ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Summary

Description

JEE Web Fundamentals is a five-day hands-on JEE / Java EE training course geared for experienced Java developers new to JEE, who need to get up and running with essential dynamic web development skills. Created in collaboration with several leading JEE / Java EE s authors and industry experts, this comprehensive course teaches students how to design and program servlets and JSPs, including all the important concepts and hands on labs that will have you building working server-side applications in no time flat. This course provides core JEE knowledge and skills that can be used as the foundation for developing production-quality web applications to a basic level.

Objectives
At the end of this course, students will be able to:

- Design and build web applications from both business and technical requirements
- Build web interfaces with JSPs and Servlets, using the latest technologies in JEE 5.
- Write maintainable web applications that separate HTML and Java
- Understand the design and development of web applications using Servlets and JSPs
- Work JEE's version of dependency injection
- Make Servlets cooperate and share data
- Store and process session information
- Deal with concurrency issues
- Understand and create JavaServer Pages (JSPs)
- Link Servlets and JSPs
- Use the built in JSP objects
- Embed JavaBeans in a JSP
- Use JavaBeans in a JSP
- Use servlets and JSP together
- Access databases with JDBC
- Understand and work with JEE 5 persistence options, including JPA
- Work with new annotations included in JEE 5
- Expose web components as web services
- Recognize basic web security vulnerabilities and implement effective defenses

Topics

- JEE Application Architecture
- JDBC (Java Database Connectivity)
- Servlets
- Java Server Pages (JSP), EL, and JSTL
- JavaServer Faces
- EJB3 and Persistence
- Messaging
- Web Services
- JEE Security
- Web Application Security

Audience

This is an intermediate level Java EE (JEE) developer course, designed for experienced Java developers, new to JEE, that need to further extend their skills in web development and Struts.

Prerequisites

Attendees should have an extensive working knowledge in developing basic Java applications.

Duration

Five days

ProTech

protechtraining.com

"Charting the Course ...

... to Your Success!"

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Outline

Session: JEE Application Architecture

I. Technical Overview of JEE
- A. What is JEE?
- B. Common Themes In JEE Framework
- C. JEE Containers and Components
- D. Servlets: Components Running in Web Container
- E. Java Server Pages (JSP) are JEE Web Component
- F. EJBs and Web Services are Components As Well
- G. JEE Containers
- H. Compliant JEE Framework Ready for an Application
- I. JEE Application Modules
- J. The JEE 5 Specification
- K. What was New in JEE 5
- L. JEE Platform Roles

II. JEE Application Architectures
- A. JEE and the Web
- B. JEE 1st Generation DB-centric Web Application
- C. Adding JSPs to Separate Presentation
- D. Eliminating Java code from view
- E. The Model 2 Architecture
- F. Using EJBs
- G. JEE Infrastructure Supporting Web Applications

Session: Web Applications

III. Understanding Web Applications
- A. JEE Application Modules
- B. The Truth about Archives
- C. Enterprise Application Archive (EAR)
- D. Enterprise JavaBean Archive (JAR)
- E. JEE Application Client (JAR)
- F. Resource Adapter Archive (RAR)
- G. Web Application Archive (WAR)
- H. Directory Structure
- I. Application Assembly
- J. Assembling JEE Applications
- K. JEE Module Structures
- L. Assemblying Web Applications
- M. Assemblying Enterprise Applications
- N. Deployment Scenarios
- O. Managing Server Resources

IV. Configuring Web Applications
- A. Mapping an HTTP Request to a Resource
- B. The web.xml File
- C. Structure of web.xml
- D. Declaring Servlets and JSPs
- E. Servlet Mapping
- F. Servlet Init Parameters
- G. Web Application init Parameters
- H. Welcome Page
- I. Error Page
- J. More Elements
- K. Annotations in Servlets

Session: Developing Servlets

V. Introduction to Servlets
- A. Servlet Overview
- B. Life Cycle of Servlets
- C. Servlet Lifecycle is Handled by Web Container
- D. HttpServlet
- E. Writing the init Method
- F. HttpServlet doXXX Methods
- G. The HelloWorld HTTP Servlet
- H. Writing a Simple HTTP Servlet
- I. HttpServletRequest Methods
- J. ServletResponse
- K. HttpServletResponse
- L. Servlet I/O Classes
- M. Return a Status Code
- N. Building the Output Document
- O. Sending Binary Content

