

## Programmation Réactive avec WebFlux

### Programmation Réactive avec WebFlux — Construire des flux réactifs avec Spring

Avec Spring WebFlux, la galaxie Spring prend une orientation nouvelle : la construction d'applications web (API) et de microservices basés sur une stack réactive.

Cette formation présente les concepts fondamentaux de la programmation réactive et les met en pratique en s'appuyant sur les éléments proposés par le framework WebFlux.

#### Détails

- **Code** : JF-PRW
- **Durée** : 3 jours ( 21 heures )
- **Public**
  - Développeurs Back-End
- **Pré-requis**
  - Connaissances du framework Spring Boot
  - Connaissances de Java 8

#### Objectifs

- Comprendre Reactor (Mono & Flux et leurs opérateurs)
- Réaliser un microservices réactif de l'api REST jusqu'à la couche de persistance
- Savoir implémenter un client Http réactif avec le WebClient
- Monitorer son système réactif
- Mettre en place des tests unitaires et d'intégrations

#### Programme

##### Introduction

- La programmation réactive
- Le Manifest
- Systèmes non bloquants
- Les patterns
- Principe de l'Event-Loop

##### En Java

- Reactive Stream et la backpressure
- Les interfaces Reactives de Java 9
- Tour d'horizon des Frameworks
- Java NIO
- Netty

##### Spring WebFlux

- Présentation
- Architecture
- Cas d'utilisation

##### Reactor

- Flux et Mono
- Séquences et Souscriptions
- Les opérations : reduce, zip, merge, flatmap, etc
- La gestion d'erreur
- La backpressure
- Le modèle de thread scheduling de Reactor

##### API Rest Reactive avec WebFlux

- WebFlux vs Web MVC

- Cas d'usage
- Les Router et les Handler
- La Gestion d'erreur

##### Accéder aux données de façon réactive

- Avec R2DBC pour les bases relationnelles
- Reactive Repository et le NoSQL
- Alternatives pour les repositories synchrones

##### Le WebClient

- Appeler des API Rest de façon réactive
- Gestion des erreurs et du Retry
- Configuration proxy et OAuth2

##### Monitorer son application réactive

- Spring Boot Actuator
- Micrometer
- Tracing avec Sleuth/Zipkin

##### RSocket

- Cas d'usage
- RSocket vs gRPC
- RSocket et Spring

##### Tester son application réactive

- StepVerifier
- WebClient

##### Conclusion

## Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages