Formation Java Avancé : Programmation réactive

Durée :	2 jours
Public :	Développeurs Java
Pré-requis :	Avoir suivi le stage "Java initiation+approfondissement" ou posséder les connaissances équivalentes - Notions en programmation fonctionnelle
Objectifs :	Comprendre l'intérêt de la programmation réactive - Connaître la spécification Reactive Stream et ses implémentations (Reactor, RxJava, Java 9 Flow)- Maîtriser la librairie Reactor- Traiter des problèmes de programmation concurrente - S'appuyer sur un modèle de communication asynchrone
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	JAV101968-F
Note de satisfaction des participants:	5,00 / 5

Comprendre la programmation réactive

Programmation réactive : concepts, intérêt
Tour d'horizon de l'écosystème

Découvrir la spécification Reactive Stream

La programmation fonctionnelle
Utilisation des lambdas
Pattern Observer
Les opérateurs
Publisher et Subscriber
Flux d'événements asynchrone
Implémentation par Reactor (Flux/Mono) et RxJava

Atelier : Multiples exemples d'implémentations réactives

Maîtriser les concepts avancés de Reactive Stream

La souscription avec subscriber() Gestion de la contre-pression (backpressure) Gestion du multithreading et la concurrence avec Reactor Les Publishers de Reactor Eager vs lazy : comparatifs entre just(), defer() et ses dérivés push vs pull : stratégies de gestion de la contre-pression (backpressure) hot vs cold : cas d'usage avec les processors Comparatif avec RxJava Les opérateurs de Reactor Transformer et filtrer les événements Combiner plusieurs sources Écrire son propre opérateur Tests unitaires avec StepVerifier

Atelier : Mise en place de Reactor - Utilisation des interfaces publisher et subscriber - Utilisation des Schedulers - Gestion des erreurs et tests unitaires