Hibernate

Course Summary

Description

Hibernate is an open source object/relational (OR) persistence and query service for Java. Hibernate lets you develop persistent classes following common Java idioms - including association, inheritance, polymorphism, composition and the Java collections framework.

The Hibernate Query Language, designed as a minimal object-oriented extension to SQL, provides an elegant bridge between the object and relational worlds. Hibernate also allows you to express queries using native SQL or Java-based Criteria and Example queries.

Hibernate is now the most popular OR mapping solutions for Java, and it has become a de facto standard in Java OR mapping. JBoss has integrated Hibernate into its JEMS (Java Enterprise Middleware System) product line.

The Java Persistence API (EJB 3) specification derives a great deal of its architecture from Hibernate, and the Hibernate annotations are compatible with the Java Persistence annotations. This promises to make Hibernate an even more important technology.

This course covers everything you need to know to begin working with Hibernate in a very short time. It covers all the important concepts necessary to access and update data stored in relational databases. It includes an extensive series of labs to exercise all major capabilities.

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

- Understand the benefits of Hibernate
- Understand the Hibernate architecture
- Create Hibernate based applications
- Understand and use Hibernate mapping to map persistent objects to the database
- Understand and work with collections & associations
  - Value and Entity Types
  - Bidirectional and unidirectional
  - 1-1, 1-N, N-N
- Use Hibernate's versioning support
- Map inheritance hierarchies using Hibernate
- Work with Hibernate queries, HQL, and Criteria
- Performance tune your Hibernate applications
- Understand Hibernate transaction support
- Understand the relationship between Hibernate and the Java Persistence API (JPA)
- Use the JPA annotations to do OR mapping
- Configure second level caching
- Integrate Hibernate with Java Web Apps and EJB
- Integrate Hibernate and Spring

Hibernate

Course Summary (cont'd)

Topics

- Introduction to Hibernate
- Updates and Queries
- The Persistence Lifecycle
- Relationships
- Additional Querying Capabilities
- Hibernate and Java Persistence / EJB 3
- Caching
- Integration Considerations
- Additional Topics
- Hibernate and Spring Integration

Audience

This course is ideal for anyone who wants to learn everything you need to know to begin working with Hibernate in a very short time.

Prerequisites

There are no prerequisites for this course.

Duration

Three to four days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Hibernate

Course Outline

I. Introduction to Hibernate
- A. Issues with Persistence layers and Object-Relational Mapping (ORM)
- B. Hibernate Overview and Benefits
- C. Hibernate architecture overview
- D. Configuring Hibernate
  1. hibernate.cfg.xml file, Connection properties, Database dialect
  2. SessionFactory, Configuration, and Session
- E. Mapping a Class
  1. Persistent Entity Class, Hibernate Mapping File, Mapping the Entity Class
  2. Primary keys: Id property, Generated Id
  3. Hibernate Type System
- F. Working with sessions and Persistent Objects
- G. Logging: hibernate.show_sql, log4j Overview and configuration for Hibernate

II. Updates and Queries
- A. Inserting, Updating, and Deleting Entities
- B. HQL - Hibernate Query Language Overview
- C. The Query Interface
- D. Creating and working with queries
- E. Named Queries, Projection Queries, Aggregate Queries

III. The Persistence Lifecycle
- A. Transaction Overview and Transactions in Hibernate
- B. Hibernate Transaction API (in Managed and Non-managed Environments)
- C. The lifecycle of managed objects
- D. Persistent, transient, and detached objects
- E. The Persistence (Session) Context (Lifespan, Relation to Managed Objects, Propagation)
- F. Contextual Sessions
- G. Synchronization to the Database
- H. The Session as cache
- I. Optimistic Locking / Versioning
  1. Detached Objects and Optimistic Locking
  2. Versioning overview and Using Versioning
  3. Locking Objects

IV. Relationships
- A. Object Relationship Overview
- B. Mapping Collections of Value Objects
- C. Entity Relationships: 1-N, N-1, N-N, 1-1
- D. Mapping Entity Relationships
- E. Uni and Bi-directional Relationships
- F. The Relationship "inverse"
- G. Cascading Over Relationships
- H. Queries Across Relationships (Lazy and Eager)
- I. Inheritance Mapping
  1. Entity Inheritance with Hibernate
  2. Table-per-class mapping
  3. Table per Subclass mapping
  4. Table per Concrete Class mapping

V. Additional Querying Capabilities
- A. Projection Queries, Aggregate queries, Bulk updates and deletes, Native SQL Queries
- B. Query Filters
- C. The Criteria API
  1. Overview of the Criteria API
  2. Working Querying with the Criteria API
  3. Query by Example

VI. Hibernate and Java Persistence / EJB 3
- A. Overview of Java Persistence / EJB 3
- B. Relationship between Java Persistence and Hibernate
- C. Mapping Entities with JPA Annotations
- D. The EntityManager, Persistence Context and Persistence Unit

Hibernate

Course Outline (cont'd)

E. Working with Transactions - EntityTransaction, Managed, and Unmanaged Environments
F. Inserts and Updates
G. JPQL - Java Persistence Query Language
H. Versioning
I. Relationships

VII. Caching
A. Understand caching advantages and behavior
B. The persistence context as first level cache
C. The Second Level Cache
D. Entity and Collection Caching

VIII. Integration Considerations
A. Data Access Objects
B. Integrating Hibernate and Java Web Apps
C. Open Session in View Pattern
D. Hibernate / EJB-JPA Integration

IX. Additional Topics
A. Components and Multi-Table Mapping
B. equals() and hashCode()
C. Design Considerations
D. Hibernate Toolset

X. Hibernate and Spring Integration
A. Spring Introduction
B. Dependency Injection Overview
C. Spring's Hibernate Support
D. Spring Transaction Management