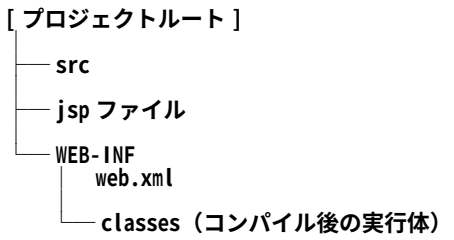
基本形１

Servlet 2.5 までの基本的な形

ディレクトリ構成



サンプルソース

参照

http://www.atmarkit.co.jp/fjava/javafaq/jsp/jsp05.html

Bean

```
package searchtel;
import java.util.Vector;
import java.sql.*;

public class SearchTelBean {
  private String tel = null;
  /** 結果を取得するメソッド */
  public String getTel() {
    return tel;
  }
  public void setTel(String tel) {
    this.tel = tel;
  }
  /** 氏名から電話番号を検索するメソッド */
  public void searchTels(String name) {
    String result = "XX-XXXX-XXXX";
    // 氏名から電話番号を検索し文字列 result をセット
    // するロジックを実装する
    setTel(result);
  }
}
```

Servlet

```
package searchtel;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class SearchTelServlet extends HttpServlet {
 /**HTTP Get リクエストの処理 */
  public void doGet(HttpServletRequest request,
   HttpServletResponse response) throws ServletException,
   IOException {
    String name = request.getParameter("name");
    SearchTelBean searchTelBean = new SearchTelBean();
    // 電話番号を検索
    searchTelBean.searchTels(name);
    //Bean を JSP に渡すために HttpServletRequest オブジェクトにセット
    request.setAttribute("searchTelBean", searchTelBean);
    //View である JSP を呼び出す
    RequestDispatcher rDispatcher =
     request.getRequestDispatcher("/result.jsp");
    rDispatcher.forward(request, response);
  }
```

```
}
```

JSP

```
<HTML>
<HEAD>
<jsp:useBean id="searchTelBean" scope="request"
class="searchtel.SearchTelBean" />
<TITLE> 検索結果 </TITLE>

<BODY>
検索結果：<jsp:getProperty name="searchTelBean" property="tel" />
</BODY>
</HTML>
```

Web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
<!-- (1)DTD 宣言の設定 -->

<web-app>
  <!-- (2)servlet タグの設定 -->
  <servlet>
    <servlet-name>test</servlet-name>
    <servlet-class>SearchTelServlet</servlet-class>
  </servlet>
  <!-- (3)servlet-mapping タグの設定 -->
  <servlet-mapping>
    <servlet-name>test</servlet-name>
    <url-pattern>/search</url-pattern>
  </servlet-mapping>
</web-app>
```

■ jsp ファイルに直接アクセスさせない

```
<security-constraint>
  <display-name>Jsp Security</display-name>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
    <url-pattern>*.jsp</url-pattern>
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
  </web-resource-collection>
  <auth-constraint>
  </auth-constraint>
</security-constraint>
```

を web.xml に追記する

基本形２

Servlet 3.0 以降
アノテーションを使うことで、web.xml を書かなくても良くなった

ディレクトリ構成

```
[ プロジェクトルート ]
├── src
├── jsp ファイル
```

```
└── WEB-INF
    ├── web.xml（なくてもよい）
    └── classes（コンパイル後の実行体）
```

servlet

コンパイル

```
javac -cp /usr/share/java/servlet.jar src/TestServlet.java -d WEB-INF/classes/
```

web.xml

なくても動くけど、一応作る

```
<?xml version="1.0" encoding="UTF-8" ?>
<web-app xmlns:xsi="http://www.w3c.org/2001/XMLSchema-instance">
<display-name>Test Application</display-name>
</web-app>
```

基本形３

jsp だけで作ってしまう。

http://d.hatena.ne.jp/nowokay/20131108

ディレクトリ構成

```
[ プロジェクトルート ]
├── src
├── jsp ファイル
└── WEB-INF
    ├── web.xml（なくてもよい）
    └── classes（コンパイル後の実行体）
```

ロジック

ビュー