

Offre Maîtrise

Progresse en Java / Spring Boot en un mois, en petit groupe, sur un projet réel. Une formation 100 % technique avec Slah : 18h de cours, de la pratique et des exercices à chaque séance.



★★★★★ 4.9/5

Résultats en quelques jours



18h

FORMATION JAVA

1 mois

EN PETIT GROUPE

3

PARTICIPANTS
MINIMUM

1 050 €

PAR PARTICIPANT

Ce que comprend l'offre Maîtrise

- Prochaine session : du 9 novembre au 11 décembre 2026
- Formation Java / Spring Boot avec Slah : 18h de cours sur 1 mois
- Petit groupe : 3 participants minimum
- Pratique et exercices à chaque séance, sur un projet fil rouge
- Accompagnement entre les séances
- Attestation de suivi CodeBridge

Prix de lancement

1 050 €

Par participant · Paiement unique ou 3x sans frais

Étudiant ? On a un prix encore plus attractif pour toi.

Ton formateur



Slah Hassine

**FORMATEUR JAVA · TECHLEAD
SOFTWARE ENGINEER**

TechLead Java, Spring et Kafka chez BNP Paribas Leasing Solutions. 13 ans d'expérience en développement Java sur des architectures microservices et des migrations cloud, notamment chez Tinubu et Groupe Pomona.

Java / Spring Boot

Architecture hexagonale

TDD / BDD / DDD

Formation technique : Java, Spring Boot, Hibernate / JPA

Une formation pratique pour apprendre à concevoir, tester, documenter et déployer une API REST professionnelle en Java avec Spring Boot et Hibernate / JPA. Toute la formation est construite autour d'**un seul projet réel**, développé de bout en bout séance après séance, avec les méthodes et les outils utilisés en entreprise.

Durée	Rythme	Répartition	Niveau
18 heures	Du 9 novembre au 11 décembre 2026, en petit groupe	≈ 30% théorie, ≈ 70% pratique	Intermédiaire

Public visé

- Jeunes diplômés en informatique (école d'ingénieur, master, BUT) souhaitant devenir développeur backend Java
- Développeurs Java junior voulant maîtriser Spring Boot, JPA et les bonnes pratiques professionnelles
- Développeurs en reconversion disposant de bases solides en programmation orientée objet

Prérequis

- Programmation orientée objet en Java : classes, héritage, interfaces, polymorphisme, collections, exceptions
- Bases de SQL et de Git
- Un ordinateur sous Windows, macOS ou Linux, 8 Go de mémoire minimum (16 Go conseillés), avec la virtualisation activée. Un guide d'installation détaillé est fourni

i Un test de positionnement de 30 minutes est proposé avant le démarrage. En cas de bases insuffisantes, une formation préalable de remise à niveau en programmation orientée objet Java est recommandée.

Objectifs pédagogiques

À l'issue de la formation, le stagiaire est capable de :

- Modéliser un domaine métier et le protéger par des tests automatisés (DDD, TDD)
- Structurer une application Spring Boot en architecture hexagonale (ports et adaptateurs)
- Persister des données dans PostgreSQL avec JPA / Hibernate et Spring Data JPA, en maîtrisant le schéma de la base
- Exposer une API REST complète : endpoints validés, erreurs normalisées, documentation OpenAPI / Swagger
- Tester à tous les niveaux : tests unitaires, Mockito / BDD, tests de la couche web et de la persistance, tests d'intégration, Postman
- Optimiser et livrer une application : performances (N+1), packaging et déploiement dans Apache Tomcat 11
- Travailler avec les outils du métier : IntelliJ IDEA, Maven, Git / GitHub, Docker, Postman

Le principe pédagogique

Les connaissances théoriques d'un jeune diplômé sont souvent solides ; ce qui fait la différence en entreprise, c'est l'expérience d'un projet complet mené avec les pratiques professionnelles. La formation consacre donc environ 70% du temps à la pratique : le stagiaire code lui-même le projet, guidé pas à pas.

Principe	Mise en œuvre
Un projet fil rouge réel	Une API de gestion de médiathèque, construite du premier au dernier jour de la formation.
La théorie juste à temps	Chaque notion est expliquée au moment où elle sert, puis appliquée immédiatement.

Le code d'abord	L'essentiel de chaque séance est consacré au développement par le stagiaire.
Les standards du métier	Tests automatisés, architecture hexagonale, Git, documentation, déploiement.
Un suivi régulier	Point de contrôle à chaque séance, travail personnel corrigé, relecture du code sur GitHub.

Déroulement type d'une séance

Part du temps	Étape	Contenu
≈ 10 %	Reprise	Correction du travail personnel, questions.
≈ 20 %	Théorie	Notions de la séance, illustrées par des exemples.
≈ 10 %	Démonstration	Mise en œuvre en direct par le formateur.
≈ 50 %	Pratique	Le stagiaire fait avancer le projet.
≈ 10 %	Bilan	Point de contrôle, enregistrement du travail (Git).

