

# Spring et Spring Boot – Développer des applications d'entreprise

5 jours (35 heures)

Initiation

Présentiel ou classe à distance

à partir de

**3 175 € HT** par personne (TVA en vigueur) Repas inclus en présentiel

Contenu mis à jour le 30/09/2026. Document téléchargé le 07/10/2026.

Réservez la formation en Spring 5 Framework éligible au CPF et décrochez la certification M2i Langages de Programmation.

## Objectifs

A l'issue de cette formation, vous serez capable de :

- Décrire le positionnement de Spring par rapport à Jakarta JEE
- Créer une architecture en couches
- Mettre en oeuvre la programmation par aspect
- Paramétrer le conteneur léger de Spring et utiliser les annotations
- Accéder à vos données avec Spring JDBC et Spring ORM
- Créer des pages Web avec Spring MVC
- Sécuriser vos applications avec Spring Security
- Décrire les bases de la mise en oeuvre de Spring Boot
- Citer les différences de Spring 6 et 7 ainsi qu'entre Spring Boot 3 et 4
- Migrer une application de Spring 6 à 7.

Les compétences visées par cette formation font l'objet d'une évaluation.

## À qui s'adresse cette formation ?

Développeurs, architectes et chefs de projets techniques.

## Prérequis

Avoir suivi le cours JAV-SE "Java – Les fondamentaux de la programmation" ou avoir une bonne connaissance du langage Java et une vue d'ensemble de la plateforme Jakarta JEE.

# Programme

## 1 Pendant votre formation

## Spring et Spring Boot – Développer des applications d'entreprise

Jour 1 – Matin

### Vue d'ensemble

- Origine et vision d'architecture de Spring
- L'inversion de contrôle ou injection de dépendances
- Bilan des solutions apportées par Spring
- Evolution de Spring
- Compatibilité des versions de JDK et de JEE avec les versions de Spring
- L'écosystème Spring (Framework, Boot, Data, Security, Cloud)
- La séparation des responsabilités

### Exemple de travaux pratiques (à titre indicatif)

- Créer un projet vierge Spring Boot 4 sur JDK 25 et démarrer l'application

Jour 1 – Après-midi

### Spring Core

- Construction des instances
- Injection des instances
- Annotation ou configuration
- SpEL (Spring Expression Language)
- Spring profiles
- Nomme des beans et bean factory
- Bean scopes
- Autowiring
- "Application Context"
- Le cycle de vie d'un bean
- Les configurations par profil

- Activer un profil au démarrage

*Exemple de travaux pratiques (à titre indicatif)*

- Déclarer un service applicatif et l'injecter dans plusieurs consommateurs avec des scopes différents

Jour 2 – Matin

### Configuration d'une application Spring Boot

- L'univers Spring et la complexité de sa configuration
- Les composants de type Starters et auto-configuration
- Utilisation de Spring Initializr pour créer une application Web

*Exemple de travaux pratiques (à titre indicatif)*

- Initiation à Spring Boot avec un TP de lancement

### Auto-configuration

- Le mécanisme d'auto-configuration
- Inspecter les conditions d'activation
- Surcharger une auto-configuration

### Configuration externalisée

- Charger la configuration depuis application.yml ou application.properties
- Les @ConfigurationProperties typées
- Les profils dans la configuration

*Exemples de travaux pratiques (à titre indicatif)*

- Tracer les auto-configurations actives au démarrage et désactiver l'une d'elles
- Externaliser les paramètres d'une mini-application dans application.yml

Jour 2 – Après-midi

### Profils Spring

- Séparation des configurations par profil dev, test et prod
- Activer un profil au démarrage
- Composer plusieurs profils
- Les variables d'environnement et les secrets

*Exemple de travaux pratiques (à titre indicatif)*

- Configurer les profils dev, test et prod avec des sources de données distinctes et faire varier le comportement de l'application au démarrage

### Sources de configuration externes

- Gérer les secrets en production
- Les conventions de variables d'environnement

*Exemple de travaux pratiques (à titre indicatif)*

- Externaliser une clé d'API via une variable d'environnement et la consommer dans l'application

Jour 3 – Matin

### Patterns DAO et DTO

- Séparation de la couche d'accès aux données du métier
- Usage des DTO pour les frontières applicatives

*Exemple de travaux pratiques (à titre indicatif)*

- Modéliser les DTO d'une API d'articles indépendamment des entités JPA

### Le Mapping JPA

- Annoter une entité simple avec ses contraintes
- Les relations 1:1, 1:N et N:N
- Les chargements lazy et eager
- Modéliser un héritage avec JPA

