



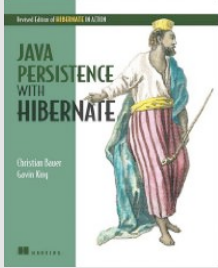
zenika  
ARCHITECTURE INFORMATIQUE

www.zenika.com

# JPA - Hibernate

## Gérer les données d'une application Java/JEE avec JPA

Repartez avec le livre



Durée  
3 jours

Répartition  
50% théorie  
50% pratique

Pré-requis  
Connaissance de Java

Public  
Architecte  
Développeur  
Chef de projet

Tarif (déjeuners inclus)  
1490 € (HT)  
Tarif (1 mois avant)  
1290 € (HT)

Lieu  
Paris 9ème

Sessions 2009  
7 au 9 janvier  
9 au 11 février  
9 au 11 mars  
22 au 24 avril  
4 au 6 mai  
3 au 5 juin  
6 au 8 juillet  
5 au 7 août  
9 au 11 septembre  
12 au 14 octobre  
2 au 4 novembre  
7 au 9 décembre

Intra-entreprise sur demande

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

### Objectifs

- Assimiler les concepts et les enjeux du mapping objet / relationnel
- Maîtriser les principales fonctionnalités de JPA
- Développer et intégrer une couche de persistance dans une application JavaSE ou JavaEE

### Contenu

#### Introduction et Architecture

- Concepts et enjeux de la persistance des données
- Les différents modes d'accès aux données : simple appels JDBC, les EJB1 et EJB2 entités, les autres ORM
- Standardisation des ORM : naissance de la spécification Java Persistence API

#### Mise en oeuvre d'une implémentation

- Les produits implémentant la spécification JPA
  - Hibernate
  - Oracle TopLink
- Autres implémentations : BEA Kodo, OpenJPA, ...
- Installation du produit et paramétrage des fichiers de configurations

#### Intégration dans l'architecture

- Intégration dans une application JEE en mode conteneur
- Intégration dans le conteneur léger Spring
- Intégration directe dans une application WEB

#### Définition d'une entité

- Pont entre le monde objet et le monde relationnel
- Une entité est avant tout un POJO matérialisé par quelques éléments : une classe Java, une définition de mapping et une clé primaire
- Comparaison entre beans Entité et beans Session
- Simplification d'accès à l'intérieur et à l'extérieur d'un container JEE

#### Manipulation d'entités

- Il ne s'agit plus d'opérations isolées de créations ou de suppressions mais d'opérations gérées par un cycle de vie
- Introduction du contexte de persistance reliant les entités à la base de données : l'EntityManager
- Détail des différents états d'une entité : neuf, managé, détaché, supprimé
- Mise en relation des différents états d'une entité avec les deux types de contexte de persistance : portée d'une transaction ou portée étendue
- Réalisation des opérations CRUD de manipulation d'une entité avec l'API EntityManager
- Utilisation des événements générés lors des transitions d'états du cycle de vie

#### Exploration des mappings

- Mise en place facilitée par les annotations JavaSE 5
- Alternative ou complémentarité des fichiers de configuration XML
- L'identité : le garant de la correspondance unique entre instances d'objets et enregistrements en base
- L'importance du choix de la stratégie de génération d'identifiants

#### Relation entre entités

- Présentation des types de relation : 1-1, 1-n, n-1 et n-m
- Choix du sens des relations : unidirectionnelle VS bidirectionnelle
- La persistance en cascade
- Les initialisations paresseuses : lazy-loading

#### Héritage

- Présentation des trois types de stratégie d'héritage entre entités
- Avantages et inconvénients de chaque mode
- Les autres modes d'héritage

#### Le langage de requête JPQL

- Un mode de requêtage du monde objet basé sur SQL
- Différence avec le langage de requêtes SQL natives
- Exploration des opérateurs du langage JPQL
- Présentation du mécanisme des requêtes dynamiques et des requêtes statiques
- La puissance des requêtes polymorphiques
- Améliorer les performances avec l'opérateur «fetch»

#### Transactions et accès concurrents

- Les propriétés d'une transaction
- Mise en place des transactions : « JTA » et « Resource-local »
- Les techniques de verrouillage : « Optimistic locking » et « Explicit read and write lock »