

Objectifs :

- consolider la connaissance des fonctionnalités de base d'Hibernate
- acquérir les bonnes pratiques de développement
- comprendre le cycle de vie des entités
- comprendre les initialisation paresseuses (lazy)
- savoir paramétrer les différents types de mapping
- intégrer Hibernate dans les architectures distribuées
- savoir paramétrer Hibernate

Durée : 4 jours

Public : développeurs, chefs de projets

Prérequis :

- connaissance du langage Java
- avoir déjà utilisé Hibernate

Démarche pédagogique :

- présentation des concepts, suivi d'exercices

Programme détaillé :

- Problématique de la persistance
 - modèles objets et relationnels
 - design patterns
 - solutions ORM
- Installation et configuration d'Hibernate
 - installation Hibernate
 - choix Hibernate ou JPA
 - configuration Hibernate ou JPA
 - la session Hibernate
- Principes de base du mapping Hibernate
 - association entre objets et base de données
 - mapping par fichier XML
 - mapping par annotation
- Manipulation des entités
 - processus de développement
 - les entités sont des POJO
 - scénarios de développement
 - top down, bottom up, middle out, meet-in-the-middle
 - opérations CRUD sur les objets
 - il ne s'agit plus de INSERT ou UPDATE mais bien du cycle de vie de l'objet
 - cycle de vie d'une entité

- transiente, persistante, détachée
- Relation entre entités
 - graphe d'objets
 - les 3 relations (1-1, 1-n, n-m), la navigabilité et leur correspondance relationnelle
 - sens de la relation
 - impacts des liens bidirectionnels
 - l'attribut mapping "inverse"
 - attribut cascade
- Relations d'héritage
 - stratégie de mapping pour la gestion de l'héritage
 - impact sur la stratégie de génération de la clé primaire
- Les composants
 - composant au lieu d'une entité
 - composant simple
 - collection de valeurs et de composants
 - lecture des objets
- Pattern Lazy Loading
 - attachement - détachement d'un objet de la session
 - conséquences et impacts sur les performances liés aux états
 - la méthode flush()
 - ré-attachement d'un objet à la session : merge(), update() et lock()
- utilisation de HQL et JPQL
- requêtes par critères
- utilisation des requêtes natives SQL et procédures stockées
- présentation de Hibernate Search
- Gestion des transactions
 - concepts de transaction
 - les transaction Hibernate, JDBC, JTA
 - gestion optimiste et pessimiste
 - verrous et concurrence
- Architectures techniques
 - design pattern : couche DAO
 - Hibernate dans une application standard
 - Hibernate dans une application Web
 - Hibernate dans une application Java EE
 - utilisation des entités JPA, ou pas
 - attention à la LazyException
 - design patterns : Transfer Object, Abstract Factory
- Interception des événements
 - la session Hibernate
 - le modèle événementiel
 - utiliser les listeners Hibernate
 - connaître le type exact des entités manipulées
- Caches
 - rôle des caches

- caches de premier et second niveau
- configuration du cache de second niveau
 - JBoss Tree Cache, EHCACHE
- Requêtes et Batch
 - utilisation des batch pour la suppression et la mise à jour
 - bien utiliser et comprendre les stratégies de fetch
 - comment charger les objets sans tenir compte du mapping
 - utilisation de l'attribut "default_batch_fetch-size"