

# Spring : Développer avec le framework Spring

DÉVELOPPER FACILEMENT DES APPLICATIONS D'ENTREPRISE JAVA EE AVEC  
SPRING FRAMEWORK



**Référence :** SFSG

**Durée :** 3 jours (21h.)

**Tarif :** 1890 € HT

**Contact :** 01 43 34 90 94

**Niveau :** Intermédiaire

**Classe à distance :** Possible

**Prochaines sessions :**

- 10 septembre au 12 septembre

- 15 octobre au 17 octobre

- 10 décembre au 12 décembre

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.

## Objectifs

### Objectif opérationnel :

Savoir maîtriser l'écosystème Spring/SpringBoot et acquérir les bonnes pratiques de développement d'applications d'entreprises avec ce framework.

### Objectifs pédagogiques :

À l'issue de cette **formation Spring**, vous aurez acquis les connaissances et compétences nécessaires pour :

- 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

## Public

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.

## Pré-requis

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.

## Contenu du cours

### 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

1/5

# Spring : Développer avec le framework Spring

DÉVELOPPER FACILEMENT DES APPLICATIONS D'ENTREPRISE JAVA EE AVEC  
SPRING FRAMEWORK



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 Cœur

### 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. @Configuration et @Bean

@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 pas 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

#### Ateliers 1, 2 et 3 (à titre indicatif)

##### Objectifs :

Démystifier Spring et visualiser le mécanisme d'IoC à l'aide de code Java très simple

Maîtriser la configuration via les annotations

Comprendre la technologie AOP et son utilisation par le framework

##### Description :

Développement de 2 beans, configuration XML

Tracing du cycle de vie des beans

Reprise de la configuration via des annotations

Mise en place de 2 profils (jdbc et csv)

Développement d'un aspect @Profilable

## Spring Boot

Le principe de l'auto-configuration, application à Java

Comment démarrer ? Spring Initializr et les starters disponibles

Un unique fichier : Gestion des propriétés de configuration

Assistance de L'IDE Spring Tools Suite

#### Atelier 4 (à titre indicatif)

##### Objectifs :

Comprendre le mécanisme d'auto-configuration et ses apports

Prendre en main toutes les facilités apportées par l'IDE STS

##### Description :

Création d'une application web via l'assistant

Assistance de l'éditeur pour la configuration de l'application

Configuration du trace

Exécution en mode DEBUG

## 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  
Requêtage et classe de support  
Les apports de l'auto-configuration SpringBoot, Un MongoDB embarqué

## **Ateliers 5 et 6 (à titre indicatif)**

### Objectifs :

Comprendre les apports de l'abstraction SpringData  
Comprendre les alternatives pour l'implémentation de la persistance avec Spring

### Description :

Récupération des classes modèles du domaine  
Déclaration d'interface JpaRepository et utilisation lors de test  
Implémentation d'une transaction métier dans un bean @Service  
Implémentation d'un MongoDBRepository et utilisation lors de test

## Applications Web avec Spring MVC

### **Spring MVC pour les applications Web**

Rappels du pattern architectural MVC, implémentation dans Spring  
Comment écrire un contrôleur Spring ?  
L'auto-configuration SpringBoot et sa personnalisation avec WebMvcConfigurer  
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

## **Ateliers 7 et 8 (à titre indicatif)**

### Objectifs :

Comprendre les rôles des controllers de SpringMVC  
Implémenter une API Rest et publier sa documentation via OpenAPI  
Développer un client REST

### Description :

Implémentation de la séquence web, affichage puis POST de formulaire  
Implémentation des fonctionnalités CRUD dans un RestController  
Gestion centralisée des exceptions  
Mise en pace SpringDoc et test de l'API via Swagger  
Développement d'un client REST

## 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

## **Ateliers 9 et 10 (à titre indicatif)**

### Objectifs :

Comprendre les deux approches de la sécurité : Stateful et Stateless

*Savoir configurer le filtre securityFilterChain*  
*Comprendre le support de Spring pour OAuth2*  
**Description :**  
*Activation de la sécurité et configuration par défaut*  
*Définition des ACLs d'une sécurité stateful*  
*Utilisation d'un gestionnaire d'authentification personnalisée*  
*OpenID/OAuth2 : « Se connecter avec Google »*  
*OAuth2Resource : Définition des ACLs par rapport à un Jeton OAuth*