*Exemple de travaux pratiques (à titre indicatif)*

- Modéliser les entités Article, Author, Tag avec leurs relations puis tester la persistance

### Le Repository Spring Data

- Déclaration d'un Repository pour une entité
- Les Query Methods dérivées du nom
- Spécification pour la recherche dynamique

*Exemple de travaux pratiques (à titre indicatif)*

- Implémenter les opérations CRUD d'Article via un Repository et observer les requêtes générées

Jour 3 – Après-midi

### Requêtes personnalisées

- Requêtes JPQL
- Les requêtes natives
- Les Named Queries au niveau de l'entité

*Exemple de travaux pratiques (à titre indicatif)*

- Implémenter une recherche complexe d'articles publiés par auteur avec pagination

### Pagination et tri

- Exploiter Pageable et Sort
- Exposer une réponse paginée

*Exemple de travaux pratiques (à titre indicatif)*

- Exposer un endpoint paginé sur la collection d'articles

### Transactions

- Les niveaux d'isolation et la propagation
- Annoter une méthode avec @Transactional
- Les rollbacks sur exception

*Exemple de travaux pratiques (à titre indicatif)*

- Implémenter un transfert métier qui doit être atomique et tester son rollback

Jour 4 – Matin

## Spring et le Web

- Spring MVC pour le Web
- Bean validation côté Web
- Conversion et affichage

## API REST

- Support des services RESTful
- Support des WebSockets
- WebFlux et programmation réactive

*Exemple de travaux pratiques (à titre indicatif)*

- Mise en oeuvre d'une application minimale Spring MVC

Jour 4 – Après-midi

## Spring Security

- Sécuriser une application Web
- La chaîne des filtres
- Protéger des routes
- Authentification :
  - Définir des utilisateurs
  - Exposer login et logeur
- Autorisation :
  - Définir des ACL

*Exemple de travaux pratiques (à titre indicatif)*

- Sécurisation des deux applications jusque-là développées

Jour 5 – Matin

## Les tests fondamentaux

- Les tests d'intégration
- Mock avec @MockitoBean
- Isoler le contexte applicatif d'un test à l'autre
- Isoler la couche persistance

- Valider les requêtes Repository et JPQL

*Exemples de travaux pratiques (à titre indicatif)*

- Tester le contexte d'une application en succès et en cas de bean manquant
- Ecrire des tests sur un Repository

### Tests Web

- Isoler la couche Web avec @WebMvcTest
- Simuler une requête HTTP sans serveur avec MockMvc
- Vérifier statuts, en-têtes et contenu de la réponse
- Mock des services métier injectés dans le contrôleur

*Exemple de travaux pratiques (à titre indicatif)*

- Tester l'API REST Article avec @WebMvcTest et MockMvc, en succès et en cas d'erreur de validation

### Tests Security

- Tester un endpoint protégé
- Valider les autorisations par méthode
- Contrôler le comportement de Spring Security en test

*Exemple de travaux pratiques (à titre indicatif)*

- Couvrir les règles d'autorisation de l'API par des tests Spring Security

Jour 5 – Après-midi

### Spring Boot Actuator

- Les endpoints de santé et d'information
- Protéger les endpoints sensibles

*Exemple de travaux pratiques (à titre indicatif)*

- Exposer l'état d'une application et observer ses indicateurs

### Migration Spring 6 / Boot 3 vers Spring 7 / Boot 4

- Les breaking changes de Spring Framework 7
- Les outils de migration
- Les dépendances Jakarta EE 11

### Spring et IA

- Les assistants disponibles
- Formuler un prompt qui précise la version Spring et les conventions
- Repérer les pièges spécifiques – versions Spring mélangées dans les exemples (3.x, 5.x, 6.x), API dépréciées, javax vs jakarta
- Vérifier systématiquement le code généré par exécution Spring Boot Test et inspection Actuator

## Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

Nous contacter pour en savoir plus →

## Modalités pédagogiques

Formation délivrée en présentiel ou distanciel\* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode\*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

\* nous consulter pour la faisabilité en distanciel

\*\* ratio variable selon le cours suivi

Accessibilité aux personnes en situation de handicap

## Détails de la session

Du 16 novembre 2026 au 20 novembre 2026 (5 jours / 35 heures)

Mulhouse

Du 16 novembre 2026 au 20 novembre 2026

**Mulhouse**

- 16 novembre 2026 ..... 9h00 – 15h00
- 17 novembre 2026 ..... 9h00 – 15h00
- 18 novembre 2026 ..... 9h00 – 15h00
- 19 novembre 2026 ..... 9h00 – 15h00
- 20 novembre 2026 ..... 9h00 – 15h00