:options collection="staffs" property="staffCd"
labelProperty="stfNameKanji" />
</html:select>
</TD>
</TR>
</TABLE>
</html:form>

<html:form name="last_information" action="/tutorial/last.do" method="post"
type="org.apache.struts.action.ActionFormBean">
<INPUT type="submit" value="最新の情報">
</html:form>

</CENTER>
</BODY>
</html:html>

```

4.2.3 ブランク (blank.jsp)

```

<HTML>
<HEAD>
<TITLE>White Screen</TITLE>
</HEAD>
<BODY bgcolor="WhiteSmoke">
</BODY>
</HTML>

```

4.2.4 詳細 (stf_body.jsp)

```
<%@ page contentType="text/html; charset=Windows-31J" pageEncoding="Windows-31J" %>
<%@ page errorPage="/j2ee/document/error/error.jsp" %>
<%@ taglib prefix="tutorial" uri="http://www.intra-mart.co.jp/taglib/core/framework" %>
<%@ taglib prefix="html" uri="/WEB-INF/struts-html.tld" %>
<%@ taglib prefix="bean" uri="/WEB-INF/struts-bean.tld" %>
<%@ taglib prefix="logic" uri="/WEB-INF/struts-logic.tld" %>

<tutorial:HelperBean id="bodyBean" class="staff.view.bean.StaffBodyHelperBean" />
<bean:define id="staffDetail" name="bodyBean" property="staff"
    type="staff.model.object.StaffModelObject" />

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html:html>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=Shift_JIS">
    <TITLE>社員情報詳細画面</TITLE>
  </HEAD>

  <BODY bgcolor="whitesmoke">
    <CENTER>
      <B>社員情報詳細画面</B>
      <BR>
      <logic:notPresent name="staffDetail">
        該当するデータが存在しません。
      </logic:notPresent>
      <logic:present name="staffDetail">
        <html:form action="/tutorial/update.do" method="post" scope="request">
          <TABLE border="1">
            <TR>
              <TD bgcolor="lightpink"><B>社員コード</B></TD>
              <TD><bean:write name="staffDetail" property="staffCd" />
                <html:hidden name="staffDetail" property="staffCd" /></TD>
            </TR>
            <TR>
              <TD bgcolor="lightpink"><B>社員名漢字</B></TD>
              <TD><html:text name="staffDetail" property="stfNameKanji" /></TD>
            </TR>
            <TR>
              <TD bgcolor="lightpink"><B>社員名カナ</B></TD>
              <TD><html:text name="staffDetail" property="stfNameKana" /></TD>
            </TR>
            <TR>
              <TD bgcolor="lightpink"><B>社員名英字</B></TD>
              <TD><html:text name="staffDetail" property="stfNameEng" /></TD>
            </TR>
          </TABLE>
          <BR>
          <INPUT type="submit" value="更新">

          <INPUT type="button" value="削除" onClick="document.delete_form.submit();">

        </html:form>
        <html:form action="/tutorial/delete.do" method="post" name="delete_form"
            type="staff.struts.action.DeleteForm" scope="request">
          <html:hidden name="staffDetail" property="staffCd" />
        </html:form>
      </logic:present>
    </CENTER>
  </BODY>
</html>
```

```
</CENTER>  
</BODY>  
</html:html>
```

4.2.5 成功 (success.jsp)

```
<%@ page contentType="text/html; charset=Windows-31J" pageEncoding="Windows-31J" %>  
<HTML>  
<HEAD></HEAD>  
<BODY bgcolor="WhiteSmoke">  
<CENTER>  
<H1>成功しました。</H1>  
</CENTER>  
</BODY>  
</HTML>
```

4.3 ActionForm

ActionForm のパッケージ名は以下のとおりとする。

```
staff.struts.action
```

これらをコンパイルしたクラス (intra-mart BaseModule の場合はソースファイルでも可) を以下のディレクトリに置く。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes/staff/struts/action
```

