Java: Pivotal - Core Spring

Course ID #: 1260-531-SP-W

Hours: 28

Course Content

Course Description:

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.

At Course Completion:

After competing this course, student will be able to:

- 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

Target Student:

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

Topics:

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

Java: Pivotal - Core Spring

Course ID #: 1260-531-SP-W

Hours: 28

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 Lufecycle
- 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

Java: Pivotal - Core Spring

Course ID #: 1260-531-SP-W

Hours: 28

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
- 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