

Formation Design patterns pour les microservices

Code : NDJ24

Durée : 3 jours

Classe : Présentiel / à distance

Public

- Cette formation s'adresse aux développeurs et ingénieurs logiciels souhaitant évoluer vers des rôles d'architecte microservices.

Prérequis

- Bonnes bases en développement logiciel
- Connaissance d'un langage comme Java
- Notions d'architecture applicative

Objectifs

- Comprendre l'architecture microservices
- Identifier les principaux design patterns
- Gérer les communications entre services
- Mettre en place des solutions scalables

Programme détaillé

1- Introduction aux microservices

- Les architectures monolithiques traditionnelles : avantages et inconvénients
- Les architectures microservices :
 - Définition(s)
 - Comparaison avec SOA
 - Scalabilité, Maintenabilité et Évolutivité
 - Processus et organisation des équipes
 - Avantages et Inconvénients
- Patterns et relations entre patterns
- Les patterns et leurs relations aux microservices

2- Stratégies de décomposition

- Approche DDD
- Approche par sous-domaines
- Définition des APIs
- Atelier:
 - Décomposition de l'application monolithique en microservices

3- Communications entre microservices

- Modes d'interaction, APIs évolutives, formats des messages
- Modèle RPC
- Modèle asynchrone
- Atelier:
 - Mise en place des deux modèles de communication

4- Gestion des transactions

- Le besoin pour des transactions distribuées
- Le pattern Saga, Coordination
- Isolation des transactions
- Atelier:
 - Illustration du pattern Saga, utilisation du message broker Kafka



Formation Design patterns pour les microservices

Code : NDJ24

Durée : 3 jours

Classe : Présentiel / à distance

5- Logique métier

- Patterns d'organisation de la logique métier
- DDD, définitions des frontières métier
- Publication des événements métier, développement piloté par les événements
- Saga et le modèle événementiel
- Atelier:
 - Saga et le modèle événementiel

6- Requêtage

- Pattern de composition d'API
- CQRS Pattern
- Vues CQRS
- Atelier :
- Mise en place de vues

7- API Externe

- Problématique de design, diversité des clients
- Pattern gateway
- Atelier:
 - Mise en place de deux gateways pour deux clients différents

8- Tests

- Tests et pipeline de déploiement
- Tests unitaires : Entity, Saga, Services métier, contrôleurs, gestionnaires de messages, Mocking
- Tests d'intégration
- Tests d'acceptance et e2e (end-to-end)
- Atelier:
 - Tests unitaires automatisés

9- Production

- Fonctionnalités demandées par la production
- Services configurables, observables et sécurisés
- Microservices Chassis, Service mesh
- Atelier:
 - Présentation Istio/Kubernetes

