Java web 开发

之 JSP + JavaBean

MyEclipse Web 工程

- 工程简介： web.xml , src/ , WebRoot/, lib/

JSP --- 基本语法

- JSP 基本语法包括：两种注释类型，3 个脚本元素，3 个指令元素，8 个动作命令。
- JSP 代码中包含 JSP 元素和 Template Data 两类。Template Data 指的是 JSP 引擎不处理的部分。即指在标记 <%.....%> 以外的部分，如代码中的 HTML 的内容等，这些数据会直接传送到客户端的浏览器。而 JSP 元素则是指将由 JSP 引擎直接处理的部分，这部分代码必须符合 JSP 语法，否则会导致编译错误。
- 示例： jspSample.jsp

基本语法 --- 两种注释类型

- HTML 注释 – 发送到浏览器，但并不是显示

  <!-- ... -->

- 如果在注释标记中包含了表达式等需要 JSP 引擎处理的部分，则返回的是处理后的内容。
- JSP 注释 – 写在 JSP 程序代码里，但不发送到浏览器

  单行注释： <%-- 这是一个单行注释 --%>

  多行注释： <%/* 这是一个多行注释 */%>

基本语法 ---3 个脚本元素

- JSP 脚本元素是最基本的语法格式，包括：
- 1. 声明 (Declaration) ：用于在 JSP 程序中声明合法的变量，其中变量类型包括 Java 基本类型以及类对象声明。

  <%! declaration;[declaration;]...%>
- 例如 :

  <%! int I = 0; %>

  <%! int a, b, c; %>

  <%! Date date; %>

基本语法 ---3 个脚本元素

- 2. 表达式(Expression)：用来定义一个符合 JSP 语法的表达式，在运行后被自动转化为字符串，然后插入到这个表达式在 JSP 文件的位置显示。
- 语法格式： <%=expression %>
- 在 JSP 代码中经常使用表达式来输出变量的值，可用在任何地方。
- 例如：

```
<%! int a, b, c; %>
<% a=12; b=a; c=a+b; %>
<p>a=<%=a %></p>
<p>b=<%=b %></p>
<p>c=<%=c %></p>
```

基本语法 ---3 个脚本元素

- 3. 脚本段 (Scriptlet) ：用来包含一个有效的 Java 程序段。一个 Scriptlet 能够包含多个 JSP 语句，方法，变量，表达式。
- 语法格式： <% 代码 %>
- 例如：见 Scriptlet.jsp

基本语法 --- 3 个指令元素

- JSP 引擎为了方便 JSP 开发而预定义了一些指令，主要包括：
- **1. page 指令**：用于定义 JSP 文件中的全局属性。
- 示例见 jspSample.jsp 。
- <%@ page %>指令作用于整个 JSP 页面，同样包括静态的包含文件，但是不能作用于动态的包含文件，比如 <jsp:include> 。
- 可以在一个页面中用上多个 <%@ page %>指令，但是其中的属性只能用一次，不过也有例外，那就是 import 属性。
- 无论把 <%@ page %>指令放在 JSP 文件的哪个地方，它的作用范围都是整个 JSP 页面。不过，为了程序的可读性以及养成良好的编程习惯，最好还是放在页面的顶部。

基本语法 --- 3 个指令元素

- **2. include 指令**：使用 include 指令可以在 JSP 中包含一个静态文件，同时解析这个文件中的 JSP 语句。
- 语法格式：<%@ include file="relativeURL" %>
- file ：这个包含文件的路径名一般来说是指相对路径，不需要什么端口号，协议或者域名。
- 如果仅仅只是用 include 来包含一个静态文件。那么这个包含的文件所执行的结果将会插入到 JSP 文件中"<%@ include %>"所处的位置。一旦包含文件被执行，那么主 JSP 文件的过程就会被恢复，继续执行下一行。
- 这个被包含的文件可以是 html 文件， jsp 文件，文本文件，或者只是一段 Java 代码，但是要注意在这个包含文件中不能使用 <html> ，</html> ，<body> ，</body> 标记，因为这将影响在原 JSP 文件中同样的标记，有时会导致错误。

基本语法 --- 3 个指令元素

- 3. taglib 指令：定义一个标签库及其自定义标签库的前缀。略过 ...
- 语法格式：

  <%@ taglib uri="URIToTagLibrary" prefix="tagPrefix"%>

基本语法 --- 8 个动作指令

- JSP 动作指令是预定义的 JSP 标签，实现了最常用的基本功能。
- 1. 页面跳转： <jsp:forward>

  <jsp:forward> 标签从一个 JSP 文件向另一个文件传递一个包含用户请求的 request 对象。
- 例如 :

```
<jsp:forward page=”test.jsp”>
    <jsp:param name=”username” value=”<%=user%>” />
    <jsp:param name=”password” value=”<%=pwd%>” />
</jsp:forward>
```