VI. Debugging JEE Applications
- A. The Debug Perspective
- B. Breakpoints
- C. Breakpoints and the Debug view
- D. Debug Mode
- E. Running in Debug Mode
- F. Running a Server in the Debug Mode
- G. Variables View
- H. Change Variable Value
- I. Set Variables Value

VII. Processing Input Data
- A. Form Processing with Servlets
- B. HTML Form
- C. LoginServlet doPost

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Outline (cont'd)

VIII. Server-Side Control
- A. Request Dispatcher
- B. Forward the processing
- C. Passing Processing on and Getting it Back
- D. Servlet Runs Within Web Container Environment
- E. Several Options for Receiving Data
- F. Init Parameters and Attributes
- G. ServletConfig
- H. ServletContext
- I. Servlet Variables are Scoped
- J. HTTP Request Information
- K. Several Options for Sharing Data
- L. configuration and Context
- M. Servlet Variables
- N. HttpServlet Request
- O. Threading and Data

IX. Client Side Control
- A. Output Buffering
- B. Setting Status Codes
- C. Setting Headers
- D. sendRedirect
- E. Disabling Client Caching
- F. Supporting Persistent Connections
- G. Setting Content Length
- H. Dynamic Content Pushing

X. Maintaining Client State – Sessions
- A. Session Management
- B. Tracking Problem – Stateless HTTP
- C. Data Problem – Session Data
- D. Solving the Tracking Problem
- E. Cookies
- F. Cookie Behavior
- G. Retrieving Cookies
- H. servlet to set Cookies
- I. Servlet to Show Cookies
- J. URL Rewriting
- K. Solving the Data Problem
- L. Web Container Manages Session Instances
- M. Sessions with Cookies
- N. Cookie-Based Sessions
- O. Basic Session Implementation
- P. Cookie Detection is not Standardized
- Q. Getting Rid of Http Sessions
- R. Session with URL Rewriting
- S. Session Using Cookie is Widely Used Approach

XI. Application and Session Events
- A. Event Listener Model
- B. Life Cycle Events in a Web-Application
- C. Declare the Listener
- D. Type of Events
- E. Context Listeners
- F. Session Listeners
- G. Session Listeners for Session-Objects

Session: Filters

XII. Overview of Filters
- A. What is a Filter
- B. Single Filter
- C. Filter Objects
- D. doFilter Method Example
- E. init Method
- F. Filter Life Cycle
- G. Cascading Filters
- H. Register a Filter

XIII. Filtering Requests and Responses
- A. Request Wrapper
- B. Process the Request
- C. Examples of Request Filters
- D. Filter the Response
- E. Response Wrapper
- F. Process the Wrapper
- G. Examples of Response Filters

Session: Developing JavaServer Pages

XIV. Introduction to JavaServer Pages
- A. Separating Presentation from Model
- B. Java Server Page (JSP): An Extension of Servlet
- C. Lifecycle of a JSP
- D. Example JSP
- E. JSP Syntax Consists of Several Types
- F. JSP Scripting: Declarations
- G. JSP Scripting: Expressions
- H. JSP Scripting: Scriptlets
- I. JSP Directives
- J. The session Attribute
- K. The errorPage/isErrorPage Attribute
- L. JSP Actions
- M. JSP Actions: Include/Forward
- N. Typical JSP Access Model
- O. JSP Action: useBean
- P. Implicit Objects
- Q. Comments in JSPs
- R. JSPs or Servlets?

