Core Spring

Duration 4 day(s) (SPRING-CORE-04)

Develop a Java / JEE application with Spring

Certifying Training

Official Training

Zenika exclusivity

Description

As part of their partnership, Zenika invites you to join the official Pivotal courses led by a consultant certified by Pivotal. The cost of Spring certification is not included in the price.

If the trainee wishes to pass the certification, he will have to apply for registration directly on the website:

https://pivotal.io/training/certification

Cost of certification: $ 200

Do not hesitate to contact us for more information.

Goals

- Use the Spring Framework to develop Java applications.
- Configure Spring with Java classes or Annotations.
- Understanding Dependency Injection and Aspect Oriented Programming (AOP)
- Test applications based on Spring, using JUnit5
- Use Spring to access data: JDBC, JPA, and Spring Data
- Use Spring transaction support.
- Develop a web application with Spring MVC.
- Expose and consume REST services with Spring MVC and RestTemplate
- Realize and configure applications faster with spring boot.
- Use Spring Security to secure applications.
- Understanding the challenges of microservices: Introdution at Spring Cloud
- Understanding responsive programming: introduction to Spring Web Reactive

Public

- Architect
- Developer
- Project Manager

Prerequisites

- Knowledge of Java

Structure

40% Theory, 60% Practice

Program

Introduction to Spring

- The Java configuration and the Spring container
- Annotations @Configuration, @Bean, @Import
- The notion of *scope*
- Launch a Spring app and get our *beans*

Spring configuration in Java: in more detail

- Outsourcing of properties, Property Sources
- The notion of Environment
- The notion of bean profile
- Spring Expression Language (SpEL)
- The operation of proxies by inheritance

The Spring configuration by annotations

- Injection and auto-discovery of components
- Java configuration or auto-discovery: when to use them?
- Annotations of the life cycle: @PostConstruct, @PreDestroy
- Annotations stereotypes, meta-annotations
- Factories (design pattern): FactoryBeans

Advanced: Spring Container Operation

- The life cycle of a Spring component
- Post-processors: BeanFactoryPostProcessor, BeanPostProcessor
- Proxies
- The typing of @Bean methods

Test a Spring application

- Spring and TDD (Test Driven Development)
- Quick presentation of JUnit 5
- Spring 5 integration tests with JUnit 5
- Application context caching and @DirtiesContext annotation
- Selection of profiles with @ActiveProfiles
- Easy implementation of data access tests with @SQL

Aspect Oriented Programming (AOP)

- What are the issues resolved by the AOP?
- Difference between spring AOP and AspectJ
- Define aspects with @Around, @Before, @After

Data access and JDBC with Spring

- How Spring integrates with existing data access technologies
- The DataAccessException exception hierarchy
- Caching with @Cacheable
- Facilitate testing with embedded databases Spring's JdbcTemplate

Transaction management with Spring

- The concept of transaction
- Implementation of transactions with Spring
- Insulation levels; strategies for spreading and backtracking transactions
- Transactions in integration tests

JPA with Spring and Spring Data

- Quick introduction to ORM with JPA
- Use JPA with Spring: the benefits
- The implementation of JPA with Spring

Spring Boot

- Minimize configuration with Spring Boot
- Simplify dependency management with *starter POMs*
- Redefine the default configurations of Spring Boot in a simple way

Spring JPA - Advanced

- Configure Spring JPA with Spring Boot
- Spring Data JPA: the automatic implementation of data access (dynamic repositories)

Spring in a web application

- Configure Spring in a web application
- Introduction to Spring MVC and Necessary Components
- Signing methods on controllers
- Views: Views and ViewResolvers
- Annotations @Controller, @RequestMapping
- Spring MVC configuration with Spring Boot
- Creation of a deliverable with Spring Boot: JAR or WAR?

Spring Boot - Advanced (Optional)

- Go beyond the default setting
- Fine customization of Spring Boot configuration
- Setting the log system
- Properties in YAML format
- Spring Boot Tests

Spring Security

- What are the issues resolved by Spring Security?
- Configure authentication and intercept URLs
- Spring Security server side support
- Security at the method level
- Understand the Spring Security filter chain

REST with Spring MVC

- Introduction to REST architecture
- Control HTTP return codes with @ResponseStatus
- Implement REST with Spring MVC and annotations @RequestBody, @ResponseBody
- Automatic content negotiation and HTTP message converters (HttpMessageConverter)

Introduction - Microservices with Spring Cloud (Optional)

- The benefits of a microservice oriented architecture
- New challenges of cloud applications
- Use Spring Cloud
- Develop a simple system with microservices

Introduction - Reactive applications with Spring (Optional)

- The concepts of reactive programming
- Reactive programming support with Spring
- Implement Spring's Reactive WebClient