

# Formation Spring Avancé : Programmation réactive avec Spring WebFlux et Spring Data Reactive

<b>Durée :</b>	3 jours
<b>Public :</b>	Développeurs Java EE - Architectes
<b>Pré-requis :</b>	Maîtriser la programmation orientée objet en Java et les bases du framework Spring Web et Spring Boot
<b>Objectifs :</b>	Comprendre les fondamentaux de la programmation réactive - Construire une API réactive avec Spring WebFlux et Spring Data Reactive
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	JAV101967-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

## Comprendre la programmation réactive

Spécification Reactive Stream et la librairie Reactor  
La programmation réactive : reactive Manifesto, architecture micro-services, event driven  
Les systèmes non bloquants  
Tour d'horizon des Frameworks : Spring Reactor, Spring WebFlux  
Programmation réactive avec Spring Reactor : présentation, principes et mise en place  
Mono et Flux  
Les opérateurs : reduce, flatmap, zip, merge ...etc  
La gestion d'erreurs  
BackPressure (la contre pression)  
Le modèle de thread Scheduling de Reactor

**Atelier: Installation de l'environnement de développement - création d'un projet Spring Boot et configuration de Spring Reactor et implémentations**

## Construire une API Rest réactive avec Spring WebFlux

Netty: principe de fonctionnement  
Spring WebFlux vs Spring web MVC  
Annotations vs programmation fonctionnelles  
Functional Web avec Spring WebFlux : principes, ends points, validation, routeurs  
Intercepteurs et gestion des erreurs  
Construction d'un client web réactif  
Implémentation de tests

**Atelier : Création d'une API REST réactive**

## **Gérer la couche de persistance avec Spring Data Reactive**

Spring Data Reactive : principe de fonctionnement

Utilisation des types Mono et Flux, BackPressure

Spring Data R2DBC pour les bases de données relationnelles Spring Data Reactive et le NoSQL

Requêtes personnalisées avec le Reactive Repository

**Atelier: Implémentation d'une couche de persistance avec Spring Data Reactive (avec une base de données SQL ou NoSQL)**

## **Utiliser les Server Sent Events (SSE) avec Spring WebFlux**

SSE : mécanisme, normalisation, cas d'usage

Server Sent Events vs Web Sockets

Schéma de fonctionnement, format des messages SSE

Implémentations : côté client, côté serveur

Stream events avec Spring WebFlux

Gestion de la connexion (fermeture, reconnexion)

Sécurité de la communication et utilisation de token

**Atelier : Implémentation d'une communication serveur-client en utilisant les SSE**