

# Formation Spring Boot

## Objectifs de la formation Spring Boot

- Spring est le framework Java le plus répandu pour le développement d'applications d'entreprises offrant une réelle alternative à Java EE.
- Ses atouts :
  - ✓ Faciliter le développement et les tests.
  - ✓ Améliorer les performances.
  - ✓ Proposer de bon design pattern aux développeurs.
- Spring s'appuie principalement sur l'intégration de trois concepts clés :
  - ✓ L'inversion de contrôle ou injection de dépendance.
  - ✓ La programmation orientée aspect.
  - ✓ Une couche d'abstraction qui permet d'intégrer d'autres frameworks et bibliothèques avec une plus grande facilité.
- Les participants à cette formation Spring apprendront les bases du framework Spring et seront sensibilisés à la facilité de mettre en œuvre des tests unitaires et d'intégration avec ce framework.



- Le premier atelier permet aux participants de mettre en place l'IDE Spring Tools Suite et d'effectuer une première configuration Spring simple.
- Les ateliers suivants s'adossent à une application web typique utilisant Spring MVC, une couche service et Spring Data.
- Les participants doivent alors compléter les sources fournies, effectuer les configurations Spring adéquates, mettre au point les tests validant la spécification.
- Tous les tiers d'une application web sont abordés et plusieurs implémentations sont proposées.
- Le dernier chapitre permet aux participants de découvrir les aspects les plus modernes du framework et les APIs vers les réseaux sociaux, les mobiles et le cloud.

## À qui s'adresse cette formation ?

### Public :

Ce cours s'adresse aux développeurs Java.

### Prérequis :

De bonnes connaissances du langage Java sont demandées pour aborder cette formation Spring.

## Contenu du cours Spring Boot :

### Introduction au framework Spring

- ✓ Présentation du framework.



- ✓ Concepts de conteneur léger.
- ✓ L'offre de Pivotal Software.
- ✓ Positionnement vis à vis de Java EE.
- ✓ Pattern IoC et injection de dépendance.
- ✓ Spring et l'injection de dépendance.

#### **Exemple basique**

- ✓ Configuration XML, par annotations, Spring Boot.

### **Atelier 1 :**

Mise en place de Spring Tools Suite, 1ère configuration Spring, Tests via Junit.

### **Spring Cœur**

#### **Conteneur IoC et Beans :**

- ✓ Les Beans, BeanFactory et Application Context.
- ✓ Modes singleton ou prototype.
- ✓ Gestion des propriétés, "collaborateurs".
- ✓ Déclaration de beans via annotations, les différents stéréotypes.
- ✓ Méthodes d'injection de dépendance, l'annotation @Autowired, injection par constructeur.
- ✓ Explorer un contexte de beans via STS.

### **Spring AOP :**



- ✓ Concepts de l'AOP
- ✓ Terminologie utilisée : aspect, joinpoint, advice, shortcut, weaving.
- ✓ Les différents types d' « advice » : around, before, throws, after.
- ✓ Objectifs et fonctionnalités AOP de Spring.
- ✓ Intégration avec le framework AspectJ.

### **Autres technologies cœur :**

- ✓ L'interface Resource.
- ✓ Validation, data-binding et conversion de types.
- ✓ SpEL.

### **Atelier 2 :**

Création de différents contextes de beans, Configuration Java, Profiling via Spring AOP.

### **Applications Web avec Spring MVC**

- ✓ Rappel MVC, architecture Spring et contrôleurs Spring.
- ✓ Web Mvc Configurer Adapter.
- ✓ Résolution des vues, Thymeleaf, intégrations avec autres techniques de vues.
- ✓ Gestion des locales et des thèmes, traitements des exceptions.
- ✓ Contrôleurs REST.
- ✓ Mise en place de la sécurité.



- ✓ CORS Filter, authentification OAuth2.

## Spring Data et l'accès aux données (JDBC / Hibernate / JPA / NoSQL)

- ✓ L'objectif de Spring Data, les interfaces Repository et Crud Repository.
- ✓ Les types de requêtes.
- ✓ Spring Data JPA.
- ✓ Requête DSL et JPA.
- ✓ JPA Template, Jdbc Template, Hibernate Template.
- ✓ Gestionnaire de transaction.
- ✓ Spring Data NoSQL MongoDB.
- ✓ Requête MongoDB et Query DSL.
- ✓ Spring Data NoSQL MongoDB.
- ✓ Utilisation du template.
- ✓ Spring Data REST.
- ✓ Spring Web Flux.

### Atelier 3 :

Implémentation de la couche d'accès aux données à base de Repository, Démarcation des transactions des services métier.

### Spring et les tests

- ✓ IoC adapté aux tests, tests unitaires et d'intégration.



- ✓ Tests unitaires et Mock Objets.
- ✓ Classes de support pour le test unitaire.
- ✓ Apports de Spring pour les tests d'intégration.
- ✓ Annotations pour les tests.
- ✓ Tests d'intégration de Spring MVC.

#### **Atelier 4 :**

Mise en place de tests unitaires et d'intégration sur l'application précédemment développée

#### **Spring et l'intégration**

- ✓ Intégration avec des services distants (JMS, JAX-WS, Rest)
- ✓ Intégration avec serveur de mail
- ✓ Planification de jobs
- ✓ Utilisation de cache

#### **Atelier 5 :**

Mise en place d'un cache, intégration avec service REST distant, planification d'une tâche