

CS320 Web and Internet Programming

JSP Scripting Elements and Page Directives

Chengyu Sun
California State University, Los Angeles

Java Server Page (JSP)

- ◆ Why?
 - It's tedious to generate HTML using `println()`
 - Separate presentation from processing
- ◆ How?
 - *Java code* embedded in *HTML documents*

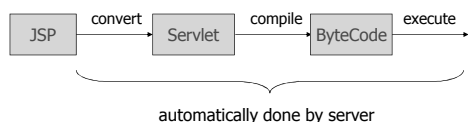
HelloJSP1.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">  
<HTML>  
<HEAD><TITLE>Hello JSP 1</TITLE></HEAD>  
<BODY>A JSP without J or S.</BODY>  
</HTML>
```

HelloJSP2.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">  
<HTML>  
<HEAD><TITLE>JSP Hello World</TITLE></HEAD>  
<BODY>Hello World on <%= new java.util.Date() %>.  
</BODY>  
</HTML>
```

How Does JSP Work?



- ◆ Look under
`$TOMCAT_HOME/work/Catalina/localhost/context_name`

Some Simple Observations about the JSP/Servlets

- ◆ In package `org.apache.jsp`
- ◆ `_jspService()` handles everything
 - replaces `service()` in `HttpServlet`
 - What happened to `doGet()`, `doPost()`, ...??
- ◆ HTML text → `out.write(...)`

JSP Components

- ◆ HTML template text
- ◆ Code elements of Java
 - Directives
 - Comments
 - Scripting elements
 - Beans
 - Expression language
 - Custom tag libraries

Directives

- ◆ `<%@ type attr="value" ... %>`
- ◆ Three types of directives
 - page
 - include
 - taglib

Some Page Directives

```
<%@ page import="java.util.*, java.util.zip.*" %>
<%@ page contentType="text/html" %>
<%@ page pageEncoding="Shift_JIS" %>
<%@ page session="false" %> <%-- default is true -->
```

Comments

- ◆ `<%-- Hidden Comments --%>`
- ◆ `<!-- Output (HTML) Comments -->`

JSP Scripting Elements

- ◆ JSP Expressions
- ◆ JSP Scriptlets
- ◆ JSP Declarations

JSP Expressions

- ◆ `<%= Java expression %>`
 - *What's an expression??*

`<%= expression %>` in JSP



`out.write(expression)` in `_jspService()`

Example: Add.jsp

- ◆ Add two integer parameters and display the sum

Pre-defined Variables

- ◆ request, response, out
- ◆ session, application
- ◆ config, pageContext
- ◆ page

The same request, response, session etc. that are used in servlets – see the Servlet API documentation for what you can do with them.

JSP Scriptlets

- ◆ `<% Java code %>`
- ◆ All code goes *inside* `_jspService()`

Example: Add.jsp Again

- ◆ Add the code to validate the parameters

Another Scriptlet Example

```
<% if( Math.random() < 0.5) { %>
<H1>Have a <I>nice</I> day!</H1>
<% } else { %>
<H1>Have a <I>lousy</I> day!</H1>
<% } %>
```

JSP Declarations

- ◆ `<%! class variables or methods %>`
- ◆ All code goes *outside* `_jspService()`

Example: RequestCounter.jsp

- ◆ Initialize a counter as a class variable
- ◆ Increment the counter each time the JSP is requested

About JSP Scripting Elements

- ◆ There is a straightforward mapping between JSP with scripting elements and servlet
- ◆ Mixing presentation and processing
 - hard to debug
 - hard to maintain
- ◆ No clean and easy way to reuse code
- ◆ Do NOT use them