



TOKYO IT SCHOOL

## Struts が提供する Action

### 目次

1. はじめに	1
2. ForwardAction	2
3. DispatchAction	3
4. LookupDispatchAction	9

## 1. はじめに

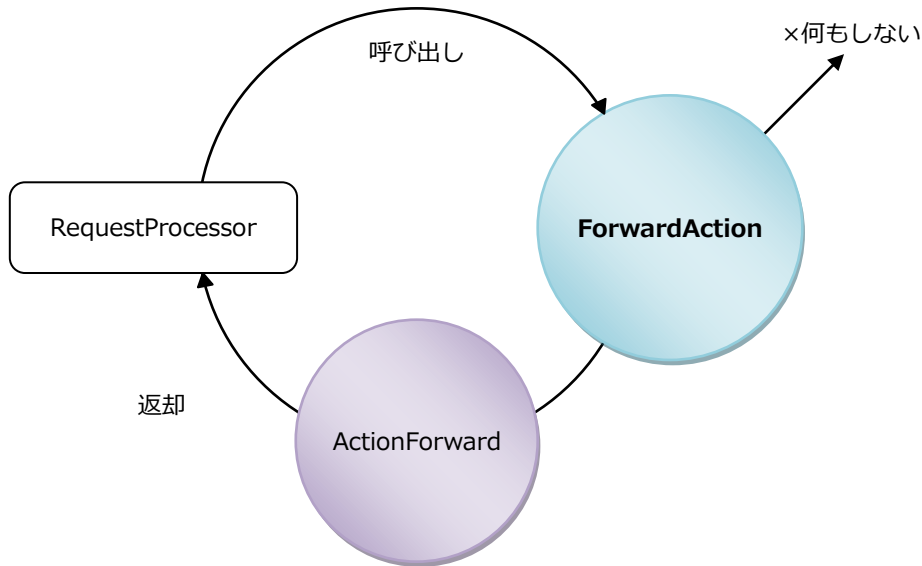
Struts にはあらかじめ便利な Action が用意されています。これらは、上手に利用することにより、開発コストを下げられるなどのメリットがありますので、適宜利用していくことをおすすめします。

## 2. ForwardAction

Web アプリケーションを作成する場合、ページを表示する際に前処理を必要とせず、単に JSP へ転送するだけでよいことも多々あります。そのような際には、JSP を直接呼び出したくなりますが、それは極力避けるべきでしょう。

このような場合に、Struts では転送だけを行う Action として ForwardAction が用意されています。ForwardAction の動作を次の図に示します。

### 【ForwardAction の動作】



ForwardAction は、通常の Action とは違ってコーディングする必要はなく、Struts 設定ファイルに次のような記述をするだけで、使用することができます。

### struts-config.xml (ForwardAction の設定例)

```
<action-mappings>
  <action path="/forwardSample"
    type="org.apache.struts.actions.ForwardAction"
    parameter="/WEB-INF/jsp/sample/forward.jsp"/>
</action-mappings>
```



action 要素の type 属性に ForwardAction を完全限定名で指定し、parameter 属性にフォワード先のパスを Web アプリケーションルートからの相対パスで指定します。

