

Mis à jour le 23/09/2026

S'inscrire

Formation Jakarta Persistence (JPA)

3 jours (21 heures)

Présentation

Jakarta Persistence (JPA) est la spécification Java de référence pour gérer la persistance des données et le mapping objet-relationnel (ORM) entre les objets d'une application et une base de données relationnelle.

Cette formation Jakarta Persistence (JPA) vous permettra de maîtriser la persistance des données dans vos applications Java avec une mise en œuvre pratique basée sur Hibernate ORM. Vous apprendrez à créer des entités, configurer leurs associations et gérer leur cycle de vie avec l'EntityManager et le contexte de persistance.

Vous serez en mesure de modéliser des relations complexes, de gérer les transactions et accès concurrents et d'interroger efficacement vos données avec JPQL et la Criteria API. Vous apprendrez également à sélectionner les stratégies de chargement adaptées aux besoins de vos applications.

La formation vous permettra d'identifier et de corriger les problèmes de performances courants tels que les requêtes N+1, les chargements inutiles ou les mappings inefficaces. Vous apprendrez à analyser les requêtes SQL générées par Hibernate et à optimiser vos accès aux données grâce aux stratégies de fetch, au batching, à la pagination et aux mécanismes de cache.

Enfin, vous apprendrez à migrer une application JPA historique vers Jakarta Persistence, notamment en maîtrisant le passage de `javax.persistence.*` vers `jakarta.persistence.*` et les évolutions de Jakarta Persistence 3.2 afin de moderniser vos applications Java existantes.

À l'issue de cette formation, vous serez capable de concevoir une couche de persistance robuste, de maîtriser les transactions et les performances de vos applications et d'exploiter efficacement Jakarta Persistence avec Hibernate.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la

Objectifs

- Comprendre l'architecture et le fonctionnement de Jakarta Persistence (JPA)
- Créer et mapper des entités et leurs associations avec une base de données relationnelle
- Maîtriser le Persistence Context, EntityManager et le cycle de vie des entités
- Gérer les transactions et les accès concurrents aux données
- Construire des requêtes avec JPQL et Criteria API
- Identifier et corriger les problèmes de performances liés au mapping et au chargement des données
- Migrer une application utilisant javax.persistence vers Jakarta Persistence 3.2

Public visé

- Développeurs Java
- Développeurs backend
- Ingénieurs logiciels
- Développeurs utilisant JPA ou Hibernate dans leurs applications
- Développeurs chargés de moderniser des applications Java EE existantes

Pré-requis

- Bonne maîtrise de la programmation Java et de la programmation orientée objet
- Connaissances de base de SQL et des bases de données relationnelles
- Compréhension des principes de classes, objets, annotations et collections Java
- Une première expérience avec JDBC, JPA ou Hibernate constitue un plus mais n'est pas obligatoire

Programme de notre formation Jakarta Persistence (JPA) : persistance des données Java

[Jour 1 - Matin]

Comprendre Jakarta Persistence et gérer le cycle de vie des entités

- Comprendre le rôle de Jakarta Persistence (JPA) dans les applications Java et son intégration à Jakarta EE
- Distinguer la spécification Jakarta Persistence de ses implémentations telles que Hibernate ORM
- Configurer une unité de persistance, une datasource et les principales propriétés du provider
- Créer des entités, définir leurs identifiants et comprendre les stratégies de génération des clés
- Maîtriser le Persistence Context, EntityManager et les états transient, managed, detached et removed

- Atelier pratique : configurer Jakarta Persistence avec Hibernate et réaliser les premières opérations CRUD sur une base relationnelle.

[Jour 1 - Après-midi]

Maîtriser le mapping des entités et des associations

- Mapper les propriétés Java vers les tables, colonnes, contraintes et types de données SQL
- Configurer les associations OneToOne, OneToMany, ManyToOne et ManyToMany
- Maîtriser les relations bidirectionnelles, opérations en cascade et suppressions orphelines
- Utiliser les Embeddable, collections de valeurs et clés primaires composites
- Mettre en œuvre les stratégies d'héritage et le mapping d'entités sur plusieurs structures relationnelles
- Atelier pratique : modéliser un domaine métier comportant plusieurs entités, associations, objets embarqués et règles de cascade.

[Jour 2 - Matin]

Gérer les transactions, la concurrence et le cycle de persistance

- Comprendre le fonctionnement des transactions et leur interaction avec le contexte de persistance
- Utiliser les opérations persist, merge, remove, find, refresh, flush et detach de l'EntityManager
- Comprendre le dirty checking et la synchronisation entre les objets Java et la base de données
- Maîtriser le verrouillage optimiste et pessimiste pour gérer les accès concurrents
- Utiliser les callbacks et listeners du cycle de vie pour réagir aux événements de persistance
- Atelier pratique : implémenter un traitement transactionnel et résoudre un scénario de modification concurrente des données.

[Jour 2 - Après-midi]

Interroger les données avec JPQL et Criteria API

- Écrire des requêtes avec Jakarta Persistence Query Language (JPQL)
- Utiliser les paramètres, jointures, agrégations, projections, sous-requêtes et requêtes typées
- Construire des requêtes dynamiques avec la Criteria API
- Mettre en œuvre la pagination et choisir la stratégie de requêtage adaptée au besoin
- Exploiter les fonctionnalités actuelles de Jakarta Persistence 3.2, notamment `getSingleResultOrNull()` et les évolutions de JPQL et Criteria
- Atelier pratique : construire plusieurs requêtes JPQL et Criteria puis comparer leur pertinence selon différents besoins métier.

[Jour 3 - Matin]

Optimiser les performances avec Jakarta Persistence et Hibernate

- Comprendre les stratégies LAZY et EAGER et leurs conséquences sur les accès aux données
- Identifier et corriger le problème N+1 avec les fetch joins et stratégies de chargement adaptées
- Analyser les requêtes SQL générées par Hibernate et diagnostiquer les accès inefficaces
- Optimiser les traitements avec le batching, la pagination et les mécanismes de cache appropriés
- Identifier les anti-patterns ORM et résoudre les principaux problèmes liés aux sessions et entités détachées
- Atelier pratique : diagnostiquer une application présentant des requêtes excessives puis optimiser son mapping et ses stratégies de chargement.

[Jour 3 - Après-midi]

Moderniser et migrer vers Jakarta Persistence 3.2

- Comprendre l'évolution de JPA vers Jakarta Persistence et le passage de javax.persistence.* vers jakarta.persistence.*
- Mettre à jour les dépendances, imports, configurations et versions de Hibernate d'une application existante
- Exploiter les types Java modernes et privilégier java.time pour la gestion des données temporelles
- Utiliser les évolutions de Jakarta Persistence 3.2 telles que les records Embeddable, PersistenceConfiguration et SchemaManager
- Tester la compatibilité des mappings, requêtes et transactions après migration et préparer les tests de non-régression
- Atelier pratique : migrer une application utilisant javax.persistence vers Jakarta Persistence 3.2 avec Hibernate et corriger les incompatibilités rencontrées.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des

séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.