

Formation Hibernate, Maîtriser et optimiser les performances

Objectifs de la formation Hibernate

La mise en place d'une couche de mapping objet-relationnel entre un modèle métier « objet » et une base de données relationnelle facilite considérablement la problématique de persistance d'une application. Cependant, les deux modèles de données et de programmation sous-jacente n'ont rien à voir, ce qui occasionne des difficultés connues sous le terme d'« impudence mismatch ». Ces difficultés sont visibles lorsqu'une application Java devient peu performante ou instable. Avant d'en arriver là, la connaissance précise de l'outil de mapping et le respect de ses règles de développement sont impératives. Réaliser un mapping correctement adapté à ses besoins reste une étape très délicate qui peut nécessiter plusieurs ajustements de sa stratégie de persistance afin de trouver le bon compromis entre performances, stabilité et impédance faible, en fonction du contexte applicatif. Hibernate offre un large panel d'options de configuration et de représentation ainsi que des fonctionnalités spécifiques qui s'inscrivent dans cette recherche d'optimisation. Ce cours permettra aux stagiaires de bien comprendre et d'expérimenter les avantages et inconvénients de chaque solution offerte pour choisir le compromis adapté face à certaines situations récurrentes.

Cette formation Hibernate vous permettra de:

- Connaître et assimiler les problématiques classiques de performance liées au mapping objet-relationnel
- Maîtriser les concepts avancés d'Hibernate liés à la performance
- Maîtriser les bonnes pratiques de développement Hibernate
- Savoir utiliser les différents caches Hibernate.

À qui s'adresse cette formation ?

Public :

Ingénieurs Architectes Chefs de projets

Prérequis :

Très bonne pratique de Java Connaissance et pratique Hibernate de base

Contenu du cours Hibernate

Objectifs

Mapping Objet-Relationnel et « impudence mismatch »

Problématiques liées au chargement des données

Le lazy loading ou chargement par nécessité

Notion et utilisation de Proxy,

Les stratégies de fetch

Utilisation des caches d'Hibernate

Le cache de session

Le cache de second niveau

Le cache mapping

Les stratégies de cache,

Avantages et inconvénients des différentes implémentations,

Le cache de requête.

Partage des données

Problématiques liées à la concurrence d'accès

Verrouillage optimiste/pessimiste,

Clustering,

JBossTrecache

Optimisation des associations

Présentation des principes et techniques
Classes techniques de type Stub ou Skeleton
Pattern proxy
Illustration au travers de RMI, CORBA et EJB

Persistance des composants

Cas des associations bidirectionnelles
Gestion de l'attribut inverse
Associations polymorphes

Problématiques liées à l'héritage

Une table par hiérarchie
Une table par sous-classe
Une table par classe concrète

Monitoring des performances

Suivi d'une Session Factory
Métriques