Programme détaillé

N°	Étape	Contenu	Théorie / pratique
1	Lancement : HTTP, REST et Java moderne	Architecture front/back, protocole HTTP, principes REST, histoire de Java de la version 8 à la 25 ; conception des endpoints du projet ; exercices de Java moderne (records, lambdas, Streams, Optional, java.time).	40% / 60%
2	Conception du domaine : DDD et TDD	Langage ubiquitaire, entités, agrégats, modèle riche ; tests unitaires JUnit et AssertJ ; écriture des règles métier en TDD.	30% / 70%
3	Spring Boot et architecture hexagonale	Inversion de contrôle, injection de dépendances, auto-configuration ; ports et adaptateurs, règle de dépendance ; création et structuration du projet.	35% / 65%
4	Persistance (1) : JPA et PostgreSQL	ORM, JPA, Hibernate et Spring Data ; PostgreSQL sous Docker ; mapping des premières entités.	30% / 70%
5	Persistance (2) : relations et contexte de persistance	Associations, chargement LAZY/EAGER, dirty checking ; traduction entre entités JPA et objets du domaine.	35% / 65%
6	Spring Data JPA et gestion du schéma	Repositories, requêtes dérivées, JPQL, pagination ; schéma maîtrisé par scripts SQL ; tests de la couche de persistance.	30% / 70%
7	Cas d'usage et transactions	Ports d'entrée, services applicatifs, transactions et leurs pièges ; implémentation de tous les cas d'usage métier.	30% / 70%

8	API REST : contrôleurs, DTO et validation	Spring MVC, codes HTTP, DTO, Bean Validation ; exposition des 22 endpoints de l'API.	25% / 75%
9	Gestion des erreurs et documentation OpenAPI	Erreurs normalisées (RFC 9457), gestion centralisée des exceptions ; documentation Swagger / OpenAPI.	30% / 70%
10	Tester l'API : Postman et tests web	Collections Postman automatisées ; tests de la couche web avec MockMvc.	25% / 75%
11	Tests avancés : Mockito, BDD, intégration	Doublures de test, scénarios BDD, tests d'intégration de bout en bout, test automatique de l'architecture.	30% / 70%
12	Performance, packaging et déploiement	Problème N+1, packaging WAR, déploiement dans Apache Tomcat 11 ; démonstration finale du projet.	25% / 75%



Sur l'ensemble de la formation : environ 5 heures de théorie (cours et démonstrations) et 13 heures de pratique (développement du projet).

Le projet fil rouge

Une médiathèque de quartier souhaite remplacer son cahier d'emprunts par une application web. Le stagiaire conçoit et livre l'**API REST backend** que cette application utilisera.

Fonctionnalité	Contenu
Catalogue	Auteurs, catégories, livres et exemplaires ; recherche paginée
Adhérents	Inscription, modification, désactivation
Emprunts	Emprunt, retour, calcul des pénalités de retard, liste des retards, historique

22

ENDPOINTS REST

7

RÈGLES MÉTIER

5

ENTITÉS DU DOMAINE

6

NIVEAUX DE TESTS

Technologies et outils

Domaine	Technologies
Langage	Java 25 (LTS)
Framework	Spring Boot 4.1 (Spring Framework 7)
Persistance	JPA, Hibernate 7, Spring Data JPA, PostgreSQL 18 (Docker)
API	REST, JSON, Bean Validation, OpenAPI / Swagger UI
Tests	JUnit, AssertJ, Mockito, MockMvc, Postman

Architecture	Architecture hexagonale (ports et adaptateurs), DDD
Outillage	IntelliJ IDEA, Maven, Git / GitHub, Docker
Déploiement	Packaging WAR, Apache Tomcat 11

Modalités de la formation Java

Rubrique	Détail
Méthodes pédagogiques	Apports théoriques ciblés et démonstrations (≈ 30%), développement guidé d'un projet réel (≈ 70%), travail personnel entre les séances (30 à 45 min).
Supports remis	Support de cours (une fiche par séance : l'essentiel, le cours, le code de référence, le vocabulaire), guide d'installation de l'environnement, aide-mémoire Java moderne, code source du projet.
Évaluation	Test de positionnement à l'entrée ; point de contrôle à chaque séance ; relecture du code ; démonstration finale du projet (API, tests, documentation, déploiement).
Validation	Attestation de fin de formation ; projet complet publié sur GitHub, valorisable dans un portfolio.
Format	Petit groupe, 3 participants minimum. Pour la même formation en individuel, voir l'offre Maîtrise Plus.

Ton formateur



Slah Hassine

Formateur Java · TechLead Software Engineer

TechLead Java, Spring et Kafka chez BNP Paribas Leasing Solutions. 13 ans d'expérience en développement Java sur des architectures microservices et des migrations cloud, notamment chez Tinubu et Groupe Pomona. Des monolithes aux architectures microservices et hexagonales, il fait du TDD, du BDD et du DDD sa pratique quotidienne.

Au-delà du code, il cadre les choix d'architecture, rédige les spécifications techniques, valide les merge requests et coordonne les mises en production au sein d'équipes Scrum. Il intègre des systèmes d'information complexes et maîtrise toute la chaîne de livraison : GitLab CI, Docker, Kubernetes, supervision.

13 ans

dev Java

18h

Formation Java

[LinkedIn](#)

Prêt à passer à la Maîtrise ?

Réserve un appel découverte gratuit avec Zahir pour voir si l'offre Maîtrise est faite pour toi, ou si Maîtrise Plus (la même formation en 1:1) te correspond mieux.

[Réserver mon appel découverte](#)

Zahir Haniche · CodeBridge · haniche.zahir@gmail.com · 06 62 02 85 52