

## Hibernate tu connais ... mais en fait tu connais pas

Speaker : Emmanuel Bernard (JBoss)

Format : Conférence

Date : 22 avril 2016

Pendant cette conférence, Emmanuel présente les nouveautés de la version 5 des projets Hibernate.



## Hibernate 5 ORM

- Bootstrap plus propre
  - Prend des classes d'un peu partout : classes entité, XML, Proxy, DataSource
  - Modularisation améliorée
  - Découpage en 3 blocs
    - StandardServiceRegistry : configuration globale
    - Metadata : hbm.xml, entité, stratégie de nommage
    - SessionFactory
- Compatible Java 8
  - Pas de lambdas
  - Mais support Data & Time : nouveau module hibernate-java8 à ajouter
  - Amélioration sur les Generics et AutoClosable
    - Plus besoin de `downcast session.get()`
- Augmentation par Bytecode
  - Détection des changements plus intelligents pour la comparaison entre les propriétés d'objet. Le bytecode enhancement permet de limiter les objets à comparer (dirty tracking)
  - Outillage maven : lazy initialisation, dirty tracking ...
  - Par défaut, toutes les propriétés sont chargées, sauf lorsqu'on appelle `@Lazy`. Lors du premier appel à une propriété `@Lazy`, toutes les propriétés sont chargées. Introduction d'une notion de groupe de propriété via `@LazyGroup`
  - Management des associations bidirectionnelles : la mise à jour d'un objet d'un côté est mis à jour de l'autre côté
- Cache de second niveau
  - Moins d'objets créés par Hibernate (ex : la clé est partagée entre Hibernate et le cache)

- Cache par référence : tous les les objets qui ne bougent jamais (objets de référence immuables) sont partagés par les différentes sessions Hibernates (plus de copie). A activer avec `user_reference_entries`
- Divers
  - UUID : super important pour le @NoSQL car peut être géré par le client.
  - Naming strategy : stratégie implicite traduit le nom d'une classe en un nom de table. On peut désormais personnaliser le nommage.
  - Documentation : gros effort apporté pour rafraichir la documentation. Ils sont repartis de zéro et ont ajouté plein d'exemples. Blog sur [in.relation.to](http://in.relation.to)

## Hibernate Search

- Indexation full-text déléguée à Lucene
- S'accroche à une source de donnée, le plus connue étant Hibernate ORM, pour mettre à jour l'index
- Cluster master/save
- Récemment :
  - Montée de version à Lucene 5
  - Améliorations sur les performances : utilisation d'une queue pour le flush. Se rapproche des perfs NRT
  - Backend Elasticsearch (se référer à la conférence *Elasticsearch et Hibernate sont sur un bateau*)
- Annotations : `@Indexed`, `@Spatial`, `@Field`, `@Facet`, `@NumericField`, `@Latitude`, `@Longitude`, `@IndexedEmbedded`

## Hibernate OGM

- Utilise les annotations JPA pour persister les entités dans des bases NoSQL
- Support : MongoDB, Infinispan, Neo4j, Redis, EhCache, Cassandra, CouchDB
- Pratique pour démarrer avec le NoSQL sans trop de changements pour le développeur expérimenté en JPA

## Hibernate Validator

- Implémentation de référence de Bean Validation
- Nouveau : pas de lambda, noms réels des paramètres, Date et Time, Optional, gestion des collections : `List<@MyNotNull @Email String> emails`
- Les contraintes peuvent être déclarées par programmation (DSL propre à Hibernate Validator) => permet d'appliquer dynamiquement les contraintes

## Hibernate Spatial

- Extension d'Hibernate ORM intégrant des fonctions spatiales natives aux bases de données.
- Utilisation d'un dialect spéciale
- La requête JQL change un peu

## Questions / Réponses

- Support d'Hazelcast prévu comme cache de 2<sup>nd</sup> niveau ? Se concentre sur JCache.
- Support d'Hbase dans Hibernate OGM ? oui mais problème de bande passante. Préfère se concentrer sur Cassandra.
- Pourquoi Elasticsearch et pas Solr ? L'objectif est d'avoir les 2. Il fallait commencer par un moteur.
- Intégration d'Hibernate 5 dans Spring Boot ? Sûr dans Spring Data. A vérifier dans Spring Boot.