



JPA (Java Persistence API)

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

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 »

Durée
3 jours (21h)

Répartition
50% théorie
50% pratique

Pré-requis
Connaissance de Java

Public
Architecte
Développeur
Chef de projet

Tarif (déjeuners inclus)
1 450 € (HT)

Sessions 2008 - Paris
19 au 21 mai
16 au 18 juin
21 au 23 juillet
18 au 20 août
15 au 17 septembre
13 au 15 octobre
17 au 19 novembre
15 au 17 décembre

Intra-entreprise sur
demande

Renseignements et
inscription
+33(0)1.45.26.19.15
training@zenika.com
www.zenika.com