

Durée : 2 jours soit 14h**PRESENTATION :**

La formation est composée de 4 ateliers. Chacun comprend un volet théorique, permettant aux stagiaires d'acquérir les bases de résolution du problème à partir d'un symptôme commun, et un volet pratique, dans lequel les stagiaires sont invités à diagnostiquer puis corriger les problèmes sur une application réelle.

- Atelier 1 : Les temps de réponse de mon application sont mauvais
- Atelier 2 : Les temps de réponse de mon application se dégradent avec la charge
- Atelier 3 : Mon application ne répond plus
- Atelier 4 : J'ai une fuite mémoire

ROGRAMME**ATELIER 1 : LES TEMPS DE RÉPONSE DE MON APPLICATION SONT MAUVAIS**

Durant cet atelier, nous aborderons les sujets suivants :

- Généralités sur la notion de performance des applications Web
- Top Ten des problèmes de performances des applications Web Java
- Méthodologie et outils pour l'analyse des performances d'une application Java
- Route Cause Analysis : identifier l'origine d'un problème de performance
- Architecturer et concevoir des applications performantes

ATELIER 2 : LES TEMPS DE RÉPONSE DE MON APPLICATION SE DÉGRADENT AVEC LA CHARGE

Durant cet atelier, nous aborderons les sujets suivants :

- Principes et patterns de scalabilité des applications Web
- Les différents types de contentions; Méthodologie et outils pour les tests de charge; Typologie des tests de charge : pic, limite, endurance, simulation
- Identifier et supprimer les contentions
- Architecturer et concevoir des applications scalables

ATELIER 3 : MON APPLICATION NE RÉPOND PLUS

Durant cet atelier, nous aborderons les sujets suivants :

- Monitorer les ressources contraintes : pools de threads, de connexions, d'objets, caches, etc.
- Dead-locks java, dead-locks système, couplage : les pièges des architectures distribuées
- Outils et techniques pour identifier l'origine d'un blocage
- Principes et patterns des architectures asynchrones

ATELIER 4 : J'AI UNE FUITE MÉMOIRE

Durée : 2 jours soit 14h

Durant cet atelier, nous aborderons les sujets suivants :

- Principales sources des fuites mémoire java
- Bonne pratiques de développement pour la gestion mémoire en Java
- Heap dumps et profilage : diagnostiquer une fuite mémoire
- Comprendre les messages du Garbage Collector