## 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

### **Atelier 11 (à titre indicatif)**

**Objectifs :**  
*Comprendre les intentions des tests système et des tests d'intégration*  
*Savoir tester en isolation toutes les couches d'une application web*

**Description :**  
*Test couche persistance*  
*Test sérialisation Json*  
*Test couche contrôleur*  
*Test système complet*

## 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

### **Atelier 12 (à titre indicatif)**

**Objectifs :**  
*Savoir construire un package pour la production*  
*Comparer les alternatives de déploiement*

**Description :**  
*Mise en place actuator, configuration Maven*  
*Création d'exécutable jar et exécution*  
*Création image docker et exécution*

## Moyens pédagogiques et techniques

Les formations PLB sont conçues et animées par des experts en activité.  
Les **cours pratiques** alternent **travaux pratiques** , concrets et progressifs construits sous forme de projet fil rouge, **et apports théoriques** (en moyenne 60% de travaux pratiques, 40% de théorie).  
Les **séminaires** font alterner **études de cas et démonstrations** concrètes, issues de l'expérience terrain de nos formateurs, **et apports théoriques**.  
Le support de cours et le cahier d'exercices incluant les corrigés sont fournis en français, au format PDF, et sont téléchargeables.  
Chaque participant, en présentiel ou en classe virtuelle, accède aux travaux pratiques ou aux démonstrations via un poste de travail qui lui est dédié, configuré et équipé des logiciels et outils spécifiques présentés dans cette formation, dans leur dernière version.  
Les cours et examens de certification officiels suivent les conditions de l'éditeur, du constructeur ou du certificateur. Ils sont susceptibles d'évoluer à tout moment.  
La plupart des formations peuvent être suivies indifféremment en présentiel ou en classe à distance

synchrone depuis votre entreprise ou votre domicile (voir les [Modalités pédagogiques, techniques et de suivi](#) spécifiques aux formations à distance).

Les salles sont équipées pour accueillir ces deux modalités: vidéoprojecteur, écran/tableau interactif, webcam et sonorisation, accès Internet très haut débit et espace documentaire partagé.

L'accès aux ressources pédagogiques, techniques et de suivi s'effectue via un espace personnel 100% digital. Le service d'assistance technique est joignable par chat et téléphone avant et pendant la formation.

Nos bâtiments sont classés ERP 5. En cas de handicap, contactez-nous par téléphone ou via l'adresse [accessibilite@plb.fr](mailto:accessibilite@plb.fr) afin de mettre en place l'équipement et l'accompagnement adaptés.

## Modalités de suivi et d'évaluation

Les pré-requis sont évalués **par QCM** avant l'entrée en formation.

Les participants réalisent, **en début et en fin de formation, une auto-évaluation** de leurs connaissances et compétences au regard des objectifs pédagogiques de la formation.

L'évaluation des acquis en cours de formation est réalisée au fur et à mesure **des études de cas et/ou Travaux Pratiques** (50% du temps minimum pour les cours pratiques) **et/ou sous forme de QCM**.

L'évaluation en fin de formation est réalisée de la même façon et/ou via **le questionnaire d'auto-évaluation** qui permet de mesurer l'évolution par rapport au début de la formation et/ou par le passage de l'examen de certification, le cas échéant.

Les convocations, avec horaires définitifs, lieu et plan d'accès, sont envoyées deux semaines avant le début de la formation.

Les participants en classe virtuelle reçoivent leurs éléments de connexion par e-mail.

La feuille de présence numérique est à signer par demi-journée.

En fin de formation, les participants remplissent un questionnaire de satisfaction global.

Une attestation de réalisation est remise à la fin de la formation.

Si les conditions le permettent, nous pouvons inscrire jusqu'à 24h avant le début de la session. En cas d'inscription via le site [Moncompteformation](http://Moncompteformation), un délai de 11 jours ouvrés minimum est à respecter entre la proposition de commande et l'entrée en formation.

**INSCRIVEZ-VOUS  
À CETTE FORMATION**