

JSPs

CS174

Chris Pollett

Nov 20, 2006.

Outline

- Java Server Pages

Java Server Pages (Review from before midterm)

- Java Server Pages are an approach to web programming which is built on top of Servlets
- One problem with servlets is that web code is embedded in print statements.
- This requires a person who creates a servlet to both be an expert web designer as well as a Java programmer.
- JSP allow one to embed short Java code snippets within an XHTML document.
- These code snippets typically make small requests for things like the value of a variable, so do not demand to much of a web programmer.
- Behind the scenes the maintenance of variables might live a be done in a Java bean which consists of pure Java code and is not responsible for any I/O.
- A coder writes the bean/ other objects and doesn't have to worry about understanding web design.

JSP Documents

- When requested by a browser, a JSP document is processed by a program called a JSP container.
- Some JSP containers compile the document only when the document is loaded on the server; other compile them only when they are requested.
- This compilation process translates the JSP document into a servlet object.
- A JSP document consists of four different kinds of elements: directives, traditional XHTML or XML, action elements, and scriptlets.
- The XHTML code is used to produce the content that is fixed about the page. It is called template text.
- Action elements dynamically create content.
- The result of the JSP is an XHTML document consisting of the template part and the filled in output of the action elements.

More on JSPs

- The general form of an action element is:

`<jsp: element>`

- As an example `<jsp:include page="my.jsp" >` includes a file in the current file at the given location.
- This is a so called standard action element, there are also custom action elements created by users and JSTL action elements.
- action elements are like programming language statements
- directives are like messages to the JSP container.
- For example: `<%@ page contentType = "text/html" %>`
- Directives have the syntax `<%@ .. %>`
- Another possible directive is:
`<%@ taglib prefix="c" uri="http://www.sun.com/jsp/jstl/core" %>`

Scriptlets (New Stuff)

- Scriptlets are Java code that can be embedded in a JSP document.
- There are four types of scriptlets: comments, scriptlet code, expressions, and declarations.
- Declarations are rarely used and won't be discussed.
- A scriptlet expression takes the form:
`<%= expression %>`
And inserts the value of the expression at that location in the document.
For example, `<%= bob %>` inserts the value of variable bob (after converting it to a string) at this location in the document.
For scoping reasons you could imagine variables as living in say the `_jspervice` method.
- Scriptlet code is delimited by `<% .. %>`.
- Scriptlet comments are just Java comments within a scriptlet block. They are different from JSP comments `<%-- --%>`

Example JSP with Scriptlet

```
<html>
<head><title>test</title></head>
<body>
  <% String bob=request.getParameter("bob");%>
  <!-- notice variable request and response are
        available since it is as if we are running in_jsp
        service() -->
  <p>Bob=<%=bob %>
</p>
</body>
</html>
```

Implicit variables

- As we saw in the last example that a JSP page can make use of implicit variables such as response.
- Other variables which are available are: out --a JSPWriter to the client, page, pageContext, request, response, session, config, application.

New Tag Libraries

- JSP support the ability to create new tag libraries.
- We are now going to look at one particular tag library briefly:
- The Java Standard Tag library.
- You can download this from:
<http://jakarta.apache.org/taglibs/>
- You want to move the two jar files standard.jar and jstl.jar in the standard/lib subfolder of what you download into your tomcat's common/lib directory. At which point after restarting Tomcat things should work.

JSTL

- The Java Standard Template library makes use of two technologies: JSTL tags and the JSP expression language.
- The JSP expression language (EL) is similar to Javascript. These serve as the values of attributes for tags from the tag library.
- An EL expression is introduced with a \$ sign and delimited with braces: `${expression}`
- An EL expression consists of literal, the usual arithmetic operators, implicit variables, and normal variables.
- One source of EL data is the implicit variable `param` which has information about form variables sent.
- If there was a form variable `address` one could have the expression: `${param.address}` to use it. You could also write `${param['address']}`.
- To explicitly say that you want a EL expression to be output you can use:

```
<c:out value = "${param.address}" />
```

Expression Language Example

```
<%@page contentType="text/html" %>
```

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

```
<html xmlns="http://www.w3.org/1999/xhtml"  
xml:lang="en" lang="en">  
<head><title>test</title></head>  
<body>
```

```
<p>bob=<c:out value = "${param.bob}" />  
</p>  
</body>  
</html>
```

JSTL control action elements

```
<c:if test = "boolean">
```

```
</c:if>
```

For example,

```
<c:if test = "${pageContext.request.method=='Post'}">
```

```
</c:if>
```

```
<c:forEach begin = "1" end = "10" var = "i">
```

```
</c:forEach>
```

```
<c:forEach items = "${paramValues.topping}" var = "top">
```

```
<c:out value = "${top}"><br /></c:forEach>
```