

Introduction à la Persistance de données avec JPA

Description

Le cours Java Persistence API est une introduction, étape par étape afin de créer des applications utilisant le framework de persistance Java. La Java Persistence API est également connue sous le nom persistance EJB3 ou persistance JEE 5. C'est un framework léger pour la création d'applications de base de données Java. Il vous permet de développer les classes persistantes et objets POJO à partir d'une base de données relationnelles à l'aide de concepts tels que association, héritage, polymorphisme, composition et le framework Java collections.

Reference

JPA

Contenu du cours

Module 1 : Introduction à l'API de persistance Java (JPA)

- Leçon 1 : Vue d'ensemble
 - Couches de persistance, mappage objet-relationnel (ORM), JDBC
 - Vue d'ensemble de JPA
- Leçon 2 : Mapping avec JPA
 - Entities et @Entity, ids et @Id
 - Valeurs d'ID générées
 - Types de mappage de base
- Leçon 3 : Unité de persistance et EntityManager
 - Persistance vers la base de données, l'API EntityManager
 - Unités de persistance, configuration, contexte de persistance
 - Recherche d'entités persistantes avec find ()
- Leçon 4 : En savoir plus sur les mappages
 - Mappages par défaut, @Basic, @Column
 - Field vs Property Access
 - Mappages temporels (date / heure)
- Leçon 5 : Options de journalisation (basée sur le fournisseur)

Module 2 : Mises à jour et requêtes

- Leçon 1 : Insertion et mise à jour - Nouvelles entités persistantes, mise à jour d'une instance, suppression d'une instance
- Leçon 2 : Querying et JPQL
 - Requêtes basées sur l'entité, SELECT, WHERE
 - Interface de requête, Exécution de requêtes, Requêtes génériques (JPA 2)
 - Opérateurs, expressions et paramètres JPQL
 - Nomées de requêtes
- Leçon 3 : Conditions de requête supplémentaires
 - Projection de requête, commande, agrégation de requête, génération de mise à jour et suppression
- Leçon 4 : Embedded Objects
 - @Embeddable, @Embedded
 - Définir et utiliser des objets incorporés
 - Clés primaires composées - @EmbeddedID, @IDClass, Définition des clés composées

Module 3 : Le cycle de vie de la persistance

- Leçon 1 : Vue d'ensemble et transactions dans JPA
 - Aperçu de l'API Transaction
 - EntityTransaction (y compris JTA et Resource-local EntityManager)
- Leçon 2 : Le cycle de vie de la persistance
 - États d'entité JPA (Nouveau, Géré, Détaché, Supprimé) et Diagramme d'état d'entité
- Leçon 3 : Contexte de persistance - durée de vie, propagation
 - Synchronisation vers la base de données
- Leçon 4 : Version et verrouillage optimiste
 - Présentation, instances isolées
 - Version, @Version, Verrouillage Optimiste
- Leçon 5 : Rappel de cycle de vie
 - @PrePersist, @PostPersist, etc.
 - Entity Listeners, @EntityListeners

Module 4 : Relations d'entités

- Leçon 1 : Relations Vue d'ensemble: relations d'objet, participants, rôles, directionnalité, cardinalité
- Leçon 2 : Cartographie des relations :
 - Vue d'ensemble de la cardinalité (1-1, 1-N, N-1, N-N)
 - Unidirectionnel et bidirectionnel
 - @ManyToOne, @OneToMany, Structures de table
 - Relationship Inverse - Owning Side
 - Collection Types (Liste, Set, etc)
 - Cascading sur les relations (y compris orphanRemoval - JPA 2)
 - @ManyToMany, @OneToOne
 - Lazy and Eager Chargement
 - Queries Across Relationships (jointures internes, jointures externes, jointures de récupération)
- Leçon 3 : Mappage d'héritage d'entité
- Leçon 4 : Vue d'ensemble
 - Mappage de table unique
 - Mappage joint (Table par sous-classe)
 - Table par mappage de classe concrete
 - Avantages et Inconvénients
- Leçon 5 : Collections d'Eléments (JPA 2)
 - Vue d'ensemble, Collections d'objets valeur, @ElementCollection, @CollectionTable
 - Utilisation des collections d'éléments
 - Collections d'Embeddables

Module 5 : L'API Critères (JPA 2)

- Leçon 1 : Vue d'ensemble de l'API Criteria
- Leçon 2 : Expressions de cheminement, création de requêtes (CriteriaBuilder, CriteriaQuery, Subquery, Predicate, Expression, Order, Selection, Join)
- Leçon 3 : Exécuter des requêtes et accéder aux résultats

Module 6 : Capacités JPA supplémentaires

- Leçon 1 : Fichiers XML de Mapping
- Leçon 2 : Validation des Beans (JPA 2)
- Leçon 3 : Meilleures pratiques
- Leçon 4 : Clés primaires, requêtes nommées, chargement paresseux / avide, sémantique

transactionnelle, encapsulation, requêtes de rapport

Module 7 : Intégration

- Leçon 1 : Intégration des objets d'accès aux données (DAO) et Java SE (facultatif) :
 - Vue d'ensemble de la DAO
 - Classe JpaUtil pour la gestion d'EntityManager dans Java SE
 - Considérations de cycle de vie
- Leçon 2 : Intégration avec EJB (facultatif) :
 - Utilisation de JPA avec des beans de session
 - Gestion d'entités gérées par le Container (Injected)
 - JTA Transactions et considérations liées au cycle de vie
 - Contextes de persistance étendus
- Leçon 3 : Utilisation de JPA avec Java Web Apps :
 - Utilisation d'EntityManager dans les applications Web - recherche de portée
 - Lazy Loading - Open EntityManager dans View Pattern
- Leçon 4 : Intégration avec Spring (optionnel)
 - Injection d'EntityManager, EntityManagerFactory
 - LocalEntityManagerFactoryBean
 - JPA / Spring DAO

Documentation

- Support de cours numérique inclus

Profils des participants

- Développeurs débutants qui veulent développer des applications POJO qui interagissent avec la base de données relationnelle

Connaissances Préalables

- Connaissance intermédiaire de Java
- Une expérience dans la création d'applications Web de base en utilisant servlet et JSP
- Une certaine familiarité avec SQL, les bases de données relationnelles, et les interfaces Java Database Connectivity (JDBC)

Objectifs

- Expliquer comment persister leur classe POJO avec une base de données relationnelle en utilisant la technologie JPA de mapping objet-relationnel
- Mettre en œuvre et maintenir une persistance objet-relationnel dans leur application JPA, et tous les clients nécessaires côté client et côté serveur

Niveau

Intermédiaire

Prix de l'inscription en Présentiel (CHF)

2300

Prix de l'inscription en Virtuel (CHF)

2150

Durée (Nombre de Jours)

3