

Java EE, Part 4 of 8: EJB 3.0 in the Enterprise

page 1

Meet the expert: Susan Bryant is an experienced IT trainer and consultant with a broad array of skills. She has over 15 years experience in information systems with roles including systems consulting, project management, staff management, staff mentoring, and certified technical trainer. Susan has a strong technical knowledge of IBM WebSphere Application Server, WebSphere Portal Server, WebSphere Process Server, Lotus Domino, and web application development technologies including, Struts, JSF, EJBs and AJAX.

Prerequisites: This course is intended for intermediate to advanced Java programmers. System architects will find the advanced concepts especially beneficial in designing a framework. Students should have good Java programming knowledge and should be familiar with web application programming, including servlets and JSPs.

Runtime: 08:57:39

Course description: This course takes you beyond the fundamentals of Enterprise Java Bean (EJB 3.0) programming using Eclipse. In this course, you will learn about building message-driven Beans, EJB Security, utilizing Web Services technologies like JAX-WS, and managing distributed transactions. You will also be exposed to Java Interceptors, EJB Timer service, Hibernate JPA caching, EJB design patterns, and deployment.

Course outline:

JPAQL

- Introduction
- JPAQL
- Interface
- Refactor
- Local Interface
- Summary

Intro to Relationships

- Introduction
- Relationship Between Entities
- Anatomy of a Relationship
- Foreign Key
- Example Schema
- One-to-One Unidirectional
- Creating Entity Instances
- Traversing the Relationship
- The Cascade Behavior
- One-to-One Bidirectional
- Traversing the Relationship
- Demo: Intro to Relationships
- Summary

Complex Relationships

- Introduction
- One-to-Many and Many-to-One
- Many-to-One Unidirectional
- Creating Entity Instances
- Traversing the Relationship
- Modeling One-to-Many
- Creating Entity Instances

- Traversing the Relationship
- Many-to-Many
- Modeling Many-to-Many
- Creating Entity Instances
- Traversing Relationship
- Unidirectional One-to-Many
- Unidirectional One-to-Many
- Creating Entity Instances
- Traversing the Relationship
- Demo: Complex Relationships
- Summary

Relationships and JPAQL

- Introduction
- Relationships in JPAQL
- Example: Relationship
- Fetching Optimizations
- Lazy vs. Eager Initialization
- Lazy Initialization
- Lazy Initialization Problems
- Fetch Join Query
- Demo: Relationships and JPAQL
- Summary

Entity and Inheritance

- Introduction
- Inheritance
- Inheritance Example
- Inheritance and Entities
- Inheritance Strategies
- Single Table Per Class

- Single Table Pros and Cons
- Demo: Entity and Inheritance
- Summary

More Inheritance Strategies

- Introduction
- Table Per Concrete Class
- JBoss Warning
- Table Structure
- Pros and Cons
- Joined Subclasses
- Table Structure
- Pros and Cons
- Which Approach to Use
- Demo: Inheritance Strategies
- Summary

MDB Intro

- Introduction
- Trouble with RMI/IIOP
- Messaging to the Rescue
- Messaging Features
- Message Domains
- Publish/Subscribe
- Point-to-Point
- Java Message Service
- Overview
- JMS Interfaces
- Integrating JMS and EJB
- MDBs are Different
- MDBs Cannot Respond

- MDBs are Stateless
- Demo: MDBs
- Summary

MDB API

- Introduction
- Message-Driven Bean Interfaces
- javax.jms.MessageListener
- javax.jms.Message
- Lifecycle
- Example: Message Drive Bean
- MDB - Client Example
- Transactions
- Security
- Load Balancing
- Clustering and Topics
- Clustering and Queues
- A Few Tips
- Building a Response
- Potential Problems
- A Simple Alternative
- Type Checking and Messages
- Testing Message-Driven Beans
- Demo: MDB API
- Summary

MDB Advanced Topics

- Introduction
- Message-Oriented Middleware
- Durable Subscription
- javax.jms.Message

(Continued on page 2)

Java EE, Part 4 of 8: EJB 3.0 in the Enterprise

page 2

- Specialized Message Types
- Poison Messages
- Avoiding Poison Messages
- Summary

EJB Security

- Introduction
- Introduction
- How EJB Security Works
- Protecting Web Resources
- Setting Method Permission
- Defining Roles
- Defining Roles: Annotations
- Specify Methods Permission
- Disable Security Check
- Excludes List
- RunAs Security Identity
- RunAs: Configuration
- Programmatic EJB Security
- Security Role Reference
- Demo: EJB Security
- Summary

Web Services and EJBs

- Introduction
- Web Services
- Operation and Implementation
- WSDL
- Typical Development Workflow
- Advantages of Web Services
- Web Service Clients
- JAX-WS Overview
- JAX-WS Framework
- Java to XML Data Conversion
- Main Goals of JAX-WS
- JAX-WS Server
- WSDL Overview
- WSDL Document Tags
- Web Services and EJB
- Web Service EJBs
- Annotate the Session Bean
- Annotate the Methods
- Service Endpoint Interface
- Package and Deploy
- Demo: Web Services and EJBs
- Summary

Timer Service

- Introduction
- Container Managed Timer Service
- EJB Timer Service
- Timer Service and EJB
- Timer Service API
- The TimerService Interface
- Obtaining the Timer Service

- Creating a Timer
- Getting All Timers
- Timeout Callback Method
- Timer Interface
- Example: TimerTestBean
- Timer and Transaction
- Limitations of EJB Timers
- Demo: Timer Service
- Summary

Transactions

- Introduction
- Need for Transactions
- Transactions
- ACID Properties
- Transaction Components
- Distributed Transactions
- Two Phase Commit
- Java Transaction API
- Object Transaction
- EJB Transaction Basics
- Transaction Propagation
- Transaction Outcome
- Container Managed Transaction
- Container MT Settings
- Interacting with Container MTs
- Example: Container MT
- Transaction Attributes Support
- Bean Managed Transaction
- Client Managed Transaction
- Transaction Isolation
- Isolation Level
- Demo: Transactions
- Summary

Interceptors

- Introduction
- Introduction
- Interceptors
- Internal Interceptor
- Example: Internal Interceptor
- External Interceptors
- Example: Default Interceptor
- Another Example
- Class-Level Interceptor
- Pre & Post Processing
- Excluding Interceptors
- Interceptors on Lifecycle Call
- Blocking Calls
- Demo: Interceptors
- Summary

EJB3 Design Patterns

- Introduction
- Introduction

- Message Facade Pattern
- Message Facade Using JMS
- Message Facade Using Timer
- Notes About Message Facade
- The Command Pattern
- Implementing Command Pattern
- Example Command Pattern
- Summary

Design Patterns and EJB Data

- Introduction
- Session Facade Pattern
- JPA and Session Facade Pattern
- Facade Example
- Data Transfer Object Pattern
- JPA Entity as DTO
- Problem with JPA Entity as DTO
- Version Number Pattern
- JPA and Version Number Pattern
- Primary Key Generation
- JPA and Primary Key Generation
- Fast Lane Pattern
- JPA and Fast Lane Pattern
- Summary