

Introduction to Spring 4, Spring MVC, and Spring REST

Course Summary

Description

Over the years, Spring has added many new capabilities, in line with the increasing capabilities of the Java platform. As well as being current with Spring 4, this course introduces techniques for using these powerful capabilities. It includes complete 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. It also provides guidelines for when and how to use each one.

The course starts with in-depth coverage on using the powerful capabilities 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 integrating persistence layers (e.g. Hibernate/JPA) with Spring, using Spring's powerful Aspect Oriented Programming (AOP) to program cross-cutting concerns in a safe and maintainable way, and using Spring's declarative transaction capabilities.

The course includes integration of Spring with Java EE Web applications, and an introduction to Spring MVC. Spring MVC utilizes a Model-View-Controller pattern for building Web applications, and the introduction covers the basics of Spring MVC, and how it supports organizing your Web applications in a highly structured, loosely coupled manner. This includes an introduction to REST (Representational state transfer), and how to use Spring MVC to build RESTful resources and invoke them from Ajax-based front ends.

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.

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 can be made available for all major development environments, including IBM RAD and IntelliJ.

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
- Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as Hibernate or JPA.
- Understand and use Spring's powerful AOP capabilities for programming cross-cutting concerns across multiple points in an application
- Learn safe and maintainable techniques for programming with AOP
- 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 REST, and use Spring MVC to build RESTful services
- Use Ajax-based front ends with Spring MVC / Spring REST

Introduction to Spring 4, Spring MVC, and Spring REST

Course Summary (cont'd)

Topics

- Introduction to Spring
- Configuration in Depth
- Wiring in Depth
- Database Access with Spring
- Aspect Oriented Programming (AOP)
- Spring Transaction (TX) Management
- Spring Web Integration and Intro to Spring MVC
- Additional Spring MVC Capabilities
- RESTful Services with Spring
- XML Specific Configuration
- Appendix: Maven and Spring

Prerequisites

A working knowledge of Java is required. Topics for integrating Hibernate and JPA with Spring require a working knowledge of Hibernate and/or JPA.

Duration

Five days

Introduction to Spring 4, Spring MVC, and Spring REST

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) Examining Dependencies
 - 1. Dependency Inversion
 - 2. Dependency Injection (DI) in Spring - Basic Configuration and Usage

II. Configuration in Depth

- A. Annotation Driven Configuration •JSR 330 (@Named) and Spring (@Component) annotation styles
 - 1. @Named/@Component, @Inject/@Autowired, @Repository, @Service
 - 2. Configuring Beans and Autowiring with Annotations
 - 3. Enabling Annotations - context:component-scan
 - 4. Pros and Cons
- B. Java Based Configuration (@Configuration) Overview - code-centric Configuration
 - 1. @Configuration and @Bean

- 2. Dependency Injection
- 3. Resolving Dependencies on Other Beans, Injecting Configuration Classes
- 4. Pros and Cons
- C. Integrating Configuration Types
 - 1. Choosing a Configuration Style
 - 2. Integrating Configuration Styles
 - 3. Importing: @Import and
 - 4. Scanning with @Configuration style
- D. Bean Scope and Lifecycle
 - 1. Bean Scope Defined - singleton, prototype, and Other Scopes
 - 2. Configuring Scope
 - 3. Bean Creation Lifecycle, Lifecycle Callbacks
 - 4. BeanPostProcessor, Event Handling

III. Wiring in Depth

- A. Value Injection
 - 1. Configuring Value Properties, Property Conversions
 - 2. Externalizing Values in Properties Files
 - 3. Constructor Injection
 - 4. Constructor Injection Overview
 - 5. Configuration - @Configuration and XML
 - 6. p: and c: namespaces for XML configuration
- B. Qualifiers / Domain Specific Language (DSL) Limitations of Autowiring
 - 1. Qualifiers and DSL
 - 2. Creating and Using an Annotation-Based DSL for Bean Configuration
 - 3. Benefits of Qualifiers for Bean Configuration
- C. Profiles
 - 1. Profiles Overview
 - 2. Configuring Profiles (XML and @Configuration)
 - 3. Activating Profiles
- D. Overview of SpEL

