

Gérer la persistance des données avec JPA/Hibernate

Programme (Mis à jour le 26/04/2024)

ORM, JPA et Hibernate : de quoi parle t'on ?

- Rappels rapides sur l'API JDBC
- Quels sont les limites de JDBC
- Qu'est-ce qu'un ORM et qu'est-ce qu'il apporte de plus ?
- La solution d'ORM standardisée par Java EE : JPA (Java Persistence API)
- Liens avec l'API EJB 3
- Liens entre Hibernate et JPA
- Les autres implémentations JPA : Open JPA et Eclipse LINK

Démarrage d'Hibernate

- Déploiement d'une base de données MySQL
- Démarrage du logiciel Hibernate
- Définir le mapping via un fichier XML (non standardisé)
- Utilisation de l'API native Hibernate
- Travaux pratiques : définition et utilisation d'un mapping XML avec l'API Hibernate
- Définir le mapping via les annotation JPS
- Utilisation de l'API JPA
- Travaux pratiques : définition et utilisation d'un mapping avec JPA
- Comparaison Mapping XML VS Mapping JPA
- Comparaison API native Hibernate VS API JPA
- Outils JPA/Hibernate proposés par Eclipse
- Travaux pratiques : utilisation des outils proposés par Eclipse
- Dans la suite de la formation, seule l'API JPA sera développée, car elle est standardisée et recommandée par Java EE et Oracle.

Configuration du moteur JPA et des logs

- Configuration JPA
- Intégration avec Log4J
- Activation de certains Logger (ordres SQL produits, paramètres, ...)
- Travaux pratiques : journalisation des requêtes SQL générées par l'ORM JPA

Le mapping JPA

- Mapping des associations unidirectionnelles sans table d'association (OneToOne, ManyToOne et OneToMany)
- Mapping des associations unidirectionnelles avec table d'association (OneToOne, ManyToOne, OneToMany et ManyToMany)
- Mapping des associations bidirectionnelles
- Mapping des collections
- Travaux pratiques : mapping d'un ensemble de classes relatives à la gestion d'un site de vente en ligne.
- Mapping de l'héritage via les trois stratégies disponibles
- Travaux pratiques : mapping des liens d'héritage présents dans notre modèle de classes
- Gestion des cascades

Requêtes JPQL

- Comparatif JPQL (Java Persistence Query Language) et HQL (Hibernate Query Language)

Référence

THIL1985

Durée

3 jours / 21 heures

Prix HT / stagiaire

2175€

Objectifs pédagogiques

- Concevoir ce qu'est un ORM (Object Relational Mapping)
- Analyser les liens entre la notion d'ORM, l'API JPA, et le logiciel Hibernate.

Niveau requis

- Garantir avoir les connaissances des formations : « Java – Programmation orientée Objet : Initiation » et « Java – Programmation orientée Objet : Perfectionnement », ou avoir les connaissances équivalentes.

Public concerné

- Développeurs ainsi qu'aux chefs de projets informatiques

Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

Conditions d'accès à la formation

Délai : 3 mois à 1 semaine avant le démarrage de la formation dans la limite des effectifs indiqués

Moyens pédagogiques et techniques

Salles de formation (les personnes en situation de handicap peuvent avoir des besoins spécifiques pour suivre la formation. N'hésitez pas à nous contacter pour en discuter) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD

Documents supports de formation projetés
Apports théoriques, étude de cas concrets et exercices

Mise à disposition en ligne de documents supports à la suite de la formation

Dispositif de suivi de l'exécution de l'évaluation des résultats de la formation

Feuilles d'émargement (signature électronique privilégiée)

Evaluations formatives et des acquis sous forme de questions orales et/ou écrites (QCM) et/ou mises en situation

Questionnaires de satisfaction (enquête électronique privilégiée)

- Requêtes JPQL simples
- Requêtes JPQL polymorphiques
- Utilisation de fonctions d'agrégation
- Sous-requêtes JPQL
- Gestion des différents types de jointures
- Travaux pratiques : productions et exécution de différents type de requêtes afin d'extraire des commandes de la base de données
- Utilisation de l'API de criteria
- Travaux pratiques : mise en oeuvre d'un criteria

L'API JPA

- Définition et utilisation de paramètres nommés : « Named parameters »
- Définition et utilisation de requêtes nommées : « Named queries »
- Travaux pratiques : mise en oeuvre de « Named parameters » et de « Named queries »
- Gestion des requêtes par lots : « Batch queries »
- Travaux pratiques : réalisation de « Batch queries »
- Cycle de vie d'un objet persistant
- API de manipulation de vos objets persistants
- Travaux pratiques : utilisation de l'API JAP
- Rappels sur le concept de transaction
- Gestion des transactions avec JPA
- Travaux pratiques : utilisation de transactions

Les différentes stratégies de chargement

- Introduction
- Chargement paresseux (Lazy loading)
- Rapatriement immédiat (Eager fetching)
- Chargement explicite
- Bonnes pratiques

Utilisation du cache de premier niveau

- Différences entre les méthodes load et get
- Différences entre les méthodes save et persist
- Lecture et écriture par lots
- Suppression de données dans le cache

Mise en oeuvre d'un cache de second niveau

- Aspects Généraux
- Utilisation de l'implémentation Ehcache
- Utilisation d'un cache de requêtes
- Autres types de caches