

Java: Under the Hood

Enterprise JavaBeans Development

Course Outline

Course Length:

5 days

Audience:

This course is intended for experienced Java developers who want to learn about Enterprise JavaBeans and how to write EJB applications.

Prerequisites:

Attend the *Java: Under the Hood – Advanced Java Programming* course, or possess the equivalent knowledge of Java programming, serialization, RMI and JDBC.

Objectives:

- Understand what Enterprise JavaBeans are.
- Understand the EJB architecture.
- Become familiar with the J2EE specification.
- Understand EJB servers and containers.
- Be able to assemble and deploy EJB's.
- Become familiar with component models and component transaction models.
- Understand entity and session beans.
- Be able to write an entity bean.
- Be able to write a session bean.
- Understand the Java Message Service (JMS).
- Be able to write a message-driven bean.
- Understand the responsibilities of the Home interface and EJBObject.
- Understand the purpose of a deployment descriptor.
- Be able to write a deployment descriptor.
- Understand the EJB transaction model and the Java Transaction API.
- Learn the difference between container managed transactions and bean managed transactions.
- Understand the security features of EJB and how to implement them.
- Learn how exceptions are handled in EJB applications.

Java: Under the Hoodä

Enterprise JavaBeans Development

Module Descriptions

Module 1: An Overview of the Java 2 Platform, Enterprise Edition

- ❑ **The J2EE Specification:** A superset of the EJB specification.
- ❑ **The EJB Specification:** A description of services provided by an EJB server.
- ❑ **Java Servlets:** Applications that run in a Web server.
- ❑ **Java Database Connectivity:** Connecting to a database from within a Java application.
- ❑ **JavaServer Pages:** Used for creating dynamic Web page content.
- ❑ **XML:** The eXtensible Markup Language
- ❑ **JNDI:** The Java Naming and Directory Interface.
- ❑ **Transactions:** The Java Transaction Service and the Java Transaction API.
- ❑ **Java Message Service:** An API for enterprise messaging needs.
- ❑ **Java IDL:** The Interface Definition Language and RMI-IIOP.
- ❑ **Java Mail:** An overview of the Java Mail API.
- ❑ **Connectors:** Integrating existing information systems with new applications.
- ❑ **Summary**

Module 2: The Enterprise JavaBeans Architecture

- ❑ **Why EJBs?:** The benefits of using Enterprise JavaBeans.
- ❑ **The Roles of EJB Development:** The developer, deployer, application assembler, container and server.
- ❑ **EJB's vs. JavaBeans:** A comparison of these Java component models.
- ❑ **Implementing the EJB Specification:** The server vendor.
- ❑ **Servers and Containers:** Servers are applications that manage containers.
- ❑ **The Remote Interface:** The business methods exposed to the client.
- ❑ **The Home Interface:** Contains methods for creating the bean.
- ❑ **The Bean Class:** Session and Entity Beans.
- ❑ **Three Types of Enterprise JavaBeans:** Session, entity and message-driven beans.
- ❑ **The EJBObject:** The class representing the bean's container.
- ❑ **The Ejb-jar File:** Deploying the bean in the server.
- ❑ **Summary**

Module 3: An Introduction to Enterprise JavaBeans

- ❑ **“Hello, EJBs!”:** A simple Enterprise JavaBean.
- ❑ **Step 1: The Remote Interface:** Determine the business methods the client needs.
- ❑ **Step 2: Write the Home Interface**
- ❑ **Step 3: Write the EJB Class:** Implement the methods of the Remote interface.
- ❑ **Step 4: Create the Deployment Descriptor**
- ❑ **Step 5: Create the JAR File**
- ❑ **Step 6: Deploy the Bean**
- ❑ **Step 7: Write the Client**
- ❑ **Step 8: Running the Client**
- ❑ **Summary**

Module 4: Session Beans

- ❑ **An Overview of Session Beans:** Providing the workflow of the EJBs.
- ❑ **The Lifecycle of a Session Bean:** How the client views a session bean.
- ❑ **The SessionBean Interface:** A session bean must implement these methods.
- ❑ **Swapping Session Beans:** Saving the conversational state of a session bean.
- ❑ **Stateful Session Beans:** The lifecycle of stateful beans.
- ❑ **An Example of a Session Bean**
- ❑ **The SessionSynchronization Interface:** The lifecycle of stateless beans.
- ❑ **Stateless Session Beans:** The lifecycle of stateless beans.
- ❑ **The Session Context:** The bean’s interface to the container.
- ❑ **The Session Bean Class:** Specifics about writing a session bean class.
- ❑ **Summary**

