

Web Application Development Using Spring, Hibernate and JPA

Duration: 5 Days (*Face-to-Face & Remote-Live*), or 35 Hours (*On-Demand*)

Price: CDN\$3,275 (*Face-to-Face & Remote-Live*), or CDN\$1,995 (*On-Demand*)

Discounts: We offer multiple discount options. [Click here](#) for more info.

Delivery Options: Attend face-to-face in the classroom, [remote-live](#) or [on-demand training](#).

Students Will Learn

- The core principles of Spring and of Dependency Injection (DI)/Inversion of Control
- Using the Spring core module and DI to configure and wire application objects (Beans) together
- Using the different types of metadata (XML, @Component, and @Configuration)
- The complete capabilities of the core module
- Using the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as Hibernate or JPA
- Spring's transaction support
- Integration of Spring with Java EE Web applications
- The features and benefits of Hibernate
- How to architect applications using the Hibernate framework
- Structuring code using Hibernate protocols
- Using Hibernate Mapping to map persistent objects to the database
- Working with collections and associations
- Using Hibernate's versioning support
- Mapping inheritance hierarchies using Hibernate
- Working with Hibernate Queries, HQL, and Criteria
- Hibernate transaction support
- The relationship between Hibernate and the Java Persistence API (JPA)
- How to architect and code using JPA2
- Exploiting AJAX integration
- Exploiting Web services

Course Description

This course provides a comprehensive introduction to JPA (the Java Persistence API), the Spring and Hibernate open source frameworks as well as Web Services and AJAX. Suitable for both Spring3/Hibernate3 and Spring4/Hibernate4, the course includes coverage of the core Spring and Hibernate capabilities, as well as the integration capabilities provided by Spring.

This course introduces techniques for using the many new and powerful capabilities that Spring4 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 JEE Web technologies.

The course covers all important capabilities of the Hibernate 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 covers fundamentals of JPA (the Java Persistence API) including JPA Annotations and JPQL (Java Persistence Query Language).

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency. Attendees build working Spring/Hibernate applications utilizing the Eclipse IDE, providing a knowledge foundation applicable to all major development environments.

Course Prerequisites

Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML, HTML, and JavaScript is helpful but not required. HOTT's course [Java Programming](#) or equivalent knowledge provides a solid foundation.

Course Overview

Introduction to Spring

- Overview of Spring Technology
 - Challenges for Modern Applications
 - Motivation for Spring
 - Spring Architecture
 - The Spring Framework API
- Spring Fundamentals
 - Managing Beans
 - Inversion of Control/IoC
 - Dependency Injection/DI
 - Configuration Metadata
 - Configuring Beans (XML)
- The Spring Container
 - Function of the Spring Container
 - ApplicationContext Overview
 - ClassPathXmlApplicationContext
 - FileSystemXmlApplicationContext
 - AnnotationConfigApplicationContext
 - API and Usage
- Dependencies and Dependency Injection (DI)
 - Examining Dependencies
 - Dependency Inversion
 - Dependency Injection (DI) in Spring
 - Basic Configuration and Usage

Configuration in Depth

- Annotation Driven Configuration
 - JSR 330 (@Named) and Spring (@Component) annotation styles
 - @Named/@Component, @Inject/@Autowired, @Repository, @Service
 - Configuring Beans and Autowiring with Annotations
 - Enabling Annotations - context:component-scan
 - Annotation Pros and Cons
- Java Based Configuration (@Configuration)
 - Overview of Code-Centric Configuration
 - @Configuration and @Bean
 - Dependency Injection
 - Resolving Dependencies on Other Beans
 - Injecting Configuration Classes
 - Pros and Cons
- Integrating Configuration

Types

- Choosing a Configuration Style
- Integrating Configuration Styles
- Importing: `@Import`
- Scanning with `@Configuration` Style
- Bean Scope and Lifecycle
 - Bean Scope Defined
 - Singleton, Prototype and Other Scopes
 - Configuring Scope
 - Bean Creation Lifecycle
 - Lifecycle Callbacks
 - `BeanPostProcessor` and Event Handling

Wiring in Depth

- Value Injection
 - Configuring Value Properties
 - Property Conversions
 - Externalizing Values in Properties Files
- Constructor Injection
 - Constructor Injection Overview
 - Configuration - `@Configuration` and XML
 - `p:` and `c:` Namespaces for XML Configuration
- Qualifiers and Domain Specific Language (DSL)
 - Limitations of Autowiring
 - Qualifiers and DSL
 - Creating and Using an Annotation-Based DSL for Bean Configuration
 - Benefits of Qualifiers for Bean Configuration
- Profiles
 - Profiles Overview
 - Configuring Profiles (XML and `@Configuration`)
 - Activating Profiles
- Introduction to SpEL

Spring/Hibernate Integration

- Overview of Spring Database Support
- Configuring a DataSource
- Using Contextual Sessions
 - Spring/Hibernate Configuration with

