Introducing JavaServer Pages

Core Servlets & JSP book: www.coreservlets.com
More Servlets & JSP book: www.moreservlets.com
Servlet and JSP Training Courses: courses.coreservlets.com

Agenda

• Why we need JSP
• How JSP works
• Benefits of JSP
• Setting up your environment for JSP
• A simple example
The Need for JSP

- **With servlets, it is easy to**
  - Read form data
  - Read HTTP request headers
  - Set HTTP status codes and response headers
  - Use cookies and session tracking
  - Share data among servlets
  - Remember data between requests
  - Get fun, high-paying jobs

- **But, it sure is a pain to**
  - Use those println statements to generate HTML
  - Maintain that HTML

The JSP Framework

- **Idea:**
  - Use regular HTML for most of page
  - Mark servlet code with special tags
  - Entire JSP page gets translated into a servlet (once), and servlet is what actually gets invoked (for each request)

- **Example:**
  - JSP
    - Thanks for ordering
      `<I><%= request.getParameter("title") %></I>`
  - URL
    - http://host/OrderConfirmation.jsp?
      `?title=Core+Web+Programming`
  - Result
    - Thanks for ordering Core Web Programming
Benefits of JSP

- Although JSP technically can't do anything servlets can't do, JSP makes it easier to:
  - Write HTML
  - Read and maintain the HTML
- JSP makes it possible to:
  - Use standard HTML tools such as Allaire HomeSite, Macromedia DreamWeaver, or Adobe GoLive.
  - Have different members of your team do the HTML layout than do the Java programming
- JSP encourages you to
  - Separate the (Java) code that creates the content from the (HTML) code that presents it

Advantages of JSP Over Competing Technologies

- Versus ASP or ColdFusion
  - Better language for dynamic part
  - Portable to multiple servers and operating systems
- Versus PHP
  - Better language for dynamic part
  - Better tool support
- Versus pure servlets
  - More convenient to create HTML
  - Can use standard tools (e.g., HomeSite)
  - Divide and conquer
  - JSP programmers still need to know servlet programming
Advantages of JSP (Continued)

- **Versus Velocity or WebMacro**
  - Standard

- **Versus client-side JavaScript (in browser)**
  - Capabilities mostly do not overlap with JSP, but
    - You control server, not client
    - Richer language

- **Versus server-side JavaScript**
  - (eg, LiveWire, BroadVision, JRun)
  - Richer language

- **Versus static HTML**
  - Dynamic features
  - Adding dynamic features no longer "all or nothing" decision

### Setting Up Your Environment

- **Set your CLASSPATH.** Not.
- **Compile your code.** Not.
- **Use packages to avoid name conflicts.** Not.
- **Put JSP page in special directory.** Not.
  - `install_dir\webapps\ROOT\` (HTML and JSP -- Tomcat)
  - `install_dir\servers\default\default-app` (JRun)
  - Some servers reserve certain parts of Web hierarchy for JSP pages. Tomcat 3 and JRun (standalone) don't.
- **Use special URL to invoke JSP page.** Not.
- **Caveats**
  - Previous rules about CLASSPATH, install dirs, etc., still apply to regular Java classes used by a JSP page
Example

<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>JSP Expressions</TITLE>
<META NAME="author" CONTENT="Marty Hall">
<META NAME="keywords"
CONTENT="JSP,expressions,JavaServer,Pages,servlets">
<META NAME="description"
CONTENT="A quick example of JSP expressions.">
<LINK REL=STYLESHEET
HREF="JSP-Styles.css"
TYPE="text/css">
</HEAD>

<BODY>
<H2>JSP Expressions</H2>
<UL>
   <LI>Current time: <%= new java.util.Date() %>
   <LI>Your hostname: <%= request.getRemoteHost() %>
   <LI>Your session ID: <%= session.getId() %>
   <LI>The <CODE>testParam</CODE> form parameter:
       <%= request.getParameter("testParam") %>
</UL>
</BODY>
</HTML>

Example (Continued)

<BODY>
<H2>JSP Expressions</H2>
<UL>
   <LI>Current time: <%= new java.util.Date() %>
   <LI>Your hostname: <%= request.getRemoteHost() %>
   <LI>Your session ID: <%= session.getId() %>
   <LI>The <CODE>testParam</CODE> form parameter:
       <%= request.getParameter("testParam") %>
</UL>
</BODY>
</HTML>
Example Result

- **If location was**
  - C:\jakarta-tomcat-4.0\webapps\ROOT\Expressions.jsp or
  - C:\Program Files\Allaire\JRun\servers\default\default-app\Expressions.jsp
- **URL would be**
  - http://localhost/Expressions.jsp

Most Common Misunderstanding
Forgetting JSP is Server-Side Technology

- **Very common question**
  - I can’t do such and such with HTML. Will JSP let me do it?
- **Why doesn’t this question make sense?**
  - JSP runs entirely on server
  - It doesn’t change content the client (browser) can handle
- **Similar questions**
  - How do I put a normal applet in a JSP page?
    Answer: send an `<APPLET>` tag to the client
  - How do I put an image in a JSP page?
    Answer: send an `<IMG>` tag to the client
  - How do I use JavaScript/Acrobat/Shockwave/Etc?
    Answer: send the appropriate HTML tags
2nd Most Common Misunderstanding
Translation/Request Time Confusion

- **What happens at page translation time?**
  - JSP constructs get translated into servlet code.

- **What happens at request time?**
  - Servlet code gets executed. *No* interpretation of JSP occurs at request time. The original JSP page is totally ignored at request time; only the servlet that resulted from it is used.

- **When does page translation occur?**
  - Typically, the first time JSP page is accessed after it is modified. This should never happen to real user (developers should test all JSP pages they install).
  - Page translation does *not* occur for each request.

---

**The JSP Lifecycle**

<table>
<thead>
<tr>
<th>JSP page translated into servlet</th>
<th>Servlet compiled</th>
<th>Servlet instantiated and loaded into server's memory</th>
<th>init (or equivalent) called</th>
<th>doGet (or equivalent) called</th>
<th>Request #1</th>
<th>Request #2</th>
<th>Request #3</th>
<th>Request #4</th>
<th>Request #5</th>
<th>Request #6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
ofoto.com:

- print and manage digital and conventional photos.

First USA Bank:

- largest credit card issuer in the world;
- most on-line banking customers
JSP/Servlets in the Real World

• Delta Airlines: entire Web site, including real-time schedule info

• American Century Investments: more than 70 mutual funds, $90 billion under management, two million investors
JSP/Servlets in the Real World

- Excite: one of the top five Internet portals; one of the ten busiest sites on the Web

Summary

- JSP makes it easier to create and maintain HTML, while still providing full access to servlet code
- JSP pages get translated into servlets
  - It is the servlets that run at request time
  - Client does not see anything JSP-related
- You still need to understand servlets
  - Understanding how JSP really works
  - Servlet code called from JSP
  - Knowing when servlets are better than JSP
  - Mixing servlets and JSP
- Other technologies use similar approach, but aren’t as portable and don’t let you use Java for the "real code"