Introduction to Spring 4, Spring MVC, and Spring REST

Course Outline (cont'd)

IV. Database Access with Spring

- 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
 - 4. Using Spring with JPA
 - 5. Managing the EntityManager (EM)
 - 6. LocalContainerEntityManagerFactoryBean and Container-managed EMs
 - 7. JEE and JNDI Lookup of the EM
 - 8. Configuration and Vendor Adaptors
 - 9. Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext

V. Aspect Oriented Programming (AOP)

- A. Overview of AOP
 - 1. Crosscutting Concerns
 - 2. AOP Basics, Aspect, Joinpoint, Advice, Pointcut
- B. Spring AOP Introduction
 - 1. Configuration - XML and @AspectJ
 - 2. Defining an Aspect, Pointcut, and Advice
 - 3. How Advice is Triggered
- C. Pointcut Expressions and Advice
 - 1. Pointcut Expression Overview
 - 2. The execution() Designator
 - 3. Other Designators (within, target, args, @target, ...)
 - 4. Kinds of Advice - before, after, around, after-returning, after-throwing
- D. Marker Annotations (Rubber Stamp AOP)
 - 1. Issue with AOP Configuration
 - 2. Defining an AOP Marker / Rubber Stamp
 - 3. Configuring AOP Using a Marker

- 4. Advantages of Marker Annotations

- E. @AspectJ Based AOP Support
- F. @AspectJ Annotations Overview
 - 1. Defining an Aspect, Pointcut, and Advice
- G. Other Considerations
 - 1. Spring AOP Proxies and Self-Invocation Issues
 - 2. Load-Time Weaving
 - 3. Caveats of AOP

VI. Spring Transaction (TX) Management

- A. Intro to Spring Transaction Management
 - 1. Spring Transaction Managers
 - 2. Spring Declarative TX Management
 - 3. Spring TX Scope and Propagation
 - 4. Spring TX Attributes (REQUIRED, SUPPORTS, etc)
- B. XML Configuration of Transactions
 - 1. Specifying Advice, TX Attributes, and Methods
 - 2. Linking Advice with Pointcuts
 - 3. Benefits of XML Configuration of TX Behavior

VII. Spring Web Integration and Intro to Spring MVC

- A. Integrating Spring with Java EE Web Apps - ContextLoaderListener and WebApplicationContext
- B. Spring Web MVC Overview, Capabilities, Architecture (Front Controller, MVC Pattern)
- C. Spring MVC Basics
 - 1. DispatcherServlet, Configuration (@EnableWebMvc, Servlet 3 initialization), mvc Namespace
 - 2. Controllers, @Controller, @RequestMapping (Handler Methods)
 - 3. @RequestParam and Parameter Binding

Introduction to Spring 4, Spring MVC, and Spring REST

Course Outline (cont'd)

- 4. View Resolvers Controller Details,
@RequestMapping,
@RequestParam, @PathVariable
 - 5. Model Data, @ModelAttribute,
Model/ModelAndView Classes
- VIII. Additional Spring MVC Capabilities**
- A. Reference Data with @ModelAttribute
 - B. Forms and Binding, Spring Form Tags
 - C. Session Attributes,
@SessionAttributes
 - D. Validation / JSR-303
- IX. RESTful Services with Spring**
- A. REST Overview
(Characteristics/Capabilities, URI
Templates, REST vs SOAP)
 - B. REST and Spring MVC
 - 1. Spring support for REST
 - 2. @RequestMapping/@PathVariable,
@RequestBody,
@ResponseBody, HTTP Method
conversion
 - 3. Writing RESTful Controllers
 - 4. Returning XML and JSON data
 - C. Client-side Access to RESTful
Services
 - 1. Ajax access
(Browser/JavaScript/jQuery)
 - 2. Using Spring's RestTemplate
 - D. Programming Common REST
Patterns
 - 1. GET: Read
 - 2. POST: Create
 - 3. PUT: Update
 - 4. DELETE: Delete
- X. XML Specific Configuration**
- A. Collection Valued Properties -
Configuring and using 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
- XI. Appendix: Maven and Spring**