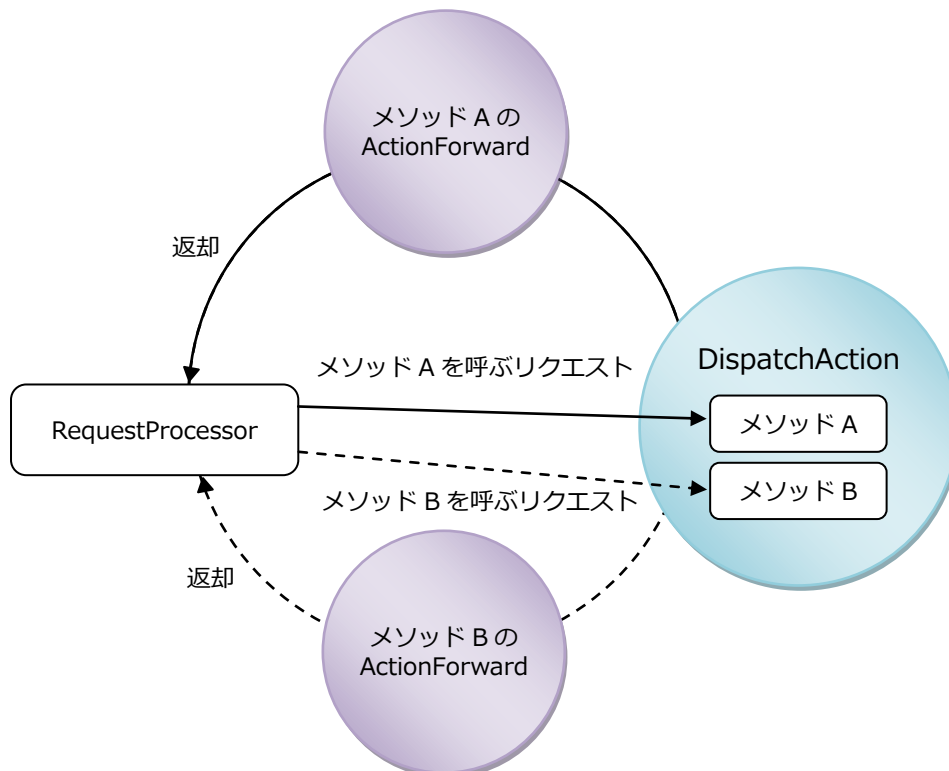
ForwardAction では通常の Action を経由しないで、直接 RequestProcessor でフォワードしています。

### 3. DispatchAction

互いに関連する複数の操作を 1 つの Action にまとめたい場合があります。たとえばユーザーの新規作成、編集、削除などを 1 つのページから行う場合などです。

通常これらは複数の Action に分割するか、リクエストパラメータで呼び出すメソッドを判断するロジックを組み込むか、いずれかの方法で対処する必要があります。DispatchAction は関連する複数のメソッドを 1 つのクラスにまとめる機能を持ち、リクエストパラメータの値に応じて自動的に正しいメソッドを選択してくれます。ここで選択されるメソッドをディスパッチメソッドといいます。

