

Web Application Development Using JEE, Enterprise JavaBeans and JPA

Duration: 5 days

Price: £995

Discounts: For multiple course purchases, please [contact us](#) for applicable discounts.

Delivery Options: Perform training at your own pace via our [on-demand training](#) option or attend regularly scheduled live courses via [remote-live attendance](#).

Students Will Learn

- The features and benefits of the EJB 3 Architecture and API
- How to use EJB 3 Annotations
- Creating, deploying and using Stateful and Stateless session beans
- Using CDI (Contexts and Dependency Injection) to initialize resources
- Using Interceptors (Lifecycle and Business Method)
- JNDI (Java Naming and Directory Interface)
- Architecture and coding of EJB clients
- Deploying and using Message-Driven beans
- Distributed transactions, the Java Transaction API, and the EJB transaction model
- Practical architectural issues associated with EJB applications
- The Java Persistence API (JPA)
- The EJB security model
- Creating, deploying and using JPA persistent entities
- Mapping relational schemas to persistent entities
- Using the EntityManager
- Java Persistence Query Language (JPQL) syntax
- Using optimistic locking and versioning
- Advanced JPA capabilities
- Best practices associated with JPA applications
- AJAX integration with Java
- Web Services integration
- Using JAX-WS
- Using SOAP/Http
- Using REST-Based Web Services

Course Description

This hands-on course provides participants with the knowledge and experience required to

develop and deploy Enterprise JavaBeans, Web Services and robust JEE (Java Enterprise Edition) web applications. The Enterprise JavaBeans 3 specification is a deep overhaul of the EJB specification that improved the EJB architecture by reducing its complexity from the developer's point of view. It leverages annotations and Object-Relational Mapping (ORM) technologies to eliminate the dependence on complex EJB APIs, allow POJO (Plain Old Java Object) based development, and provide an effective technology for creating distributed, transactional components including mapping relational data to an object schema. The EJB 3.2 release is part of the JEE 7 specification and adds additional refinements and capability.

This course covers architectural design issues as well as specific coding models for EJB3 components. It starts with the basic concepts and APIs of EJB and then continues on with complex topics such as message driven beans and transactions. Newer concepts such as the use of annotations and the use of CDI / Dependency Injection to initialize references are covered in depth. The course also includes thorough coverage of managing persistence using the Java Persistence API 2 (JPA2). Security, transaction management, inter-component communication and deployment issues are discussed in detail, with hands-on labs to solidify understanding. The course includes content on how to expose EJBs as standards-based (JAX-WS, SOAP/Http) and REST-based web services.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency with this complex technology. Students will come away with an understanding of the role and architecture of the most important containers in the Java EE specification: the Web Container (which hosts HTML pages, Servlets and JSP Pages) and the EJB Container (which hosts Enterprise JavaBean components).

Course Prerequisites

Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML is helpful but not required. HOTT's course [Java Programming](#) or equivalent knowledge provides a solid foundation.

Course Overview

Introduction

- Overview of EJB and Java Persistence API (JPA)
 - Role of EJB in the Java EE Architecture
 - Role of Containers
- EJB 3.2 Overview
 - Session Beans
 - Persistent Entities
 - Message Driven Beans
 - Entity Beans
 - Server Setup and Introduction
 - Annotations and EJB 3.2
 - Annotation Syntax
 - Annotation Placement

Session Beans

- Session Bean Overview
 - Services Provided
 - Stateless and Stateful Beans
 - Defining a Session Bean - EJB 3 Annotations
 - Bean Implementation
 - Remote and Local Business Interface
- Packaging and Deployment
 - ejb-jar file
 - Deployment Descriptors in EJB 3
 - EAR file
 - JEE 6/JEE 7 WAR Files and EJB
 - The EJB Container
 - Creating a Stateless Session

- Bean
- JNDI Overview
 - Distributed Naming and Lookup
 - Context and InitialContext
 - Using JNDI
- Writing an EJB 3 Client
 - Client View of a Session Bean
 - JEE 6 Portable EJB Lookup Names
 - Running the Client
 - Creating an EJB Client

Additional Capabilities

- Resources and Dependency Injection
 - EJB referencing another EJB
 - Injecting with CDI/@Inject and with @EJB
 - CDI Qualifiers and Producers
 - Referencing Resources
 - Environment Entries
 - Connection Factories (DataSource and others)
 - Using Dependency Injection
 - Creating and Using Environment Entries
- Session Bean Lifecycle and Interceptors
 - Stateless Session Bean Lifecycle
 - Business Method Interceptors
 - InvocationContext
 - Lifecycle Callback Interceptors
 - Interceptor Classes
 - Working with Interceptors
- Asynchronous Methods and Singleton Session Beans
- Stateful Session Beans
 - Overview
 - Defining Client Relationship
 - Lifecycle, Activation and Passivation
 - Implementing Stateful Session Beans
- Timer Service
 - Programmatic Timers
 - Calendar-Based Timers
 - Working with Timers

