

## **EJB 3.2 and JPA2**

### **Course Summary**

#### **Description**

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, and for mapping relational data to an object schema. The EJB 3.2 release is part of the Java EE 7 specification and adds additional refinements and capability.

This course provides thorough coverage of the EJB3 technology - presented in a clear and effective manner. It starts with the basic concepts and APIs of EJB and then continues on with complex topics such as message driven beans and transactions. New 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). This course provides additional coverage of JPA as compared to our Fast Track to EJB3 course.

The course uses hands-on labs and a well-paced approach to make this complex technology understandable in an accelerated fashion. You will come away with a comprehensive understanding of EJB and the important issues that need to be considered to use it in real world applications. This course can be customized to your requirements

#### **Objectives**

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

- Understand the EJB 3 architecture and API, and how it fits into the overall Java EE architecture
- Understand and use the EJB 3 annotations
- Create, deploy & use stateful & stateless session beans
- Use CDI (Contexts and Dependency Injection) to initialize resources
- Understand and use Interceptors (Lifecycle and Business Method)
- Use JNDI (Java Naming and Directory Interface) and be familiar with current portable naming conventions
- Write EJB clients (remote and local)
- Understand, deploy and use message-driven beans
- Understand distributed transactions, the Java Transaction API, and the EJB transaction model
- Understand and use the EJB security model
- Understand practical architectural issues associated with EJB applications
- Understand the new Java Persistence API (JPA)
- Create deploy and use JPA persistent Entities
- Map relational schemas to persistent entities, including the use of primary keys
- Understand and use the EntityManager
- Understand and use Java Persistence Query Language
- Use optimistic locking and versioning
- Use advanced JPA capabilities such as entity relationships (1-1, 1-N, N-N, unidirectional, bidirectional), inheritance, and embeddable classes
- Learn best practices associated with JPA applications

## EJB 3.2 and JPA2

### Course Summary (cont'd)

#### Topics

- Introduction
- Session Beans
- Additional Capabilities
- Message-Driven Beans
- Transactions and Security
- Exceptions
- Introduction to Java Persistence API V2 (JPA2 )
- Updates and Queries
- Entity Relationships
- Additional JPA Capabilities

#### Audience

This course is designed for developers.

#### Prerequisites

There are no prerequisites for this course.

#### Duration

Five days

## EJB 3.2 and JPA2

### Course Outline

- I. Introduction**
  - A. Overview of EJB and Java Persistence API (JPA)
    - 1. Goals of EJB, EJB in the Java EE architecture
  - B. EJB 3.2 Overview
    - 1. Session Beans, Persistent Entities, Message Driven Beans
  - C. Lab: Server setup and introduction
  - 2. Business Method Interceptors, InvocationContext, Lifecycle Callback Interceptors, Interceptor Classes
  - E. Lab: Working with Interceptors
  - F. Asynchronous Methods, Singleton Session Beans
  - G. Stateful Session Beans
    - 1. Overview
    - 2. Defining, Client Relationship
    - 3. Lifecycle, Activation, Passivation
  - H. Lab: Stateful Session Beans
  - I. Timer Service
    - 1. Programmatic Timers
    - 2. Calendar-Based Timers
  - J. Lab: Working with Timers
- II. Session Beans**
  - A. Session Bean Overview
    - 1. Services provided, Stateless and Stateful Beans
    - 2. Defining a Session Bean - EJB 3 Annotations
    - 3. The Bean Implementation, Remote and Local Business Interface
  - B. Packaging and Deployment
    - 1. ejb-jar file, Deployment Descriptors in EJB 3
    - 2. EAR file
    - 3. JEE 6+ WAR files and EJB
    - 4. The EJB Container
  - C. Lab: Creating a Stateless Session Bean
  - D. JNDI Overview
    - 1. Distributed Naming and Lookup
    - 2. Context and InitialContext
    - 3. Using JNDI
  - E. Writing an EJB 3 Client
    - 1. Client View of a Session Bean
    - 2. JEE 6 Portable EJB Lookup Names
    - 3. Running the Client
  - F. Lab: Creating an EJB Client
- III. Additional Capabilities**
  - A. Resources and Dependency Injection
    - 1. EJB referencing another EJB
    - 2. Injecting with CDI/@Inject and with @EJB
    - 3. CDI Qualifiers and Producers
    - 4. Referencing Resources, Environment Entries
    - 5. Connection Factories (DataSource and others)
  - B. Lab: Using Dependency Injection
  - C. Lab: Creating and Using Environment Entries
  - D. Session Bean Lifecycle and Interceptors
    - 1. Stateless Session Bean Lifecycle
- IV. Message-Driven Beans**
  - A. Overview of Messaging Systems
    - 1. Messaging, Loose Coupling
    - 2. Pub/Sub, Point2Point
  - B. Overview of JMS API
    - 1. Overview and Structure
    - 2. ConnectionFactory and Destination
    - 3. JMS Producer and Consumer Client example
    - 4. JMS Messages
  - C. Message-Driven Beans (MDB)
    - 1. Overview and Goals
    - 2. @MessageDriven and MDB example
    - 3. Configuring with activationConfig
    - 4. State Diagram and Interceptors
  - D. Lab: Message Driven Beans
- V. Transactions and Security**
  - A. Overview of Transactions and Transactional Systems
    - 1. ACID, Transaction Managers, Resource Managers
  - B. Transactions in EJB 3
    - 1. Declarative Transaction Management
    - 2. Transaction Attributes and Transactional Scope
    - 3. Transaction Scenarios and Bean-Managed Tx
  - C. Lab: Working with Transactions
  - D. Security
    - 1. Java EE Security Overview (Role-based)
    - 2. @RolesAllowed, @PermitAll, Role \*\*\*\*
    - 3. Programmatic Security
  - E. Lab: Integrating EJB and Java EE Security

