

Maîtriser JPA pour la persistance des données en Java

Programme (Mis à jour le 30/01/2026)

Fondations JPA et environnement de travail (1h30)

- Clarifier EntityManager, persistance context, unité de persistance
- Configurer datasource, dialecte, DDL auto, logs SQL
- Comprendre JDBC vs ORM et impacts sur le modèle
- (ATELIER) Démarrer un projet JPA avec base MariaDB

Mapping de base et contraintes relationnelles (2h30)

- Définir @Entity, @Id, @GeneratedValue, @Column, contraintes
- Mapper types Java, dates, enums, converters JPA
- Gérer @Embedded, @Embeddable, clés composées
- Travaux pratiques : Mapper un domaine simple et valider le schéma généré

Associations et chargements maîtrisés (2h30)

- Modéliser @OneToMany, @ManyToOne, @ManyToMany, @OneToOne
- Choisir fetch EAGER/LAZY et éviter N+1
- Configurer cascade, orphanRemoval, owning side, join table
- (ATELIER) Cartographier commandes-lignes-produits avec contraintes métier

Cycle de vie des entités et opérations CRUD (3h)

- Manipuler états managed, detached, removed, transient
- Utiliser persist, merge, remove, flush, refresh
- Gérer validation Bean Validation et erreurs de contraintes
- Travaux pratiques : Implémenter service CRUD transactionnel avec tests

JPQL essentiel pour requêtes métier (3h)

- Écrire SELECT, JOIN, WHERE, ORDER BY, fonctions JPQL
- Paramétrer requêtes typées et sécuriser les paramètres
- Implémenter pagination et tri avec setFirstResult
- (ATELIER) Construire un moteur de recherche multi-critères en JPQL

JPQL avancé et optimisation des accès (3h)

- Utiliser fetch join, projections DTO, agrégations
- Exploiter named queries et hints de requêtes
- Diagnostiquer SQL généré et plans d'exécution
- Travaux pratiques : Optimiser un écran liste en supprimant N+1

Transactions, verrous et cohérence des données (2h30)

- Comprendre ACID, propagation, isolation, rollback
- Utiliser @Transactional et frontières de transaction
- Gérer optimistic locking avec @Version
- (ATELIER) Traiter une mise à jour concurrente avec conflit contrôlé

Bonnes pratiques et mini-projet de synthèse (3h)

- Appliquer patterns Repository, Service, DTO, mapping propre
- Structurer packages et conventions pour portabilité Java EE
- Éviter anti-patterns ORM et dette technique de persistance

Référence

THIL3648

Durée

3 jours / 21 heures

Prix HT / stagiaire

1875€

Objectifs pédagogiques

- Réaliser un mapping objet-relationnel JPA conforme au schéma relationnel
- Implémenter le cycle de vie des entités et les opérations CRUD
- Concevoir des associations JPA fiables avec contraintes et cascade
- Écrire des requêtes JPQL efficaces avec paramètres et pagination
- Diagnostiquer et réduire les problèmes de performance liés à l'ORM
- Gérer les transactions JPA et la cohérence des données concurrentes
- Appliquer des bonnes pratiques pour portabilité et maintenabilité Java EE

Niveau requis

- Bonne connaissance du langage Java et de l'orienté objet
- Bases SQL et concepts SGBD relationnel

Public concerné

- Développeurs Java travaillant sur applications Java EE/Spring
- Chefs de projets techniques encadrant des développements Java

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

- Travaux pratiques : Implémenter un cas production complet et restitution d'équipe