



## Spring Core

Durée 4 jour(s) (SPRING-CORE-04)

Développer une application Java/JEE avec Spring

Formation Certifiante



Formation officielle



Exclusivité Zenika



### Description

Dans le cadre de leur partenariat, Zenika vous invite à vous joindre aux formations Pivotal officielles animées par un consultant certifié par Pivotal. Le coût de la certification Spring n'est pas inclus dans le tarif.

Si le stagiaire souhaite passer la certification, il devra faire sa demande d'inscription directement sur le site web :

<https://pivotal.io/training/certification>

Cout de la certification : 200\$

N'hésitez pas à nous contacter pour plus d'information.

### Objectifs

- Utiliser le Framework Spring pour développer des applications Java.
- Configurer Spring avec des classes Java ou par Annotations.
- Comprendre l'injection de dépendances et la programmation orientée aspect (AOP)
- Tester des applications basées sur Spring, en utilisant JUnit5
- Utiliser Spring pour accéder aux données: JDBC, JPA, et Spring Data
- Utiliser le support de transaction de Spring.
- Développer une application Web avec Spring MVC.
- Exposer et consommer des services REST avec Spring MVC et RestTemplate
- Réaliser et configurer des applications plus rapidement avec spring Boot.
- Utiliser Spring Security pour sécuriser des applications.
- Comprendre les enjeux des microservices : Introduction à Spring Cloud
- Comprendre la programmation réactive : introduction à Spring Web Reactive

### Public

- Architecte
- Développeur
- Chef de projet

### Prérequis

- Connaissance de Java

### Répartition

40% Théorie, 60% Pratique

## Programme

### Introduction à Spring

- La configuration Java et le conteneur Spring
- Les annotations @Configuration, @Bean, @Import
- La notion de *scope*
- Lancer une application Spring et obtenir nos *beans*

### La configuration Spring en Java : plus en détail

- Externalisation des propriétés, Property Sources
- La notion d'Environnement
- La notion de profil des beans
- Spring Expression Language (SpEL)
- Le fonctionnement des proxys par héritage

### La configuration Spring par annotations

- Injection et auto-découverte des composants
- Configuration Java ou auto-découverte : quand les utiliser?
- Les annotations du cycle de vie : @PostConstruct, @PreDestroy
- Annotations stéréotypes, méta-annotations
- Les fabriques (patron de conception) : FactoryBeans

### Avancé : Fonctionnement du conteneur Spring

- Le cycle de vie d'un composant Spring
- Les post-processeurs : BeanFactoryPostProcessor, BeanPostProcessor
- Les proxies
- Le typage des méthodes @Bean

### Tester une application Spring

- Spring et le TDD (Test Driven Development)
- Présentation rapide de JUnit 5
- Les tests d'intégration Spring 5 avec JUnit 5
- La mise en cache du contexte applicatif et l'annotation @DirtiesContext
- La sélection des profils avec @ActiveProfiles
- Une mise en oeuvre facile des tests d'accès aux données avec @SQL

### La programmation orientée aspect (AOP)

- Quelles sont les problématiques résolues par l'AOP?
- Différence entre spring AOP et AspectJ
- Définir des aspects avec @Around, @Before, @After

### L'accès aux données et JDBC avec Spring

- Comment Spring s'intègre avec les technologies existantes d'accès aux données
- La hiérarchie d'exceptions DataAccessException
- La mise en cache avec @Cacheable
- Faciliter les tests avec des bases de données embarquées
- Le JdbcTemplate de Spring

### La gestion des transactions avec Spring

- La notion de transaction
- Mise en oeuvre des transactions avec Spring
- Niveaux d'isolation; stratégies de propagation et de retour arrière des transactions
- Les transactions dans les tests d'intégration

### JPA avec Spring et Spring Data

- Introduction rapide aux ORM avec JPA
- Utiliser JPA avec Spring : les avantages
- La mise en oeuvre de JPA avec Spring

### Spring Boot

- Réduire la configuration au minimum avec Spring Boot
- Simplifier la gestion des dépendances avec les *starter POMs*
- Redéfinir de façon simple les configurations par défaut de Spring Boot

### Spring JPA - Avancé

- Configurer Spring JPA avec Spring Boot
- Spring Data JPA : l'implémentation automatique des accès aux données (dynamic repositories)

### Spring dans une application Web

- Configurer Spring dans une application Web
- Introduction à Spring MVC et composants nécessaires
- La signature des méthodes sur les contrôleurs
- Les vues : Views et ViewResolvers
- Les annotations @Controller, @RequestMapping
- Configuration de Spring MVC avec Spring Boot
- Création d'un livrable avec Spring Boot : JAR ou WAR?

### Spring Boot - Avancé (Optionnel)

- Aller au-delà du paramétrage par défaut