

JAV113

Persistence de données en Java avec JPA

Durée

3 jours.

Pré-requis

Connaissance du langage Java et du langage SQL. La connaissance de JDBC serait un plus.

Orientation

Développeurs impliqués dans la construction d'applications Java EE ou Java SE liées à des bases de données relationnelles.

Dates

En intra entreprise

Objectifs

Comprendre les mécanismes de persistance au travers de JPA. Savoir définir un modèle d'objets persistants. Intégrer avec JPA la couche de persistance dans une application Java SE et Java EE. Comparer les différentes fonctionnalités (requêtes, cache) de la bibliothèque.

Organisation du cours

Théorie : 55%

Pratique : 45%

Configuration matérielle

Une machine par stagiaire. Environnement de développement Eclipse + plugin JPA, Serveur d'application Tomcat, Glassfish ou JBoss.

Documentation fournie

Support de cours et exercices corrigés.

Tarif H.T. *

* nous consulter

Programme

Introduction

- Définition et principe de la persistance, techniques de gestion de la persistance et du mapping objet/relationnel
- Les différentes solutions sur le marché
- Liens avec JDBC

Principe de JPA (Java Persistence API)

- Couche d'abstraction, standard de développement
- Les implantations : Hibernate, Oracle TopLink, Apache OpenJPA
- Liens avec les EJB 3

Le gestionnaire de persistance : EntityManager

Configuration de JPA, le fichier persistence.xml

- Gestion de la connexion JDBC
- Intégration avec Spring et Hibernate
- Intégration dans l'application Java

Objets persistants : entités

- Définition d'une entité
- Les fichiers descripteurs de la correspondance et/ou annotations dans la classe
- Définition d'un mapping simple classe/table
- Clé, identité, mapping des champs

Cycle de vie des entités

- Objets persistants et transitoires, notion de session
- Détachement et attachement des objets
- Comment créer, rechercher, modifier, supprimer une donnée ?

Relations entre objets

- Relations unidirectionnelle et bidirectionnelle
- Relations 1-1, relations 1-n et n-m, classes d'association
- Héritage (les différentes stratégies)
- Les classes incluses
- Persistance transitive (mise en cascade)

Le langage de requêtes JPQL (Java Persistence Query Language)

- Langage à base d'objets
- Requêtes simples et paramétrées
- Tri et jointure (utilisation des relations entre objets)
- Sous-requêtes, optimisations avec fetch
- Requêtes natives en SQL (JDBC)

Gestion des transactions

- Rappels sur les transactions
- Gestion par le conteneur (API JTA) ou "locale"
- Verrouillage optimiste ou pessimiste

Optimisations

- Gestion efficace du chargement des objets liés : optimisation des relations en cascade (eager fetching)
- Le chargement paresseux (lazy-loading)

Architecture et Intégration

- Application Java EE dans un conteneur d'EJB
- Application Java Web dans un conteneur Web comme Tomcat
- Application autonome Java SE