JSPs and JSTLs

Duration 5 days.

Overview JavaServer Pages (JSPs) are used to create web content containing both static and dynamic components. JSPs have the dynamic capabilities of Java Servlet technology but with a more natural approach for creating the static content of the presentation layer. JSP is similar to PHP and ASP, but under the covers, it uses the Java programming language to create dynamically generated web pages based on HTML, XML, or other document types. JSP Standard Tag Library (JSTL) is a collection of useful tags you can use in your JSPs.

Prerequisites The ability to use an Eclipse-based IDE, such as RAD, IDz, etc. Java programming experience is required as is the ability to code and run simple servlets. Knowledge of HTML and some CSS and JavaScript is required.

Objectives Upon successful completion of this course, you will be able to:

- Code and run JSPs that invoke servlets and other JSPs.
- Code and run a servlet that forwards to a JSP and retrieves data from Javabeans using session, request and application scope.
- Use different scopes within a web application.
- Code JSTL and JSP tags in your JSPs.
- Use the MVC (Model View Controller) architecture.
- Write servlets and JSPs that receive and send data to web pages.

Participants Software developers who want to create dynamic web pages based on HTML.

Format Lecture and discussion with many hands-on exercises to reinforce the course content.

Topic Outline

JSP Basics

What is a JSP?
JSP - JavaServer Pages: Overview
Life Cycle of a JSP Page
Servlets vs JSPs vs HTML
Create and use a JSP Folder in an Eclipse-based IDE
Create and run a JSP file
Test JSPs on Multiple Browsers
Typical Client/ Server Flow
Web Application Architecture; MVC
Division of Labor
JSP invokes Servlet
HTML invokes JSP
Servlet invokes JSP
servlet gets <form> element values
"Nick" Names for servlets in web.xml
Troubleshooting
Why is my update not in browser???
Browser Cache vs Server Cache vs generated class files
Context Root
Change context name
HTML and JSP files in folders - context
Get Parameter Names
Servlet with getParameterNames
Output to browser

Topic Outline

JSP Basics (continued)
- Legacy JSP Code (for maintenance)
- Special Characters and Escape Characters
- HTTP is Stateless

Threads and Thread Safety
- Thread Safety with shared objects and shared resources
- Threads and JSPs
- Thread-safe servlets and JSPs
- Multiple Ways to Synchronize Code

Combining Servlets and JSPs
- JSP Logic Flow
- Servlet Forward to JSP
- <form> to launch Servlet
- Manage line breaks in Java Eclipse/ Windows/ Pop-ups
- Servlet Logic - session
- Servlet Logic - forward

JSP Tags
- JSP Scriptlet and Expression
- JSP Scriptlet containing "if"
- JSP gets data from a bean: Old Ways/ Newer Ways
- JavaBeans
- JavaBean Specification
- useBean tag – setting bean properties
- Using beans in JSPs
- <jsp:useBean attributes
- JSP Retrieves bean from request
- useBean tag - loading
- useBean tag - getProperty
- useBean tag - setProperty
- <jsp: include . . .> tag
- Other Include Techniques
- Tips for Links with web.xml
- back link versus back arrow
- JSP Action Tags
- Get and Post Methods
- HTTP is Stateless. How to "remember"?
- Instantiating Java objects inside JSPs
- Using built-in Java classes
- JSP 2.0 Enhancements
- JSPs: Simple to Complex Application Strategies

Error Handling
- Untrapped Error - NullPointerException
- Typical Website Errors
- HTTP 404 error
- HTTP 500 error
- JSP Runtime Error Processing
- JSP's Nine Implicit Objects
- Catching Exceptions – 2 Ways
- IE "limitation" for error-pages !!!

Topic Outline **Error Handling (continued)**

Catching Errors #1 – using web.xml
Catching Errors #1 – errorpage1.jsp
Catching Errors #1 – null_1.jsp
Catching Errors #1 – Browser Output
Catching Errors #2 – using <@page
Catching Errors #2 – JSP with error
Catching Errors #2 – Browser output
Catching Errors #3 – stack trace
Catching Errors #3 – Browser Output
Catching Errors #4 – Formatted
Catching Errors #4 – Browser Output

Topic Outline **More JSP Tags**

Implicitly Defined JSP Objects
Directives: <%@
<%@page directive attributes and examples
Exception Handling
Response and Page Encoding
<%@include directive
<%@taglib directive
Multi-thread considerations
Declarations <%!
JSP Expressions <%=
Scriptlets
Comments
Import JSP / HTML code

EL - Expression Language

Expression Language
Objects implicit to EL
EL to get param from web.xml
EL sessionScope and requestScope
Application Scope
EL Syntax and Operators
EL Precedence of Operators
EL Reserved Words
EL Examples
Handling Exceptions with EL
Oracle's Best Practices for Servlets and JSPs
JSTL and EL for Map contents

JSTL Tags

What is JSTL? - JSP Standard Tag Libraries
May need jstl jar files
Typical custom tags from the JSTL
Core "c" tags in JSPs
<c:out >
<c:set >
JSTL to prevent XSS (cross-site scripting) hacking
Import using JSTL
<c:if>
<c:choose>,<c:when>,<c:otherwise>

Topic Outline

JStL Tags

- <c:forEach>
- <c:forTokens>
- <c:catch>
- Alternating browser line colors using CSS

Hacking

- DoS and DDoS Attacks
- DoS Symptoms
- Methods of Attack
- Forging Sender Addresses
- Prevent Hacking

Miscellaneous Topics

- JSON
- JSON one-level Example in JavaScript
- JSON Array Example in JavaScript
- JSON Nested Objects in JavaScript
- JSON from a File
- Unicode
- <meta charset="utf-8">
- Favicon
- Create a favicon file
- Adding a favicon to your web page
- Favicon Output in 3 Browsers
- Same HTML running in Chrome vs IE vs Firefox
- Running html from html subfolder
- Running JSP from jsp subfolder
- Web Development Considerations
- Usability; 508 Compliance
- Browser Compatibility
- Responsive Web Design

What is the Internet?

- Internet concept and architecture
- Who owns the Internet?
- How old is the Internet?
- Internet Terminology
- URL – Web Suffixes naming conventions