Transactions and Security

- Overview of Transactions and Transactional Systems
 - ACID, Transaction Managers,

Message-Driven Beans

- Overview of Messaging Systems
 - Loose Coupling
 - Pub/Sub, Point2Point
- Overview of JMS API
 - JMS Architecture
 - ConnectionFactory and Destination
 - JMS Producer and Consumer Client Example
 - Using JMS Messages
- Message-Driven Beans (MDB)
 - Features and Benefits
 - @MessageDriven and MDB Architecture
 - Configuring with activationConfig
 - State Diagram and Interceptors
 - Working with Message Driven Beans

Exception Handling

- Exceptions Overview
 - Checked and Unchecked Exceptions

- Resource Managers
- Transactions in EJB 3
 - Declarative Transaction Management
 - Transaction Attributes and Transactional Scope
 - Transaction Scenarios and Bean-Managed Tx
 - Working with Transactions
- Security
 - Java EE Security Overview (Role-based)
 - @RolesAllowed, @PermitAll, Role "***"
 - Programmatic Security
 - Integrating EJB and Java EE Security
- Exceptions in EJB 3
 - Application Exceptions, System Exceptions
- EJB 3 Best Practices
 - When to Use Coarse-Grained Business Interfaces
 - Session Façade
 - Transaction Guidelines
 - Clustering

Introduction to Java Persistence API V2 (JPA2)

- Overview
 - Persistence Layers
 - Object-Relational Mapping (ORM)
 - JDBC Overview
 - JPA Overview
- Mapping with JPA
 - Entities and @Entity, IDs and @Id,
 - Generated ID Values
 - Basic Mapping Types
 - Mapping an Entity Class
- EntityManager and Persistence Context
 - Persistence Unit and persistence.xml
 - Persisting to the DB, the EntityManager
 - Injecting an EntityManager
 - Retrieving Persistent Entities
 - Using the EntityManager to Persist and Find an Entity
- Exploiting Mappings
 - Refining your Mappings

Entity Relationships

- Relationships Overview
 - Object Relationships
 - Participants
 - Roles
 - Directionality
 - Cardinality

Updates and Queries

- Inserting and Updating Data
 - Transient, Persistent, Detached, Removed
 - Persisting New Entities, Updating a Persistent Instance
 - Inserting and Updating an Entity
- Querying and JPQL
 - Fundamental Query Syntax
 - Object Based Queries
 - SELECT Statements
 - WHERE Clause
 - Named Queries
 - Creating and Using JPQL Queries
- Criteria API
 - Criteria Querying
- The Persistence Lifecycle
 - JPA Entity States
 - Lifecycle and Persistence Context
- Versioning and Optimistic Locking
 - Locking Concepts
 - Detached Entities
 - Creating Versioned Entities
 - Optimistic Locking

Additional JPA Capabilities

- Advanced Queries
 - Projection
 - Aggregation
 - Bulk Update/Delete
- Extended Persistence Contexts
- XML Mapping Files

Relationship Mapping

- Mapping Overview (1-1, 1-N, N-1, N-N)
- Unidirectional and Bidirectional
- Mapping One-One, One-Many
- Join Columns
- Relationship Inverses
- Mapping Entity Relationships
- Many-Many Relationships
- Lazy and Eager Loading
- Cascading
- Querying Across Relationships
 - Inner Joins, Outer Joins, Fetch Joins
- Inheritance Mapping
 - Entity Inheritance
 - Single Table
 - Joined (Table per Subclass)
 - Table per Concrete Class
 - Pros and Cons of Inheritance Mapping
- Other Mapping Capabilities
 - Embedded Objects
 - Compound Primary Keys
 - Element Collections

EJB and Java SE

- Best Practices
 - Primary Keys
 - Named Queries
 - Lazy/Eager Loading
 - Transactional Semantics
 - Encapsulation
 - Report Queries

Exposing EJBs as Web Services

- Overview of Web Services
- Advantages of Web Services
- Web Services Technologies: SOAP and WSDL
- Web Services in Java EE and Java SE
- SOAP/Http Web Services
- Working with WSDL Files
- Writing a JAX-WS Web Service
 - The Service Implementation Bean
 - JAX-WS Deployment Descriptors
 - JAX-WS Client-Side Programming
 - JAX-WS Tools
- RESTful Web Services

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Hands On Technical Training
The Best Way to Transfer Technology Skills

14 Fletcher Street
Chelmsford, MA 01824
United States

www.traininghott.co.uk

Copyright© 2019 Hands On Technical Training