#### 【DispatchAction の動作】

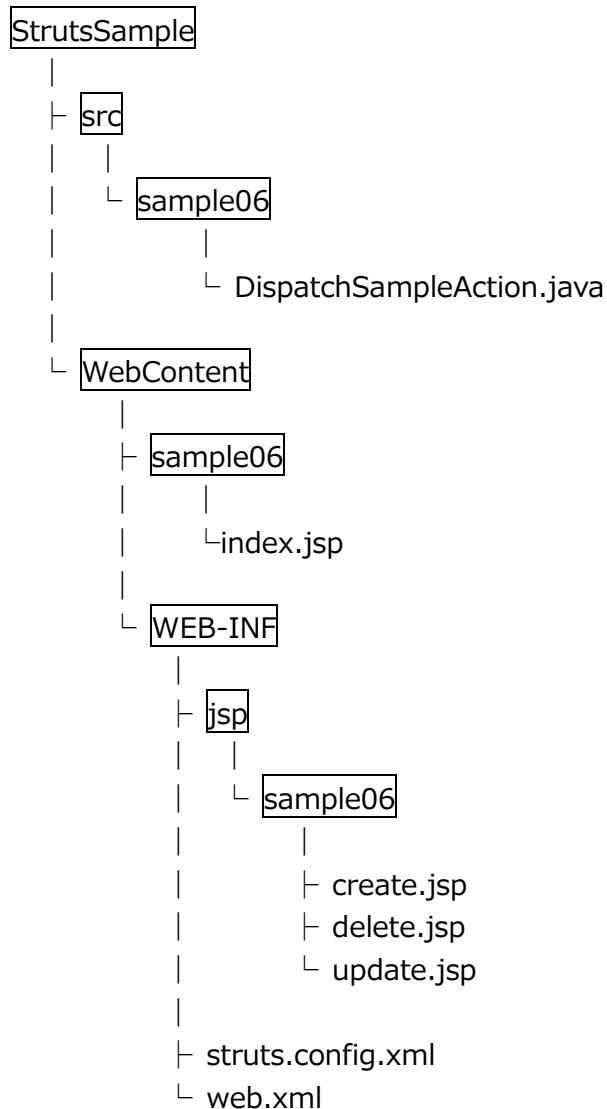


それでは、実際に DispatcherAction を実装してみます。作成、および変更するファイルの一覧を以下に示します。

ファイル名	解説	ロケーション
struts-config.xml	Struts 設定ファイル	/WEB-INF/
DispatchSampleAction.java	Action	/src/sample06/
create.jsp	登録画面用 JSP	/WEB-INF/jsp/sample06/
update.jsp	変更画面用 JSP	/WEB-INF/jsp/sample06/
insert.jsp	削除画面用 JSP	/WEB-INF/jsp/sample06/
index.jsp	入力フォーム	/sample06/

Eclipse 上から見たディレクトリ構成を以下に示します。

## 【ディレクトリ構成】



DispatchAction は、org.apache.struts.actions.DispatchAction を継承したクラスを作成し、execute()メソッドをオーバーライドせずにディスパッチメソッドを実装します。ディスパッチメソッドの引数は Action の execute()メソッドと同じ引数を取ります。

#### 【DispatchSampleAction.java】

```
package sample06;

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 org.apache.struts.actions.DispatchAction;

public class DispatchSampleAction extends DispatchAction {
    public ActionForward create(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {
        //ここに処理を書きます
        return mapping.findForward("create");
    }

    public ActionForward update(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {
        //ここに処理を書きます
        return mapping.findForward("update");
    }

    public ActionForward delete(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {
        //ここに処理を書きます
        return mapping.findForward("delete");
    }
}
```

DispatchAction を使用する場合の Struts 設定ファイルは、以下のように記述します。

#### 【Struts-config.xml】

```
... (略) ...
<form-beans>
    ... (略) ...
    <form-bean name="sample06_dynaForm"
        type="org.apache.struts.action.DynaActionForm">
        <form-property name="id" type="java.lang.String" />
    </form-bean>
</form-beans>
<action-mappings>
    ... (略) ...
    <action path="/sample06/dispatch"
        name="sample06_dynaForm"
        type="sample06.DispatchSampleAction"
        parameter="method">
        <forward name="create"
            path="/WEB-INF/jsp/sample06/create.jsp"/>
        <forward name="update"
            path="/WEB-INF/jsp/sample06/update.jsp"/>
        <forward name="delete"
            path="/WEB-INF/jsp/sample06/delete.jsp"/>
    </action>
</action-mappings>
... (略) ...
```

action 要素の parameter 属性には、振り分けのための任意のパラメータ名を指定します。DispatchAction の実行を要求する際に、リクエストパラメータに"パラメータ名=ディスパッチメソッド名"を付加すると、指定されたディスパッチメソッドが呼び出されます。

遷移元 JSP の実装は次のようになります。

#### 【index.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="/tags/struts-html" prefix="html" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html:html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>メニュー画面</title>
</head>
<body>
```

