

## Formation Spring : Développer avec le framework Spring



SII-298 3 Jours (21 Heures)



### Description

Spring est actuellement le framework Java le plus utilisé pour le développement d'applications d'entreprises. Il a toujours offert une alternative à la complexité de Java EE. Ses atouts : augmenter la productivité du développeur, faciliter les tests, améliorer les performances, faciliter l'intégration de technologies.

Spring est construit principalement autour de trois designs patterns clés : l'inversion de contrôle ou injection de dépendance, la programmation orientée aspect et des couches d'abstraction identifiant tous les besoins d'une application d'entreprise (Persistance, Messaging, Sécurité, Tracing, etc.).

Cette **formation Spring** démarre par une revue en profondeur des concepts de bases du framework Spring. Ce chapitre primordial permet de démystifier l'aspect magique de Spring et de comprendre les mécanismes sous-jacentes utilisées dans les chapitres suivants. Ensuite, chaque chapitre est consacré aux différentes couches logicielles des applications développées actuellement en entreprise, la persistance des données, la couche web, la sécurité et le test. Les mécanismes d'auto-configuration apportés par SpringBoot, l'accélérateur de projet Spring, sont détaillés dans chaque chapitre.

### À qui s'adresse cette formation ?

#### Pour qui

Ce cours s'adresse aux développeurs Java souhaitant utiliser le framework Spring pour faciliter le développement et les tests de leurs applications. Sont aussi concernés les architectes Java SE / Java EE ayant à penser l'infrastructure des applications Java d'entreprise.

#### Prérequis

Les participants doivent disposer d'une bonne connaissance de Java concrétisée dans un ou plusieurs projets professionnels. Pour profiter pleinement de tous les chapitres, ils doivent être sensibilisés aux problématiques web, d'accès aux données (persistance) et de tests d'application.

### Les objectifs de la formation

- Comprendre les apports du framework Spring (IoC, AOP) Savoir mettre en place une IDE productif et bonnes pratiques de développement Savoir gérer la configuration des propriétés et des composants d'une application Être capable d'implémenter toutes les couches d'une application Web d'entreprise (Web/Rest, Service, Accès aux données...) Savoir mettre en œuvre la sécurité au sein de son développement Spring Savoir mettre en œuvre des tests avec Spring

## Programme de la formation

### Introduction au framework Spring

- La genèse du projet, son positionnement vis à vis de Java EE
- L'offre de Pivotal Software, les cas d'usage du framework, les 10 commandements d'un projet Spring
- IoC et l'injection de dépendance : Le design pattern au cœur du framework
- Les types d'injections supportés par Spring
- Spring : en tant qu'Usine à Bean, de conteneur léger
- Comment configurer le framework : XML, annotations, l'auto-configuration avec Spring Boot

### Spring : Un conteneur léger

- Parcours des APIs BeanDefinition, BeanFactory et ApplicationContext
- Le cycle de vie des beans
- L'interface Environment : propriétés de configuration et profils
- Les autres fonctionnalités cœur : SpEL, Messages, Chargement de ressources

### Spring et les annotations

- Configuration via annotations.
- Component et ses différents stéréotypes
- Injecter des dépendances via les annotations : Autowired, Resource, Inject
- L'injection implicite par constructeur
- Annotations pour les profils, leur cas d'usage, les modes d'activation

### Spring AOP

- Les concepts de l'AOP, les problèmes résolus par l'AOP
- La terminologie utilisée : aspect, joinpoint, advice, shortcut, weaving
- Les différents types d' « advice » : around, before, throws, after
- Quel est le support de l'AOP dans Spring ?
- Configuration XML, bénéfices/inconvénients
- Intégration avec le framework AspectJ, l'AOP via les annotations

### Spring Boot

- Le principe de l'auto-configuration, application à Java
- Comment démarrer ? Spring Initializ et les starters disponibles
- Un unique fichier : Gestion des propriétés de configuration
- Assistance de L'IDE Spring Tools Suite

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

- L'objectif de Spring Data, les abstractions Repository et CrudRepository
- Les alternatives pour le requêtage des objets du domaine
- Application à JPA : Spring Data JPA
- Les classes de support fournis : JdbcTemplate, HibernateTemplate
- Comment gérer les transactions, l'annotation @Transactional
- Les apports de l'auto-configuration SpringBoot pour configurer les DataSource
- Application à NoSQL : Spring Data MongoDB
- Requête et classe de support
- Les apports de l'auto-configuration SpringBoot, Un MongoDB embarqué

## Spring MVC pour les applications Web

- Rappels du pattern architectural MVC, implémentation dans Spring
- L'auto-configuration SpringBoot et sa personnalisation avec WebMvcConfigurer
- Comment écrire un contrôleur Spring ?
- Résolution des vues, l'exemple de Thymeleaf,
- Gestion des locales et des thèmes, traitements des exceptions, Page d'erreur

## Spring MVC pour les APIs RESTful

- Les spécificités des contrôleurs REST
- Configuration typique : le CORS
- Ne pas négliger la sérialisation JSON : Personnalisation via les annotations de la librairie Jackson
- Documenter automatiquement son API avec SpringDoc et OpenAPI
- Interroger une API Restful avec la classes de support RestTemplate

## Spring Security

- Les apports de SpringSecurity et Concepts de base
- L'auto-configuration de SpringBoot, les filtres de protection
- Les différentes alternatives pour l'authentification, l'objet SecurityContext
- La sécurité Web via le bean SpringSecurityFilterChain, comprendre sa configuration, savoir debugger la sécurité
- GrantedAuthority vs Role
- Mise en place : Sécurité stateful ou sécurité Stateless ?
- Le protocole OAuth2 et le standard JWT
- Le support d'OAuth2 dans Spring5
- Protection de la couche métier, mécanisme et mise en place

## Spring et les tests

- Les apports de Spring-test , Le contexte SpringBoot
- L'annotation @SpringBootTest, quel est le mécanisme de détection de configuration, les cas d'usages qui en résultent
- Mocking des beans : souvent deux usages
- Tests d'intégration par couche avec les test auto-configurés,
- Tester une API sécurisée par SpringSecurity

## Déploiement des applications Spring

- Pour les Ops : le starter actuator
- Se préparer pour la production, l'objectif build-info du plugin Maven
- Les différents format de packaging, le support du plugin Maven ou Gradle
- La SISE en service d'application SpringBoot
- Création d'une image docker, document manifest kubernetes typique