

Mis à jour le 07/06/2025

S'inscrire

Formation Spring Boot : Développer des microservices

3 jours (21 heures)

PRÉSENTATION

Notre formation « Spring Boot, Développer des microservices » vous apprendra à maîtriser l'un des frameworks les plus performants pour la création de microservices. Spring Boot s'est imposé comme la référence Java pour développer, tester et livrer rapidement des applications cloud-native.

Découvrez notre formation avancée pour maîtriser les architectures microservices, pensée pour vous permettre de développer, sécuriser et déployer efficacement des microservices robustes et évolutifs.

Comprenez les fondements essentiels : responsabilité unique, autonomie et isolation des services. Identifiez clairement les avantages concrets en matière de scalabilité, de résilience et d'agilité opérationnelle.

Explorez les principaux frameworks du marché : Spring Boot, Quarkus, Micronaut ou encore Helidon. Vous serez capables de comparer ces technologies afin de sélectionner celle qui répondra le mieux aux besoins spécifiques de vos projets. Notre programme de formation vous enseignera pourquoi Spring Boot est devenu le ??? framework du marché pour les microservices en le comparant avec ses concurrents directs. Vous apprendrez ensuite les compétences nécessaires pour concevoir, sécuriser et livrer un micro-service Spring Boot en autonomie complète.

Maîtrisez l'art des communications inter-services sécurisées et résilientes grâce à des pratiques telles que CQRS, Event Sourcing, ainsi que les patterns de résilience avancés comme le Circuit Breaker ou Retry, pour une robustesse maximale.

Enfin, approfondissez les stratégies modernes de sécurité et de monitoring : authentification JWT, OAuth2, orchestration avec Docker et Kubernetes, ainsi qu'une intégration complète dans un pipeline DevOps automatisé. Devenez expert dans la conception et l'exploitation optimale d'architectures microservices performantes.

Comme toutes nos formations, elle se déroulera sur la dernière version de l'outil : [Spring Boot 3.5](#).

OBJECTIFS

- Comprendre l'architecture des microservices et identifier ses avantages
- Analyser différents frameworks et identifier le plus approprié au déploiement d'un microservice
- Distinguer les différents conteneurs (Docker, Azure) et leurs avantages pour les microservices
- Sécuriser la communication entre microservices
- Développer un microservice

PUBLIC VISÉ

- Développeurs, architectes

Pré-requis

- Connaissances du langage Java

Programme de notre Formation Spring Boot : Développer des microservices

Fondamentaux et principes clés d'une architecture Microservices

- Définition claire des microservices (responsabilité unique, autonomie, isolation)
- Bounded Context et Domain Driven Design (DDD)
- Avantages clés : scalabilité, résilience, évolutivité
- Patterns de communication (synchrone REST/gRPC, asynchrone Event-Driven)
- Exposition des API avec API Gateway (équivalent Zuul, Kong, Traefik)
- Gestion centralisée de configuration (ex. Config Server, Hashicorp Vault)
- Atelier pratique : Découpage d'une application monolithique en microservices par domaines métiers.

Architecture micro-services et avantages

- Monolithe vs micro-services : découplage, scalabilité, résilience
- Bounded Contexts (DDD) & principe « Single Responsibility Service »
- Transactions distribuées : sagas, CQRS, cohérence éventuelle

- Comparaison entre différents Frameworks microservices et pourquoi Spring Boot
- Atelier participatif : benchmark entre différents frameworks

État de l'art sur les principaux frameworks Microservice

- Spring Boot / Spring Cloud (Java, complet, mature)
- Quarkus (Java natif, léger, rapide)
- Micronaut (Java/Kotlin, très performant)
- Helidon (Java natif, léger, modulaire)
- Impact sur le cold-start serverless et les coûts
- Écosystème Spring (Data, Security, Actuator) comme facteur de choix
- Atelier participatif : comparer deux POC puis justifier Spring Boot

Gestion avancée des interactions et résilience des Microservices (CQRS & Event Sourcing)

- CQRS (Command Query Responsibility Segregation) : séparation commande / requête
- Event Sourcing : gérer les états et l'historique via des événements métiers
- Patterns de résilience avancés (Circuit Breaker, Retry, Bulkhead, Timeout)
- Communication inter-services (client-side load balancing type Ribbon, abstraction API type Feign)
- Service Discovery : concepts et outils (Eureka, Consul, DNS-based)
- Traçage distribué (Zipkin, Jaeger) pour l'observabilité et le débogage
- Atelier participatif : Design d'un microservice utilisant CQRS et Event Sourcing (ex. gestion des commandes dans un e-commerce)

Opérations, sécurité et monitoring avancé en architecture microservices

- Patterns de sécurité avancés (OAuth2, JWT, mutual TLS entre microservices)
- Stratégies d'autorisation et gestion des accès inter-services
- Leader Election et gestion de coordination (équivalent Spring Cloud Cluster, Zookeeper)
- Déploiement automatisé et orchestrateurs (Docker, Kubernetes, Azure Container Apps)
- Stratégies avancées de monitoring et alerting (Prometheus, Grafana, ELK Stack)
- Bonnes pratiques DevOps : Continuous Integration & Delivery (CI/CD)
- Atelier participatif : Quiz interactif et scénario pratique sur les bonnes pratiques de sécurité et monitoring dans une architecture microservices.

Spring Boot démarrage express

- Spring Initializr, starters, auto-configuration
- Actuator : health-checks, métriques, endpoints personnalisés
- Démo « Hello-World » REST & tests unitaires
- Atelier pratique : implémenter un micro-service "Catalogue" basique

Contract-first & gestion d'API

- Spécification OpenAPI 3, génération contrôleurs / DTO

- Validation JSON-Schema côté serveur et client
- Spring Cloud Gateway : routage, Circuit Breaker léger, rate-limiting
- Atelier pratique : écrire un contrat OpenAPI puis scaffolder le code

Dockeriser un micro-service Spring Boot

- Dockerfile multi-stage, Buildpacks, images slim
- Variables d'environnement, secrets et logs STDOUT
- Probes liveness, readiness, startup
- Atelier pratique : builder, tagger et pousser l'image sur Docker Hub

Orchestration sur Azure AKS

- Concepts AKS : node pool, Managed Identity, auto-scaling
- Helm Charts, déploiement Blue/Green, rollback
- Pipeline GitHub Actions : build ? push ? deploy
- Atelier pratique : déployer le service "Catalogue" sur AKS

Sécuriser les API Spring Boot

- Spring Security 6 : OAuth 2.1 / OpenID Connect, JWT stateless
- Politique CORS, en-têtes sécurisées, Key Vault pour secrets
- Atelier pratique : protéger l'API "Catalogue" avec JWT & RBAC

Communication inter-services protégée

- Service Mesh Istio : mTLS strict, authentification mutuelle
- Circuit Breaker & retry avec Resilience4j
- Tracing Zipkin/Jaeger, correlation-id
- Atelier pratique : injecter sidecars Istio et activer mTLS

Build, Test, Deploy

- Tests unitaires & contract-tests dans CI
- Build image, push, déploiement Helm, autoscaling HPA
- Monitoring Grafana / Prometheus, alertes Slack
- Atelier pratique : livraison finale du micro-service et rétrospective.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.