

J2EE Web Development using JSP & Servlets

Course Duration: 5 day; / 35 hours;
Instructor-led

Time: 9.00am – 5.00pm

Break: 10.15am – 10.30am / 3.15pm – 3.30pm

Lunch: 1.00pm – 2.00pm

WHAT YOU WILL LEARN

JSP Training Overview This JSP and Servlets training course teaches beginning and early intermediate Java developers how to build database-driven Web applications using Java Server Pages and Java Servlets.

PREREQUISITES:

This 5 days Servlets/JSP training course can be tailored to any level of Java experience. In general, we recommend that all attendees be fluent in HTML and have a prior knowledge of object-oriented or object-based programming, as well as some knowledge of SQL and relational databases.

METHODOLOGY

This program will be conducted with interactive lectures, PowerPoint presentations, discussions and practical exercises

COURSE OBJECTIVE

- Design and build robust and maintainable web applications
- Create dynamic HTML content with Servlets and JavaServer Pages, using the JSP Standard Tag Library (JSTL)
- Make Servlets and JSP work together cleanly
- Use JSTL and other Custom Tag Libraries to separate Java and HTML code
- Access databases with JDBC
- Use Enterprise JavaBeans effectively from Servlets and JSP

COURSE OUTLINES

Module 1: Java EE Introduction

- Java EE Overview

- Java EE Technologies
- Java EE Architecture
- MVC

Module 2: Servlets and JSP with JSP Standard Tag Library (JSTL)

- Web Application Basics
 - a) How the Web works, Thin Clients, TCP/IP
 - b) HTTP overview, Brief HTML review
 - c) Overview of Java EE, servlets & Web applications.
 - d) Servlet Basics
- Servlet API
 - a) HTML Forms
 - b) HTTP: Request-response, headers, GET, POST
 - c) Overview: How Servlets Work
 - Servlet Lifecycle: init(), service(), destroy()
 - Requests and responses
 - d) Core Servlet API: GenericServlet, ServletRequest, and ServletResponse
 - e) HTTP Servlets: HttpServletRequest, HttpServletResponse and HttpServlet
 - f) Accessing Parameters
- Additional Servlet Capabilities
 - a) HTTP headers and MIME types
 - b) RequestDispatcher: Including and forwarding
 - c) Sharing data with the request object attributes
 - d) Sharing data with ServletContext attributes
 - e) Error Handling
- JavaServer Pages
 - a) Basics and Overview
 - JSP architecture
 - JSP tags and JSP expressions
 - Fixed Template Data
 - Lifecycle of a JSP
 - b) Model View Controller (MVC)
 - Model 1/Model 2 Architecture
 - c) Data Sharing among servlets & JSP
 - Object scopes or "buckets"

- Request, application, session and page scope
- Predefined JSP implicit objects (request, session, application, page)
 - <jsp:useBean>
 - <jsp:getProperty>, <jsp:setProperty>
 - <jsp:include>, <jsp:forward>
- Using Custom Tags
 - a) Custom tags overview
 - b) Reducing JSP complexity
 - c) Tag Libraries
 - Tag Library Descriptor (TLD)
 - Loading a tag library in a web app
 - d) The JSTL
 - e) JSP Expression Language (EL)
 - f) Using custom tags
 - The c:url, c:param, c:forEach, c:out tags
- More JSP Capabilities and Session Management
 - a) HTTP as a stateless protocol
 - b) Hidden form fields
 - c) Cookies: Overview, API, Using cookies
 - d) Session overview: Cookies and session tracking
 - e) HttpSession
 - Putting data into a session object
 - Retrieving data from a session object
 - f) Using session data in servlets and JSPs
 - g) Additional JSP Capabilities
 - Exception handling and error pages
 - Directives (page, include, others)
 - h) Import declarations
 - i) Multithreading considerations and data safety
 - SingleThreadModel interface
- Additional JSP Capabilities
 - a) JSP Directives
 - b) JSP Error Pages
 - c) JSP and Java
 - Declarations, Scriptlet overview, Scriptlet syntax
- More JSTL
 - a) Overview of JSTL libraries
 - b) The JSTL Expression Language
 - Expressions, Type Coercion, Operators, String concatenation, Implicit Objects
 - c) The Core JSTL Library

- General Purpose: c:out, c:set, c:catch
- Conditional: c:if, c:choose,
- d) Overview of other capabilities

- Additional Topics
 - a) Servlet Filter overview
 - Filtering examples, lifecycle, & filter chains
 - Filter API, Modifying a request, Modifying a response
 - b) Struts Overview
 - Advanced MVC – Struts overview
 - Command and State patterns
 - Struts View and Controller elements

Module 3: JDBC

- Introduction
 - a) Relational Database and JDBC Overview
 - Overview, Table Relationships, Web Based Data Access, JDBC Characteristics
 - b) JDBC Architecture, JDBC API Overview
 - DriverManager, JDBC Drivers
 - Naming databases with JDBC URLs
 - c) Connecting to a database
 - Connection interface, Establishing a connection
 - d) DataBaseMetaData
 - e) Handling Database Exceptions
- Data Access
 - a) DAO – Data Access Objects, O-R Mapping, Value Objects
 - b) Processing Database Data
 - Executing statements, precompiled statements and stored procedures
 - Processing ResultSets,
 - Dealing with Null data
 - Updating, inserting, retrieving data
 - Controlling Transactions
 - c) JDBC Driver Types
 - d) DataSource
 - Java EE and DataSource, Using JNDI
 - e) Connection Pooling
 - Overview, Usage, Advantages
- Advanced Topics
 - a) Other Database Access Technologies

- SQLJ, O-R Mapping Tools, JDO (Java Data Objects), EJB Entity Beans
- b) Advanced JDBC Features
 - Batch Updates, Scrollable Result Sets, Rowsets, User Defined Types, BLOBS, CLOBS

Module 4: Component Integration and EJB

- Database Integration
 - a) Web Architecture choices
 - b) Connecting servlets to a database via Data Source
- EJB
 - a) Overview of EJB
 - b) High-level EJB architecture
 - c) Roles within EJB
 - d) Client view of a session bean
 - e) JNDI (Java Naming and Directory Interface)
 - f) Remote and Home Interfaces
 - g) Writing an EJB client
- Other Technologies
 - a) XML and Web Services
 - b) JMS – Java Message Service
- Labs
 - a) Accessing a database from a servlet
 - b) Accessing an EJB from a servlet

Module 5: Conclusion