

## JPA

### Mettre en œuvre la persistance avec Java Persistence API (JPA)

L'intérêt d'implémenter une couche de mapping objet-relationnel entre un modèle métier « objet » et une base de données relationnelle n'est probablement plus à démontrer : le code qui contient les requêtes SQL est isolé dans un endroit unique, permettant une adhérence réduite au schéma de la base ; le modèle métier peut respecter les principes de la programmation objet, utiliser l'héritage et la navigation.

Issue des travaux du groupe d'experts de la JSR 220 (EJB 3), JPA est une spécification clé de Java EE. Loin d'être un framework de plus, JPA établit une norme standard simplifiée pour la persistance afin de garantir une meilleure portabilité des applications Java EE. JPA reprend naturellement les meilleures pratiques de frameworks établis tels Hibernate ou TopLink. EJB 3 s'appuie sur JPA pour la persistance des entités.

Cette formation permet de comprendre et d'expérimenter JPA par la pratique. Elle est un pré-requis indispensable au cours EJB3.

#### Détails

- **Code** : JE-JPA
- **Durée** : 3 jours ( 21 heures )

#### Public

- Architectes
- Chefs de projets
- Consultants
- Développeurs
- Ingénieurs

#### Pré-requis

- Bonne pratique de Java

#### Objectifs

- Connaître les concepts de la norme JPA,
- Savoir définir et utiliser un modèle persistant,
- Connaître les problématiques liées à la persistance d'objets sur bases relationnelles,
- Comprendre et savoir utiliser le langage de requêtes objet JPQL,
- Connaître les techniques et compromis permettant de garantir de bonnes performances

#### Programme

##### La problématique de persistance

- Le mapping objet-relationnel
- Intérêt de JPA, positionnement
- JPA et les serveurs d'applications
- Historique des solutions, Hibernate, TopLink, EJB
- Java EE : historique
- Les serveurs Java EE

##### Le modèle de persistance

- Définition d'une entité persistante, concept de POJO, les attributs, les relations
- Entité et Identité
- Cycle de vie des entités
- Mise en place d'un mapping
- Mapping sans annotations
- Mapping par annotations
- Déclaration des attributs
- Représentation des différents types de relation entre entités
  - Classe embarquée
  - Relation 0..1
  - Relation 1..N et N..1

##### ◦ Relation N..M

- Manipulations CRUD des entités
- Gestion de la bidirectionnalité (ou relation inverse)
- Gestion de l'héritage et du polymorphisme
- Mapper une entité sur plusieurs tables

#### TP

- Mise en place de la persistance d'un modèle illustrant les différentes configurations présentées

#### Manipulation des Entités

- Le concept d'Entity Manager
- Gestion des Transactions, JTA (java Transaction API)
- Les requêtes : requêtes nommées et requêtes dynamiques, JavaPersistence Query Language
- Définition d'une requête
- Exécution d'une requête
- Requêtes paramétrées
- Syntaxe JPQL

#### TP

- Utilisation des fonctionnalités de requêtage de JPA

## Performances

- Utilisation de Cascade
- Optimisation du chargement et Lazy Loading
- Le verrouillage Optimiste

## Conclusion

- Panorama des implémentations
- Futur des serveurs d'application Java et de la persistance

## Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages