

LP340: Integrating Hibernate and Spring (5 days)

Integrating Hibernate and Spring is a five day in-depth course geared for experienced Java developers who need to understand what Hibernate is in terms of today's systems and architectures, and how to apply Hibernate to persistence requirements in Java and J2EE applications. The course then covers to the Spring framework and how to integrate Hibernate as the persistence layer for Spring applications.

Hibernate is a powerful enabler that addresses object/relational persistence in the Java world. Hibernate offers all the advantages of developing in Java plus a comprehensive suite of capabilities for mapping object-oriented features to the relational model. This course tells you what you need to know to design and build your own Hibernate-enabled applications. You'll learn the details of the key Hibernate capabilities and how to leverage their strengths, with a special focus on using Hibernate with the Spring framework.

The Spring framework is an application framework that provides a lightweight container that supports the creation of simple-to-complex components in a non-invasive fashion. Spring's flexibility and transparency is congruent and supportive of incremental development and testing. The framework's structure supports the layering of functionality such as persistence, transactions, view-oriented frameworks, and enterprise systems and capabilities. Spring's Aspect-Oriented Programming (AOP) framework enables developers to declaratively apply common features and capabilities across data types in a transparent fashion.

Spring makes J2EE development easier. Spring simplifies commons tasks and encourages good design based on programming to interfaces. Spring makes your application easier to configure and reduces the need for many J2EE design patterns. Spring puts the OO design back into your J2EE application, and it integrates nicely with JSF.

If you want to deliver an application using Hibernate within the Spring framework, you'll find this course essential.

What You'll Learn

This course provides coverage of Hibernate concepts and practices for interacting between Java and relational databases. The goal of this course is to empower developers with the knowledge they need to design and implement Spring applications that effectively and transparently use Hibernate to manage data persistence. The areas addressed in this course range from data/class mapping and persisted object lifecycle and management to how Hibernate can be used in the Spring framework.

At the conclusion of this course, attendees will be able to:

- Explain how the issues associated with object persistence in a relational model are addressed by Hibernate
- Understand the relationships between SQL, Java, Spring, and Hibernate
- Discuss the challenges to adopting Hibernate in the enterprise
- Write applications that take advantage of the Hibernate Persistence Manager.
- Map Java classes to relational tables.
- Capture both relational and inheritance associations in metadata using either XML or the Java 5 Annotations mechanism.
- Create and use mappings between Java classes and relational databases.
- Understand how identity and keys are handled in Hibernate.
- Understand the persistent object lifecycle and how that relates to transactions and concurrency.
- Take advantage of Hibernate's data filtering and interception.
- Explain the issues associated with complex frameworks such as J2EE and how Spring addresses those issues

- Write applications that take advantage of the Spring container and the declarative nature of assembling simple components into applications.
- Work with Spring's support for transactions
- Understand how to use Hibernate within the Spring framework

Course Overview: Hands-On Learning

This class is “technology-centric”, designed to train attendees in essential J2EE development skills coupling the most current, effective techniques with the soundest coding practices.

Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review.

This workshop is about 50% hands-on lab and 50% lecture. Multiple detailed lab exercises are laced throughout the course, designed to reinforce fundamental skills and concepts learned in the lessons. Because these lessons, labs and projects are presented in a building block fashion, students will gain a solid understanding of not only the core concepts, but also how all the pieces fit together in a complete application. At the end of each lesson, developers will be tested with a set of review questions to ensure that he/she has fully understands that topic.

Who Should Attend

This an **intermediate** level Hibernate and Spring training course, designed for developers that need to understand how and when to use Hibernate in Spring applications.

Pre-Requisites

Attendees should have Java development experience or have taken an introductory-level Java course.

Course Details

Hibernate Overview

- ORM Mapping Issues
- Hibernate Architecture
- Persistence, Identity, and Equality in Hibernate
- Domain Models and Metadata Options

Hibernate QuickStart

- Basic Mapping
 - Class/properties to Table/Columns
- Basic Configuration
- Mapping a POJO to a Database
- CRUD operations
- Basics of Hibernate Session
- Working with Persisted Objects

Types in Hibernate

- Hibernate Type System
- Entity and Value Types
- Components

- Collections
- Implementing Complex Types

Associations in Hibernate

- Relational Associations
 - Composition
 - Bidirectional
 - Unidirectional
- Inheritance Associations
 - Mapping Strategies
 - Single, Class, and Concrete Tables

Working with Persisted Objects

- Object states and lifecycles
- Hibernate Persistence Manager API
- Data Filtering
- Interception
- Transactions
- Concurrency

Hibernate Query Functionality

- Hibernate Query API
- Hibernate Query Language (HQL)
- Grouping
- Subqueries
- Injection Vulnerabilities and Defenses

Introduction to the Spring Framework

- Inversion of Control
- Dependency Injection
- Spring Overview
- Spring Application Architectures
- Spring Container
 - Managing the Container
 - Access to Services and Resources
 - Application Contexts
- Beans as Components
 - Beans and Factories
 - XML Bean Configuration
 - Bean Definition and Dependencies
 - Bean Lifecycle
- Customization Options
 - Post-Processors
 - Property Editors

Spring and Persistence

- Data Access Pattern
- Overview of Persistence Layer and Transactions
- Spring JDBC
 - Spring JDBC Architecture
 - Working with JDBC Template
 - Database Operations

- Handling JDBC Exceptions
- Hibernate
 - Spring - Hibernate Architecture
 - ORM Mapping Overview
 - DAO Implementation
 - Working with Hibernate DAOs in Spring
 - Hibernate Template

Spring and Transactions

- Transaction Overview
- Spring Transactions
- Defining Spring Transactions
- Working with Demarcation
- Managing Spring Transactions

Spring and Extensibility

- AOP Concepts
- Spring's AOP Framework
- Interceptor Chain
- Integration with Spring Container
- Working with Proxies

Appendix: Basic ORM

- Relational to Object Data Models
- SQL and JDBC Basics

Appendix: Hibernate Annotations

- Java 5 Annotations
- Using Annotations for Metadata