



Spring Workshop - Les backends Java sans soucis

Durée 3 jour(s) (BACKEND-JAVA)

Apprendre à développer un backend Java

Description

Pour l'élaboration de backends REST modernes, l'écosystème Java demeure un incontournable. En particulier, le framework Spring fournit une bibliothèque de fonctionnalités couvrant l'ensemble des besoins techniques typiquement mis en oeuvre pour la mise en place de backends robustes, et offre une aide substantielle aux développeurs en termes de productivité et d'assistance à la production de code standardisé et conforme à l'état de l'art. Dans cette formation nous allons introduire les bonnes pratiques fondamentales à l'élaboration d'un backend de qualité ainsi qu'un tour d'horizon du framework Spring. Le stagiaire découvrira Spring tout au long du workshop un backend complet en transformant une unique classe rudimentaire en un backend REST complet.

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)

Objectifs

- Un tour d'horizon sur les bonnes pratiques de l'ingénierie Java
 - Principes SOLID
 - Les tests
 - Maven
- Découverte de Spring
 - Fondamentaux (Spring Code)
 - REST à travers Spring MVC
 - Sécurisation des backends (Spring Security)
 - Persistence via Spring Data illustrée à travers Spring Data JPA et Spring Data Elasticsearch
 - Spring Batch
 - Spring Boot

Public

Développeurs

Prérequis

Connaître le langage Java.

Répartition

30% Théorie, 70% 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

Clean code

- Principes SOLID
- Packaging d'application
- Inversion de controle (IOC)

Testing

- Valeur
- Différents types de tests
- Frameworks de test: junit, AssertJ, Mockito

Maven

- Grands principes (artifacts, scopes, organisation, pom.xml)
- Cycle de vie
- Plugins
- Génération de code, illustrée avec OpenAPI

Spring Core

- Injection de dépendances
- Injection de propriétés
- Classes de configurations
- Stéréotypes
- Scopes

Spring MVC

- Modèle MVC
- Mise en oeuvre
- Contrôleurs
- REST avec Spring MVC (binding, gestion des erreurs)

Spring Security

- Mise en oeuvre
- Authentification / autorisation
- Composants techniques essentiels (AuthenticationManager / UserDetailsService)
- Sécurisation du trafic
- Sécurisation du code

Spring Data

- Vision des différents modules Spring Data disponibles
- Notion de Repository
- Fonctionnalités auto-générées
- Query methods
- Repositories composites
- Focus sur Spring Data JPA
- Exemple d'un autre module Spring Data avec Spring Data Elasticsearch

Spring Batch

- Notions de Jobs and Steps
- Steps Tasklet et Steps "chunks"
- ItemReader, ItemProcessor, ItemWriter
- Création d'un job par classe de configuration
- Exécution d'un job

Spring Boot

- Spring Initializr
- Auto configuration
- Configuration via application.yaml
- Serveur web embarqué
- Starter modules (et comment créer le sien)
- Actuator