

## Programme de formation Spring

Cette formation à Spring offre aux participants une expérience pratique dans la manipulation des fonctionnalités du framework Java. Ils apprendront les meilleures pratiques pour configurer Spring, accéder aux données, créer des applications Web et REST, ou encore gérer la sécurité applicative grâce à Spring Security. Au cours de nombreux ateliers réalisés durant la formation Spring à distance ou en présentiel, ils mettront en application leurs nouvelles connaissances et seront en mesure de développer des applications d'entreprise robustes et performantes, tirant pleinement parti de la puissance du framework.

### Prérequis

Des connaissances en programmation Java.

### Durée

5 jours

### Public

Concepteurs, architectes-logiciels, chefs-de-projet, développeurs

### Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

### Modalités d'évaluation

- En amont de la session de formation, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation (variable selon la formation suivie).
- En cours de formation, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience.
- En fin de session, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.

## Programme de formation

### Introduction à la formation sur le framework Spring

Présentation générale et objectifs pédagogiques  
Comprendre l'écosystème Spring et son évolution  
Concepts de conteneur léger, avantages et inconvénients du framework  
Le design pattern d'inversion de contrôle (IoC) et l'injection de dépendances (DI)  
La Programmation Orientée Aspect (AOP)  
Vue d'ensemble de l'architecture et des modules Spring : Core, test, Web (MVC)...  
Nouveautés apportées par Spring 5 : Java 8, Hibernate 5, programmation réactive avec WebFlux, Kotlin, JUnit 5...  
Vue d'ensemble des projets Spring : formation à Spring Boot, Spring Data, Spring Batch, Spring Cloud...  
Exemples de cas pratiques : tour de table, présentation des apprenants et du formateur ou de la formatrice, échanges sur les différents contextes professionnels et perspectives d'application des acquis pédagogiques de cette formation Spring. Installation de l'environnement de développement (Spring Tools 4), mise en place des bibliothèques et configuration d'un premier projet Spring.

### Spring Core : configuration des beans et injection de dépendances

Les beans et l'approche POJO (Plain Old Java Object)  
3 manières de définir les configurations : XML, annotations ou code Java  
Les interfaces BeanFactory et ApplicationContext  
La portée des beans (scope) : modes singleton et prototype  
Cycle de vie d'un bean, les hooks Init et Destroy  
L'injection de dépendances et les annotations @Autowired, @Qualifier et @Resource  
Ajouter l'annotation @Component dans les classes  
Annotations @Configuration, @Bean et @Import  
Utiliser les profils  
Ecrire des expressions avec le langage Spring

SpEL (Spring Expression Language)  
Exemples de cas pratiques : création de beans, configuration via XML et annotations, instanciations, injection des dépendances par le constructeur, par un setter, utilisation des différents types d'autowiring, activation des profils, externalisation des propriétés.

### Spring AOP, les apports de la Programmation Orientée Aspect

Limites de la POO (Programmation Orientée Objet), quels problèmes résout l'AOP ?  
Concepts principaux et terminologie : aspect, advice, join point et pointcut  
Les différents types d'advices  
Spring AOP vs AspectJ  
Implémenter un advice avec @Around, @Before et @After  
Meilleures pratiques AOP : expressions pointcut, proxies, mise en cache...  
Exemples de cas pratiques : mise en oeuvre des fonctionnalités de programmation Spring orientée aspect via le module Spring AOP, définition d'aspects, implémentation des advices, intégration avec le framework AspectJ.

### Gestion des données : JDBC, Hibernate, JPA et transactions

Les patterns DAO (Data Access Object, ou objet d'accès aux données) et Template  
Comprendre l'intérêt du JdbcTemplate de Spring  
Choisir une approche pour l'accès aux données  
Configurer une DataSource, accéder aux ressources JNDI  
Le mapping objet-relationnel avec Hibernate  
La gestion des transactions avec Spring : définition, annotations utiles  
Propagation des transactions  
Travailler avec JPA, définir une interface Repository  
Introduction à Spring Data : architecture, configuration, utilisation de CrudRepository  
Exemples de cas pratiques : configuration d'une source de données, exécution de requêtes SQL avec Spring JDBC et implémentation d'une couche d'accès aux

données, utilisation de Spring avec Hibernate, avec JPA, configuration de Spring Data.

## **Spring MVC et la couche Web**

Retour sur le pattern Model-View-Controller

Comprendre le flux de données et les composants Spring MVC

La validation des beans côté Web et les annotations utilisées

Créer des contrôleurs : @Controller et

configuration du DispatcherServlet

Résolution de vues et mappings

L'annotation @Async

Gestion des exceptions

Utiliser le moteur de template Thymeleaf

Coder en Spring via programmation réactive avec WebFlux

Introduction à Spring Boot

Exemples de cas pratiques : création d'un projet web, configuration d'une application avec le framework Spring MVC, création d'une classe avec @Controller, définition des méthodes de mappage, création de pages JSP, prise en main de Thymeleaf et création d'une application REST avec les composants RestController et WebClient.

## **Spring Boot pour simplifier le déploiement d'applications**

L'apport du module Spring Boot

Utiliser l'annotation @EnableAutoConfiguration

Importer un ensemble de dépendances avec les starters

Propriétés des configurations

Utiliser CommandLineRunner

Exemple de cas pratiques : création d'une application Boot avec Spring Initializr.

## **Sécurité applicative avec Spring Security**

Pourquoi utiliser Spring Security ?

Configurer l'authentification

Comprendre le principe de chaîne de filtres

Exemples de cas pratiques : configuration des dépendances de Spring Security, configuration

d'un filtre de sécurité pour les requêtes

entrantes, ajout de contrôleurs et mise en

oeuvre de la sécurité sur l'application

développée précédemment.