

(WS2) Microservices en Java (Spring)




Microservices, concepts, patterns, solutions
et mise en oeuvre avec Spring Cloud et Spring Boot

Cette formation couvre tous les aspects des architectures microservices en introduisant les concepts et les patterns, pour aboutir au développement et déploiement de services distribués résilients et scalables. Elle permet d'acquérir une compréhension des enjeux des architectures microservices afin d'être mises en place sur vos projets.

Ce parcours vous apprendra à modéliser, développer, déployer, monitorer, consommer et exposer des microservices avec la plate-forme Spring Cloud et Spring Boot.

Détails

Durée : 5 jours

- 2 jours pour introduire et développer une architecture microservices en Java (FR )
- 2 jours pour assister à des conférences autour des microservices (EN )
- 1 jour pour développer des microservices avec Java EE et MicroProfile (EN )

Public

- Architectes
- Développeurs

Pré-requis

- Connaissance du langage Java
- Avoir déjà développé des applications web en Java

Objectifs

- Comprendre les patterns des architectures microservices
- Comprendre comment s'articulent les différentes briques techniques
- Découvrir les concepts liés aux systèmes distribués
- Comprendre les contraintes de migration des monolithes vers les microservices
- Développer des microservices en utilisant Spring Cloud et Spring Boot

Dates :

- 25 au 31 octobre 2018

Tarifs :

- 5 jours 1490 €HT

Programme du jeudi 25 et vendredi 26 octobre

Ces deux journées vous donnerons les bases théoriques pour comprendre les architectures microservices et les différencier des monoliths. Une fois les concepts et patterns présentés, vous aurez à développer une application en prenant en compte les contraintes des architectures distribuées. Enfin, vous terminerez ces deux jours en résumant les avantages et inconvénients des microservices afin de savoir si vos futurs projets pourront en tirer profits.

Il y aura des ordinateurs à votre disposition, mais vous pouvez amener le votre si vous préférez.

Principe des architecture microservices

- Agilité et monolithes
- Agilité et microservices
- Définitions
- Avantages et inconvénients

Patterns, vocabulaire et concepts

- Organisation
- Stockage des données
- Interface utilisateurs
- Services et services distribués
- Sécurité
- Scalabilité et disponibilité
- Monitoring
- Déploiement et environnements

Services REST

- Développer et exposer des microservices REST
- Documenter les services avec Open API
- Consommer les services avec Swagger Codegen et Feign
- Gérer les CORS
- Afficher des données sur une SPA Angular + TypeScript + Bootstrap
- Enregistrer et découvrir les services avec Consul

Monitorer les microservices

- Centraliser les logs
- Utiliser ELK
- Configurer à chaud les microservices

Gestion des pannes et Scale des microservices



- Circuit Breaker Hystrix
- Load balancer Ribbon
- Scaler horizontalement

Sécuriser les microservices

- Gateway
- OAuth 2 et JWT

Enjeux des architectures microservices

- Enjeux business
- Enjeux techniques
- Enjeux de production
- Avez-vous besoin d'une architecture microservices ?
- Patterns de migration

Intervenants : Antonio Goncalves (Java Champion), Jean-Michel Doudoux (Java Champion), Rémi Nguyen (Oxiane), Jean-François Lefevre (Oxiane)

Programme du lundi 29 et mardi 30

Durant ces deux jours vous participerez à la conférence Voxxed Days Microservices. Vous aurez ainsi la possibilité d'assister à plus de 20 conférences en anglais sur les sujets liés aux architectures microservices:

- Architecture Microservice : définition, stratégies de migration, patterns et anti-patterns
- Modélisation : (DDD) Domain Driven Design
- Stockage de données : Bases de données, patterns de synchronisation de données, (CDC) Capture Data Change, Event Sourcing et CQRS
- Remote Procedure Invocation : REST, gRPC, APIs, contrats d'API
- Messages : messaging, evenements
- Integration : Proxy, gateway, circuit breaker, load balancer, service registry, CORS
- Interfaces utilisateurs : techniques et patterns d'agrégation d'interface utilisateurs
- Cloud, Containers & Infrastructure : Docker, Kubernetes, Istio, Vagrant, Cloud, PaaS
- Monitoring : Health check, aggregation des logs, exception tracking, metriques, audit, alerting, tracing distributé
- Sécurité : autorisation, authentification, OAuth2, Open ID, JWT
- Tests : testers les microservices en isolation, tests d'APIs, tests d'integration, Chaos testing
- Scaling : horizontal, vertical, disponibilité
- Organisation & Culture : methodologies de developpement, culture DevOps et DevSecOps, agilité
- Retours d'expérience : mise en place de microservices, comment découper un monolith

Intervenants : Josh Long (Pivotal), Ken Finnigan (RedHat), Emily Jiang (IBM), Bridget Kromhout (Microsoft), David Pilato (Elastic), Cyrille Martraire, Gunnar Morling (RedHat), Katherine Stanley (IBM), David Gageot (Google), Mete Atamel (Google), etc.

Programme du mercredi 31 octobre

Durant ce workshop d'une journée, vous développerez une architecture microservices assez complexe avec la plateforme Spring Boot, Spring Cloud et Netflix OSS.

Après une introduction théorique à JHipster, Spring Cloud et Spring Boot, vous aurez toute une journée pour développer plusieurs microservices. De la base de données (relationnelle et orientée documents) à l'interface utilisateur (Angular ou React), en passant par la couche de persistance (Spring Data), vous mettrez en place tous les patterns de résilience liés aux microservices. Enfin, vous aurez à packager et à déployer vos microservices sur le cloud.

Générateur de code JHipster

- Architecture et fonctionnalités
- Historique et fonctionnement du projet
- Technologies utilisées
- Workflows de développement proposés
- Introduction aux technologies principales: Spring Boot, Angular et React

Architecture microservices

- Gateway (Zuul, Traefik)
- Registry et service discovery (Eureka, Consul et Istio)
- Monitoring (stack Elastic, Prometheus)
- Cache distribué (Hazelcast, Ehcache, Infinispan, Memcached)
- Sécurité avec Spring Security (et un détail des modifications/améliorations proposées par JHipster)

Développement de plusieurs microservices

- Spring Boot et Spring Cloud pour démarrer rapidement une gateway et des microservices
- Netflix Eureka ou Consul pour gérer l'annuaire de services
- Spring Cloud Config pour configurer l'ensemble de l'architecture
- Sécurité JWT ou OAuth2 (avec Keycloak)
- API gateway avec Zuul
- Monitoring avec la stack Elastic
- Docker pour tester facilement l'ensemble de l'architecture localement, et pour passer en production

Intervenants : Julien Dubois (créateur de JHipster), Pierre Besson et Pascal Grimaud (tous deux membres de la "core team" de JHipster)