```
<h2>メニュー画面</h2>
<html:form method="POST" action="/sample06/dispatch.do">
  ID: <html:text property="id" /><br>
  <html:submit value="create" property="method" />
  <html:submit value="update" property="method" />
  <html:submit value="delete" property="method" />
</html:form>
</body>
</html:html>
```

処理を振り分けるボタンを表示するための `html:submit` 要素の `property` 属性に、Struts 設定ファイルで `action` 要素の `parameter` 属性に指定したパラメータ名（この例では"method"）を記述し、`value` 属性にはそれぞれのボタンが呼び出すディスパッチメソッド名を指定します。最後に遷移先の JSP です。こちらは単純な実装となります。

### 【create.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>登録画面</title>
</head>
  <body>
    <h2>登録画面</h2>
    <p>登録画面です。</p>
  </body>
</html>
```

### 【update.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>変更画面</title>
</head>
  <body>
    <h2>変更画面</h2>
    <p>変更画面です。</p>
  </body>
```



```
</html>
```

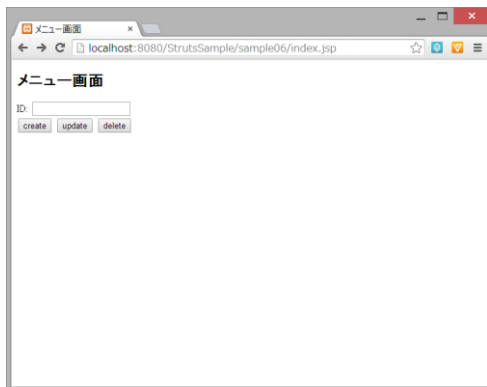
### 【delete.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>削除画面</title>
</head>
  <body>
    <h2>削除画面</h2>
    <p>削除画面です。</p>
  </body>
</html>
```

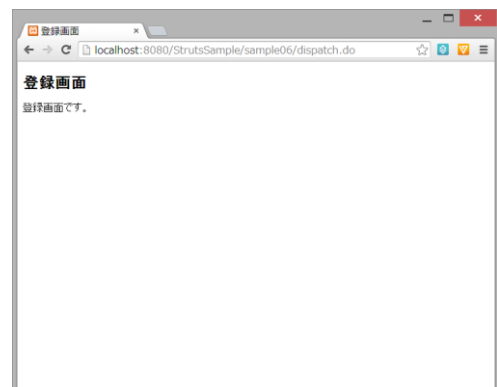
それでは、動作を確認しましょう。Tomcat を起動し、下記 URL をブラウザで表示します。

<http://localhost:8080/StrutsSample/sample06/index.jsp>

#### ■メニュー画面



#### ■登録画面



押下したボタンにより呼び出されるメソッドが変化していることが確認できました。

## 4. LookupDispatchAction

DispatchAction を使ってボタンで処理を振り分ける場合、input 要素の value 属性の値をディスパッチメソッド名と一致させる必要があるため、ボタンのラベルに日本語を使用することができません。そこで用意されているのが LookupDispatchAction です。

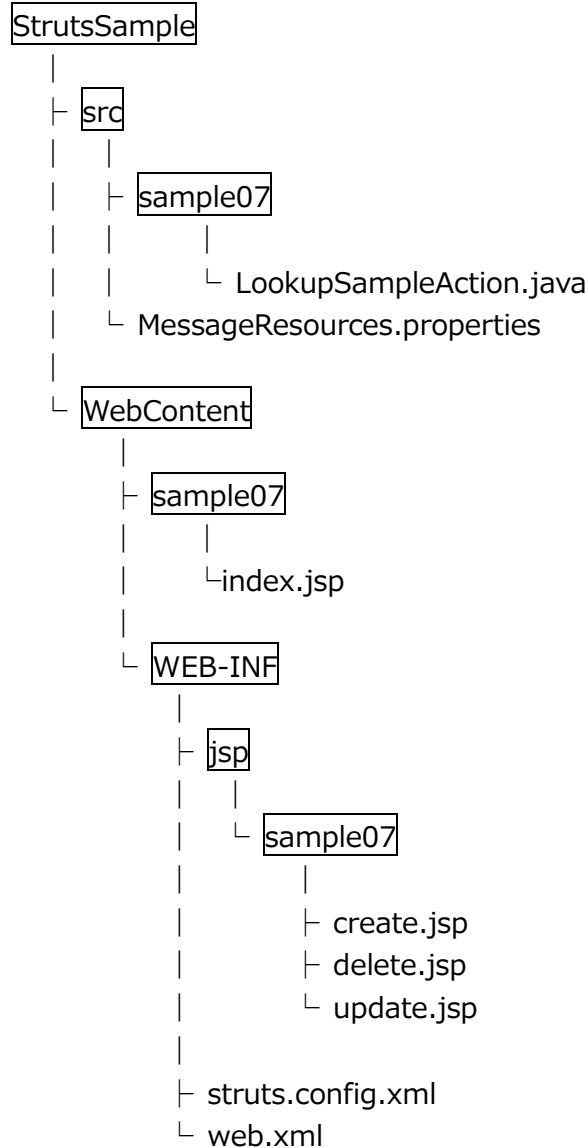
LookupDispatchAction は DispatchAction と同様の機能を実現しますが、DispatchAction がリクエストパラメータの値を用いて呼び出すメソッドを決定するのに対し、LookupDispatchAction はリクエストパラメータの値を用いてメッセージリソースを逆引きし、得られたキーを用いて呼び出すメソッドを決定します。これによりメッセージの国際化対応が可能になっています。また、org.apache.struts.actions.LookupDispatchAction を継承して作成する点が、DispatchAction と異なります。

それでは、実際に LookupDispatchAction を実装してみます。作成、および変更するファイルの一覧を以下に示します。

ファイル名	解説	ロケーション
struts-config.xml	Struts 設定ファイル	/WEB-INF/
LookupSmapleAction.java	Action	/src/sample07/
create.jsp	登録画面用 JSP	/WEB-INF/jsp/sample07/
update.jsp	変更画面用 JSP	/WEB-INF/jsp/sample07/
insert.jsp	削除画面用 JSP	/WEB-INF/jsp/sample07/
index.jsp	入力フォーム	/sample07/
MessageResources.properties	リソースファイル	/src/

Eclipse 上から見たディレクトリ構成を以下に示します。

**【ディレクトリ構成】**



まずは、メッセージリソースファイルに各ボタンに対応する任意のキーと表示名を記述します。

**【Messages.properties】**

```
button.create=登録
button.update=変更
button.delete=削除
```

続いて Action です。LookupDispatchAction はもともと DispatchAction を継承したクラスですので、実装方法もほとんど同じですが、getKeyMethodMap()メソッドを実装する必要があります。

getKeyMethodMap()メソッド内では Map を生成し、メッセージリソースのキーと対応するメソッド名を Map に追加します。Map の put()メソッドの第 1 引数にメッセージリソースのキーを、第 2 引数に対応するメソッド名を指定します。

その他のメソッドは DispatchAction の場合と同様です。

#### 【LookupSampleAction.java】

```
package sample07;

import java.util.HashMap;
import java.util.Map;

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 org.apache.struts.actions.DispatchAction;
import org.apache.struts.actions.LookupDispatchAction;

public class LookupSampleAction extends LookupDispatchAction {

    protected Map getKeyMethodMap() {
        Map map = new HashMap();
        map.put("button.create", "create");
        map.put("button.update", "update");
        map.put("button.delete", "delete");
        return map;
    }

    public ActionForward create(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {
        // ここに処理を書きます
        return mapping.findForward("create");
    }

    public ActionForward update(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {
```

```
// ここに処理を書きます
return mapping.findForward("update");
}

public ActionForward delete(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    // ここに処理を書きます
    return mapping.findForward("delete");
}
}
```

Struts 設定ファイルでは、以下のように記述します。DispatchAction 同様、action 要素の parameter 属性に、リクエストパラメータ名を指定します。ActionForm は Sample06 と同一のものを使用しています。