Module 5: Entity Beans

- ❑ **An Overview of Entity Beans:** A bean representing persistent data.
- ❑ **The Features of Entity Beans:** Long-lived, persistent data objects.
- ❑ **Developing Entity Beans:** The responsibilities of the bean developer.
- ❑ **The Lifecycle of an Entity Bean:** How the container manages entity beans.
- ❑ **The Home Interface:** Creating and finding entity beans.
- ❑ **Primary Key:** The primary key uniquely identifies entity beans.
- ❑ **The EntityBean Interface:** An entity bean must implement these methods.
- ❑ **The Entity Bean Class:** Implementing the business methods and create and finder methods.
- ❑ **Bean-managed Persistence:** The bean developer writes the database code.
- ❑ **The Entity Context:** The bean’s interface to the container.
- ❑ **Container-managed Persistence:** The container writes the database code.
- ❑ **Deploying CMP Entity Beans:** Specifying CMP fields and finder methods.
- ❑ **Summary**

Module 6: Transactions

- ❑ **Overview of Transactions:** How the EJB specification provides support for transactions.
- ❑ **Managing Transactions:** Transactions are either bean-managed or container-managed.
- ❑ **Container-Managed Transactions (CMT):** The bean does not worry about creating or managing a transaction.
- ❑ **Transaction Attributes for CMTs:** Setting the transactional behavior of the bean.
- ❑ **Deployment Descriptors for CMTs:** Applying the transaction attributes.
- ❑ **Other CMT Issues:** The container's responsibilities.
- ❑ **Bean-Managed Transactions (BMT):** The bean creates and manages transactions.
- ❑ **The Benefits of BMT:** Allows for more complicated transactions.
- ❑ **The Java Transaction API:** The *javax.transaction.UserTransaction* interface.
- ❑ **Transaction Status for BMTs:** Determining the status of a transaction.
- ❑ **Summary**

Module 7: MessageDriven Beans

- ❑ **Overview of JMS:** An overview of the Java Message Service.
- ❑ **Topics vs. Queues:** Understanding point-to-point and publish/subscribe domains.
- ❑ **Using the JMS:** An overview of connections, sessions, producers, consumers, listeners and selectors.
- ❑ **Messages:** The different types of messages in JMS.
- ❑ **MessageDriven Beans:** An overview of message-driven beans.
- ❑ **A MessageDriven Bean Example:** Developing message-driven beans.
- ❑ **Summary**

Module 8: Exception Handling

- ❑ **EJB Exceptions:** An overview of exception handling within an EJB application.
- ❑ **Application Exceptions:** Exceptions designed for the client to handle.
- ❑ **System Exceptions:** Exceptions designed for the container to handle.
- ❑ **The Client's View:** How the client handles exceptions.
- ❑ **Summary**

Module 9: Enterprise Bean Environment

- ❑ **The Enterprise Bean Environment:** An overview of the goals of the bean environment.
- ❑ **Defining Environment Variables:** The *<env-entry>* tag for the deployment descriptor.
- ❑ **Locating Bean Environment Variables:** Using the JNDI API to locate environment variables.
- ❑ **Resource References:** Obtaining a reference to server resources.
- ❑ **Bean References:** Creating a reference to the home of another bean.
- ❑ **Summary**

Module 10: Security

- ❑ **Overview of EJB Security:** An overview of the security management in the EJB architecture.
- ❑ **The Security Context:** Determining if a method is accessible to the client.
- ❑ **Defining Security Role References:** The `<security-role-ref>` tag.
- ❑ **Defining Security Roles:** Creating logical groups of roles.
- ❑ **Defining Method Permissions:** The `<method-permission>` tag.
- ❑ **The Client Application:** Authenticating the user.
- ❑ **Summary**

Java: Under the Hoodä

Enterprise JavaBeans Development

Weekly Schedule

The following is a tentative schedule for the pacing of the course. The actual flow of the course may vary.

Day One

Module 1: An Overview of Enterprise JavaBeans

Module 2: The Enterprise JavaBeans Architecture

Module 3: An Introduction to Enterprise JavaBeans

Day Two

Module 4: Session Beans

Day Three

Module 5: Entity Beans

Day Four

Module 6: Transactions

Module 7: Message-driven Beans

Day Five

Module 8: Exception Handling

Module 9: Enterprise Bean Environment

Module 10: Security