

Formation Design patterns pour les micro services

Description de la formation Design Patterns pour Micro services

Les approches et les objectifs de la démarche DevOps ont eu une influence sur les architectures des systèmes informatiques. En conséquence, les systèmes migrent désormais d'architectures monolithiques vers des architectures micro services.

Cette migration n'est pas sans risque et de nombreuses décisions doivent être prises, concernant notamment : la décomposition du monolithe, l'organisation des données et la persistance, les modes d'interaction entre services, la sécurité, les aspects transactionnels, etc.

Pour répondre à toutes ces problématiques, il existe différents « Design patterns » (modèles de conception) spécifiques. Cette **formation Design patterns pour les micro services**, technique et destinée à des développeurs/architectes, vous propose de les parcourir et de les illustrer par des ateliers pratiques.

Objectifs

Objectifs opérationnels :

Connaître et appliquer les bons design patterns pour répondre aux problématiques soulevées par la migration d'une architecture monolithique vers une architecture micro services.

Objectifs pédagogiques :

À l'issue de cette **formation Design Patterns pour Micro services**, vous aurez acquis les connaissances nécessaires pour :

- Comprendre les bénéfices attendus d'une architecture micro services
- Démarrer une migration depuis une architecture monolithique
- Appréhender les problématiques à résoudre
- Connaître les solutions architecturales avec leurs avantages et inconvénients

À qui s'adresse cette formation ?

Public :

Ce cours Design Patterns pour Micro services cible principalement des développeurs SSII souhaitant devenir architectes.

Prérequis :

Pour suivre cette formation Design patterns dans des conditions optimales, il est conseillé d'avoir de solides connaissances dans au moins un langage de développement (Java par exemple).

Contenu du cours Design Patterns pour Micro services

Introduction aux micro services

Influence du DevOps sur les architectures et les infrastructures

Les architectures monolithiques traditionnelles : avantages et inconvénients

Les architectures micro services : - 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 micro services



ITgate

Training

Your Gateway to Excellence

Stratégies de décomposition

Approche DDD

Approche par sous-domaines

Définition des APIs

Atelier

Décomposition de l'application monolithique en micro services

Communications entre micro services

Modes d'interaction, APIs évolutives, formats des messages

Modèle RPC

Modèle asynchrone

Atelier

Mise en place des deux modèles de communication

Gestion des transactions

Le besoin pour des transactions distribuées

Le pattern Saga, Coordination

Isolation des transactions

Atelier

Illustration du pattern Saga

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 évènementiel

Requêtage

Pattern de composition d'API

CQRS Pattern

Vues CQRS

Atelier

Mise en place de vues

API Externe

Problématique de design, diversité des clients

Pattern Gateway

Atelier

Mise en place de deux Gateway pour deux clients différents

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

Production

Fonctionnalités demandées par la production

Services configurables, observables et sécurisés

Micro services Châssis, Service mesh

Atelier

Présentation Istio/Kubernetes

Travaux Pratiques

Un atelier sous forme de projet servira de fil conducteur durant toute la durée de la formation.