

Formation Spring Cloud : architecture microservices avec Spring Boot

Introduction à la formation Spring Cloud

Présentation des objectifs de la formation.

Comprendre la place de Spring Cloud dans l'écosystème Spring.

Identifier les limites d'une application monolithique.

Comprendre les cas où une architecture microservices est pertinente.

Introduction aux microservices

Qu'est-ce qu'un microservice ?

Monolithe, monolithe modulaire et microservices.

Avantages et limites d'une architecture microservices.

Scalabilité, autonomie des équipes, déploiement indépendant.

Complexité réseau, observabilité, sécurité et gestion des erreurs.

Découpage d'une application en microservices

Identifier les domaines fonctionnels.

Comprendre les bounded contexts.

Définir les responsabilités de chaque service.

Éviter la base de données partagée.

Définir les contrats d'API entre services.

Atelier pratique :

Analyse d'une application existante.

Identification des services potentiels.

Définition des responsabilités et des échanges entre services.

Création de microservices avec Spring Boot

Structure d'un projet microservices.

Création de plusieurs applications Spring Boot.

Exposition d'API REST.

Gestion des profils d'environnement.

Configuration par service.

Atelier pratique :

Création de deux microservices Spring Boot.

Mise en place des premiers endpoints REST.

Configuration des services.

Communication entre microservices

Communication synchrone avec REST.

Utilisation de RestClient ou WebClient.

Gestion des erreurs réseau.

Timeouts et réponses d'erreur.

Propagation d'un identifiant de corrélation.

Atelier pratique :

Faire communiquer deux microservices.

Gérer un cas d'erreur entre services.

Ajouter une logique simple de diagnostic.

Service discovery avec Spring Cloud

Pourquoi utiliser un service registry.

Présentation de Eureka.

Enregistrement automatique des services.

Découverte de services.

Appels par nom logique.

Atelier pratique :

Créer un serveur Eureka.

Enregistrer les microservices.

Valider la découverte des services.

API Gateway avec Spring Cloud Gateway

Rôle d'une API Gateway.

Point d'entrée unique.

Routage vers les microservices.

Filtres de requêtes et de réponses.

Gestion centralisée de certains comportements transversaux.

Atelier pratique :

Créer une API Gateway.

Configurer les routes.

Exposer les services à travers la Gateway.

Configuration centralisée avec Spring Cloud Config

Problèmes liés à la configuration distribuée.

Principe du Config Server.

Configuration par environnement.

Gestion des propriétés communes et spécifiques.

Atelier pratique :

Créer un Config Server.

Centraliser la configuration de plusieurs services.

Modifier une configuration sans changer le code applicatif.

Résilience des microservices

Comprendre les pannes dans un système distribué.

Timeouts, retries et circuit breakers.

Introduction à Resilience4J.

Dégradation contrôlée du service.

Atelier pratique :

Simuler une panne de service.

Ajouter une stratégie de résilience.

Tester le comportement de l'application.

Communication asynchrone et événements

Différence entre communication synchrone et asynchrone.

Introduction à Kafka ou à une approche événementielle.

Producteurs, consommateurs et événements métier.

Cas d'usage dans une architecture microservices.

Atelier pratique :

Présentation ou mise en place simple d'un échange événementiel.

Publication et consommation d'un événement.

Observabilité des microservices

Pourquoi l'observabilité est essentielle.

Logs applicatifs.

Correlation ID.

Health checks.

Métriques.

Introduction à Micrometer, Prometheus, Grafana ou outils équivalents.

Atelier pratique :

Activer les endpoints de santé.

Lire les logs de plusieurs services.

Suivre une requête entre plusieurs composants.

Tests dans une architecture microservices

Tests unitaires.

Tests d'intégration.

Tests d'API.

Validation des contrats entre services.

Stratégies de test adaptées aux architectures distribuées.

Atelier pratique :

Tester un endpoint REST.

Tester une communication entre services.

Valider un scénario complet.

Déploiement local avec Docker Compose

Conteneurisation des services.

Organisation d'un environnement local.

Services applicatifs, base de données, Gateway et registry.

Atelier pratique :

Exécuter l'architecture microservices avec Docker Compose.

Valider le démarrage complet de l'environnement.

Tester un scénario de bout en bout.