4.3.1 社員選択 (StaffSelectForm.java)

```
/*
 * StaffSelectForm.java
 *
 * Created on 2003/03/31, 12:00
 */

package staff.struts.action;

import org.apache.struts.action.ActionForm;

/**
 * 選択された社員の情報です。
 */
public class StaffSelectForm extends ActionForm {

    /**
     * 社員コード
     */
    private String staff_cd;

    /**
     * 社員コードを設定します。
     *
     * @param staff_cd 社員コード
     */
    public void setStaff_cd(String staff_cd) {
        this.staff_cd = staff_cd;
    }

    /**
     * 社員コードを取得します。
     *
     * @return 社員コード
     */
    public String getStaff_cd() {
        return this.staff_cd;
    }
}
```

4.3.2 更新データ (UpdateForm.java)

```
/*
 * UpdateForm.java
 *
 * Created on 2003/03/31, 12:00
```



```
*/  
  
package staff.struts.action;  
  
import org.apache.struts.action.ActionForm;  
  
/**  
 * 社員情報の更新内容です。  
 */  
public class UpdateForm extends ActionForm {  
  
    /**  
     * 社員コード  
     */  
    private String staffCd;  
  
    /**  
     * 社員名 (カナ)  
     */  
    private String stfNameKana;  
  
    /**  
     * 社員名 (漢字)  
     */  
    private String stfNameKanji;  
  
    /**  
     * 社員名 (英語)  
     */  
    private String stfNameEng;  
  
    /**  
     * 社員コードを設定します。  
     *  
     * @param staffCd 社員コード  
     */  
    public void setStaffCd(String staffCd) {  
        this.staffCd = staffCd;  
    }  
  
    /**  
     * 社員コードを取得します。  
     *  
     * @return 社員コード  
     */  
    public String getStaffCd() {  
        return this.staffCd;  
    }  
  
    /**  
     * 社員名 (カナ) を設定します。  
     *  
     * @param stfNameKana 社員名 (カナ)  
     */  
    public void setStfNameKana(String stfNameKana) {  
        this.stfNameKana = stfNameKana;  
    }  
}
```

```
/**
 * 社員名（カナ）を取得します。
 *
 * @return 社員名（カナ）
 */
public String getStfNameKana() {
    return this.stfNameKana;
}

/**
 * 社員名（漢字）を設定します。
 *
 * @param stfNameKanji 社員名（漢字）
 */
public void setStfNameKanji(String stfNameKanji) {
    this.stfNameKanji = stfNameKanji;
}

/**
 * 社員名（漢字）を取得します。
 *
 * @return 社員名（漢字）
 */
public String getStfNameKanji() {
    return this.stfNameKanji;
}

/**
 * 社員名（英語）を設定します。
 *
 * @param stfNameEng 社員名（英語）
 */
public void setStfNameEng(String stfNameEng) {
    this.stfNameEng = stfNameEng;
}

/**
 * 社員名（英語）を設定します。
 *
 * @return 社員名（英語）
 */
public String getStfNameEng() {
    return this.stfNameEng;
}
}
```

4.3.3 削除データ (DeleteForm.java)

```
/**
 * DeleteForm.java
 *
 * Created on 2003/03/31, 12:00
 */

package staff.struts.action;

import org.apache.struts.action.ActionForm;
```

```
/**
 * 削除する社員の情報です。
 */
public class DeleteForm extends ActionForm {

    /**
     * 社員コード
     */
    private String staffCd;

    /**
     * 社員コードを設定します。
     *
     * @param staffCd 社員コード
     */
    public void setStaffCd(String staffCd) {
        this.staffCd = staffCd;
    }

    /**
     * 社員コードを取得します。
     *
     * @return 社員コード
     */
    public String getStaffCd() {
        return this.staffCd;
    }
}
```

4.4 Action

Action のパッケージ名は以下のとおりとする。

```
staff.struts.action
```

これらをコンパイルしたクラス (intra-mart BaseModule の場合はソースファイルでも可) を以下のディレクトリに置く。

```
<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes/staff/struts/action
```

4.4.1 一覧表示 (StaffSelectAction.java)

```
* StaffSelectAction.java
*
* Created on 2003/03/31, 12:00
*/

package staff.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import jp.co.intra_mart.framework.extension.common.util.ServiceUtils;

import java.io.IOException;
import javax.servlet.ServletException;
import jp.co.intra_mart.framework.system.exception.SystemException;

/**
 * 一覧から社員を選択した時の Action です。
 */
public class StaffSelectAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException {
        // 社員コードの設定
        StaffSelectForm selectForm = (StaffSelectForm)form;
        request.setAttribute("staff_cd", selectForm.getStaff_cd());

        // 次の画面の準備
        return mapping.findForward("on_change");
    }
}
```

4.4.2 更新 (UpdateAction.java)

```
/*
 * UpdateAction.java
 *
 * Created on 2003/03/31, 12:00
 */
```

```
package staff.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import jp.co.intra_mart.framework.extension.common.util.ServiceUtils;

import staff.model.event.UpdateStaffEvent;
import staff.model.object.StaffModelObject;

import java.io.IOException;
import javax.servlet.ServletException;
import jp.co.intra_mart.framework.system.exception.SystemException;

/**
 * 社員情報を更新する時の Action です。
 */
public class UpdateAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException {
        // イベントの取得
        UpdateStaffEvent event = null;
        try {
            event = (UpdateStaffEvent)ServiceUtils.createEvent("tutorial",
                                                                "update",
                                                                request,
                                                                response);
        } catch (SystemException e) {
            throw new ServletException(e.getMessage(), e);
        }

        // イベントに社員情報を格納
        UpdateForm updateForm = (UpdateForm)form;
        StaffModelObject modelObject = new StaffModelObject();
        modelObject.setStaffCd(updateForm.getStaffCd());
        modelObject.setStfNameKana(updateForm.getStfNameKana());
        modelObject.setStfNameKanji(updateForm.getStfNameKanji());
        modelObject.setStfNameEng(updateForm.getStfNameEng());

        // イベントの設定
        event.setModelObject(modelObject);
        try {
            ServiceUtils.dispatchEvent(event);
        } catch (Exception e) {
            throw new ServletException(e.getMessage(), e);
        }

        // 次の画面の準備
        return mapping.findForward("update");
    }
}
```

```
}
```

4.4.3 削除(DeleteAction.java)

```
/*
 * DeleteAction.java
 *
 * Created on 2003/03/31, 12:00
 */

package staff.struts.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import jp.co.intra_mart.framework.extension.common.util.ServiceUtils;

import staff.model.event.DeleteStaffEvent;

import java.io.IOException;
import javax.servlet.ServletException;
import jp.co.intra_mart.framework.system.exception.SystemException;

/**
 * 社員情報を削除する時の Action です。
 */
public class DeleteAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException {
        // イベントの取得
        DeleteStaffEvent event = null;
        try {
            event = (DeleteStaffEvent)ServiceUtils.createEvent("tutorial",
                                                                "delete",
                                                                request,
                                                                response);
        } catch (SystemException e) {
            throw new ServletException(e.getMessage(), e);
        }

        // イベントの設定
        DeleteForm deleteForm = (DeleteForm)form;
        event.setStaffCode(deleteForm.getStaffCd());
        try {
            ServiceUtils.dispatchEvent(event);
        } catch (Exception e) {
            throw new ServletException(e.getMessage(), e);
        }

        // 次の画面の準備
        return mapping.findForward("delete");
    }
}
```

```
}
}
```

4.5 データベース

intra-mart のサンプルをインストールしたときにできるテーブル「m_sample_stf」を使用する。

4.6 Struts の設定

Struts 上でサンプルを動かすための設定を以下のファイルで行う。

<Application Runtime のインストールパス>/doc/imart/WEB-INF/struts-config.xml

struts-config.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 1.0//EN"
    "http://jakarta.apache.org/struts/dtds/struts-config_1_0.dtd">

<struts-config>

  <!-- ===== Form Bean Definitions ===== -->
  <form-beans>

    <!-- Configuration for Staff -->
    <form-bean name="selectForm"
      type="staff.struts.action.StaffSelectForm"/>
    <form-bean name="updateForm"
      type="staff.struts.action.UpdateForm"/>
    <form-bean name="deleteForm"
      type="staff.struts.action.DeleteForm"/>
  </form-beans>

  <!-- ===== Global Forward Definitions ===== -->
  <global-forwards>

    <!-- Configuration for Staff -->
    <forward name="on_change"
      path="/struts/staff/stf_body.jsp"/>
    <forward name="update"
      path="/struts/staff/success.jsp"/>
    <forward name="delete"
      path="/struts/staff/success.jsp"/>
  </global-forwards>

  <!-- ===== Action Mapping Definitions ===== -->
  <action-mappings>

    <!-- Configuration for Staff -->
    <action path="/tutorial/index_next_staff"
      forward="/struts/staff/stf_ndp.jsp"/>
    <action path="/tutorial/index_next_stfhead"
      forward="/struts/staff/stf_head.jsp"/>
    <action path="/tutorial/index_next_blank"
      forward="/struts/staff/blank.jsp"/>
    <action path="/tutorial/on_change"
      type="staff.struts.action.StaffSelectAction"
```

```
        name="selectForm"
        scope="request"
        validate="false"/>
<action path="/tutorial/last"
        forward="/struts/staff/stf_head.jsp"/>
<action path="/tutorial/update"
        type="staff.struts.action.UpdateAction"
        name="updateForm"
        scope="request"
        validate="false"/>
<action path="/tutorial/delete"
        type="staff.struts.action.DeleteAction"
        name="deleteForm"
        scope="request"
        validate="false"/>
</action-mappings>
</struts-config>
```

4.7 Web アプリケーションの設定

Struts を動かす、そして Struts と im-J2EE Framework を連携するための設定を以下のファイルで行う。

<Application Runtime のインストールパス>/doc/imart/WEB-INF/web.xml

web.xml に追加する内容

```
...

<!-- Struts Configuration -->
<filter-mapping>
  <filter-name>EncodingFilter</filter-name>
  <servlet-name>action</servlet-name>
</filter-mapping>
<filter-mapping>
  <filter-name>EncodingFilter</filter-name>
  <servlet-name>intramart-struts connector</servlet-name>
</filter-mapping>
<filter-mapping>
  <filter-name>SessionFilter</filter-name>
  <servlet-name>action</servlet-name>
</filter-mapping>

...

<!-- Action Servlet Configuration -->
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>intramart-struts connector</servlet-name>
  <servlet-class>
    jp.co.intra_mart.framework.extension.struts.action.StrutsConnectServlet
  </servlet-class>
</servlet>

...
```



```

<!-- Action Servlet Mapping -->
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>intramart-struts connector</servlet-name>
  <url-pattern>/StrutsConnectServlet</url-pattern>
</servlet-mapping>

. . .

<!-- Struts tag libraries -->
<taglib>
  <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>
<taglib>
  <taglib-uri>/WEB-INF/struts-template.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-template.tld</taglib-location>
</taglib>

. . .

```

4.8 メニュー

intra-mart の[システム設定]-[メニュー]で以下のメニューを登録する。メニューツリーのどこに登録してもかまわないが、サンプルを使うユーザが見えるところに追加する。

項目	内容
フォルダ・ページ名	Struts 連携サンプル
説明	Struts 連携のサンプルです。
ソート番号	(なし)
PC/mobile	" PC 用ページとしてのみ使用"
形式	ページ [Servlet or JSP]
ページ URL	/imart/StrutsConnectServlet
ページ引数	/imart/tutorial/index_next_staff.do
ロール	super

4.9 その他の im-J2EE Framework

intra-martwをインストールしたときのサンプルを利用する。

4.9.1 イベント

4.9.1.1 プロパティ

イベントフレームワークは以下のディレクトリに `EventConfig_tutorial.properties` を設定して利用する。

`<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes`

`EventConfig_tutorial.properties` の設定内容

```
event.class.delete=staff.model.event.DeleteStaffEvent
factory.class.delete=jp.co.intra_mart.framework.base.event.StandardEventListenerFactory
factory.param.delete.listener=staff.model.event.DeleteStaffEventListener

event.class.update=staff.model.event.UpdateStaffEvent
factory.class.update=jp.co.intra_mart.framework.base.event.StandardEventListenerFactory
factory.param.update.listener=staff.model.event.UpdateStaffEventListener

event.class.get_staff=staff.model.event.GetStaffDetailEvent
factory.class.get_staff=jp.co.intra_mart.framework.base.event.StandardEventListenerFactory
factory.param.get_staff.listener=staff.model.event.GetStaffDetailEventListener

event.class.get_staff_list=staff.model.event.GetStaffListEvent
factory.class.get_staff_list=jp.co.intra_mart.framework.base.event.StandardEventListenerFactory
factory.param.get_staff_list.listener=staff.model.event.GetStaffListEventListener
```

4.9.1.2 クラス

`<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes/staff/model/event` 以下のクラスを使う。

4.9.2 データ

4.9.2.1 プロパティ

データフレームワークは以下のディレクトリに `DataConfig_tutorial.properties` を設定して利用する。

`<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes`


`DataConfig_tutorial.properties` の設定内容

```
dao.class.staff=staff.model.data.StaffIntramartDBDAO
dao.connector.staff=intra_mart_db
```

4.9.2.2 クラス

`<Application Runtime のインストールパス>/doc/imart/WEB-INF/classes/staff/model/data` 以下のクラスを使う。

4.10 実行

`intra-mart` を起動し、管理者権限 (ロール=`super`) を持つユーザでログインすると「4.8 メニュー」で登録したメニューが使えるようになる。このメニューをクリックすると「 4-1 サンプル」のような画面が出る。

社員情報検索画面

社員名

社員1

最新の情報

社員情報詳細画面

社員コード	st1001
社員名漢字	社員1
社員名カナ	シヤイン1
社員名英字	staff-1

更新 削除

図 4-1 サンプル

付録 A im-J2EE Framework と Struts

im-J2EE Framework のサービスフレームワークと Struts は完全には一致しないが、おおよそ「表 A-1 Struts と im-J2EE Framework(サービスフレームワーク)」のように分類することができる。

表 A-1 Struts と im-J2EE Framework(サービスフレームワーク)

比較項目	im-J2EE Framework	Struts
設定	ServiceConfig~.properties	struts-config.xml
コントローラ	ServiceServlet	ActionServlet
入力チェック	ServiceController の check メソッド	ActionForm の validate メソッド
Web 層の処理	ServiceController の service メソッド	Action の perform メソッド
キーによる遷移先の決定	Transition の getNextPage	ActionMapping の findForward メソッド
ファイルアップロード	ServiceControllerAdapter の getEntity メソッド	ActionForm の getMultipartRequestHandler メソッド

付録 B Struts のライセンス

本書で紹介されている imart-struts.zip には Apache Software Foundation (<http://www.apache.org/>)によって開発されたソフトウェアが含まれています。Struts を使用する場合、そのライセンスや使用条件は Struts に付随するものに準じます。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

```
/*
 * $Header: /home/cvs/jakarta-struts/LICENSE,v 1.2 2001/02/02 00:38:31 craigmcc Exp $
 * $Revision: 1.2 $
 * $Date: 2001/02/02 00:38:31 $
 *
 * =====
 *
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1999-2001 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution, if
 * any, must include the following acknowledgement:
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgement may appear in the software itself,
 * if and wherever such third-party acknowledgements normally appear.
 *
 * 4. The names "The Jakarta Project", "Struts", and "Apache Software
 * Foundation" must not be used to endorse or promote products derived
 * from this software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *
```

* 5. Products derived from this software may not be called "Apache"
* nor may "Apache" appear in their names without prior written
* permission of the Apache Group.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* =====
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <<http://www.apache.org/>>.
*
* /

intra-mart im-J2EE Framework
Struts 連携ガイド 版

初版 : March 31, 2003

Copyright(C) NTT DATA INTRAMART CO.,LTD.

TEL: 03-5549-2821

FAX: 03-5549-2816

URL: <http://www.intra-mart.co.jp/>