

Spring and Hibernate

Course Summary

Description

This course provides comprehensive introductions to the Spring and Hibernate open source frameworks. Suitable for both Spring3/Hibernate3 and Spring4/Hibernate4, it includes coverage of all the core Spring and Hibernate capabilities, as well as the integration capabilities provided by Spring.

The course includes an extensive set of hands-on labs that reinforce all the important concepts and capabilities. It will enable you to build working Spring/Hibernate applications, and give you an understanding of the important concepts and technology in a very short time.

Spring: As well as being current with recent Spring releases, this course introduces techniques for using the many new and powerful capabilities that Spring supports. It includes complete coverage of the three main configuration styles (@Configuration, @Component, XML), and guidelines for their usage. It also covers more advanced capabilities such as support for JDBC and persistence frameworks like Hibernate, Spring declarative transactions, and Spring integration with Java EE Web technologies.

Hibernate: The course covers all important capabilities of this open source object/relational persistence and query service for Java. Coverage includes developing persistent classes in Java, as well as using associations/relationships, inheritance, polymorphism, composition and collections. It also provides an overview of JPA (the Java Persistence API).

The standard platform does all labs with the **Eclipse IDE**, and the lab instructions include detailed directions for setting up and using it. The course is available for all major development environments, including IBM RAD.

Objectives

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

- Understand the core principles of Spring, and of Dependency Injection (DI)/Inversion of Control
- Use the Spring Core module and DI to configure and wire application objects (beans) together
- Know the different types of metadata (XML, @Component, and @Configuration), and how and when to use them
- Understand and use the complete capabilities of the Core module, such as lifecycle events, bean scopes, and the Spring API
- Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as Hibernate or JPA.
- Understand and use Spring's transaction support, including the easy-to-use Java annotation support, as well as the tx/aop XML configuration elements
- Integrate Spring with Java EE Web applications
- 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
- Understand Hibernate transaction support
- Understand the relationship between Hibernate and the Java Persistence API (JPA), and use JPA annotations for OR mapping

Spring and Hibernate

Course Summary (cont'd)

Topics

- Introduction to Spring
- Configuration in Depth
- Wiring in Depth
- Introduction to Hibernate
- Spring/Hibernate Integration
- Updates and Queries
- Transaction (TX) Management
- The Hibernate Persistence Lifecycle
- Relationships
- Introduction to Spring Web Integration
- Hibernate and JPA (Java Persistence API)

Audience

This course is designed for developers.

Prerequisites

There are no prerequisites for this course.

Duration

Five days

Spring and Hibernate

Course Outline

- I. Introduction to Spring**
 - A. Overview of Spring Technology
 - B. Challenges for Modern Applications
 - C. Motivation for Spring, Spring Architecture
 - D. The Spring Framework
 - E. Spring Introduction
 - F. Managing Beans
 - G. Inversion of Control / IoC, Dependency Injection / DI
 - H. Configuration Metadata Overview, Configuring Beans (XML)
 - I. The Spring Container
 - J. Overview of the Spring Container
 - K. A Simple Spring Example
 - L. ApplicationContext Overview
 - M. ClassPathXmlApplicationContext, FileSystemXmlApplicationContext, AnnotationConfigApplicationContext
 - N. API and Usage
 - O. Dependencies and Dependency Injection (DI)
 - P. Examining Dependencies
 - Q. Dependency Inversion
 - R. Dependency Injection (DI) in Spring - Basic Configuration and Usage
- II. Configuration in Depth**
 - A. Annotation Driven Configuration
 - B. JSR 330 (@Named) and Spring (@Component) annotation styles
 - C. @Named/@Component, @Inject/@Autowired, @Repository, @Service
 - D. Configuring Beans and Autowiring with Annotations
 - E. Enabling Annotations - context:component-scan
 - F. Pros and Cons
 - G. Java Based Configuration (@Configuration)
 - H. Overview - code-centric Configuration
 - I. @Configuration and @Bean
 - J. Dependency Injection
 - K. Resolving Dependencies on Other Beans, Injecting Configuration Classes
 - L. Pros and Cons
 - M. Integrating Configuration Types
 - N. Choosing a Configuration Style
 - O. Integrating Configuration Styles
 - P. Importing: @Import and
 - Q. Scanning with @Configuration style
 - R. Bean Scope and Lifecycle
- III. Wiring in Depth**
 - A. Value Injection
 - B. Configuring Value Properties, Property Conversions
 - C. Externalizing Values in Properties Files
 - D. Constructor Injection
 - E. Constructor Injection Overview
 - F. Configuration - @Configuration and XML
 - G. p: and c: namespaces for XML configuration
 - H. Qualifiers / Domain Specific Language (DSL)
 - I. Limitations of Autowiring
 - J. Qualifiers and DSL
 - K. Creating and Using an Annotation-Based DSL for Bean Configuration
 - L. Benefits of Qualifiers for Bean Configuration
 - M. Profiles
 - N. Profiles Overview
 - O. Configuring Profiles (XML and @Configuration)
 - P. Activating Profiles
 - Q. Overview of SpEL
- IV. 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
 - E. hibernate.cfg.xml file, Connection properties, Database dialect
 - F. SessionFactory, Configuration, and Session
 - G. Mapping a Class
 - H. Persistent Entity Class, Hibernate Mapping File, Mapping the Entity Class
 - I. Primary keys: Id property, Generated Id
 - J. Hibernate Type System
 - K. Working with sessions and Persistent Objects
 - L. Logging: hibernate.show_sql, log4j Overview and configuration for Hibernate
 - M. Hibernate
- V. Bean Scope Defined - singleton, prototype, and Other Scopes**
- T. Configuring Scope**
- U. Bean Creation Lifecycle, Lifecycle Callbacks**
- V. BeanPostProcessor, Event Handling**

