ProTech Professional Technical Services, Inc.

protechtraining.com

Course Outline

Introduction to Spring 5, Spring Boot, and Spring REST (2022)

Course Summary

Description

Spring 5 provides an evolutionary advance of Spring’s powerful capabilities. This course introduces the many Spring Core capabilities, as well as providing guidelines on when and how to use them. It also goes into considerable depth on Spring Boot for dependency management and auto-configuration, as well as Spring REST for creating RESTful resources.

This course has been completely revised to utilize Spring Boot's easy configuration and auto-configuration wherever possible. "Classic" Spring configuration (usually more verbose and complicated) is optionally covered in abbreviated form.

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 JPA persistence layers with Spring and Spring Data, and using Spring’s declarative transaction capabilities.

The course includes a solid introduction to Spring REST, and coverage of building RESTful resources. It also covers many of the details of Spring Boot, including how to create Boot-based POMs (maven) for simplified dependency management, customizing Boot behavior, and understanding/managing Boot’s auto-configuration.

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

After taking 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
- 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
- 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

Topics

- Introduction to Spring
- Configuration in Depth
- Spring Boot Overview
- Spring Testing
- Database Access with Spring/Boot
- Spring Transaction (TX) Management
- RESTful Services with Spring
- Working with JSON and XML
- Java Clients for RESTful Services
- Common REST Patterns
- Boot Configuration and Customization
- Spring Boot Web/Security
- Additional Spring/Boot Features

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.

Introduction to Spring 5, Spring Boot, and Spring REST (2022)

Course Summary (cont)

Audience

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

Prerequisites

Working knowledge of Java programming, including use of inheritance, interfaces, and exceptions.

Duration

Five Days

Introduction to Spring 5, Spring Boot, and Spring REST (2022)

Course Outline

I. Introduction to Spring

- A. Overview of Spring Technology
- B. Motivation for Spring, Spring Architecture
- C. The Spring Framework
- D. maven and Spring
- E. Spring Introduction
- F. Declaring and Managing Beans
- G. ApplicationContexts - The Spring Container
- H. XML and @Component/@Named Config
- I. Dependencies and Dependency Injection (DI)
- J. Examining Dependencies
- K. Dependency Inversion / Dependency Injection (DI)
- L. DI in Spring - XML and @Autowired
- M. Spring Boot Quickstar

II. Configuration in Depth

- A. Java Based Configuration (@Configuration)
- B. Overview, @Configuration, @Bean
- C. Dependency Injection
- D. Resolving Dependencies
- E. Integrating Configuration Types
- F. XML and @Component Pros/Cons
- G. @Configuration Pros/Cons
- H. Choosing a Configuration Style
- I. Integrating with @Import and <import>
- J. Bean Scope and Lifecycle
- K. Singleton, Prototype, and Other Scopes
- L. Configuring Scope
- M. Bean Lifecycle / Callbacks

III. Spring Boot Overview

- A. Spring Boot Overview
- B. Spring POMs with Boot Parents
- C. Spring Boot Starters
- D. SpringApplication – Apps With main()
- E. CommandLineRunner and ApplicationRunner
- F. Working with Properties
- G. Boot Property Files
- H. Using Application Properties
- I. Customizing Behavior with Boot Properties

IV. Spring Testing

- A. Testing and JUnit 5 Overview
- B. Writing Tests - Test Classes, asserts, Naming Conventions
- C. Running Tests - IDE, maven, ...
- D. Test Fixtures - setup and teardown
- E. Spring TestContext Framework
- F. Overview
- G. Configuration
- H. Running Tests

V. Database Access with Spring/Boot

- A. Overview of Spring/Boot database support
- B. DataSources, Boot Auto-Configuration, and Custom Configuration
- C. Boot - Embedded Database
- D. Using Spring/Boot with JPA
- E. Spring Boot Auto-Configuration and Scanning
- F. Customizing the Configuration
- G. Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext
- H. Spring Data Overview
- I. Overview and Architecture
- J. Configuring Spring Data
- K. Repositories and JPA Repositories
- L. Using CrudRepository
- M. Using Spring Data
- N. Naming Conventions for Querying
- O. Creating more Complex Queries
- P. Query Configuration
- Q. [Optional] Configuration Without Boot
- R. Managing the EntityManager (EM)
- S. LocalContainerEntityManagerFactoryBean and Container-managed EMs
- T. JEE and JNDI Lookup of the EM
- U. Configuration and Vendor Adaptors

Introduction to Spring 5, Spring Boot, and Spring REST (2022)

Course Outline (cont.)

VI. Spring Transaction (TX) Management

A. Overview
B. Declarative TX Management (REQUIRED, etc.)
C. TX Scope and Propagation
D. Configuration and Boot Auto-Configuration
E. Pointcut-based Configuration of Transactions

VII. RESTful Services with Spring

A. REST Overview and Principles
B. DispatcherServlet - Boot Auto-Config and Customization
C. Requests and Responses - GET, POST, PUT, DELETE
D. Spring's REST API
E. Spring support for REST
F. @RequestMapping/@PathVariable, @RequestBody, @ResponseBody
G. URI Templates and @PathVariable
H. Controllers with @RestController
I. Ajax Overview

VIII. Working with JSON and XML

A. Generating JSON
B. JSON Overview
C. JSON Representations for Resources
D. Message Converters
E. [Optional] Generating XML
F. JAXB and Jackson Message Converters for XML
G. JAXB / @XmlRootElement
H. Content Negotiation

IX. Java Clients for RESTful Services

A. Client Requirements and Spring's RestTemplate
B. getForObject() / getForEntity()
C. Other RestTemplate Methods
D. Accessing Headers / exchange()

X. 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)

XI. Boot Configuration and Customization

A. Logging and its Configuration
B. Profiles
C. Other Configuration

XII. Spring Boot Web/Security

A. Spring Boot Web
B. Boot's Embedded Servers
C. Classic Spring MVC Configuration
D. Spring Boot Security
E. Spring Boot Data REST

XIII. Additional Spring/Boot Features

A. Updates to Spring Core
B. WebFlux / Reactive Web Framework
C. Boot Actuator