



Spring Core

Durée 3 jour(s) (SPRING-CORE-04)

Développer une application Java/JEE avec Spring

Exclusivité Zenika



Description

Découvrez notre programme de formation Spring spécialement conçu pour les développeurs, architectes qui cherchent à approfondir leurs compétences. Notre formation allie théorie solide et pratique intensive (60% de travaux pratiques) pour vous fournir des compétences réellement applicables et réutilisables dans votre quotidien. Nous couvrons des sujets essentiels tels que l'injection de dépendances, la création d'API REST avec Spring Boot, l'utilisation de Spring Data JPA, et bien plus, toujours avec une approche orientée vers les dernières évolutions de ces technologies.

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

- Identifier et Expliquer les concepts de base de Spring, Spring Boot et de l'Injection de Dépendances.
- Construire et Configurer une application Spring Boot utilisant Spring Data JPA.
- Explorer et Analyser le fonctionnement interne de Spring.
- Gérer et Optimiser la gestion des Transactions avec Spring.
- Concevoir et Implémenter des applications REST avec Spring Boot, @RestController, et OpenAPI.
- Comparer et Choisir les méthodes d'appel d'une API REST avec Spring Boot.
- Créer et Exécuter des tests unitaires pour des applications Spring Boot.
- Appliquer certains principes de Domain Driven Design pour structurer le code.
- Comprendre et Exploiter les bases de Spring dans les systèmes distribués.

Public

- Architecte
- Développeur
- Chef de projet

Prérequis

- Connaissance de Java
- Lire et comprendre des consignes en Anglais

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.

Introduction à Spring Boot

- Comment le framework Spring a changé la façon de développer les applications Java
- Spring Boot: comment ce framework de "Convention over Configuration" a permis de simplifier l'utilisation de Spring
- Création d'une première application Spring Boot avec start.spring.io
- Mise en place de JUnit et AssertJ pour les tests automatisés

L'Injection de Dépendances

- L'injection de dépendances, à quoi ça sert ? (exemples avec les Singletons, les dépendances contextuelles, Les Proxies)
- Injection par constructeur ou par setter: quelles sont les bonnes pratiques ?
- Les annotations @Scope, @Qualifier et @Value
- Le cycle de vie des applications Spring

La Programmation Orientée Aspect avec Spring

- Présentation de la Programmation Orientée Aspect
- Les Aspects qui sont intégrés dans le framework Spring: Transactions, Security, Caching...
- Mode de fonctionnement de la Programmation Orientée Aspect: comment Spring utilise les Proxies pour l'injection de code
- Créer ses propres Aspects (exemples avec les Logs et la gestion des exceptions)

Accès aux bases de données relationnelles avec Spring Data JPA

- JDBC, JPA, Hibernate, Spring Data: pourquoi tant de couches différentes ?
- Rappels sur l'utilisation de JPA (Entités, Relations, cascade, lazy loading)
- Mise en place Spring Data JPA pour simplifier les interactions JPA
- Préparation aux Microservices : comment utiliser les "loose relationships" et les Agrégats pour avoir des requêtes plus simples, et pouvoir - facilement passer aux microservices si nécessaire

Les Transactions avec Spring

- Transactions programmatiques ou déclaratives : quelle différence ?
- L'approche déclarative de Spring avec @Transactional
- Bonnes pratiques pour la gestion du rollback
- Utilisation des Transactions dans les tests unitaires
- Bonnes pratiques pour le chaînage et la propagation des transactions
- Préparation aux microservices: Transaction IDs et idempotence

Exposer une API REST avec Spring

- Exposition d'un service REST avec Spring MVC (get/post/put/delete)
- Construire une API Spring avec OpenAPI et Swagger-UI
- Validation des requêtes entrantes (avec ou sans l'annotation @Valid)
- Personnalisation de la sérialisation JSON avec les annotations Jackson
- Bonnes pratiques: qu'est ce que le pattern Data Transfer Object et quand faut-il l'utiliser ?
- Gestion des exceptions : comment toujours renvoyer un message clair en cas d'erreur

Tests automatisés et Gestion des Profils

- Passage en revue des outils de Test Unitaires utilisés dans les Travaux Pratiques
- Bonne pratique : utilisation du cache pour que les tests s'exécutent plus rapidement
- Introduction aux Composants Spring Cloud pour les microservices

La certification “Spring-certified professional”

- Présentation de la certification “Spring-certified professional”
- Démarches pour passer la certification
- Utilisation des “Spring Profiles” pour définir des contextes Développement / Tests / Production

Spring Security

- Exposition/Restriction de l'accès aux endpoints avec les requestMatchers
- Rôles et Utilisateurs
- Persistance des utilisateurs en base de données
- Introduction à l'authentification par token JWT

Observabilité

- Différence entre les Logs, les Métriques et les Traces
- Utilisation de Spring Boot Actuator
- Comment surveiller le bon fonctionnement d'une application
- Comment changer le niveau de Logs sans redémarrer le serveur d'applications
- Utilisation de Spring Boot Admin

Introduction aux Microservices avec Spring

- Différence entre Monoliths et Microservices. Faut-il toujours avoir une architecture Microservices ?
- Bonne pratique : organisation du code dès le départ pour un passage facile aux microservices
- Introduction aux Composants Spring Cloud pour les microservices