Introduction to Spring 5, Spring MVC and Spring REST

Duration: 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

Price: $2495 *(Face-to-Face & Remote-Live)*, or $895 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom, remote-live or via on-demand training.

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 Springs reactive programming model for repositories and Web resources.

This course is hands on with labs throughout the course 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.

Students desiring coverage of Spring Boot rather than Spring MVC may instead be interested in attending the Introduction to Spring 5, Spring Boot and Spring REST course.

Prerequisites

Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML is helpful but not required. The course Java Programming or equivalent knowledge provides a solid foundation.

Course Overview

Introduction to Spring

- Overview of Spring Technology
  - Motivation for Spring, Spring Architecture
  - The Spring Framework
- 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`

Configuration in Depth

- Java Based Configuration (@Configuration)
  - Overview of `@Configuration` and `@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 and Callbacks
- Externalizing Properties
  - Properties Files
  - `@PropertySource` property-placeholder
  - Using `@Value`
  - SpEL
- Profiles
  - Overview and Configuration
  - Activating Profiles

Spring Boot Overview

- Maven and Spring

Spring Testing

- Testing and JUnit Overview

- Spring Boot Structure
- Spring POMs with Boot Parents
- Spring Boot Starters
    - 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 and 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
- Pointcut-Based Configuration of Transactions

Spring Web Integration and Introduction to Spring MVC

- Java EE Web App Integration
- ContextLoaderListener and WebApplicationContext
- Web MVC Overview
- Spring MVC Basics
    - Configuration and the DispatcherServlet

Additional Spring MVC Capabilities

- `@ModelAttribute` and Reference Data
- Forms and Binding, Spring Form Tags
- Sessions and `@SessionAttributes`

`@Controller`, `@RequestMapping` (Handlers)
- `@RequestParam` and Parameter Binding
- View Resolvers
- Controller Details: `@RequestParam and @PathVariable`
- Model Data and `@ModelAttribute`

Validation / JSR-303

RESTful Services with Spring

- REST Overview, URI Templates
- REST and Spring MVC
  - Spring Support for REST
  - `@RequestMapping`/`@PathVariable`, `@RequestBody and @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

Java Clients for RESTful Services

- Client Requirements and Spring's RestTemplate
- `getForObject()` and `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)

What's New in Spring 5

- Updates to Spring Core
- WebFlux
- Reactive Web Framework

XML Specific Configuration

- Collections: Lists, Sets, etc.
- Additional Capabilities
  - Factory Classes and Factory Methods
  - Definition Inheritance (Parent Beans)
  - AutoWiring with XML
  - Inner Beans and Compound Names

Copyright © 2021 Software Skills Training, Inc.