Spring and Hibernate

Course Outline (cont'd)

- V. Spring/Hibernate Integration**
 - A. Overview of Spring database support
 - B. Configuring a DataSource
 - C. Using Contextual Sessions
 - D. Spring/Hibernate configuration with LocalSessionFactoryBean and SessionFactory configuration
 - E. Creating Spring/Hibernate DAO classes using Contextual Sessions
 - F. Overview of Template Approach
 - H. Detached Objects and Optimistic Locking
 - I. Versioning overview and Using Versioning
 - J. Locking Objects
- VI. 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
- VII. Transaction (TX) Management**
 - A. Hibernate Transaction Management
 - B. Transaction Overview and Transactions in Hibernate
 - C. Hibernate Transaction API (in Managed and Non-managed Environments)
 - D. Intro to Spring Transaction Management
 - E. Spring Transaction Managers
 - F. Spring Declarative TX Management
 - G. Spring TX Scope and Propagation
 - H. Spring TX Attributes (REQUIRED, SUPPORTS, etc)
 - I. XML Configuration of Spring Transactions
 - J. Specifying Advice, TX Attributes, and Methods
 - K. Linking Advice with Pointcuts
 - L. Benefits of XML Configuration of TX Behavior
- VIII. The Hibernate Persistence Lifecycle**
 - A. The lifecycle of managed objects
 - B. Persistent, transient, and detached objects
 - C. The Persistence (Session) Context (Lifespan, Relation to Managed Objects, Propagation)
 - D. Contextual Sessions
 - E. Synchronization to the Database
 - F. The Session as cache
 - G. Optimistic Locking / Versioning
- IX. 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
 - J. Entity Inheritance with Hibernate
 - K. Table-per-class mapping
 - L. Table per Subclass mapping
 - M. Table per Concrete Class mapping
- X. Introduction to Spring Web Integration**
 - A. Integrating Spring with Java EE Web Apps
 - B. ContextLoaderListener
 - C. WebApplicationContext
 - D. Using Spring beans in Wep app controller logic
- XI. Hibernate Additional Topics**
 - A. Components and Multi-Table Mapping
 - B. equals() and hashCode()
 - C. Caching and Efficiency
 - D. Design Considerations
 - E. Session 13: Hibernate and JPA (Java Persistence API)
 - F. Overview of the Java Persistence API (JPA) / EJB 3
 - G. Relationship between JPA and Hibernate
 - H. Mapping Entities with JPA Annotations
 - I. The EntityManager, Persistence Context and Persistence Unit
 - J. Working with Transactions - EntityTransaction, Managed, and Unmanaged Environments
 - K. Inserts and Updates
 - L. JPQL - Java Persistence Query Language
 - M. Versioning
 - N. Relationships