

This 3-day course covers everything you need to know to begin working with the Java Persistence API in a very short time. It covers all the important concepts necessary to access and update data stored in relational databases. As part of the complete overhaul of the Enterprise JavaBeans (EJB) specification, database persistence was broken out into a completely separate specification, the Java Persistence API (JPA). JPA replaces entity beans with powerful new Object-Relational Mapping (ORM) capabilities based on proven technologies such as Toplink and Hibernate. This course includes all important features from JPA 2, and is also suitable for users of the 1.0 release.

JPA supports a POJO (Plain Old Java Object) based model using annotations which lets you develop persistent classes following common Java idioms. It supports entity relationships, inheritance, polymorphism, composition, and much more. The Java Persistence Query Language (JPQL), which is based on SQL but operates on the object model, provides a powerful bridge between the object and relational worlds. JPA also allows you to express queries using native SQL, including the capability to map the SQL query results to persistent entities.

Course Objectives:

- Understand the JPA architecture
- Create JPA based applications
- Understand and use JPA to map persistent objects to the database
- Use the JPA EntityManager
- Understand and work with collections & associations
- Use versioning support for optimistic locking
- Map inheritance hierarchies using JPA
- Performance tune your JPA applications
- Understand the relationship between JPA /EJB3

Audience: Java developers who need to develop their understanding of JPA.

Prerequisites: Proficiency in Java and object-oriented programming, knowledge in relational databases, and some familiarity with Spring® is recommended.

Number of Days: 3 days

1 Introduction to Java Persistence API (JPA)

Overview

Persistence Layers, Object-Relational Mapping (ORM), JDBC

JPA Overview

Mapping with JPA

Entities and @Entity, ids and @Id,

Generated Id Values

Basic Mapping Types

Persistence Unit and EntityManager

Persisting to the DB, the EntityManager API

Persistence Units, Config, Persistence Context

Retrieving Persistent Entities with find()

More About Mappings

Default Mappings, @Basic, @Column

Field vs. Property Access

Temporal (Date/Time) Mappings

Logging Options (Provider based)

<p>2 Updates and Queries Inserting and Updating - Persisting new Entities, Updating an Instance, Removing an Instance Querying and JPQL Entity Based Queries, SELECT ,WHERE Query Interface, Executing Queries, Generic Queries (JPA 2) JPQL Operators, Expressions, and Parameters Named Queries Additional Query Capabilities Projection query, Ordering, Aggregate Query, Build Update and Delete Embedded Objects @Embeddable, @Embedded Defining and using Embedded Objects Compound Primary Keys - @EmbeddedID, @IDClass, Defining Compound Keys</p> <p>3 The Persistence Lifecycle Transaction Overview and Transactions in JPA Transaction Overview EntityTransaction API (including JTA and resource-local EntityManager) The Persistence Lifecycle JPA Entity States (New, Managed, Detached, Removed), and Entity State Diagram Persistence Context - Lifespan, Propagation Synchronization to the DB Versioning and Optimistic Locking Overview, Detached Instances Versioning, @Version, Optimistic Locking Lifecycle Callbacks @PrePersist, @PostPersist, etc. Entity Listeners, @EntityListeners</p> <p>4 Entity Relationships Relationships Overview: Object Relationships, Participants, Roles, Directionality, Cardinality</p>	<p>Relationship Mapping Unidirectional and Bidirectional @ManyToOne, @OneToMany, Table Structures Relationship Inverse - Owning Side Collection Types (List, Set, etc) Cascading Over Relationships (including orphanRemoval - JPA 2) @ManyToMany, @OneToOne Lazy and Eager Loading Queries Across Relationships (Inner Joins, Outer Joins, Fetch Joins) Entity Inheritance Mapping Single Table Mapping Joined (Table per Subclass) Mapping Table per Concrete Class Mapping Pros and Cons Element Collections (JPA 2) Using Element Collections Collections of Embeddables</p> <p>5 The Criteria API (JPA 2) Overview of the Criteria API Path Expressions, Building Queries (CriteriaBuilder, CriteriaQuery, Subquery, Predicate, Expression, Order, Selection, Join) Executing Queries and Accessing results</p> <p>6 Additional JPA Capabilities XML Mapping files Bean Validation (JPA 2) Best Practices</p> <p>7 Integration Data Access Objects (DAO) and Java SE Integration (Optional) DAO Overview JpaUtil Class for EntityManager management in Java SE Lifecycle Considerations Integration with EJB (Optional) Using JPA with Session Beans Container Managed (Injected) Entity Manger JTA Transactions and Lifecycle Considerations Extended Persistence Contexts Using JPA with Java Web Apps</p>
--	--

Using EntityManager in Web apps -
request scoping
Lazy Loading - Open EntityManager in
View Pattern
Integration with Spring (Optional)
Injection of EntityManager,
EntityManagerFactory
LocalEntityManagerFactoryBean
JPA/Spring Based DAO