

The Enterprise JavaBeans 3.0 specification is a deep overhaul of the EJB specification that is intended to improve the EJB architecture by reducing its complexity from the developer's point of view. It leverages annotations (introduced in Java 5) 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 mapping relational data to an object schema.

This course provides thorough coverage of the EJB3 technology including new concepts such as the use of annotations and the use of Dependency Injection to initialize references. This course also includes in-depth coverage of managing persistence using the Java Persistence API (JPA). It 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.

Course Objectives:

- 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, and use stateful and stateless session beans.
- Use EJB 3 dependency injection to initialize resources
- Understand and use Interceptors.
- Use the Java Naming and Directory Interface (JNDI).
- Write remote and local EJB clients.
- 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 persistence entities.
- Understand and use the Entity Manager.
- Use advanced JPA capabilities such as entity relationships, inheritance, and embeddable classes.

Audience: Java developers who want to use EJB 3.

Prerequisites: One year of Java programming experience (or equivalent) is preferred. Knowledge of relational databases and JDBC is strongly recommended.

Number of Days: 4 days

1. Overview What is EJB? EJB Goals Types of Enterprise JavaBeans	Java Persistence API EJB and Java EE (Enterprise Editions) EJB in Java EE Architecture
--	--

- SOA and EJB
- SOA with Web Services and EJB
- 2. **EJB 3.0**
 - EJB 3.0 Overview
 - EJB 2.X Problems
 - EJB 3.0 Goals
 - Session Bean Usage
 - Persistent Entity Usage
 - MDB Usage
- 3. **Session Bean Overview**
 - What are Session Beans?
 - Stateless Session Beans (SLSB)
 - Stateful Session Beans (SFSB)
 - Session Bean can be Distributed
 - Defining a Session Bean
 - Stateless Session Bean Definition
 - Calculator Remote Business Interface
 - Remote and Local Business Interfaces
 - Calculator Bean Local Business Interface
 - A Brief Note on Annotations
 - How Annotations Work
 - Annotation Definition
 - Using Annotations
 - What Else is Needed?
- 4. **Packaging and Deployment**
 - J2EE Packaging
 - EJB-JAR File
 - Deployment Descriptor (DD)
 - Deployment Descriptors in EJB 3
 - EJB-JAR File Structure
 - Enterprise Archive (EAR File)
 - application.xml File
 - Server Deployment
 - EJB Container
 - The EJB Container
 - Server Deployment
- 5. **JNDI Overview**
 - How do Remote Clients get EJB Access?
 - JNDI – Java Naming and Directory Interface
 - EJB Container Binds a Reference into JNDI
 - Client Looks up Reference in JNDI
 - JNDI Tree Structure
- JNDI API Overview
- JNDI API
- The Context Interface
- The InitialContext Class
- Specifying the InitialContext Properties
- Using JNDI
- 6. **EJB Client**
 - Client View of a Session Bean
 - Client Invocation of a Session EJB
 - PortableRemoteObject and Casting
 - Running a Client
- 7. **Dependency Injection**
 - Dependency Injection
 - The JavaTunes Online Music Store
 - An EJB Referencing Another EJB
 - CatalogMaintenance
 - Injection of CatalogMaintenance
 - What is Happening
- 8. **Deployment Descriptors**
 - More About Deployment Descriptors
 - The XML Deployment Descriptor
- 9. **The EJB Environment**
 - Issues with Obtaining Resources
 - Referencing Resources External to EAR
 - Reference & Deployment Descriptor Usage
 - Resolving a JNDI Name
 - Simple Environment Entries
 - Declaring Simple Environment Entries
 - Setter Injection
 - Resource Manager Connection Factories
 - More on the Enterprise Bean Environment
 - The Environment is a Level of Indirection

Looking up Resources in the Environment

Looking up via EJBContext

Looking up via JNDI API

More on the @Stateless Annotation

More on the @EJB Annotation

More on the @Resource Annotation

Deployment Descriptor vs Annotation

10. **Stateless Session Bean Lifecycle & Interceptors**

Stateless Session Bean State Diagram

Lifecycle of SSB

Client Call of a Stateless SB Method

Interceptors

Business Method Interceptors

InvocationContext Interface Details

Interceptor Method Details

Interceptor Class

Using Interceptor Classes

Method Level Interceptors

Lifecycle Callback Interceptors

Lifecycle Interceptor in the Bean Class

Lifecycle Interceptor in a Separate Class

11. **Stateful Session Beans**

Stateful Session Bean (SFSB) Overview

Coding a Stateful Session Bean

Stateful Session Bean Clients

Stateful Session Bean Removal

Stateful Session Passivation/Activation

When to Use Stateful Session Beans

@PrePassivate and @PostActivate Callbacks

Stateful Session Bean State Diagram

12. **The Timer Service**

Bean Requirements

The javax.ejb.Timer Interface

The javax.ejb.TimerService Interface

How the Timer Works

Issues with Using the Timer Service

13. **Overview of Messaging Systems**

What is Messaging?

Loose Coupling

When is Messaging Used?

Two Messaging Models

Publish/Subscribe – Illustrated

More on Publish/Subscribe

Point-to-Point – Illustrated

More on Point-to-Point (P2P)

Message Delivery – Push versus Pull

14. **Overview of JMS API**

What is Java Message Service?

API Structure

JMS Interfaces

Administered Objects

Administered Objects and JNDI – Illustrated

Client Workflow

Synchronous Queue Consumer Client

Asynchronous Queue Consumer Client

JMS Message Types

Message Header Fields

15. **Message-Driven Beans**

J2EE Message Producers and Consumers

Message-Driven Bean (MDB) Overview

Goals of Message-Driven Beans

MDB Consumption of a Message

@MessageDriven Details

Activation Configuration Properties

Specifying a Destination for an MDB

Specifying a Destination Using a DD

16. **Message-Driven Bean Lifecycle**

Lifecycle Overview

MDB State Diagram

Interceptor Methods

17. **Transaction Definition**

Transaction Overview

Transaction Lifecycle

Transactions Clarify Systems

18. **Transactional System Overview**

Overview of a Transactional System

Transactional System

Components

- Transactional Object
- EJB Transaction Support
- 19. Transactions in EJB**
- EJB Declarative Transaction Management
- Transactional Scope
- EJB Transaction Attributes
- Specifying Transaction Attributes
- NOTSUPPORTED
- SUPPORTS
- REQUIRED
- REQUIRESNEW
- MANDATORY
- NEVER
- Beans Have a Say in Transactions
- Beans can be Notified of Transaction Status
- Transaction Attributes – Some Choices
- Explicit/Bean-Managed Transactions
- Transaction Isolation Levels
- Multi-Process TX and Two Phase Commit
- 20. Security in EJB**
- Security Requirements
- J2EE Security
- Roles
- J2EE Security Overview
- EJB Security Overview
- Authentication
- Programmatic Security
- Transport Level Security with SSL
- 21. Exception Handling**
- Overview of Exceptions
- Exception Hierarchy
 - Application Exceptions in EJB
- Defining Application Exceptions
- Container Handling of Application Exception
- Bean Throwing of Application Exception
- Client Handling of Application Exceptions
- System Exceptions Indicate Failure
- Container Handling of System Exception
- Client Handling of System Exceptions
- 22. EJB 3 Best Practices**
- When to Use EJB
- Keep Business Interfaces Coarse Grained
- Session Façade Structure
- Use Container-Managed Transactions
- Transaction Duration
- Local and Remote Business Interface
- Tuning
- Session Bean Tuning
- Clustering
- Clustering Session Beans
- 23. JPA Overview**
- The Issues with Persistence Layers
- Object-Relational Mapping (ORM) Issues
- Issues with JDBC Alone
- Java Persistence API Overview
- JPA Benefits
- Java Persistence Environments
- 24. Mapping a Simple Class**
- Entity Classes
- Entity Class Requirements
- The Entity Declaration
- The Event Class
- The Id Property
- Field Access or Property Access
- The EVENTS Table
- Generated Id Property
- Mapping Properties
- Basic Mapping Types
- Persisting to the Database
- 25. Entity Manager and Persistence Context**
- The Entity Manager & Persistence Context
- The EntityManager
- The EntityManager Interface
- Persistence Unit
- Injecting an EntityManager
- Retrieving Persistent Objects
- Container-Managed Entity Manager
- The Persistence Unit