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Outline (cont'd)

XV. Implicit Objects
- A. Implicit Objects
- B. Page Object
- C. Config Object
- D. Request Object
- E. Response Object
- F. Out Object
- G. Output Buffer
- H. Session Object
- I. Application Object
- J. PageContext Object
- K. Attributes
- L. Session Attributes
- M. Exception Handling
- N. Exception Handling Example

XVI. Actions, Java Beans,™ and Custom Tags
- A. Standard Actions
- B. Forwarding
- C. Including
- D. Using JavaBeans™ and JSP
- E. Declaring to use a Bean
- F. Using a Bean, Example
- G. Setting and Getting Properties
- H. Example of Using a Bean
- I. What are Custom Tags?
- J. Benefits
- K. Create and Use a Custom TagLib
- L. Usage Example

XVII. JSPs in Depth
- A. JSP 2.1 Specification
- B. The web.xml in JEE 5
- C. <jsp-config>
- D. JSP Format Rules
- E. JSP Error Pages

XVIII. The Unified Expression Language
- A. The Expression Language
- B. The Expression Language (JSP 2.0)
- C. JSP 2.1 Introduces Unified EL
- D. The Unified EL Defined
- E. Enable/Disable Unified EL
- F. Variables
- G. Literals and Operators in the Unified EL
- H. Implicit Objects in Unified EL
- I. Reserved Words
- J. Unified EL Functions
- K. Developing the Function
- L. Declaring the Function in the TLD
- M. Using the Function
- N. Pre-Built JSTL Functions

Session: JSTL

XIX. JSTL Introduction and Core Library
- A. Introduction to JSTL
- B. Unified EL in JSTL
- C. Review of JSP Bean Tags
- D. Tag Collaboration
- E. JSTL Core
- F. <c:choose> <c:when> <c:otherwise>
- G. <c:if>
- H. <c:import>
- I. <c:forEach>
- J. <c:forTokens>
- K. <c:out>
- L. <c:param> <c:catch>
- M. <c:redirect>
- N. <c:remove>
- O. <c:set>
- P. Using c:set to pass HTML
- Q. <c:url>

XX. JSTL Format Library
- A. JSTL Format
- B. <fmt:requestEncoding>
- C. <fmt:setLocale>
- D. <fmt:timeZone>
- E. <fmt:setTimeZone>
- F. <fmt:bundle> <fmt:setbundle>
- G. <fmt:message>
- H. <fmt:param>
- I. <fmt:formatNumber>
- J. <fmt:formatNumber> (Example)
- K. <fmt:parseNumber>
- L. <fmt:formatDate>
- M. <fmt:parseDate>

Session: JavaServer Faces

XXI. Working With JavaServer Faces (JSF)
- A. Typical Web-User Interactions
- B. Web Application Development Challenges
- C. JSF Addresses Some of Those Challenges
- D. JSF Solves Common Web App Problems
- E. JSF Architecture Components
- F. Nominal Flow of Processing
- G. Reference Implementation and Alternatives
- H. Creating a JSF Application

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Outline (cont'd)

Session: Database Integration: JDBC and JEE

XXII. JDBC and Its Position in JEE
- A. Recap of JDBC
- B. JDBC Overview
- C. The JDBC API
- D. JDBC in JEE
- E. Programming with DataSources
- F. DataSource Programming Best Practices

XXIII. JDBC Data Access API
- A. Structured Query Language (SQL)
- B. Statements
- C. PreparedStatements
- D. ResultSet
- E. Executing Inserts, Updates, and Deletes
- F. Mapping SQL Types to Java Types
- G. CallableStatement

XXIV. The DAO Pattern
- A. Data Access Object (DAO)
- B. DAO Structure
- C. DAO Example: ProductDAO
- D. The DAOFactory
- E. Complete UML Diagram
- F. Using the ProductDAO

Session: Persistence and EJB 3

XXV. Java Persistence API (JPA)
- A. Object-Relational Mapping (ORM)
- B. Typical ORM Approaches
- C. ORMs Use Lazy Loading
- D. ORM Without Lazy Loading
- E. Benefits of Using ORM Framework Like
- F. Hibernate
- G. Example of Lazy Loading
- H. Loading an Entire Object Graph
- I. Java Persistence Overview (JPA)
- J. JPA Package

