

Pivotal - Core Spring

Code:	SP-CORE
Length:	4 days
URL:	View Online

This course offers hands-on experience with Spring and its major features, including configuration, data access, web and REST applications, Spring Boot, Spring Security and using Spring Cloud to build a small microservices application. On completion, participants will have a foundation for creating enterprise-ready applications. This course prepares students for the Spring Professional certification exam. Certification exams are sold separately.

This course helps prepare students for the Spring Professional Certification Exam.

One (1) Spring Professional exam attempt within 90 days of the training is included in the cost of the Core Spring course. Certification candidates may also purchase the exam separately at <https://pivotal.io/training/certification>.

Skills Gained

Upon completion of this course, participants will understand how to implement the following:

- Spring configuration using Java Configuration and Annotations
- Aspect oriented programming with Spring
- Testing Spring applications using JUnit 5
- Spring Data Access - JDBC, JPA and Spring Data
- Spring Transaction Management
- Building Web Applications with Spring MVC
- Setup Spring applications with Spring Boot
- REST with Spring MVC and RestTemplate
- Spring Security
- Microservices with Spring Cloud
- Reactive Programming with Spring

Who Can Benefit

Application developers who want to increase their understanding of Spring with hands-on experience and a focus on fundamentals.

Prerequisites

- Basic understanding of application development using Java
- IDE (Eclipse or STS preferred); STS is used in the course

Course Details

Course Modules

Course Modules

INTRODUCTION TO SPRING

- Java configuration and the Spring application context
- @Configuration and @Bean annotations
- @Import: working with multiple configuration files
- Defining bean scopes
- Launching a Spring Application and obtaining Beans

SPRING JAVA CONFIGURATION: A DEEPER LOOK

- External properties & Property sources
- Environment abstraction
- Using bean profiles
- Spring Expression Language (SpEL)
- How it Works: Inheritance based proxies

ANNOTATION-BASED DEPENDENCY INJECTION

- Autowiring and component scanning
- Java configuration versus annotations, mixing.
- Lifecycle annotations: @PostConstruct and @PreDestroy
- Stereotypes and meta-annotations

FACTORY PATTERN IN SPRING

- Using Spring FactoryBeans

ADVANCED SPRING: HOW DOES SPRING WORK INTERNALLY?

- The Spring Bean Lifecycle
- The BeanFactoryPostProcessor interception point
- The BeanPostProcessor interception point
- Spring Bean Proxies
- @Bean method return types

TESTING A SPRING-BASED APPLICATION

- Spring and Test Driven Development
- Brief overview of JUnit 5
- @ContextConfiguration and @RunWith annotations
- Application context caching and the @DirtiesContext annotation
- Profile selection with @ActiveProfiles
- Easy test data setup with @Sql

ASPECT-ORIENTED PROGRAMMING

- What problems does AOP solve?
- Differences between Spring AOP and AspectJ
- Defining pointcut expressions
- Implementing an advice: @Around, @Before, @After

DATA ACCESS AND JDBC WITH SPRING

- How Spring integrates with existing data access technologies
- DataAccessException hierarchy
- Implementing caching using @Cacheable
- Embedded databases for testing
- Spring's JdbcTemplate

DATABASE TRANSACTIONS WITH SPRING

- Transactions overview
- Transaction management with Spring
- Isolation levels, transaction propagation and rollback rules
- Transactions and integration testing

JPA WITH SPRING AND SPRING DATA

- Quick introduction to ORM with JPA
- Benefits of using Spring with JPA
- JPA configuration in Spring

SPRING BOOT

- Using Spring Boot to bypass most configuration
- Simplified dependency management with starter POMs
- Easily overriding Spring Boot defaults

ADVANCED SPRING JPA

- Configuring Spring JPA using Spring Boot
- Spring Data JPA dynamic repositories

SPRING IN A WEB APPLICATION

- Configuring Spring in a Web application
- Introduction to Spring MVC, required configuration
- Controller method signatures
- Views and ViewResolvers
- Using @Controller and @RequestMapping annotations
- Configuring Spring MVC with Spring Boot
- Spring Boot packaging options, JAR or WAR

SPRING BOOT - GOING FURTHER (OPTIONAL)

- Going beyond the default settings
- Customizing Spring Boot configuration
- Logging control
- Configuration properties using YAML
- Boot-driven testing

REST WITH SPRING MVC

- An introduction to the REST architectural style
- Controlling HTTP response codes with `@ResponseStatus`
- Implementing REST with Spring MVC, `@RequestBody`, `@ResponseBody`
- Spring MVC's `HttpMessageConverters` and automatic content negotiation

SPRING SECURITY

- What problems does Spring Security solve?
- Configuring authentication and intercepting URLs
- The Spring Security tag library for JSPs
- Security at the method level
- Understanding the Spring Security filter chain

REST WITH SPRING MVC

- An introduction to the REST architectural style
- Controlling HTTP response codes with `@ResponseStatus`
- Implementing REST with Spring MVC, `@RequestBody`, `@ResponseBody`
- `ResponseBody`
- Spring MVC's `HttpMessageConverters` and automatic content negotiation

MICROSERVICES WITH SPRING CLOUD

- Microservice Architectures
- Challenges with cloud-native applications
- Using Spring Cloud
- Developing a simple microservice system

REACTIVE APPLICATIONS WITH SPRING

- Overview of Reactive Programming concepts
- Reactive Programming support in Spring
- Using Spring's reactive `WebClient`

Corporation and Tech Data Corporation, respectively
Copyright ©2019 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.