- persistence.xml
- 26. **Inserting and Updating**
 - Transient, Persistent, Detached Instances
 - Removed Instances
 - Persisting a New Entity
 - Synchronization to the Database
 - Updating a Persistent Instance
 - Detached Entities
- 27. **Querying and Java Persistence Query Language (JPQL)**
 - Java Persistence Query Language
 - JPQL Basics
 - Executing a Query
 - Where Clause
 - Query Parameters
 - Using Query Parameters
 - Named Queries
- 28. **Versioning/Optimistic Locking**
 - Optimistic Locking
 - Using a Detached Instance
 - Versioning
 - Version Property in Java Class
 - Locking Objects
 - Lock Modes
- 29. **Entity Relationships**
 - Relationships Overview
 - Object Relationships
 - Characteristics of Relationships
 - Directionality
 - Characteristics of Relationships
- 30. **Mapping Relationships**
 - Mappings Overview
 - Unidirectional Many-To-One Relationship
 - The Table Structure – Many-To-One
 - The Owning Side
 - @JoinColumn
 - Bidirectional One-To-One Relationship
 - Using the Relationship
 - More on the Inverse Side
 - Bidirectional One-To-Many Relationship
 - Mapping the One-To-Many Relationship
 - Other Collection Types
 - Many-To-Many Relationship
 - Defining Many-To-Many Relationship
 - Mapping Many-To-Many Relationships
 - Specifying the Join Table
 - Lazy and Eager Loading
 - Cascading Operations
 - The Cascade Element
 - Queries Across Relationships
 - OUTER and FETCH JOIN
- 31. **Mapping Inheritance**
 - Entity Inheritance
 - Details of Entity Inheritance
 - Single-Table Strategy
 - Entity Definitions for Single-Table
 - Single Table: Pros and Cons
 - Joined (Table per Subclass)
 - Entity Definitions for Joined
 - Joined: Pros and Cons
 - Table per Concrete Class
- 32. **Embedded Objects**
 - Using Embedded Objects
 - Embeddable Class
 - Reusing Embeddable Classes
 - Overriding Embedded Class
 - Attributes
 - Compound Primary Keys
 - Compound Key with Embedded Id Class
 - Using an Embedded Id Class
- 33. **Additional Java Persistence Capabilities**
 - More on Querying
 - Projection Queries
 - Aggregate Queries
 - Bulk Update and Delete
 - Native SQL Queries
- 34. **Extended Persistence Contexts**
 - Stateful Session Beans with Entity State
 - Extended Persistence context
 - Issues with Extended Persistence Context
- 35. **XML Mapping Files**

- XML Mapping Files
- A Simple Entity Class
- JPA XML Mapping File
- JPA XML Mapping File – Mapping Entities
- JPA XML Mapping File – Named Queries

36. Java Persistence with Java SE

- Using JPA with Java SE
- Java SE APIs

37. Java Persistence Best Practices

- Primary Key Considerations
- Use Named Queries
- Use Lazy/Eager Loading Appropriately
- Be Aware of Transaction Semantics
- Encapsulate JPA Code
- Use Report Queries Where Applicable
- Optimize Read-Only/Mostly Data Access
- Paging Data
- Consider Going Outside of Java Persistence
- Know Your Provider Implementation