## EJB 3.2 and JPA2

### Course Outline (cont'd)

- VI. Exceptions**
  - A. Exception Overview
    - 1. Checked and Unchecked Exceptions
  - B. Exceptions in EJB 3
    - 1. Application Exceptions, System Exceptions
  - C. EJB 3 Best Practices
    - 1. When to Use, Coarse-Grained Business Interfaces, Session Façade, Transaction Guidelines, Clustering
- VII. Introduction to Java Persistence API V2 (JPA2)**
  - A. Overview
    - 1. Persistence Layers, Object-Relational Mapping (ORM), JDBC
    - 2. JPA Overview
  - B. Mapping with JPA
    - 1. Entities and @Entity, ids and @Id,
    - 2. Generated Id Values
    - 3. Basic Mapping Types
  - C. Lab: Mapping an Entity Class
  - D. EntityManager and Persistence Context
    - 1. Persistence Unit and persistence.xml
    - 2. Persisting to the DB, the EntityManager,
    - 3. Injecting an EntityManager
    - 4. Retrieving Persistent Entities
  - E. Lab: Using the EntityManager to persist and find an Entity
  - F. More About Mappings
  - G. Lab: Refining your Mappings
- VIII. Updates and Queries**
  - A. Inserting and Updating
    - 1. Transient, Persistent, Detached, Removed
    - 2. Persisting new Entities, Updating a Persistent Instance
  - B. Lab: Inserting and Updating an Entity
  - C. Querying and JPQL
    - 1. Object Based Queries, Select statements, WHERE clause
    - 2. Named Queries
  - D. Lab: Creating and Using JPQL Queries
  - E. Criteria API
  - F. Lab: Criteria Querying
- G. The Persistence Lifecycle - JPA Entity States, Lifecycle, and Persistence Context**
- H. Versioning and Optimistic Locking**
  - 1. Overview
  - 2. Detached Entities
  - 3. Creating Versioned Entities
- I. Lab: Optimistic Locking**
- IX. Entity Relationships**
  - A. Relationships Overview
    - 1. Object Relationships, Participants, Roles, Directionality, Cardinality
  - B. Relationship Mapping
    - 1. Mapping Overview (1-1, 1-N, N-1, N-N)
    - 2. Unidirectional and Bidirectional
    - 3. Mapping One-One, One-Many
    - 4. Join Columns
    - 5. Relationship Inverses
  - C. Lab: Mapping Entity Relationships
    - 1. Many-Many Relationships
    - 2. Lazy and Eager Loading
    - 3. Cascading
    - 4. Queries Across Relationships (Inner Joins, Outer Joins, Fetch Joins)
  - D. Lab: Querying Across Relationships
  - E. Inheritance Mapping
    - 1. Entity Inheritance
    - 2. Single Table, Joined (Table per Subclass), Table per Concrete Class
    - 3. Pros and Cons
  - F. Lab: Mapping Inheritance
  - G. Other Mapping Capabilities - Embedded Objects, Compound Primary Keys, and Element Collections
- X. Additional JPA Capabilities**
  - A. Queries - Projection, Aggregate, Bulk Update/Delete
  - B. Extended Persistence Contexts
  - C. XML Mapping Files
  - D. EJB and Java SE
  - E. Best Practices
    - 1. Primary Keys, Named Queries, Lazy/Eager Loading, Transactional Semantics, Encapsulation, Report Queries