

Developing Java EE Web Applications (JEE6/JEE7), plus JPA and REST

Course Summary

Description

Java Enterprise Edition (JEE) is a powerful platform for building web and database-driven applications. This course provides the information you need to design and build your own data-driven web applications. You'll learn the details of the core JEE Web and database technologies and how to leverage the strengths of each. You'll also be introduced to other important web-based technologies such as creating REST services with JEE.

Throughout the course, students will create code for a online store. Students will learn not only specific topics and APIs but also how to fit the pieces together into a complete application.

Objectives

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

- Design and build robust and maintainable web applications
- Create dynamic HTML content with Servlets and Java ServerPages, 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 and the JPA (Java Persistence API)
- Structure a clean data access layer.
- Gain a high level understanding of REST services and JAX-RS
- See the full descriptions for detailed content

Topics

- Java EE Introduction
- Servlet Basics
- Additional Servlet Capabilities
- JavaServer Pages
- Using Custom Tags
- HTTP Session Tracking
- Web Security
- Additional Capabilities
- JDBC Introduction and Architecture
- Resource Integration
- Introduction to JPA
- Introduction to REST
- Introduction to JAX-RS

Audience

This course is designed for developers needing to write Java EE Web Applications, and REST resources.

Prerequisites

Students should have working knowledge of Java programming, including use of inheritance, interfaces, and exceptions.

Duration

Five days

Developing Java EE Web Applications (JEE6/JEE7), plus JPA and REST

Course Outline

- I. *Java EE Introduction*
 - A. Java EE Overview and Technologies
 - B. JEE Architecture Choices
 - C. Java Web App Introduction
- II. *Servlet Basics*
 - A. Servlet Basics and Capabilities
 - B. Servlet API - Writing a Servlet
 - C. HTML Forms Review
 - D. HTTP Review: Request-response, headers, GET, POST
 - E. Overview: How Servlets Work
 - F. Servlet Lifecycle and API - Servlet, HttpServlet, @WebServlet
 - G. Requests and Responses - HttpServletRequest and HttpServletResponse
 - H. Accessing Parameters
 - I. web.xml
- III. *Additional Servlet Capabilities*
 - A. Working with HttpServletResponse (Errors, Headers, MIME types)
 - B. Initialization
 - C. Overview
 - D. Using ServletConfig and ServletContext
 - E. Init Parameters - Servlet and Web App
 - F. Error Handling: Error Pages and Their Configuration
- IV. *Java Server Pages*
 - A. Basics and Overview
 - B. JSP architecture
 - C. JSP tags and JSP expressions
 - D. Fixed Template Data
 - E. Lifecycle of a JSP
 - F. Model View Controller (MVC)
 - G. Overview
 - H. Java EE Model 2 Architecture
 - I. Servlets as Controllers, RequestDispatcher, Forwarding and Including
 - J. Data Sharing in a Web App
 - K. Object scopes or "buckets"
 - L. Using JavaBeans to Hold Data
 - M. Using the Scope Objects - get/set/remove Attributes
 - N. Request, application, session and page scope
- O. JSP Expression Language (EL) and Data Access
- P. JSP EL Overview
- Q. JSP Expressions, and Accessing Data
- R. Predefined JSP EL implicit objects (pageContext, param, header, scope objects)
- S. pageContext in Detail
- T. jsp:include, jsp:forward, the page Directive
- U. JSP Error Pages
- V. *Using Custom Tags*
 - A. Custom tags overview
 - B. Tag Libraries Overview
 - C. taglib Directive - Using a Tag Library
 - D. JSTL
 - E. Overview
 - F. c:out, c:forEach
 - G. c:url, c:param
 - H. More about JSTL
 - I. Other Useful Core Tags< - c:if, c:choose
 - J. Formatting: formatNumber, formatDate, Resource Bundles
- VI. *HTTP Session Tracking*
 - A. HTTP and Client State
 - B. Cookies - Overview, Servlet Access, Usage, Issues
 - C. Sessions
 - D. Servlet/JSP Session Support, HttpSession
 - E. Using Sessions - Putting Data in, Retrieving Data From
 - F. How Sessions Work
- VII. *Web Security*
 - A. JEE Security Overview - Role Based and Declarative Security
 - B. web.xml Security Elements
 - C. Web Authentication - Basic and Form-Based
- VIII. *Additional Capabilities*
 - A. Servlet Filter overview - Example and lifecycle
 - B. Custom Tags Using Tag Files - Writing and Using Simple Tag Files
 - C. Event Listeners

Developing Java EE Web Applications (JEE6/JEE7), plus JPA and REST

Course Outline (cont'd)

IX. *JDBC Introduction and Architecture*

- A. JDBC Overview
- B. JDBC Architecture and JDBC Drivers
- C. Connecting to a database
- D. Working with JDBC - Executing Statements and Processing Results

X. *Resource Integration*

- A. DataSources and Connection/Statement Pooling
- B. Resource Injection, @Resource, and JNDI
- C. CDI and Dependency Injection

XI. *Introduction to JPA*

- A. JPA Architecture and Programming View
- B. Entity Classes and Annotations
- C. Mapping an Entity Class
- D. EntityManagerFactory and EntityManager
- E. Inserting and Updating with JPA
- F. Querying and JPQL
- G. Entity Lifecycle
- H. Integrating JPA with the Web Tier

XII. *Introduction to REST*

- A. Overview and Principles
- B. REST Characteristics
- C. Resources and Operations
- D. REST Principles
- E. Requests and Responses
- F. REST APIs
- G. URI Templates
- H. GET, POST, PUT, DELETE
- I. Safe and Idempotent Methods

XIII. *Introduction to JAX-RS*

- A. APIs and Implementations
- B. JAX-RS Overview, Annotations
- C. JAX-RS Implementations
- D. Runtime Environment
- E. Application Server, Servlet-Only Container
- F. Architectural and Implementation Perspectives
- G. Configuring the Application
- H. Ajax-JavaScript Clients
- I. Overview
- J. Classic vs. Ajax Interactions
- K. Working with Ajax-JavaScript