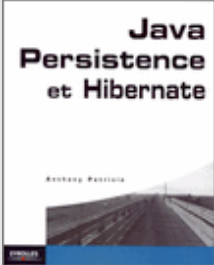




zenika  
ARCHITECTURE INFORMATIQUE

www.zenika.com

Repartez avec le livre



Durée  
2 jours

Répartition  
50% théorie  
50% pratique

Pré-requis  
Connaître Hibernate

Public  
Architecte  
Développeur  
Chef de projet

Tarif (déjeuners inclus)  
1200 € (HT)  
Tarif (1 mois avant)  
1100 € (HT)

Lieu  
Paris 9ème

Sessions 2009  
29 au 30 janvier  
19 au 20 mars  
11 au 12 mai  
29 au 30 juin  
17 au 18 septembre  
19 au 20 novembre

Intra-entreprise sur  
demande

Inscription et  
renseignements  
+33(0)1.45.26.19.15  
training@zenika.com  
www.zenika.com

# Hibernate performance

## Maîtriser et optimiser Hibernate

### Contexte

Comment limiter les accès à la base de données ? Maîtriser l'initialisation paresseuse ? Mesurer l'activité de l'outil ? Que fait la méthode « update » de la Session ?

Cette formation est construite du retour d'expériences projets de nos consultants et certaines des notions abordées ces deux jours sont mises en pratique dans la librairie [ZenTracker](#), proposée en licence LGPL.

### Objectifs

- Maîtriser les fonctionnalités avancées d'Hibernate
- Améliorer l'intégration d'Hibernate en environnement JEE
- Savoir paramétrer finement Hibernate et maîtriser les accès à la base de données

### Contenu

#### Configuration

- Comment configurer Hibernate
- Quels sont les paramètres importants
- Définition des « Naming Strategy »
- Comment mesurer l'activité d'Hibernate
- Apprendre à activer et utiliser les statistiques

#### Opérations de Session

- Comment Hibernate gère les entités
- Définition des 3 états d'une entité et comment passer d'un état à un autre (Transient, Persistent, Detached)
- Conséquences et impacts sur les performances liés à chaque état
- Comment fonctionne la méthode flush()
- Ré-attacher un objet à la session et différence entre les méthodes merge(), update() et lock()

#### Relations avancées

- Fonctionnement de l'attribut cascade
- Comment propager le changement d'état d'une entité vers ses associations
- Impacts et conséquences liés à l'utilisation de liens bidirectionnels
  - Code Java et performance
- L'attribut de mapping « inverse »

#### Interception et événements

- S'immiscer au coeur de la session Hibernate
- Comprendre le modèle événementiel
- Comment rajouter et enrichir les listener par défaut d'Hibernate
- Comment connaître le type exact des entités manipulées par la Session
- Quels sont les champs modifiés d'une entité mise à jour, valeur avant et après

#### Batch

- Utilisation de batch pour la suppression et la mise à jour de masse
- Stratégie de génération d'identifiants : Impact sur le mode batch

#### Cache

- Comment configurer le cache de second niveau
- Comment configurer et utiliser le cache de requêtes

#### Requêtes

- Comment optimiser ses requêtes afin de limiter les accès à la base de données
- Bien utiliser et comprendre les stratégies de fetch
- Comment charger des objets sans tenir compte du mapping
- Optimiser l'utilisation des Criteria
- Projections et ResultTransformer, intérêt pour l'optimisation et la performance
- Gestion de la pagination
- Les requêtes SQL
- Bien utiliser l'attribut default\_batch\_fetch\_size
- Comment éviter un ResultSet dont la taille correspond au produit cartésien des lignes des tables de jointures

#### Mappings avancés

- Les filtres, utilisés comme des « vues » de base de données paramétrables
- Différence entre les filtres exécutés au niveau base de données et les filtres exécutés au niveau mémoire
- Comment parcourir une collection chargée de manière paresseuse, sans déclencher le chargement complémentaire
- Cas d'utilisation des formula