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

```html
<!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

```html
<!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?

1. JSP
2. Convert to Servlet
3. Compile to Bytecode
4. Execute

- Automatically done by server

Eclipse Work Directory

- `$WORKSPACE/.metadata/.plugins/org.eclipse.wst.server.core/tmp0`
- Or `tmp1, tmp2, ...` if you created multiple server runtimes in Eclipse

Look under
- `$TOMCAT_HOME/work/Catalina/localhost/context_name`
Some Simple Observations about the JSP/Servers

- `HttpJspBase` is a subclass of `HttpServlet`
- `_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 Expression
- JSP Scriptlet
- JSP Declaration
Elements of a Programming Language

- Comments
- Literals
- Variables and Types
- Operators
- Expressions: anything that evaluates to a single value
- Statements
- Functions
- Classes
- Packages

JSP Expressions

- `<%= Java 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

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

Example: RequestCounter.jsp

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

About JSP Scripting Elements

- JSP Scripting Elements
  - JSP Expression
  - JSP Scriptlet
  - JSP Declaration
  - There is a straightforward mapping between JSP with scripting elements and servlet

... About JSP Scripting Elements

- Mixing presentation and processing
  - Hard to read, debug, or maintain
- No clean and easy way to reuse code
- Do NOT use them

JSP Declarations

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