



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.

La certification n'est pas intégrée dans la formation ; les stagiaires doivent faire leur demande auprès de l'Editeur

- Prépaiement carte bleue, estimation 211 €
- Il s'agit d'un voucher qui donne accès à un QCM (45 questions)
- L'obtention de la certification est validée par l'Editeur
- < <https://www.vmware.com/learning/certification/purchase-vouchers.html>

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

Formation animée en présentiel

La formation en présentiel se déroule sur des jours consécutifs

Formation disponible en mode "formation à distance"

La formation à distance se déroule de préférence sur des jours consécutifs (contactez nous si besoin de décomposer en demies journées) En inter-entreprises comme en intra-entreprises, l'outil de visio-conférence privilégié est Microsoft Teams.

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

Evaluations des acquis

L'évaluation des acquis de la formation se fera en séance au travers d'ateliers, d'exercices et/ou de travaux pratiques. Dans le cas d'une formation officielle éditeur, veuillez nous consulter afin que nous vous fassions part des modalités d'évaluation.

A l'issue de la formation, vous sera transmis une évaluation à chaud de l'action de formation qui vous permettra de nous faire part de vos retours quant à votre expérience apprenant avec Zenika.

Ressources pédagogiques

Les ressources pédagogiques proviennent de productions des équipes Zenika et/ou de la documentation éditeur dans le cas d'une formation "Officielle". Les documents sont en français ou en anglais.

RQTH et ma formation Zenika

Si vous êtes sujet à un handicap, prenez contact avec nos équipes pour que nous puissions définir ensemble comment nous pourrions aménager la session afin que vous puissiez vivre une expérience en formation inchangée.

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
- Personnalisation fine de configuration Spring Boot
- Paramétrage du système de log
- Les propriétés au format YAML
- Les tests Spring Boot

Spring Security

- Quelles sont les problématiques résolues par Spring Security?
- Configurer l'authentification et intercepter les URLs
- Le support Spring Security côté serveur
- Sécurisation au niveau des méthodes
- Comprendre la chaîne de filtrage Spring Security

REST avec Spring MVC

- Introduction à l'architecture REST
- Contrôler les codes de retour HTTP avec @ResponseStatus
- Mettre en oeuvre REST avec Spring MVC et les annotations @RequestBody, @ResponseBody

- Négociation automatique de contenu et convertisseurs de messages HTTP (HttpMessageConverter)

Introduction - Les microservices avec Spring Cloud (Optionnel)

- Les bénéfices d'une architecture orientée microservices
- Les nouveaux défis des application "cloud"
- Utiliser Spring Cloud
- Développer un système simple avec des microservices

Introduction - Les applications réactives avec Spring (Optionnel)

- Les concepts de la programmation réactive
- Le support de la programmation réactive avec Spring
- Mettre en oeuvre le Reactive WebClient de Spring