Introduction to Hibernate

- Issues with Persistence Layers and Object-Relational Mapping (ORM)
- Hibernate Overview and Benefits
- Hibernate Architecture
- Configuring Hibernate
 - `hibernate.cfg.xml` File
 - Connection Properties
 - Database Dialect
 - `SessionFactory`
 - Configuration and Session
- Mapping a Class
 - Persistent Entity Class
 - Hibernate Mapping File
 - Mapping the Entity Class
 - Primary keys: ID property, Generated ID
 - Hibernate Type System
- Working with Sessions and Persistent Objects
- Logging:
`hibernate.show_sql,`
- `log4j` Overview and Configuration for Hibernate

Updates and Queries

- Inserting, Updating and Deleting Entities
- HQL - Hibernate Query Language Fundamentals
- The Query Interface

LocalSessionFactoryBean and
SessionFactory

- Creating Spring/Hibernate DAO Classes Using Contextual Sessions
- Overview of Template Approach

Transaction (TX) Management

- Hibernate Transaction Management
 - Transaction Overview and Transactions in Hibernate
 - Hibernate Transaction API (in Managed and Non-Managed Environments)
- Intro to Spring Transaction Management
 - Spring Transaction Managers
 - Spring Declarative TX Management
 - Spring TX Scope and Propagation
 - Spring TX Attributes (REQUIRED, SUPPORTS, etc)
- XML Configuration of Spring Transactions
 - Specifying Advice, TX Attributes, and Methods
 - Linking Advice with Pointcuts
 - Benefits of XML Configuration of TX Behavior

Relationships

- Object Relationship Overview
- Mapping Collections of Value Objects
- Entity Relationships: 1-N, N-1, N-N, 1-1
- Mapping Entity Relationships
- Uni and Bi-Directional Relationships
- The Relationship "Inverse"
- Cascading Over Relationships
- Queries Across Relationships
 - Lazy Loading vs. Eager Loading
- Inheritance Mapping
 - Entity Inheritance with Hibernate
 - Table-per-Class Mapping
 - Table-per-Subclass Mapping
 - Table-per-Concrete Class Mapping

Hibernate Additional Topics

- Components and Multi-Table Mapping
- `equals()` and `hashCode()`
- Caching and Efficiency
- Design Considerations

- Creating and Working with Queries
 - Named Queries
 - Projection Queries
 - Aggregate Queries

The Hibernate Persistence Lifecycle

- The Lifecycle of Managed Objects
- Persistent, Transient, and Detached Objects
- The Persistence (Session) Context (Lifespan, Relation to Managed Objects, Propagation)
- Contextual Sessions
- Synchronization to the Database
- The Session as Cache
- Optimistic Locking/Versioning
 - Detached Objects and Optimistic Locking
 - Versioning Overview and Using Versioning
 - Locking Objects

Spring Web Integration

- Integrating Spring with Java EE Web Apps
 - `ContextLoaderListener`
 - `WebApplicationContext`
 - Using Spring Beans in Web App Controller Logic

Hibernate and JPA (Java Persistence API)

- Overview of the Java Persistence API (JPA)/EJB 3
- Relationship Between JPA and Hibernate
- Mapping Entities with JPA Annotations

The EntityManager, Persistence Context and Persistence Unit

- Working with Transactions
 - EntityManager, Managed and Unmanaged Environments
- Inserts and Updates
- JPQL - Java Persistence Query Language
- Versioning
- Relationships

Introduction to AJAX

- AJAX Architecture and Capabilities
 - Client Side
 - Server Side
- Circumventing the Page Reload Paradigm
- CSS, HTML and AJAX
- JavaScript and DOM
- XMLHttpRequest Object
 - readyState and responseXML Properties
- Making AJAX Asynchronous Calls
- AJAX Function Calling Conventions
- Response Handling with JavaScript
- Browser Compatibility Issues
- Server-Side and Client-Side AJAX

Adding AJAX to Existing Web Applications

- Issues Adding AJAX to Existing Apps
- Dealing with Asynchronous Responses
- Cross Browser Libraries and Frameworks
- Working with AJAX Toolkits
 - Dojo Toolkit
 - Prototype
 - DWR (Direct Web Remoting)
 - Google Web Toolkit
- AJAX and REST Design Compatibility Issues
- Security Issues

Web Services on the Client Side

- Consuming a Web Service
- Client Side Artifacts
- JAX-WS Clients
- Java/WSDL Mapping
- RESTful Web Services
- RESTful Web Services in Java

Java and AJAX

- Callback Functions/Methods
- Built-In Objects
- Parsing AJAX Responses
- XML and JSON Response Formats
- Servlet Code to Handle AJAX Requests
- Server Side Persistence
- AJAX Patterns and Best Practices
- Limitations of AJAX
- Debugging AJAX

Web Services on the Server Side

- Overview of Web Services
- Advantages of Web Services
- Creating a Web Service
- Deploying a Web Services
- Requirements for a JAX-WS Web Service Endpoint Implementation Class
- The Web Service Endpoint Implementation Class
- Working with WSDL Files
- Web Service Interoperability

The Best Way to Transfer Technology Skills

1 Village Square, Suite 8
14 Fletcher Street
Chelmsford, MA 01824

Copyright © 2021 Hands On Technology Transfer, Inc.