

# Spring, développer des applications d'entreprise

4 j (28 heures)

Ref : FSPR

## Public

Développeurs Java, architectes Java SE / Java EE, chefs de projets techniques...

## Pré-requis

Niveau : Bonnes connaissances du langage Java

Techniques (formations en classe virtuelle) : Vous devez disposer d'un ordinateur connecté à internet, d'un micro et d'une caméra

## Moyens pédagogiques

Modalité : Formation présentielle ou Formation distancielle (classe virtuelle) - Inter / Intra - Groupes de 4 à 12 stagiaires

Méthodes : Présentation des concepts, discussion technique, démonstrations, exercices et TP

Matériel :

*Présentiel* : Un poste informatique par stagiaire connecté à internet, à une imprimante en réseau et au réseau informatique,

Les salles sont équipées d'un tableau interactif ou d'un vidéoprojecteur et d'un paperboard

*Distanciel* : Aelion met à disposition de chaque stagiaire

- Un PC équipé des outils et logiciels nécessaires à la formation auquel vous accédez via un outil de prise en main à distance

- Un accès à un outil de classe virtuelle (Meet)

Support de formation : Un support de formation sera remis à chaque stagiaire en fin de formation : plateforme collaborative intégrant le code source des exercices réalisés en formation, webographie, mémos

## Modalités de suivi et d'évaluation

Questionnaire d'évaluation des pré-requis, suivi des connaissances tout au long de la formation, Evaluation des acquis en fin de formation

Questionnaire d'évaluation de la satisfaction en fin de stage, feuille de présence émarginée par demi-journée par les stagiaires et le formateur, Attestation de fin de formation

## Action Collective ATLAS

AELION est sélectionné par l'OPCO ATLAS dans le cadre des Actions Collectives Nationales (ACN) DEVELOPPEMENT.

Formation 100 % pris en charge par votre OPCO ATLAS (sous réserve de fonds disponibles et de validation de votre projet par votre conseiller-ère ATLAS).

Le Framework Spring permet d'écrire des applications d'entreprise Java avec un style allégé. Il propose un jeu complet de composants à la carte pour gérer les données d'une entreprise et les exploiter avec des applications web MVC, des web services ou des API Rest. La qualité est de mise avec la prise en charge des tests, l'inversion de contrôle des composants et la sécurité. A l'issue de la formation, vous serez capable de mettre en œuvre le Framework Spring et les bonnes pratiques associées, dans les projets de développement d'applications.

## Objectifs

- Comprendre le fonctionnement de Spring et son positionnement dans les technologies Java
- Gérer la configuration des composants d'une application avec Spring
- Implémenter un service d'accès aux données
- Connaître les bonnes pratiques de développement avec Spring
- Connaître les apports de la Programmation Orientée Aspect (AOP)
- Sécuriser l'application

## Programme détaillé

### **ANALYSER LES DIFFERENCES ENTRE SPRING ET SPRING BOOT**

---

- Rappel sur le framework SPRING , enjeux et objectifs
- Spring Boot, extension de Spring
- Installation de Spring Tools Suite (Eclipse)
- Maven ou Gradle

### **CONFIGURER UN PROJET SPRING BOOT**

---

- Annotations SPRING
- Auto-Configuration, custom configurations
- Développement en couches : contrôleur, service, persistance
- Injection de dépendances

### **UTILISER LES ANNOTATIONS POUR FACILITER LA CONFIGURATION**

---

- Annotations SPRING
- Auto-Configuration, custom configurations
- Développement en couches : contrôleur, service, persistance
- Injection de dépendances

### **CREER UNE APPLICATION MVC AVEC SPRING BOOT**

---

- Principe MVC: Modèle vue controleur
- Ajouter Thymleaf aux dépendances
- Premier template

### **UTILISER UN MOTEUR DE TEMPLATE POUR INDUSTRIALISER LA PRODUCTION DES PAGES**

---

- Syntaxe THYMLEAF : conditions, itérations
- Passer le modèle à la vue
- Récupérer les données de la vue

## **UTILISER L'ABSTRACTION DE L'APPLICATION PAR RAPPORT AUX BASES DE DONNEES**

---

ORM PATTERN

application.properties : configuration

Entities / Repositories

## **GERER LES RELATIONS ENTRE LES ENTITES ET UTILISER LES DEPOTS DE DONNEES**

---

Relations : MANYTOONE, ONETOMANY, MANYTOMANY, ONETOONE

Récupération des entités : lazy ou eager

Custom repositories

## **TRAVAILLER AVEC DES BASES ORIENTEES DOCUMENTS**

---

Rappel sur le NO-SQL

Installation d'un serveur de données No-SQL,

Création des entités

Manipulation des collections

## **METTRE EN PLACE UNE ARCHITECTURE REST ET PRODUIRE UNE API**

---

Retour sur l'architecture RESTFULL

Configuration des endpoints d'API

DTO

HTTP Requests / HTTP Responses

Mode synchrone ou asynchrone

Tests unitaires

Mocks

## **SECURISER UNE API AVEC SPRING SECURITY**

---

Objectifs et principes

Authentification

Autorisations