基本语法 --- 8 个动作指令

- page 属性是一个表达式或是一个字符串用于说明将要定向的文件或者 URL 。这个文件可以是 JSP ，程序段，或者其他能够处理 request 对象的文件。
- <jsp:param /> 向一个动态文件发送一个或多个参数，这个文件必须是动态文件，能够处理参数。

基本语法 --- 8 个动作指令

- page 属性是一个表达式或是一个字符串用于说明将要定向的文件或者 URL 。这个文件可以是 JSP ，程序段，或者其他能够处理 request 对象的文件。
- <jsp:param /> 向一个动态文件发送一个或多个参数，这个文件必须是动态文件，能够处理参数。

基本语法 --- 8 个动作指令

- 2. 包含页面：<jsp:include>
- 包含一个静态或者动态文件。
- 示例：

```
<jsp:include page="scripts/logout.jsp" flush="true" />
<jsp:include page="copyright.html" flush="true" />
<jsp:include page="/index.html" flush="true" />
<jsp:include page="scripts/login.jsp" flush="true">
    <jsp:param name="username" value="xiayf" />
</jsp:include>
```

基本语法 --- 8 个动作指令

- flush ：这里必须使用 flush=”true”，不能使用 false 值。而默认为 false 。
- <jsp:param> 用来传递一个或多个参数到指定的动态文件。
- <jsp:include> 元素允许包含动态和静态文件，这两种包含文件的结果是不同的。如果文件仅是静态文件，那么这种包含仅仅是把包含文件的内容加到 JSP 文件中去，类似于 <%@ include %>; 如果这个文件是动态的，那么这个被包含文件也会被 JSP 引擎执行。

基本语法 --- 8 个动作指令

- 3. 创建 Bean ： <jsp:useBean>
- 创建一个 Bean 实例并指定它的名字和作用范围。
- 例如：

```
<jsp:useBean id="cart" scope="session" class="session.Carts" />
<jsp:setProperty name="cart" property="*" />
<jsp:useBean id="checking" scope="session" class="bank.Checking">
    <jsp:setProperty name="checking" property="balance" value="0.0" />
</jsp:useBean>
```

基本语法 --- 8 个动作指令

- <jsp:useBean> 用于定位或使用一个 JavaBeans 组件。 <jsp:useBean> 首先会试图定位一个 Bean 实例，如果这个 Bean 不存在，那么 <jsp:useBean> 就会从一个 class 或模板中进行实例化。
- 5 个属性：见书本

基本语法 --- 8 个动作指令

- 4. 设置 Bean 属性： <jsp:setProperty>
- 例如 :

  <jsp:setProperty name=”mybean” property=”*” />

  <jsp:setProperty name=”mybean” property=”username” />

  <jsp:setProperty name=”mybean” property=”username” value=”Steve” />

基本语法 --- 8 个动作指令

- 5. 取得 Bean 属性： <jsp:getProperty>
- 使用 <jsp:getProperty> 可获取 Bean 的属性值，用于在页面中显示。
- 例如：

```
<jsp:useBean id=”calendar” scope=”page”
class=”empolyee.Calendar” />

    <jsp:getProperty name=”calendar” property=”username” />
```

基本语法 --- 8 个动作指令

- 6. 使用 Applet 插件： <jsp:plugin>
- 7. 插件定义参数： <jsp:param>
- 8. 插件错误提示： <jsp:fallback>

JSP 内置对象 --- Request

- Request ：请求对象，该对象封装了用户提交的信息，通过调用该对象相应的方法可以获取封装的信息，即使用该对象可以获取用户提交的信息。
- 示例： requestSample.jsp

JSP 内置对象 --- Response

- 对用户请求动态的响应，向客户端发送数据。
- 1）动态响应 contentType 属性
- 2） Response 重定向

JSP 内置对象 --- Session

- Session 对象作为内置对象，会在第一个 JSP 页面被装载时自动创建，完成会话期管理。
- 从用户打开浏览器并连接到服务器开始，到客户关闭浏览器，称为一个会话。当一个客户访问一个服务器时，可能会在这个服务器的几个页面直接切换，服务器应当通过某种办法知道这是同一个用户，就需要 session 对象。

JSP 内置对象 --- Application

- 服务器启动后就产生这个 Application 对象。
- 与 session 对象不同的是，所有用户的 Application 对象都是同一个。
- Application 常用方法：

  setAttribute(String key, Object obj)

  getAttribute(String key)

JSP 内置对象 --- Out

- Out 对象是一个输出流，用来向客户端浏览器输出数据。

JSP 内置对象 --- Cookie

JSP 内置对象 --- Config

JSP 内置对象 --- Page

JSP 内置对象 --- PageContext

JSP 内置对象 --- Exception

数据库访问 – JDBC