

**Introduction to the Spring 5 Framework**

**Course Number:** SPRG-204
**Duration:** 5 days

**Overview**

Accelebrate's Spring 5 training teaches attendees how to build Spring applications and RESTful services using the Spring framework.

**Prerequisites**

All attendees must have at least one year of full-time Java and JSP development experience. Prior experience with servlets and JSP is recommended but not required.

**Materials**

All 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
* 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

**Outline**

* Introduction to Spring
	+ Overview of Spring Technology
		- The Motivation for Spring, Spring Architecture
		- The Spring Framework
	+ Spring Introduction
		- Declaring and Managing Beans
		- ApplicationContexts
		- Using @Component/@Named
	+ Dependencies and Dependency Injection (DI)
		- Examining Dependencies
		- Dependency Inversion / Dependency Injection (DI)
		- XML Configuration of DI
		- Injection with @Autowire
* Configuration in Depth
	+ Java Based Configuration (@Configuration)
		- Overview, @Configuration, @Bean
		- Dependency Injection
		- Resolving Dependencies
	+ Integrating Configuration Types
		- XML vs. @Component
		- @Configuration Pros and Cons
		- Choosing a Configuration Style
		- Integrating Configuration Styles
	+ Bean Scope and Lifecycle
		- Singleton, Prototype, and Other Scopes
		- Configuring Scope
		- Bean Lifecycle / Callbacks
	+ Externalizing Properties
		- Properties Files
		- @PropertySource, property-placeholder
		- Using @Value
		- SpEL
	+ Profiles
		- Overview and Configuration
		- Activating Profiles
* Spring Boot Overview
	+ maven and Spring
	+ Spring Boot Structure
	+ Spring POMs with Boot Parents
	+ Spring Boot Starters
	+ Other Capabilities
* Spring Testing
	+ Testing and JUnit Overview
		- Writing Tests - Test Classes, asserts, Naming Conventions
		- Running Tests - IDE, maven, ...
		- Test Fixtures - setup and teardown
	+ Spring TestContext Framework
		- Overview
		- Configuration
		- Running Tests
* Spring and Spring Data with Hibernate/JPA
	+ Overview of Spring database support
	+ Configuring a DataSource
	+ Using Spring with Hibernate
		- High-Level Hibernate Overview
		- SessionFactory configuration, LocalSessionFactoryBean
		- Contextual Sessions and Spring Integration
	+ Using Spring with JPA
		- Managing the EntityManager (EM)
		- LocalContainerEntityManagerFactoryBean and Container-managed EMs
		- JEE and JNDI Lookup of the EM
		- Configuration and Vendor Adaptors
		- Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext
	+ Spring Data Introduction
		- Overview and Architecture
		- Configuring Spring Data
		- Repositories and JPA Repositories
		- Using CrudRepository
	+ Spring Data Querying
		- Naming Conventions for Querying
		- Creating more Complex Queries
		- Query Configuration
* Spring Transaction (TX) Management
	+ Overview
	+ Declarative TX Management (REQUIRED, etc.)
	+ TX Scope and Propagation
	+ XML Configuration of Transactions
		- Specifying Advice, TX Attributes, and Methods
		- Linking Advice with Pointcuts
		- Benefits of XML Configuration of TX Behavior
* Spring Web Integration and Intro to Spring MVC
	+ Java EE Web App Integration
	+ ContextLoaderListener and WebApplicationContext
	+ Web MVC Overview
	+ Spring MVC Basics
		- Configuration and the DispatcherServlet
		- @Controller, @RequestMapping (Handlers)
		- @RequestParam and Parameter Binding
		- View Resolvers
		- Controller Details - @RequestParam, @PathVariable
		- Model Data and @ModelAttribute
* Additional Spring MVC Capabilities
	+ @ModelAttribute and Reference Data
	+ Forms and Binding, Spring Form Tags
	+ Sessions and @SessionAttributes
	+ Validation / JSR-303
* RESTful Services with Spring
	+ REST Overview, URI Templates
	+ REST and Spring MVC
	+ Spring support for REST
	+ @RequestMapping/@PathVariable, @RequestBody, @ResponseBody
	+ URI Templates and @PathVariable
	+ Controllers with @RestController
	+ Generating JSON
		- JSON Overview
		- JSON Representations for Resources
		- Message Converters
	+ Generating XML
		- JAXB and Jackson Message Converters for XML
		- JAXB / @XmlRootElement
	+ Content Negotiation
* Working with JSON and XML
	+ Generating JSON
		- JSON Overview
		- JSON Representations for Resources
		- Message Converters
	+ 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)
* Additional New Features in Spring 5
	+ Updates to Spring Core
	+ WebFlux / Reactive Web Framework
* XML-Specific Configuration (Optional)
	+ Collections - lists, sets, etc.
	+ Additional Capabilities
		- Factory Classes and Factory Methods
		- Definition Inheritance (Parent Beans)
		- AutoWiring with XML
		- Inner Beans, Compound Names
* Conclusion