DATA SHEET

REVISED: 02.04.15

Core Spring

DELIVERY METHODS

- Instructor-led
- Live-online
- On-site training

COURSE DURATION

- Four days of instructor-led training
- 50% lecture, 50% hands-on lab

TARGET AUDIENCE

- Developers
- Architects

PREREQUISITES

Experience with developing applications using Java

PRICING

Please visit our website at pivotal.io/training

MORE INFORMATION

On-site training is also available for customers who prefer to bring a Pivotal Certified Instructor to their own facilities For additional information about on-site classes, including facility requirements, contact education@pivotal.io

COURSE OVERVIEW

Core Spring is the four-day flagship Spring Framework training. In this course, students build a Spring-powered Java application that demonstrates the Spring Framework and other Spring technologies like Spring Boot, Spring Data, AOP and Spring Security in an intensely productive, hands-on setting.

Completion of this training entitles each student to receive a free voucher to schedule an exam at a Pearson VUE Center to become a Spring Certified Professional.

COURSE OBJECTIVES

At the end of the training, you should have an understanding of Spring and associated technologies and be able to do the following:

- Use the Spring Framework to develop Java applications.
- Use dependency injection to set up and configure applications.
- Test Spring-based applications.
- Configure Spring applications using Java configuration, annotations, or XML.
- Use Spring Data JPA and JDBC to rapidly implement relational database access.
- Use Spring support for transactions.
- Use aspect-oriented programming (AOP) to declaratively add behavior to applications.
- Develop a basic Web application with Spring MVC.
- Use Spring Security to secure Web applications.
- Use Spring to easily build REST web services.
- Use Spring to easily work with messaging APIs such as JMS.
- Take the Spring Boot shortcut to productivity

COURSE MODULES

1. INTRODUCTION TO SPRING

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

2. SPRING JAVA CONFIGURATION: A DEEPER LOOK

- External properties & Property sources
- Environment abstraction
- Bean scope, bean profiles
- How it Works: Inheritance based proxies

3. ANNOTATION-BASED DEPENDENCY INJECTION

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

4. XML DEPENDENCY INJECTION

- XML syntax, constructor & setter injection
- Resource prefixes
- Factory Bean / Factory Method
- Namespaces and best practices when using
- XML profile selection
- Working with a high number of configuration files
- Bean definition inheritance

5. THE BEAN LIFECYCLE: HOW DOES SPRING WORK INTERNALLY?

- The init phase: available interceptors
- The init phase: what is the difference between XML, annotations and Java configuration?
- What happens during bean post processing
- The use and destruction phases

6. TESTING A SPRING-BASED APPLICATION

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

7. ASPECT-ORIENTED PROGRAMMING

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

8. DATA ACCESS AND JDBC WITH SPRING

- How Spring integrates with existing data access technologies
- DataAccessException hierarchy
- Implementing caching using @Cacheable
- jdbc namespace and Spring‘s JdbcTemplate

9. DATABASE TRANSACTIONS WITH SPRING

- @Transactional annotation
- Transactions configuration: XML versus annotations
- Isolation levels, transaction propagation and rollback rules
- Transactions and integration testing
- Should you use read-only transactions?

10. JPA WITH SPRING AND SPRING DATA

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

11. SPRING IN A WEB APPLICATION

- Configuring Spring in a Web application (using Spring MVC, Struts, JSF...)
- Introduction to Spring MVC, required configuration
- Handler mapping method signatures
- Views and ViewResolvers
- Using @Controller and @RequestMapping annotations

12. 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
- Customizing the Spring Security filter chain

13. SPRING BOOT

- Using Spring Boot to bypass most configuration
- Simplified dependency management with starter POMs
- Packaging options, JAR or WAR
- Easily overriding Spring Boot defaults

14. REST WITH SPRING MVC

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

15 JMS WITH SPRING

- Introduction to the JMS API
- Sending and receiving messages using the JmsTemplate
- Asynchronous message handling with @JmsListener and Listener Containers.

Pivotal

At Pivotal our mission is to enable customers to build a new class of applications, leveraging big and fast data, and do all of this with the power of cloud independence. Uniting selected technology, people and programs from EMC and VMware, the following products and services are now part of Pivotal: Greenplum, Cloud Foundry, Spring, GemFire and other products from the VMware vFabric Suite, Cetas and Pivotal Labs.

Pivotal 3495 Deer Creek Road, Palo Alto, CA 94304 **Pivotal.io**