
Préface

Traducteur(s): Vincent Ricard

Hibernate, comme tous les autres outils de mapping objet/relationnel, nécessite des méta-données qui régissent la transformation des données d'une représentation vers l'autre (et vice versa). Dans Hibernate 2.x, les méta-données de mapping sont la plupart du temps déclarées dans des fichiers XML. Une autre option est XDoclet, qui utilise les annotations du code source Javadoc et un préprocesseur au moment de la compilation. Le même genre d'annotation est maintenant disponible avec le JDK standard, quoique plus puissant et mieux pris en charge par les outils. IntelliJ IDEA et Eclipse, par exemple, prennent en charge la complétion automatique et la coloration syntaxique des annotations du JDK 5.0. Les annotations sont compilées en bytecode et lues au moment de l'exécution (dans le cas d'Hibernate, au démarrage) en utilisant la réflexion, donc pas besoin de fichiers XML externes.

La spécification EJB3 reconnaît l'intérêt et le succès du paradigme du mapping objet/relationnel transparent. La spécification EJB3 standardise les APIs de base et les méta-données requises par n'importe quel mécanisme de persistance objet/relationnel. *Hibernate EntityManager* implémente les interfaces de programmation et les règles de cycle de vie telles que définies par la spécification de persistance EJB3. Avec *Hibernate Annotations*, ce wrapper implémente une solution de persistance EJB3 complète (et autonome) au-dessus du noyau mature d'Hibernate. Vous pouvez utiliser soit les trois ensembles, soit les annotations sans le cycle de vie et les interfaces de programmations EJB3, ou même Hibernate tout seul, selon les besoins techniques et fonctionnels de votre projet. Vous pouvez à tout moment recourir aux APIs natives d'Hibernate ou même, si besoin est, à celles de JDBC et au SQL.

Cette version est basée sur la dernière version de la spécification EJB 3.0 / JPA (alias JSP-220) et prend en charge toutes les fonctionnalités de la spécification (dont certaines optionnelles). La plupart des fonctionnalités d'Hibernate et des extensions sont aussi disponibles à travers des annotations spécifiques à Hibernate. Bien que la couverture d'Hibernate en termes de fonctionnalités soit maintenant très grande, certaines sont encore manquantes. Le but ultime est de tout couvrir. Voir la section JIRA "road map" pour plus d'informations.

Si vous utilisiez une version précédente d'Hibernate Annotations, veuillez regarder <http://www.hibernate.org/371.html> pour un guide de migration.

Chapitre 1. Configurer un projet avec des annotations

1.1. Prérequis

- Téléchargez et installez la distribution Hibernate Annotations à partir du site web d'Hibernate.
- *Cette version requiert Hibernate 3.2.0.GA ou supérieur. N'utilisez pas cette version d'Hibernate Annotations avec une version plus ancienne d'Hibernate 3.x !*
- Cette version est connue pour fonctionner avec le noyau 3.2.0.CR5 et 3.2.0.GA d'Hibernate.
- Assurez-vous que vous avez le JDK 5.0 d'installé. Vous pouvez bien sûr continuer à utiliser XDoclet et avoir certains des avantages des méta-données basées sur les annotations avec des versions plus anciennes du JDK. Notez que ce document décrit seulement les annotations du JDK 5.0 et que vous devez vous référer à la documentation de XDoclet pour plus d'informations.

1.2. Configuration

Tout d'abord, paramétrez votre classpath (après avoir créer un nouveau projet dans votre IDE favori) :

- Copiez toutes les bibliothèques du noyau Hibernate3 et toutes les bibliothèques tierces requises (voir lib/README.txt dans Hibernate).
- Copiez aussi `hibernate-annotations.jar` et `lib/ejb3-persistence.jar` de la distribution Hibernate Annotations dans votre classpath.
- Pour utiliser Chapitre 5, *Intégration de Lucene avec Hibernate*, ajouter le fichier jar de lucene.

Nous recommandons aussi un petit wrapper pour démarrer Hibernate dans un bloc statique d'initialisation, connu en tant que `HibernateUtil`. Vous pourriez avoir vu cette classe sous diverses formes dans d'autres parties de la documentation Hibernate. Pour prendre en charge Annotation vous devez modifier cette classe d'aide de la manière suivante :

```
package hello;

import org.hibernate.*;
import org.hibernate.cfg.*;
import test.*;
import test.animals.Dog;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {
            sessionFactory = new AnnotationConfiguration().buildSessionFactory();
        } catch (Throwable ex) {
            // Log exception!
            throw new ExceptionInInitializerError(ex);
        }
    }
}
```