

Business Component Development with EJB Technology, Java EE 6

Duration: 3 Days

What you will learn

The latest release of the Enterprise JavaBeans (EJB) 3.1 Technology available in the Java EE 6 platform builds on previous EJB technology and further simplifies how developers approach creating business components. EJB 3.1 makes many improvements that reflect common usage patterns, including: Singletons, No-interface view, Java Naming and Directory Interface (JNDI), Asynchronous Session Bean, and Timer Service.

The Business Component Development with Enterprise JavaBeans Technology, Java EE6 course provides students with the knowledge required to build robust back-end functionality using Enterprise JavaBeans (EJB[™]) version 3.1 technology. The course provides a practical exploration of the EJB technology coding experience of session beans and message driven-beans. The course also examines EJB design, best practices, transaction management, messaging fundamentals, and security.

The course features the Java Platform, Enterprise Edition 6 (Java EE 6) technology, and uses the Java EE 6 SDK. The students perform the course lab exercises using the NetBeans Integrated Development Environment (IDE). The hands-on lab environment uses the Java EE GlassFish v3 server.

Students who can benefit from this course

- * Java Developers who are looking to build web-based and/or enterprise-based applications that incorporate EJB technology
- * Java Developers who are preparing for the Sun Certified Enterprise JavaBeans Developer certification.

Prerequisites

Required Prerequisites

Demonstrate experience with the Java programming language

Integrate existing Java code (for example, reuse existing classes created by other team members)

Suggested Prerequisites

Object-Oriented Analysis and Design Using UML (OO-226)

Java Programming Language, Java SE 6 (SL-275-SE6)

Developing Applications for the Java EE 6 Platform (FJ-310-EE6)

Course Objectives

Implement business-tier functionality using EJB technology

Describe best practices and other advanced issues in business component development with EJB technology

Assemble and deploy EJB technology business-tier components on an application server

Integrate an EJB technology-based application using the Java Messaging Service API

Create and implement timer-based services

Integrate transactions and security into an enterprise application

Course Topics

Introduction to Java EE

- Gain an understanding of the Java Platform, Enterprise Edition (Java EE)
- Examine the Java EE application architecture
- Examine Java EE container services
- Examine the EJB component types
- Evaluate the EJB Lite Container
- Compare Java EE application development with traditional enterprise application development

Implementing Session Beans

- Examine session beans
- Identify the three types of session beans
- Choose the correct session bean type given a business constraint
- Create session beans Package and deploy session beans

Accessing Session Beans

- Understand the purpose and role of JNDI in relation to EJB components
- Configure JNDI environment properties
- Use JNDI to look up a resource
- Write code that receives a resource reference through injection
- Create a session bean client
- Create a session façade
- Use dependency injection to locate an EJB

Advanced Session Bean Concepts

- Understand the relationship between the EJB container and an EJB component
- Describe the life cycle for stateless and stateful session beans
- Implement session bean life cycle methods
- Use a session bean to perform asynchronous communication
- Have fine-grained control over packaging and deployment

Singleton Session Bean

- Understand the advantages and disadvantages of using a singleton session bean
- Create a singleton session bean
- Describe the life cycle of a singleton session bean
- Implement singleton session bean life cycle methods
- Describe singleton concurrency access
- Implement a concurrency management strategy

Developing Java EE Applications Using Messaging

- Review JMS technology
- Describe the roles of the participants in the JMS API messaging system
- Create a queue message producer
- Create a synchronous message consumer

Developing Message-Driven Beans

- Understand the short-comings of using session beans as messaging consumers
- Describe the properties and life cycle of message-driven beans
- Create a JMS message-driven bean
- Create life cycle event handlers for a JMS message-driven bean
- Configure a JMS message-driven bean

Using Timer ServicesObjectives

Describe timer services
Create a timer notification callback
Process a timer notification callback Manage timer objects

Implementing Interceptor Classes and Methods

Describe interceptors and interceptor classes
Create a business interceptor method in the enterprise bean class
Create an interceptor class
Associate multiple business interceptor methods with an enterprise bean
Include life cycle callback interceptor methods in an interceptor class

Implementing Transactions

Describe transaction demarcation management
Implement CMT
Interact programmatically with an ongoing CMT transaction
Implement BMT Apply transactions to messaging

Implementing Security

Understand the Java EE security architecture
Authenticate the caller Examine Java EE authorization strategies
Use declarative authorization
Use programmatic authorization Examine the responsibilities of the deployer

Using EJB Technology Best Practices

Define best practices and state the benefits of using EJB technology best practices
Select and apply known patterns to Java EE application design
Incorporate effective exception handling into your Java EE application design