

Eclipse Persistence services (EclipseLink)

L'implémentation d'une couche de mapping objet-relationnel entre un modèle métier « objet » et une base de données relationnelle présente de nombreux avantages. La maîtrise de l'API JPA et la qualité d'une implémentation JPA telle qu'EclipseLink (Eclipse Persistence Services) permettent aux développeurs d'exploiter pleinement les intérêts du mapping objet-relationnel.

Cependant, les modèles relationnel et objet ne sont passimilaires, ce qui occasionne des difficultés connues sous le terme de « impedance mismatch ». Ces difficultés peuvent se traduire par d'importants problèmes de performance.

La connaissance précise de l'outil de mapping et le respect de ses règles de développement sont impératives. C'est pourquoi cette formation ne se limite pas à la partie standard définie dans la spécification JPA : les fonctionnalités spécifiques à Eclipse Persistence Services sont étudiées.

Programme

La persistance des objets

- Le mapping objet-relationnel et la notion "d'impedance mismatch"
- Les solutions du monde Java : Hibernate, TopLink, JDO, EJB.

- Intérêts de la spécification JPA (Java Persistence API).

- Le projet Eclipse Persistence Services et la mise en open-source de TopLink : EclipseLink

Prise en main de JPA et d'EclipseLink

- La notion de POJO
- Développer et tester une entité JPA avec Eclipse et EclipseLink. Les outils proposés par le projet Dali
- Installation et configuration d'EclipseLink. Le fichier persistence.xml de JPA
- La définition du mapping : annotations et/ou fichier XML.
- Utilisation dans un contexte Java EE et dans un contexte client

Manipulation des entités

- Le principe du CRUD
- La manipulation des entités via l'EntityManager de JPA.
- Le cycle de vie des entités : les états 'new', 'managed', 'detached' et 'removed'
- L'utilisation des 'callbacks' de JPA
- La gestion des transactions

Définition du mapping

- Mapping sans annotations
- Mapping par annotations
- Mapper une entité sur plusieurs tables
- Entités et composition
- Définition des clés. Clés multiples, clés auto-incrémentées?

- Principe de la gestion des relations

- Relations et collections

- Relations 1:1, 1:N, N:1 et M:N

- Spécificités d'EclipseLink pour le mapping des collections.

Le requêtage

- Langage de requêtes JPQL
- Définition d'une requête
- Exécution d'une requête
- Requêtes paramétrées
- Syntaxe de JPQL

Optimisation du mapping

- Définir la politique de chargement ("Fetch type"). Les modes EAGER et LAZY
- L'attribut 'Cascade' pour les relations
- Le mapping de l'héritage
- La gestion des verrous : mise en oeuvre des modes optimiste et pessimiste
- Les optimisations propres à EclipseLink
- Les mécanismes de cache

Détails

Code :

JP-EL1

Durée :

3 jours

Objectifs :

- Être capable de développer une application Java mettant en oeuvre la notion de persistance avec JPA.
 - Mettre en oeuvre JPA avec EclipseLink et connaître les spécificités d'EclipseLink.
 - Connaître les bonnes pratiques

Public :

- Ingénieurs
- Architectes
- Chefs de projets

Pré-requis :

Très Bonne pratique de Java

Tarif : 1350 € H.T. par personne (déjeuner inclus)