Java EE 7 Front-end Web Application Development Ed 2

Duración

Días: 5 Días
Horas: 30 horas

Descripción

This Java EE 7: Front-end Web Application Development training helps you explore building and deploying enterprise applications that comply with the Java Platform, Enterprise Edition 7 Web Profile. Expert Oracle University instructors will help you explore annotations, Session Enterprise JavaBeans (EJB-Lite), Java Persistence API (JPA), servlets, JavaServer Pages(JSPs), Contexts and Dependency Injection (CDI), JAX-RS RESTful web services, the Java API for WebSocket and the Java API for JSON processing.

A quién se dirige

- J2EE Developer
- Java Developers
- Java EE Developers
- Web Administrator

Requisitos

Required Prerequisites

- Able to author HTML, CSS, and JavaScript enabled web pages
- Basic understanding of database concepts and SQL syntax
- Experience with Java SE, or Java Programmer Certification
- Understand object-oriented principles
- Java SE 8 Programming

Suggested Prerequisites

- Experience with an Integrated Development Environment
- JavaScript and HTML5: Develop Web Applications

Objetivos

1. Create scalable, transacted business logic with EJB-Lite
2. Access relational databases using the Java Persistence API
3. Create web-based user interfaces using Servlet, JSP, JAX-RS, and JavaScript technologies
4. Package, deploy and debug enterprise applications
5. Identify the services provided by an Application Server
6. Create Java EE technology applications with the Java EE 7 Platform
7. Develop and run an EJB technology application
8. Select the correct Java EE Profile for a given application
9. Create and use Java annotations
10. Develop basic Java Persistence API entity classes to enable database access
11. Develop a web-based user interface using Servlets, JSPs, and JAX-RS
12. Design applications to use dependency injection
13. Use IDEs and Application Servers for Java EE development

Qué aprenderá

- Develop web-based interfaces for both desktop and mobile devices.
- Assemble an application.
- Build Java applications.
- Deploy application on application server (Java EE runtime environment).

Beneficios para usted

By taking this course, you'll gain hands-on experience building Java EE web applications. You will get the chance to create web-based user interfaces using HTML5 and JavaScript along with JSPs and servlets. Web-based user interfaces will use AJAX to communicate with RESTful web services you create; data will persist using JPA and optimistic locking.

Participate in Hands-On Labs

By learning through hands-on exercises via structured labs, you'll get a chance to explore EJB-Lite session bean components, which can be used with container-managed transactions. You'll perform lab exercises using the NetBeans IDE and WebLogic Server.

Contenido

1. Java Platform, Enterprise Edition
    - The Java EE Platform
    - The needs of enterprise application developers
    - Java EE specifications
    - A comparison of services and libraries
    - The Java EE Web Profile
    - Java EE application tiers and layers

2. Enterprise Development Tools and Applications
    - The purpose of an application server
    - Starting and stopping WebLogic Server

- Properties of Java EE components
- The development process of a Java EE application
- Configuring and packaging Java EE applications

3. JavaBeans, Annotations, and Logging
    - Java SE features used in Java EE applications
    - Creating POJO JavaBeans components
    - Using Logging and Common Java Annotations
    - Develop custom annotations
    - The role of annotations in Java EE applications

4. Java EE Web Architecture
    - The HTTP request-response model
    - Differences between Java Servlets, JSP, and JSF components
    - Application layering and the MVC pattern
    - Avoiding thread safety issues in web components
    - Use the Expression Language

5. Developing Servlets
    - The Servlet API
    - Request and response APIs
    - Set response headers
    - Two approaches to creating a response body
    - Uploading files using a servlet
    - Forwarding control and passing data
    - Using the session management API

6. Developing with JavaServer Pages
    - The role of JSP as a presentation mechanism
    - Authoring JSP view pages
    - Processing data from servlets in a JSP page
    - Using tag libraries

7. JAX-RS Web Services
    - The need for web services
    - Designing a RESTful web service

- Create methods with prescribed rules of HTTP method behavior
- Create JAX-RS resource and application classes
- Consume query and other parameter types
- Produce and consume complex data in the form of XML
- HTTP status codes

8. Java RESTful Clients
    - Pre-JAX-RS 2 Clients: HttpUrlConnection and the Jersey Client API
    - The JAX-RS 2 Client API

9. HTML5 Applications with JavaScript and AJAX
    - HTML DOM manipulation with JavaScript
    - RESTful clients with JavaScript (AJAX)
    - Limitations of JavaScript clients
    - The Same-Origin policy and CORS

10. WebSocket and the Java API for JSO Processing
    - Web Service Limitations
    - WebSocket Explained
    - Creating WebSockets with Java
    - Client-side WebSokect with JavaScript
    - Client-side WebSocket with Java
    - Consuming and Producing JSON with Java

11. Implementing a Security Policy
    - Container-managed security
    - User roles and responsibilities
    - Create a role-based security policy
    - The security API

12. POJO and EJB-Lite Component Models
    - The role of EJB components in Java EE applications
    - The benefits of EJB components
    - Operational characteristics of stateless and stateful session beans
    - Creating session beans
    - Creating session bean clients

13. The Java Persistence API
    - The role of the Java Persistence API in Java EE applications
    - Basics of Object-relational mapping
    - The elements and environment of an entity component
    - The life cycle and operational characteristics of entity components
14. Implementing a transaction policy
    - Transaction semantics
    - Programmatic vs. declarative transaction scoping
    - Using JTA to scope transactions programmatically
    - Implementing a container-managed transaction policy
    - Optimistic locking with the versioning of entity components
    - Pessimistic locking using EntityManager APIs
    - The effect of exceptions on transaction state