

## Advanced Hibernate

Advanced Hibernate training is targeted toward Java developers who wish to extract the full power of the Hibernate O/R Mapping framework. The primary target audience consists of Java developers who work with SQL-based database systems, database developers who are looking for an introduction to object-oriented software development and database administrators interested in how ORM affects performance and how to tune the performance of the SQL database management system and persistence layer. The training covers the new Hibernate 3 features.

**Course Duration:** This is a 2 day course with a half-day on-line webinar as prerequisite.

**Course Format:** The class is 70% theory and 30% hands-on lab exercises.

### Course Prerequisites

The prerequisite skills for this class are the following:

- Basic Hibernate knowledge. JBoss, Inc. provides a prerequisite on-line webinar for this purpose.
- Competency with the Java language
- Knowledge of OOAD concepts
- Familiarity with the UML
- Experience with a dialect of SQL.
- Using the JDK and creating the necessary environment for compilation and execution of a Java executable from the command line.

Experience with, or comprehensive knowledge of JNDI and JDBC. Entity EJB2.1 or EJB3.0 knowledge, while not a prerequisite, is helpful. Prior reading of the book *Hibernate in Action*, by Christian Bauer and Gavin King (published by Manning) is recommended.

## Course Prerequisite Module

We recommend and provide a web-based unattended Hibernate Essentials web-based training (WBT) tutorial (Flash player required). This three hour training covers all of the Hibernate introductory topics that will no longer be covered during the more advanced 2 day session. Good Java and basic SQL knowledge is required, the language is English. A link to this free tutorial will be provided to you upon registering for the Hibernate Advanced training class. The web-based tutorial is currently only available to Hibernate Advanced training attendees.

## Hibernate Essentials WBT

This Hibernate training is targeted at developers wanting to learn the Hibernate object/relational mapping service. The training will show you all steps necessary to write your first Hibernate application and provides an excellent starting point for your first Hibernate project.

We start the training with an introduction of the object/relational paradigm mismatch and discuss a high-level overview of the current solutions for this time consuming problem. Our conclusion is that transparent and automatic object/relational mapping as provided by Hibernate is one of the best existing options.

We then introduce Hibernate by coding an example application in the Eclipse IDE, using the JBoss Eclipse IDE plug-ins for Hibernate.

Next we have a look at a more complex Java object model and sophisticated object to relational mapping techniques. We write persistent classes, including composition and entity associations, and map them to database tables and columns.

Managing object state (i.e. loading and storing objects) with the Hibernate APIs is our next topic. We also show you how easy it is to use EJB3 persistence operations in Hibernate and how you can manage the state of your objects in a simple way with cascading persistence options.

Persisting objects requires transaction processing. You will learn how to use the Hibernate APIs for transaction demarcation and how to best integrate Hibernate into your existing transactional systems. We discuss the innovative support in Hibernate for long transactions, that is, transactions that require disconnection from the database

during user think-time. Hibernate's excellent support for optimistic locking and automatic versioning is another topic presented in detail.

Following the discussion of transactions, we have a look at different fetching strategies. Getting objects out of the database as efficiently as possible requires knowledge of Hibernate's query facilities as well as lazy, eager, and batch fetching options. HQL, Criteria, and Example queries are introduced with straightforward examples.

We finish this training with a collection of best practices for Hibernate application developers. Important design patterns such as "Open Session in View" and "ThreadLocal Session" are shown, as well as API extension points for custom functionality.

## Course Modules

- 1. Flashback.** We kick off this training with a quick Hello World example, to refresh our Hibernate basics. Right after this we delve into more advanced topics.
- 2. Advanced class and property mappings.** We have a look at the different mapping metadata facilities available for and in Hibernate. The standard XML mapping files are discussed, as well as XDoclet and JDK 1.5 Annotations for EJB persistence. Hibernate's mapping type system is an important supporting feature for fine-grained Java domain models and you'll learn how best design your persistent classes. We talk about the built-in mapping strategies for entities, simple value types, and class inheritance. Custom extension of the Hibernate type system is another topic in this training module.
- 3. Associations and collections.** Complex associations between entities, single-valued and with collections, often require advanced Hibernate mapping knowledge. We discuss various kinds of mappings, as well as Hibernate's fully polymorphic behavior. We then apply this knowledge in our lab exercise.
- 4. Transactional processing.** Hibernate defines several object states and provides APIs to change the state of an object. We show you how to use these APIs correctly, even in tricky situations. You will use Hibernate data filters to access regional and temporal data. We have a look at more advanced transaction processing concepts around optimistic locking, pessimistic locking, and long-running application transactions.

5. **Querying and fetching data.** Advanced object retrieval options and the correct usage of Hibernate's association fetching strategies are the topics of this training module. The new Criteria and Example API is also part of our discussion, as well as advanced HQL, EJB-QL (using the new EJB 3.0 persistence API), and native SQL queries. The lab includes exercises around loading and storing objects efficiently.
6. **Application design.** A popular part of the Hibernate training, this module shows you how to design and implement a persistence layer using Hibernate. You will learn how to correctly handle SessionFactory and Session in two-tiered and three-tiered applications, and the best practices for transaction handling. You will implement event interception in the persistence layer, for audit logging and other special cases. You might also be interested in Hibernate's data conversion capabilities, for automatic XML marshaling, or the representation of domain objects in nested HashMaps.
7. **Legacy system integration.** Many Hibernate developers have to work with existing database schemas and existing data. Unfortunately, these schemas are usually not in best shape, and the data might be difficult to map to Java objects. We discuss reverse engineering using the Hibernate toolset, as well as advanced Hibernate mapping techniques, such as formula-based mappings, natural- and composite keys, and triggers.
8. **Deployment and administration.** Getting Hibernate up and running is usually very easy, and deployment requires only a single configuration in most cases. We show you more advanced configuration options and tricks in this training module, including deployment of Hibernate as a JMX and JCA service in J2EE environments. Once your application is deployed, administration becomes an important issue. You will learn how to use logging efficiently and how to detect performance issues using runtime monitoring with the new statistics interfaces.
9. **Tuning.** In our last training module we'll optimize database access - our goal is to execute the optimal SQL queries in all situations. You will learn data caching basics and when to use caching. We then enable Hibernate's dual layer caching system in our labs, as well as the cache for query results.