

Introduction to Spring 5, Spring MVC, and Spring REST

Course Summary

Description

Spring 5 provides an evolutionary advance of Spring's powerful capabilities. This course introduces these capabilities, as well as providing guidelines on when and how to use them. It includes coverage of the three main configuration styles: Java-based (@Configuration), annotation-based (@Component), and the traditional XML-based configuration that may still play an important role in existing and new projects.

The course starts with in-depth coverage of Spring's Core module to reduce coupling and increase the flexibility, ease of maintenance, and testing of your applications. It goes on to cover many of the most important capabilities of Spring, including easing configuration with Spring Boot, integrating Hibernate and JPA persistence layers with Spring and Spring Data, and using Spring's declarative transaction capabilities.

The course includes integration of Spring with Java EE Web applications, a solid introduction to Spring MVC, and coverage of building RESTful resources with Spring MVC. It also provides an overview of Spring's reactive programming model for repositories and Web resources.

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

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, annotations/@Component, and Java Configuration/@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
- Use Spring Boot to simplify dependency management and configuration
- Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as Hibernate or JPA.
- Use Spring Data to automatically generate JPA-based repository classes
- 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
- Build Web applications with Spring MVC, including configuration using Java config and Servlet 3 capabilities
- Understand and use the core capabilities of Spring's Reactive programming support
- Understand REST, and use Spring MVC to build RESTful services
- Use Ajax-based front ends with Spring MVC / Spring REST

Introduction to Spring 5, Spring MVC, and Spring REST

Course Summary (cont.)

Topics

- Introduction to Spring
- Configuration in Depth
- Spring Boot Overview
- Spring Testing
- Spring and Spring Data with Hibernate/JPA
- Spring Transaction (TX) Management
- Spring Web Integration and Intro to Spring MVC
- Additional Spring MVC Capabilities
- RESTful Services with Spring
- Working with JSON and XML
- Java Clients for RESTful Services
- Common REST Patterns
- Additional New Features in Spring 5
- [XML Specific Configuration (Optional)]

Audience

Suitable for Java developers that need to quickly ramp up on the Spring framework.

Prerequisite

No programming experience necessary for this course.

Duration

Five Days

Introduction to Spring 5, Spring MVC, and Spring REST

Course Outline

I. Introduction to Spring

- A. Overview of Spring Technology
 1. Motivation for Spring, Spring Architecture
 2. The Spring Framework
- B. Spring Introduction
 1. Declaring and Managing Beans
 2. ApplicationContexts - The Spring Container
 3. XML and @Component/@Named Config
- C. Dependencies and Dependency Injection (DI)
 1. Examining Dependencies
 2. Dependency Inversion / Dependency Injection (DI)
 3. DI in Spring - XML and @Autowired

II. Configuration in Depth

- A. Java Based Configuration (@Configuration)
 1. Overview, @Configuration, @Bean
 2. Dependency Injection
 3. Resolving Dependencies
- B. Integrating Configuration Types
 1. XML and @Component Pros/Cons
 2. @Configuration Pros/Cons
 3. Choosing a Configuration Style
 4. Integrating with @Import and <import>
- C. Bean Scope and Lifecycle
 1. Singleton, Prototype, and Other Scopes
 2. Configuring Scope
 3. Bean Lifecycle / Callbacks
- D. Externalizing Properties
 1. Properties Files
 2. @PropertySource, property-placeholder
 3. Using @Value
 4. SpEL
- E. Profiles
 1. Overview and Configuration
 2. Activating Profiles

III. Spring Boot Overview

- A. maven and Spring
- B. Spring Boot Structure
- C. Spring POMs with Boot Parents
- D. Spring Boot Starters
- E. Other Capabilities

IV. Spring Testing

- A. Testing and JUnit Overview
 1. Writing Tests - Test Classes, asserts, Naming Conventions
 2. Running Tests - IDE, maven, ...
 3. Test Fixtures - setup and teardown
- B. Spring TestContext Framework
 1. Overview
 2. Configuration
 3. Running Tests

V. Spring and Spring Data with Hibernate/JPA

- A. Overview of Spring database support
- B. Configuring a DataSource
- C. Using Spring with Hibernate
 1. High Level Hibernate Overview
 2. SessionFactory configuration, LocalSessionFactoryBean
 3. Contextual Sessions and Spring Integration
- D. Using Spring with JPA
 1. Managing the EntityManager (EM)
 2. LocalContainerEntityManagerFactoryBean and Container-managed EMs
 3. JEE and JNDI Lookup of the EM
 4. Configuration and Vendor Adaptors
 5. Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext
- E. Spring Data Overview
 1. Overview and Architecture
 2. Configuring Spring Data
 3. Repositories and JPA Repositories
 4. Using CrudRepository

Introduction to Spring 5, Spring MVC, and Spring REST

Course Outline (cont.)

- F. Using Spring Data
 1. Naming Conventions for Querying
 2. Creating more Complex Queries
 3. Query Configuration
- VI. **Spring Transaction (TX) Management**
 - A. Overview
 - B. Declarative TX Management (REQUIRED, etc.)
 - C. TX Scope and Propagation
 - D. Pointcut-based Configuration of Transactions
- VII. **Spring Web Integration and Intro to Spring MVC**
 - A. Java EE Web App Integration
 - B. ContextLoaderListener and WebApplicationContext
 - C. Web MVC Overview
 - D. Spring MVC Basics
 1. Configuration and the DispatcherServlet
 2. @Controller, @RequestMapping (Handlers)
 3. @RequestParam and Parameter Binding
 4. View Resolvers
 5. Controller Details - @RequestParam, @PathVariable
 6. Model Data and @ModelAttribute
- VIII. **Additional Spring MVC Capabilities**
 - A. @ModelAttribute and Reference Data
 - B. Forms and Binding, Spring Form Tags
 - C. Sessions and @SessionAttributes
 - D. Validation / JSR-303
- IX. **RESTful Services with Spring**
 - A. REST Overview and Principles
 - B. REST and Spring MVC
 1. Spring support for REST
 2. @RequestMapping/@PathVariable, @RequestBody, @ResponseBody
 3. URI Templates and @PathVariable
 4. Controllers with @RestController
 - C. Requests and Responses
 - D. Ajax Overview
- X. **Working with JSON and XML**
 - A. Generating JSON
 1. JSON Overview
 2. JSON Representations for Resources
 3. Message Converters
 - B. Generating XML
 1. JAXB and Jackson Message Converters for XML
 2. JAXB / @XmlRootElement
 - C. Content Negotiation
- XI. **Java Clients for RESTful Services**
 - A. Client Requirements and Spring's RestTemplate
 - B. getObject() / getEntity()
 - C. Other RestTemplate Methods
 - D. Accessing Headers / exchange()
- XII. **Common REST Patterns**
 - A. GET: Read
 - B. POST: Create
 - C. PUT: Update
 - D. DELETE: Delete
 - E. Programming on server side, and client side (with RestTemplate)
- XIII. **Additional New Features in Spring 5**
 - A. Updates to Spring Core
 - B. WebFlux / Reactive Web Framework
- XIV. **[XML Specific Configuration (Optional)]**
 - A. Collections - lists, sets, etc.
 - B. Additional Capabilities
 1. Factory Classes and Factory Methods
 2. Definition Inheritance (Parent Beans)
 3. AutoWiring with XML
 4. Inner Beans, Compound Names