CSE 510
Web Data Engineering

Tag Libraries

Tag Libraries

- Collections of custom JSP tags
  - Hide Java from JSPs
  - Java classes in special format
  - Methods invoked with XML tags
  - Often looking like scripting languages
- Load one of the many on the web,
  or build your own
  - Unlikely given the plenty of choices available
- **Struts Tag Libraries**
- **JSP Standard Tag Library (JSTL)**

Struts Tag Libraries

- **Four Struts Tag Libraries**
  - **`html`** Generate HTML dynamically
  - **`bean`** Define beans, print bean properties, print localized strings
  - **`logic`** Manage conditionals, iterations, flow control
  - **`nested`**

`html` Tag Library

- Used to create input forms for your application
- A few other useful tags used in the creation and rendering of HTML-based user interfaces
  - **`html:form`**
  - **`html:errors`**
  - **`html:password`**
  - **`html:submit`**
  - **`html:text`**
  - **`html:option`**

html Tag Library: Example

```
<html:form action="/login" method="POST">
  <h1>Login</h1>
  <html:errors/>
  <table>
     <tr>  <td>User Name</td>
           <td><html:text property="userName"/></td></tr>
     <tr>  <td>Password</td>
           <td><html:password property="password"/></td></tr>
     <tr>  <td>&nbsp;</td>
           <td><html:submit value="Log in"/></td></tr>
  </table>
</html:form>
```

`bean` Tag Library

- Used for creating and accessing Java Beans and a few other general purpose uses
- **`bean:define`**

  Define a scripting variable based on the value(s) of the specified bean property
- **`bean:write`**

  Render the value of the specified bean property
- **`bean:message`**

  Render an internationalized message string to the response

bean Tag Library: Example

```
<html:html>
  <head>
    <title>Bean Define, Bean Write Tags</title>
  </head>
  <body>
    <bean:define id="message"
                 type="java.lang.String"
                 value="First message string"/>
    <p><b><bean:write name="message"/></b></p>
  </body>
</html:html>
```

`bean` Tag Library: Another Example

```
<html:html>
 <head><title>
  <bean:message key="welcome.taglib.title"/>
 </title></head>
 <body>
  <h3><bean:message key="welcome.taglib.heading"/></h3>
  <p><bean:message key="welcome.taglib.message"/></p>
 </body>
</html:html>
```

Message defined in:

`\WEB-INF\classes\MessageResources.properties`

`logic` Tag Library

- Helpful with iterating through collections, conditional generation of output, and application flow

`logic` Tag Library: Example

```
<logic:present name="itemsList”>
  Items available for the selected color
  <b><bean:write name="selectedColor"/></b>:<br/>
  <logic:iterate id="item" name="itemsList">
    <b><bean:write name="item"/></b><br/>
  </logic:iterate>
</logic:present>
<logic:notPresent name="itemsList”>
  No Items available for selected color
  <bean:writename="selectedColor"/>
</logic:notPresent>
```

JSTL

- **Core** Lib (prefix **`c`**): scripting language
- **Database** Lib (**`sql`**): support for DB
- **Functions** Lib (**`fn`**): string manipulation etc.
- **XML** Lib (**`x`**): XML support
- **Internationalization** Lib (**`fmt`**): formatting

- Installation:
  - JSTL requires **`jstl.jar`** and **`standard.jar`** located in **`apache-tomcat-6.0.20/webapps/examples/WEB-INF/lib/`**
  - Copy these two files into **`apache-tomcat-6.0.20/lib/`**

JSTL Expressions

- Many JSTL tags have attributes whose values are JSTL expressions
  - Enclosed in `${...}`
  - `<c:out value="${request.v}"/>`

    is shorthand for

    ```
    <% String attr=request.getParameter("v");%>
    ...
    <%= v %>
    ```
  - Declare on top of your JSP

    ```
    <%@ taglib prefix="c"
        uri="http://java.sun.com/jsp/jstl/core" %>
    ```

Access to Bean Properties

- Assume your JSP uses a bean named **`myBean`** that has property **`prop`** and corresponding **`getProp`** and **`setProp`**

```
<c:out value="${myBean.prop}"/>
```

stands for

```
<%= myBean.getProp() %>
```

Scripting Language Features

- Variable definition

```
<c:set var="v"
       value="${...}"
       scope="session"/>
<c:remove var="v"/>
```

- Weak typing
  - Strings to numbers
  - Integers to reals
  - and many more

Flow Control Tags

-
  ```
  <c:if test="${booleanExpression}”>
      body of tag evaluated if boolean is true
  </c:if>
  ```

-
  ```
  <c:choose>
      <c:when test="${booleanExpression}">...</c:when>
      <c:when test="${booleanExpression}">...</c:when>
      <c:otherwise>...</c:otherwise>
  </c:choose>
  ```

Iteration

- Iterate over start to end values, arrays, Collection, Iterator, Enumeration, Map, List, comma-separated string, etc.

```
<c:forEach var="i" begin="0" end="10" step="1">
    loop body
</c:forEach>
```

Collection

```
<c:forEach var="ck" items="${request.cookies}">
    loop body
</c:forEach>
```