

**Introduction to Spring 5, Spring MVC, and Spring REST**

**Course Number:** SPRG-212
**Duration:** 5 days

**Overview**

This Spring 5, Spring MVC, and Spring REST training teaches attendees how to build web applications and RESTful services using primarily Core Spring, but leveraging Spring Boot for dependency management and auto-configuration. This course covers building web applications with Spring MVC and services with Spring REST.

**Prerequisites**

All students must have working knowledge of Java programming, including use of inheritance, interfaces, and exceptions.

**Materials**

All Spring 5 training attendees receive comprehensive courseware covering all subjects in the course.

**Software Needed on Each Student PC**

* JDK 8 or later (required for Spring 5)
* IntelliJ IDEA, Eclipse with Spring Tools, or another IDE of your choice
* Tomcat 8 or later (or another servlet container, upon request)
* Related lab files that Accelebrate provides
* Other free software - please contact us if you have purchased this class

**Objectives**

* 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
* Understand and use Boot’s auto-configuration
* Customize Boot’s behavior with properties and in other ways
* Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as 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
* Build Web applications with Spring MVC, including configuration using Java config and Servlet 3 capabilities
* Understand REST, and use Spring REST to build RESTful services
* Use Ajax-based front ends with Spring REST
* Use RestTemplate to create Java REST clients

**Outline**

* Introduction to Spring
	+ Overview of Spring Technology
		- Motivation for Spring, Spring Architecture
		- The Spring Framework
		- maven and Spring
	+ Spring Introduction
		- Declaring and Managing Beans
		- ApplicationContexts - The Spring Container
		- XML and @Component/@Named Config
	+ Dependencies and Dependency Injection (DI)
		- Examining Dependencies
		- Dependency Inversion / Dependency Injection (DI)
		- DI in Spring - XML and @Autowired
	+ Spring Boot Quickstart
* Configuration in Depth
	+ Java Based Configuration (@Configuration)
		- Overview, @Configuration, @Bean
		- Dependency Injection
		- Resolving Dependencies
	+ Integrating Configuration Types
		- XML and @Component Pros/Cons
		- @Configuration Pros/Cons
		- Choosing a Configuration Style
		- Integrating with @Import and <import>
	+ Bean Scope and Lifecycle
		- Singleton, Prototype, and Other Scopes
		- Configuring Scope
		- Bean Lifecycle / Callbacks
* Spring Boot Overview
	+ Spring Boot Overview
	+ Spring POMs with Boot Parents
	+ Spring Boot Starters
	+ SpringApplication – Apps With main()
	+ CommandLineRunner and ApplicationRunner
	+ Working with Properties
		- Boot Property Files
		- Using Application Properties
		- Customizing Behavior with Boot Properties
* Spring Testing
	+ Testing and JUnit 5 Overview
		- Writing Tests - Test Classes, asserts, Naming Conventions
		- Running Tests - IDE, maven, ...
		- Test Fixtures - setup and teardown
	+ Spring TestContext Framework
		- Overview
		- Configuration
		- Running Tests
* Database Access with Spring/Boot
	+ Overview of Spring/Boot database support
		- DataSources, Boot Auto-Configuration, and Custom Configuration
		- Boot - Embedded Database
	+ Using Spring/Boot with JPA
		- Spring Boot Auto-Configuration and Scanning
		- Customizing the Configuration
		- Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext
	+ Spring Data Overview
		- Overview and Architecture
		- Configuring Spring Data
		- Repositories and JPA Repositories
		- Using CrudRepository
	+ Using Spring Data
		- Naming Conventions for Querying
		- Creating more Complex Queries
		- Query Configuration
	+ [Optional] Configuration Without Boot
		- Managing the EntityManager (EM)
		- LocalContainerEntityManagerFactoryBean and Container-managed EMs
		- JEE and JNDI Lookup of the EM
		- Configuration and Vendor Adaptors
* Spring Transaction (TX) Management
	+ Overview
	+ Declarative TX Management (REQUIRED, etc.)
	+ TX Scope and Propagation
	+ Configuration and Boot Auto-Configuration
	+ Pointcut-based Configuration of Transactions
* Web Applications with Spring MVC
	+ Java EE Overview
	+ Spring MVC Basics
		- DispatcherServlet
		- spring-boot-starter-web and auto-configuration
		- @Controller, @RequestMapping (Handlers)
		- @RequestParam and Parameter Binding
	+ View Resolvers
	+ Controller Details - @RequestParam, @PathVariable
	+ Model Data and @ModelAttribute
* More Spring MVC Capabilities
	+ @ModelAttribute and Reference Data
	+ Forms and Binding, Spring Form Tags
	+ Sessions and @SessionAttributes
	+ Validation / JSR-303
	+ External Server Deployment
	+ "Classic" Spring MVC Configuration (without Boot)
* RESTful Services with Spring
	+ REST Overview and Principles
	+ Requests and Responses - GET, POST, PUT, DELETE
	+ Spring's REST API
		- Spring support for REST (MVC-based)
		- @RequestMapping/@PathVariable, @RequestBody, @ResponseBody
		- URI Templates and @PathVariable
		- Controllers with @RestController
	+ Ajax Overview
* Generating JSON
	+ - JSON Overview
		- JSON Representations for Resources
		- Message Converters
	+ [Optional] Generating XML
		- JAXB and Jackson Message Converters for XML
		- JAXB / @XmlRootElement
	+ Content Negotiation
* Java Clients for RESTful Services
	+ Client Requirements and Spring's RestTemplate
	+ getForObject() / getForEntity()
	+ Other RestTemplate Methods
	+ Accessing Headers / exchange()
* Common REST Patterns
	+ GET: Read
	+ POST: Create
	+ PUT: Update
	+ DELETE: Delete
	+ Programming on server side, and client side (with RestTemplate)
	+ External Server Deployment
* Conclusion