

Introduction to Spring and JPA

Course Summary

Description

Spring is a lightweight Java framework for building enterprise applications. Its Core module allows you to manage the lifecycle of your objects and the dependencies between them via configuration metadata (either XML or annotations) and Dependency Injection / Inversion of Control. Its advanced capabilities provide support for JDBC and persistence frameworks like JPA2 (DAO and ORM modules), as well as integration with Java EE Web technologies.

The Java Persistence API (JPA) is a Java standard object/relational (OR) persistence and query service for Java. JPA supports a POJO (Plain Old Java Object) based model using annotations which lets you develop persistent classes following common Java idioms. It supports entity relationships, inheritance, polymorphism, composition, and much more.

This course, available for Spring 3 (and Spring 4 coming soon), includes coverage of all the core Spring and JPA capabilities, as well as the integration capabilities provided by Spring. It provides extensive coverage of using Spring and JPA together, as well as all the core concepts of Spring and JPA themselves. All capabilities are practiced via an extensive set of hands-on labs.

The course starts with the basics of Spring and in-depth coverage of using the powerful capabilities of the Core module to reduce coupling and increase the flexibility and ease of maintenance and testing of your applications. It goes on to cover the basics of using Spring to simplify the creation of a persistence layer using its JPA integration capabilities.

The JPA material covers all basic JPA concepts, including mapping persistent classes, and using EntityManager and EntityManagerFactory to access and manipulate persistent entities. Querying with the Java Persistence Query Language (JPQL) is covered in depth, from basic queries to more advanced queries like eager queries using left joins. It also includes coverage of advanced concepts such as collections of value objects, relationships, and inheritance and polymorphic queries.

This course is hands on with labs to reinforce all the important concepts. It will enable you to build working Spring/JPA applications, and give you an understanding of the important concepts and technology in a very short time.

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
- Understand and use the complete capabilities of the Core module, such as lifecycle events, bean scopes, and the Spring API
- Use Springs ORM/JPA capabilities to integrate Spring and JPA and create Spring/JPA-based DAO types
- Understand and use Spring's transaction support, including its easy to use tx/aop XML configuration elements and TX-based annotations
- Integrate Spring with Java EE Web applications
- Understand the benefits of JPA
- Understand the JPA architecture
- Create JPA based applications
- Understand and use JPA mapping to map persistent objects to the database
- Work with JPA queries and JPQL

Introduction to Spring and JPA

Course Summary (cont'd)

- Understand and work with collections and associations
 - Value and Entity Types
 - Bidirectional and unidirectional
 - 1-1, 1-N, N-N
- Use JPA's versioning support
- Map inheritance hierarchies using JPA

Topics

- Introduction to Spring
- Configuration in Depth
- Wiring in Depth
- Introduction to Java Persistence API (JPA2)
- Spring/JPA Integration
- Updates and Queries
- Transaction (TX) Management
- The JPA Persistence Lifecycle
- Relationships
- Introduction to Spring Web Integration
- JPA Additional Topics

Audience

This course is designed for those wanting to build working Spring/JPA applications, and gain an understanding of the important concepts and technology in a very short time.

Prerequisites

There are no prerequisites for this course.

Duration

Five days

Introduction to Spring and JPA

Course Outline

- I. Introduction to Spring**
 - A. Overview of Spring Technology
 - 1. Challenges for Modern Applications
 - 2. Motivation for Spring, Spring Architecture
 - 3. The Spring Framework
 - B. Spring Introduction
 - 1. Managing Beans
 - 2. Inversion of Control / IoC, Dependency Injection / DI
 - 3. Configuration Metadata Overview, Configuring Beans (XML)
 - C. The Spring Container
 - 1. Overview of the Spring Container
 - 2. A Simple Spring Example
 - 3. ApplicationContext Overview
 - 4. ClassPathXmlApplicationContext, FileSystemXmlApplicationContext, AnnotationConfigApplicationContext
 - 5. API and Usage
 - D. Dependencies and Dependency Injection (DI)
 - 1. Examining Dependencies
 - 2. Dependency Inversion
 - 3. Dependency Injection (DI) in Spring - Basic Configuration and Usage
- II. Configuration in Depth**
 - A. Annotation Driven Configuration
 - 1. JSR 330 (@Named) and Spring (@Component) annotation styles
 - 2. @Named/@Component, @Inject/@Autowired, @Repository, @Service
 - 3. Configuring Beans and Autowiring with Annotations
 - 4. Enabling Annotations - context:component-scan
 - 5. Pros and Cons
 - B. Java Based Configuration (@Configuration)
 - 1. Overview - code-centric Configuration
 - 2. @Configuration and @Bean
 - 3. Dependency Injection
 - 4. Resolving Dependencies on Other Beans, Injecting Configuration Classes
 - 5. Pros and Cons
- III. Wiring in Depth**
 - A. Value Injection
 - 1. Configuring Value Properties, Property Conversions
 - 2. Externalizing Values in Properties Files
 - B. Constructor Injection
 - 1. Constructor Injection Overview
 - 2. Configuration - @Configuration and XML
 - 3. p: and c: namespaces for XML configuration
 - C. Qualifiers / Domain Specific Language (DSL)
 - 1. Limitations of Autowiring
 - 2. Qualifiers and DSL
 - 3. Creating and Using an Annotation-Based DSL for Bean Configuration
 - 4. Benefits of Qualifiers for Bean Configuration
 - D. Profiles
 - 1. Profiles Overview
 - 2. Configuring Profiles (XML and @Configuration)

Introduction to Spring and JPA

Course Outline (cont'd)

- 3. Activating Profiles
- E. Overview of SpEL
- IV. Introduction to Java Persistence API (JPA2)**
 - A. Overview
 - 1. Persistence Layers, Object-Relational Mapping (ORM), JDBC
 - 2. JPA Overview
 - B. Mapping with JPA
 - 1. Entities and @Entity, ids and @Id,
 - 2. Generated Id Values
 - 3. Basic Mapping Types
 - C. Persistence Unit and EntityManager
 - 1. Persisting to the DB, the EntityManager API
 - 2. Persistence Units, Config, Persistence Context
 - 3. Retrieving Persistent Entities with find()
 - D. More About Mappings
 - 1. Default Mappings, @Basic, @Column
 - 2. Field vs. Property Access
 - 3. Temporal (Date/Time) Mappings
 - E. Logging Options (Provider based)
- V. Spring/JPA Integration**
 - A. Spring's DataSource Support
 - B. Managing the EntityManager (EM)
 - C. LocalContainerEntityManagerFactory Bean and Container-managed EMs
 - D. JEE and JNDI Lookup of the EM
 - E. Configuration and Vendor Adaptors
 - F. Creating a JPA Repository/DAO Bean
 - @PersistenceUnit,
 - @PersistenceContext
- VI. Updates and Queries**
 - A. Inserting Updating, and Deleting Entities
 - B. Querying and JPQL
 - 1. Entity Based Queries, SELECT, WHERE
 - 2. Query Interface, Executing Queries, Generic Queries (JPA 2)
 - 3. JPQL Operators, Expressions, and Parameters
 - 4. Named Queries
- C. Additional Query Capabilities - Projection and Aggregate Query, Embedded Objects**
- VII. Transaction (TX) Management**
 - A. Transaction Overview and Transactions in JPA
 - 1. Transaction Overview
 - 2. EntityTransaction API (including JTA and resource-local EntityManager)
 - B. Spring Transaction Management
 - 1. Overview
 - 2. TransactionManagers
 - 3. Declarative Transactions and TX Propagation
 - 4. @Transactional and its settings
 - C. XML Configured Transactions
 - 1. new <tx:*> elements, Configuring tx:advice, and tx:attributes
 - 2. Defining the TX advisor
- VIII. The JPA Persistence Lifecycle**
 - A. The Persistence Lifecycle
 - 1. JPA Entity States (New, Managed, Detached, Removed), and Entity State Diagram
 - 2. Persistence Context - Lifespan, Propagation
 - 3. Synchronization to the DB
 - B. Versioning and Optimistic Locking
 - 1. Overview, Detached Instances
 - 2. Versioning, @Version, Optimistic Locking
 - C. Lifecycle Callbacks
 - 1. @PrePersist, @PostPersist, etc.
 - 2. Entity Listeners, @EntityListeners

Introduction to Spring and JPA

Course Outline (cont'd)

IX. Relationships

- A. Relationships Overview: Object Relationships, Participants, Roles, Directionality, Cardinality
- B. Relationship Mapping
 - 1. Mapping Overview (1-1, 1-N, N-1, N-N)
 - 2. Unidirectional and Bidirectional
 - 3. @ManyToOne, @OneToMany, Table Structures
 - 4. Relationship Inverse - Owning Side
 - 5. Collection Types (List, Set, etc)
 - 6. Cascading Over Relationships (including orphanRemoval - JPA 2)
 - 7. @ManyToOne, @OneToOne
 - 8. Lazy and Eager Loading
 - 9. Queries Across Relationships (Inner Joins, Outer Joins, Fetch Joins)
- C. Entity Inheritance Mapping
 - 1. Overview
 - 2. Single Table Mapping
 - 3. Joined (Table per Subclass) Mapping
 - 4. Table per Concrete Class Mapping
 - 5. Pros and Cons
- D. Element Collections (JPA 2)
 - 1. Overview, Collections of Value Objects, @ElementCollection, @CollectionTable
 - 2. Using Element Collections
 - 3. Collections of Embeddables

X. Introduction to Spring Web Integration

- A. Integrating Spring with Java EE Web Apps
 - 1. ContextLoaderListener
 - 2. WebApplicationContext
 - 3. Using Spring beans in Wep app controller logic
- B. Open EntityManager in View
 - 1. Lazy Loading Issue in Web Apps
 - 2. Open EntityManager in View Pattern
 - 3. Using Spring's OpenEntityManagerInViewFilter/Interceptor

XI. JPA Additional Topics

- A. Criteria Queries
- B. equals() and hashCode()
- C. Design Considerations