#### 【struts-config.xml】

```
... (略) ...
<action-mappings>
    ... (略) ...
    <action path="/sample07/lookup"
        name="sample06_dynaForm"
        type="sample07.LookupSampleAction"
        parameter="method">
        <forward name="create"
            path="/WEB-INF/jsp/sample07/create.jsp"/>
        <forward name="update"
            path="/WEB-INF/jsp/sample07/update.jsp"/>
        <forward name="delete"
            path="/WEB-INF/jsp/sample07/delete.jsp"/>
    </action>
</action-mappings>
... (略) ...
```

次に遷移元の JSP を作成します。

#### **[index.jsp]**

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="/tags/struts-html" prefix="html" %>
<%@ taglib uri="/tags/struts-bean" prefix="bean" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html:html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>メニュー画面</title>
</head>
  <body>
    <h2>メニュー画面</h2>
    <html:form method="POST" action="/sample07/lookup.do">
      ID: <html:text property="id" /><br>
      <html:submit property="method">
        <bean:message key="button.create"/>
      </html:submit>
      <html:submit property="method">
        <bean:message key="button.update"/>
      </html:submit>
      <html:submit property="method">
        <bean:message key="button.delete"/>
      </html:submit>
    </html:form>
  </body>
</html:html>
```

html:submit タグの property 属性に、Struts 設定ファイルで action 要素の parameter 属性に指定した値（この例では"method"）を記述します。ここまでは DispatchAction と同じですが、LookupDispatchAction では処理を振り分けるための設定を bean:message タグの key 属性で行います。key 属性には getKeyMethodMap()メソッドで作成した Map 中で、起動したいメソッドと対になっているキー名を指定します。

最後に遷移先の JSP です。こちらはすべて Sample06 と同一の内容となっております。

#### **[create.jsp]**

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>登録画面</title>
</head>
  <body>
    <h2>登録画面</h2>
    <p>登録画面です。</p>
  </body>
</html>
```

### **【update.jsp】**

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>変更画面</title>
</head>
  <body>
    <h2>変更画面</h2>
    <p>変更画面です。</p>
  </body>
</html>
```

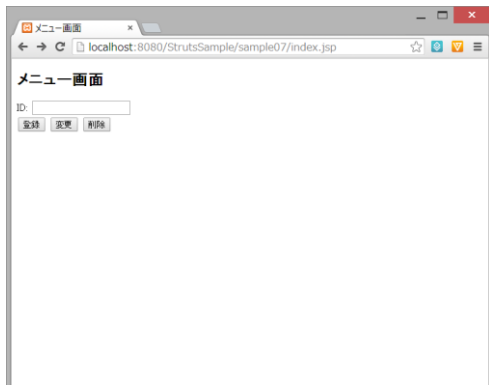
### **【delete.jsp】**

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>削除画面</title>
</head>
  <body>
    <h2>削除画面</h2>
    <p>削除画面です。</p>
  </body>
</html>
```

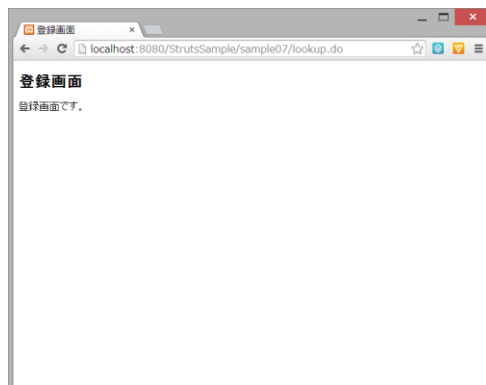
それでは、動作を確認しましょう。Tomcat を起動し、下記 URL をブラウザで表示します。

<http://localhost:8080/StrutsSample/sample07/index.jsp>

#### ■メニュー画面



#### ■登録画面



ディスパッチ機能とボタンの日本語対応の両方が機能していることが確認できました。

Struts が提供する Action についての説明は以上となります。