XXVI. Working With Enterprise JavaBeans 3.0
- A. Defining Enterprise JavaBeans
- B. JavaBeans™ vs Enterprise JavaBeans™
- C. EJB Architecture Overview
- D. EJB Container
- E. Types of EJBs
- F. Enterprise Beans
- G. Session Beans
- H. Entities
- I. Message-Driven Bean
- J. The Decorator
- K. Deployment Descriptor/Annotations
- L. EJB-Jar File

XXVII. Entities and Persistence
- A. JPA Configuration Files
- B. Example of persistence.xml
- C. Mapping Objects to the Database
- D. Example of Annotated Mapping
- E. Mapping with Annotations
- F. JPA Example
- G. EntityManager Works with Entities

XXVIII. Transactions
- A. Transaction Definitions
- B. The ACID Transaction Properties
- C. Transaction Lifecycle
- D. Overview of a Transactional System
- E. JEE Transaction Support

Session: Additional JEE Components

XXIX. Other JEE APIs
- A. Java Message Service (JMS)
- B. When is Messaging Used?
- C. Two Messaging Models
- D. More On Publish/Subscribe
- E. Logical View of Publish/Subscribe
- F. More On Point-to-Point (P2P)
- G. Logical View of Point-To-Point
- H. Message Servers
- I. JavaMail
- J. JavaMail Architecture
- K. What is XML?
- L. Benfits of XML – Example XML Document
- M. XML Separates Structure, Content, and Format
- N. Content: XML is Strict
- O. Structure: DTDs or XML Schema
- P. Format: XSLT and XSL-FO
- Q. XML Transformation to HTML
- R. Parser and API's
- S. What is JAXP?
- T. Pluggable Layers
- U. JEE and XML

JEE Programming Fundamentals (with Servlets/JSPS, JDBC, JNDI and More)

Course Outline (cont'd)

XXX. Web Services in JEE – WSEE
- A. What are Web Services?
- B. Web Services Architecturally
- C. XML and Web Service APIs
- D. Web Services for JEE - WSEE
- E. Servlets as Web Services
- F. Stateless Session EJBs as Web Services
- G. Routing SOAP requests to an EJB
- H. WSDD
- I. WSDD Example
- J. Simply Another JEE Component

Session: Accessing JEE Resources

XXXI. Java Naming and Directory Interface (JNDI)
- A. Naming Services - Overview
- B. A Unified Heterogenous Namespace
- C. Java Naming and Directory Interface (JNDI)
- D. JNDI Implementation Architecture
- E. Benefits

XXXII. Dependency Injection
- A. Dependency Injection
- B. Dependency Injection to Handle Access
- C. Dependency Injection: Inversion of Control
- D. Dependency Injection: Implementation
- E. Dependency Injection: JEE
- F. JNDI vs. Dependency Injection

Session: Security

XXXIII. JEE Security
- A. Typical JEE App Server Security Services
- B. Java 2 Platform Security Model
- C. Java 2 Security in JEE
- D. JAAS Authentication: Who are you?
- E. JEE Security Overview
- F. Authorization: Are you allowed access?
- G. High-Level Trace of JEE Authorization
- H. Deployment Descriptors Play a Large Role
- I. Declaring Secure Resources
- J. Declaring the security roles
- K. Typical JEE App Server Security Services
- L. Security on the Web
- M. Secure Web Traffic
- N. SSL In Action
- O. Responsibilities For Security
- P. CMS: Declaring HTTPS
- Q. Authentication Challenge Mechanisms

XXXIV. Web Application Security Overview
- A. Attacks are Constant and Changing
- B. Open Web Application Security Project
- C. Assets are the Targets
- D. The Context for Defensive Coding
- E. Attackers Not Hackers
- F. Cross-Site Scripting (XSS): Description
- G. Defending Against XSS Attacks
- H. SQL Injection: Description
- I. SQL Injection Example
- J. SQL Injection: Drill Down
- K. Defending Against SQL Injection Attacks

XXXV. Handling Untrusted Input
- A. Unvalidated Input: Description
- B. Protecting a Web Resource
- C. Tenacious D
- D. Defending a Web Application/Resource
- E. Responding to Error State
- F. Best Practices for Untrusted Data
- G. Additional Types of Attacks