

Object Persistence Using Hibernate and the JPA

Duration:	4 days
Type:	intermediate

Description

This course introduces object persistence using the Hibernate framework. When combined with Java 5 and EJB3 there are three ways of developing with Hibernate, firstly using the Hibernate with XML mapping files, secondly using Hibernate with JPA annotations and finally using Hibernate entirely within the confines of the JPA.

All three types of development are covered in this course, but the focus is kept on the traditional coding model. The course introduces all features of Hibernate 3.2, including managing relationships between objects, integration into JEE applications and performance tuning.

Prerequisites

Delegates must be experienced Java developers who are familiar with the JDBC API and SQL

List of Modules

The Evolution of ORM Frameworks

- Problems caused by the Object Relational Mismatch
- Key benefits of automating data access using Hibernate
- Comparing Hibernate to traditional data access methods
- The evolution of EJB3 and the Java Persistence API (JPA)

Creating Conventional Hibernate Based Projects

- Adding Hibernate and its dependencies
- Creating the *hibernate.cfg.xml* configuration file
- Choosing the driver, dialect and logging settings
- Installing the Hibernate Eclipse Plug-Ins and Ant tasks

Specifying Mapping Information

- Creating and organizing XML mapping files for classes
- Specifying mapping information using JPA annotations
- Associating classes with tables and fields with columns
- Choosing an appropriate strategy for generating object ids
- Mapping basic (many-to-one) relations between objects

Development Using the Hibernate API

- Creating *SessionFactory* and *Session* objects
- Starting and committing transactions
- Initializing and saving persistent objects
- Retrieving a persistent object via its key
- Navigating associations between persistent objects
- Finding groups of objects via HQL queries
- Using native SQL queries instead of HQL

Development Using the JPA EntityManager

- Acquiring an entity manager in JSE and JEE applications
- Entity managers, transactions and persistence contexts
- Persisting, loading, removing, detaching and merging objects
- Finding objects using the Java Persistence Query Language (JPQL)

Designing a Persistence Layer Part 1

- Differentiating between application and database transactions
- How objects are categorized as transient, persistent or detached
- Defining how Hibernate treats associations via the *cascade* attribute
- Deciding if, when and how to store sessions within a Web Application

Designing a Persistence Layer Part 2

- Producing a fine grained object model using value types and components
- Specifying and working with Sets, Lists and Maps of value types
- Modeling one-to-many and many-to-many associations
- Breaking down many-to-many relationships
- Different ways of modeling inheritance

Tuning Hibernate for Performance

- Deciding whether to load objects eagerly or lazily
- Choosing the correct level of transaction isolation
- Controlling the level 1 cache by evicting objects
- Picking and configuring a level 2 cache implementation
- Special considerations that apply to the query cache