Hibernate

Course Summary

Description

Hibernate is a popular and excellent solution for handling object persistence in J2EE apps. This course is designed to provide J2EE developers with all that they need to successfully apply Hibernate to their project. The course guides students through the subtle and tricky aspects of object persistence. The numerous labs in the course give students the critical hands on experience that will allow them to effectively utilize Hibernate in their applications. In addition to the through coverage of Hibernate the course includes a section on Hibernate best practices along with a case Study that provides students with a head start on applying Hibernate to their projects.

Topics

- Understanding the Problem that Hibernate Solves
- Developing With Hibernate
- Fundamentals of Persistent Class Mapping
- Managing Relationships With Hibernate
- Hibernate Queries
- Advanced Hibernate Mapping
- Hibernate Transactions
- Hibernate Caching
- Architecting Applications With Hibernate
- Hibernate Best Practices
- Case Study: Applying Hibernate

Audience

This course is designed to provide J2EE Developers with all that they need to successfully apply Hibernate to their project.

Prerequisites

Students should have knowledge of the Java Programming Language and knowledge of relational databases.

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Hibernate

Course Outline

I. Understanding The Problem That Hibernate Solves
- A. Object relational mapping problem
- B. Object identity
- C. Persistent object models
- D. Inheritance
- E. Associations
- F. Persistence lifecycle
- G. Persistence managers
- H. Transitive persistence
- I. Finding objects
- J. Relationships
- K. Transactions
- L. Concurrency
- M. Caching

II. Developing With Hibernate
- A. Hibernate quick start application
- B. Hibernate architecture
- C. Hibernate persistence API
- D. Basic hibernate configuring
- E. Advanced hibernate configuration
- F. Running Hibernate standalone
- G. Running Hibernate with WebSphere
- H. How Hibernate fits into the development process
- I. DB schema generation
- J. POJO code generation
- K. Metadata generation
- L. Round tripping
- M. Integrating Hibernate into your development environment
- N. Unit testing and hibernate

III. Fundamentals of Persistent Class Mapping
- A. Domain modeling
- B. Implementing domain models with Java
- C. Mapping Java classes to database tables
- D. Mapping object attributes to columns
- E. The Hibernate mapping XML schema
- F. Primary keys and object identity with hibernate
- G. Fine-grained object models
- H. Coarse-grained object models
- I. Mapping class inheritance relationships
- J. Mapping class associations

IV. Managing Relationships With Hibernate
- A. One-to-one relationships
- B. One-to-many relationships
- C. Many-to-Many relationships

V. Hibernate Queries
- A. Query interfaces
- B. Hibernate query language
- C. Named queries
- D. Binding parameters
- E. Grouping results
- F. Native SQL queries
- G. Joins
- H. Sub queries
- I. Hibernate query optimization
- J. Advanced query techniques
- K. Efficient processing of large result sets in Java

VI. Advanced Hibernate Mapping
- A. The Hibernate type system
- B. Dealing with custom data types
- C. Mapping collections of value types
- D. Class Inheritance and Mapping
- E. Extending the Hibernate mapping system

VII. Hibernate Transactions
- A. Local transactions
- B. Distributed transactions, XA and two phase commit
- C. Transaction management in Enterprise Java
- D. JDBC transactions with Hibernate
- E. JTA transactions with Hibernate
- F. Hibernate transaction API
- G. Controlling transaction isolation levels
- H. Optimistic and pessimistic locking with hibernate
- I. Transactions and performance
- J. Using Hibernate with declarative transactions frameworks (Spring, EJB)

VIII. Hibernate Caching
- A. Caching Fundamentals
- B. The Hibernate caching architecture
- C. Caching with EHChace
- D. Distributed Caching
- E. Caching with JBossCache

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

F. Caching best practices

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Hibernate

Course Outline

IX. Architecting Applications With Hibernate

A. Applying hibernate with web applications
B. Applying hibernate with EJB applications
C. Common architectural problems
D. Tricky modeling problems
E. Architecting for performance
F. Architecting for maintainability

X. Hibernate Best Practices

A. Mapping best practices
B. Transaction best practices
C. Caching best practices
D. Performance best practices
E. Modeling best practices
F. Querying best practices
G. Tooling best practices
H. Testing best practices
I. Deployment best practices

XI. Case Study: Applying Hibernate

A. In this section students work in groups to apply Hibernate and what they have learned in the course to the project they are working on. In this part of the course students bring everything that they have learned from the course to get a head start on applying hibernate on their project.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically