

Formation

Hibernate / JPA en profondeur

Formation intensive en présentiel et en ligne à la demande. Maîtrisez la persistance des données dans les applications Java d'entreprise à l'aide de JPA et Hibernate : mapping objet-relationnel, cycle de vie des entités, requêtes, performances, transactions et bonnes pratiques.

Formation 100% pratique

Exercices & cas d'usage

Formateurs expérimentés

4 jours

24-26 h

FormatAtelier
pratique**Modalité**Présentiel
ou en ligne**Niveau**Intermédiaire
avancé**INSCRIPTION / RÉSERVATION**[Je m'inscris
maintenant](#)

OBJECTIFS PÉDAGOGIQUES

- Assimiler les concepts clés de JPA et de la persistance objet-relationnelle.
- Configurer et utiliser Hibernate dans une application Java.
- Maîtriser le mapping des entités, associations et héritages.
- Écrire des requêtes avec JPQL, Criteria et SQL natif.
- Comprendre le cycle de vie des entités et la gestion des transactions.
- Optimiser les performances (lazy loading, cache, batch fetching, tuning SQL).
- Appliquer les bonnes pratiques de conception autour de JPA/Hibernate.

PUBLIC CIBLE

- Développeurs Java visant la maîtrise de JPA/Hibernate.
- Architectes et chefs de projet techniques sur applications Java d'entreprise.
- Ingénieurs logiciels expérimentés en JDBC ou frameworks Java EE/Spring.

PRÉREQUIS

- Bonne maîtrise du langage Java (classes, interfaces, exceptions, collections).
- Connaissances de base en SQL et en modélisation relationnelle.
- Une première expérience de développement d'applications Java est recommandée.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction à JPA et Hibernate

Contexte et concepts de persistance

- Problématiques de persistance dans les applications d'entreprise.
- Limitations de JDBC brut et motivations pour l'ORM.
- Présentation du standard JPA et de l'implémentation Hibernate.
- Vue d'ensemble de l'architecture Hibernate (Session, SessionFactory, EntityManager).

Mise en place de l'environnement

- Configuration du projet (Maven / Gradle).
- Fichier persistence.xml ou configuration Spring Boot (application.properties / YAML).
- Connexion à une base de données (PostgreSQL, MySQL, Oracle, etc.).
- Stratégies de génération du schéma (create, update, validate).

Mapping des entités et types de données

Entités JPA

- @Entity, @Table, @Id, @GeneratedValue, clés primaires simples.
- Types de colonnes, @Column, contraintes (@NotNull, unique, longueur).
- Valeurs par défaut, conversion de types simples.

Types embarqués et identifiants composites

- @Embeddable et @Embedded pour regrouper des attributs.
- Gestion des clés composites : @EmbeddedId vs @IdClass.

Mapping des dates, enums et types spéciaux

- Types de dates (java.time.*) et annotations associées.
- Enums : @Enumerated STRING vs ORDINAL.
- Mapping de champs larges (LOB, @Lob) et données binaires.



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Associations entre entités

Relations One-To-One et Many-To-One

- Notions de côté propriétaire et inverse.
- @OneToOne, @ManyToOne, @JoinColumn et gestion des clés étrangères.

Relations One-To-Many et Many-To-Many

- @OneToMany unidirectionnel vs bidirectionnel.
- @ManyToMany et table de jointure (@JoinTable).
- Bonnes pratiques de modélisation pour limiter la complexité.

Stratégies de chargement et de cascade

- Chargement LAZY vs EAGER : impacts sur les performances.
- CascadeType (PERSIST, MERGE, REMOVE, ALL...).
- Gestion de l'orphanRemoval et cohérence du graphe d'objets.

Héritage, polymorphisme et mapping avancé

Stratégies d'héritage

- @Inheritance : SINGLE_TABLE, JOINED, TABLE_PER_CLASS.
- Avantages et inconvénients de chaque stratégie.

Classes abstraites et interfaces

- Mapped superclasses (@MappedSuperclass).
- Facteurs communs : audit, champs techniques, etc.

Autres mappings avancés

- @SecondaryTable, vues, mapping de colonnes calculées.
- Utilisation de converters personnalisés (@Converter, AttributeConverter).

Cycle de vie des entités, contexte de persistance et transactions

États des entités

- États transient, managed, detached, removed.
- Persist, merge, remove, detach, refresh, clear.



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Contexte de persistance

- EntityManager vs Session Hibernate.
- Contexte étendu vs transactionnel.
- Synchronisation avec la base (flush, commit).

Gestion des transactions

- Transactions locales vs JTA.
- Intégration avec Spring (annotations @Transactional – sensibilisation).
- Niveaux d'isolation, locking optimiste et pessimiste (versioning, @Version).

Requêtes avec JPQL, Criteria et SQL natif

JPQL (Java Persistence Query Language)

- Syntaxe de base, sélection, filtres, jointures et agrégats.
- Paramètres nommés et positionnels.
- Projections et DTO (SELECT new ...).

API Criteria

- Construction dynamique de requêtes typées.
- Filtres conditionnels, tri, pagination.

Requêtes SQL natives

- Quand et comment utiliser des requêtes natives.
- Mapping des résultats sur des entités ou DTO.

Performances, cache et optimisation

Chargement et N+1 selects

- Comprendre et détecter le problème du N+1.
- Fetch join, entity graphs, réglages de fetch.

Cache de premier et second niveau

- Cache de contexte (first-level cache).
- Cache de second niveau (L2) et cache de requêtes.
- Stratégies de cache, invalidation et cohérence.



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Tuning et bonnes pratiques

- Analyse des requêtes SQL générées (show_sql, logs).
- Indexation, tailles de batch, pagination efficace.

Validation, audit et aspects transverses

Validation des données

- Bean Validation (JSR 380) et annotations de validation (@NotNull, @Size, etc.).
- Validation côté entité vs côté présentation.

Audit et traçabilité

- Champs d'audit (date de création, modification, utilisateur).
- Introduction à Hibernate Envers (sensibilisation).

Gestion des erreurs et robustesse

- Exceptions JPA / Hibernate, rollback, messages d'erreur utiles.

Intégration dans une architecture Java moderne & ateliers pratiques

Intégration avec Spring / Jakarta EE (sensibilisation)

- Repositories, services, couches de l'application.
- Transactions déclaratives, injection de l'EntityManager.

Étude de cas complète

- Conception du modèle d'entités pour une application métier (gestion de commandes, dossiers, etc.).
- Implémentation des mappings, associations, héritage.
- Mise en place de requêtes JPQL/Criteria pour les besoins fonctionnels.

Synthèse & bonnes pratiques

- Récapitulatif des points clés à respecter avec JPA / Hibernate.
- Check-list pour les nouveaux projets et pour l'audit d'une application existante.