

Hibernate

Première Partie (mise en place)

En premier lieu, il s'agit d'enrichir notre application de Phone Contacts. Reprenez la page **addContact.jsp**. Celle-ci doit maintenant proposer un formulaire qui contient, en plus des champs que vous avez déjà implémentés en TP Servlet/JSP, les informations suivantes : L'adresse du contact (voir classe **Address** dans le diagramme de classe ci-dessous), ces différents numéros de tél (voir classe **PhoneNumber**, on supposera qu'il en a trois max i.e. mobile, maison, bureau et qui sont optionnels) et finalement les groupes auxquels il peut appartenir (Amis, collègues, famille, etc. voir la classe **ContactGoup**, on peut imaginer que certains groupes seront proposés par défaut au moment du rajout du contact et que si ce choix est insuffisant, donner la possibilité à l'utilisateur de spécifier un nouveau groupe), sachant qu'un contact peut faire partie de plusieurs groupes en même temps. Le formulaire doit également avoir un bouton « submit » et un bouton « reset ».

A cet effet, vous devez rajouter dans votre package « **domain** » toutes les classes du diagramme donné ci-dessous (sauf Entreprise). Pour chaque classe, générez les getters et setters de ses attributs et définissez un constructeur sans paramètres. Pour les classes DAO, on verra par la suite. Dans tous les cas c'est à vous de voir si vous aurez besoin d'en définir ou pas. Pas la peine de le demander à votre prof.

Ensuite configurer votre environnement en suivant les étapes décrites dans la section Requirements.

Une fois l'environnement prêt, votre premier contact avec Hibernate sera de :

- 1- Créer un fichier de mapping (**.hbm.xml**) par classe métier présente dans le diagramme UML. **IMPORTANT : le fichier de mapping doit se trouver dans le même répertoire que la classe métier.** Vous devez prendre en compte non seulement les propriétés de la classe mais également ses associations ainsi que leurs navigabilités et leurs multiplicités. C'est ici pour vous l'occasion d'apprendre à écrire un fichier de mapping et surtout de maîtriser les one-to-one, one-to-many, etc.

ATTENTION :

- **Faire étape par étape, i.e. d'abord créer le fichier de mapping XML pour une seule classe sans association puis le tester. Ensuite rajouter les associations une à une et tester à chaque fois sinon ça devient très dur d'identifier les sources des conflits en cas d'erreurs.**
 - **Ne pas utiliser d'utilitaires pour cet exercice. Le jour de l'examen, on ressentira la différence entre ceux qui le font manuellement et ceux qui utilisent un outil!!**
- 2- Créer le fichier de configuration Hibernate (**.cfg.xml**). **IMPORTANT : ce fichier doit se trouver la racine de votre répertoire source du projet (src).** C'est dans ce fichier que vous allez donner les informations sur la data source (BD), les classes à mapper, le mode d'utilisation de la base (création, mise à jour, etc.).
 - 3- Une fois les étapes 1 et 2 réalisées, faites un test pour voir si les tables sont automatiquement générées dans la base de données. Ne passez pas à la deuxième partie tant que cette étape n'est pas réalisée. **Pour tester la création des tables par hibernate il faut :**
 - Que la base existe au niveau de votre SGBD

- Qu'elle n'est aucune table. C'est hibernate qui va les créer pour vous
- Rajouter ces lignes de code dans un des DAO (après avoir commenter tout le code JDBC). La classe *HibernateUtil* se trouve dans le ZIP du TP dans Help_Configuration:

```
Session session = HibernateUtil.getSessionFactory().getCurrentSession();
```

A ce niveau, vous aurez appris à écrire des fichiers de configuration, des fichiers de mapping classes=> tables et Java Beans.

Deuxième Partie (application)

Dans cette partie, l'idée est que vous commenciez à utiliser concrètement Hibernate dans votre code.

- 1- Commencez par enlever ou commenter votre code JDBC (qui se trouve normalement dans les classes DAOxxx) et à le remplacer par du code Hibernate (**voir exemple du cours**). Vous devez le faire pour toutes les parties de votre application (ajouter contact, supprimer ou rechercher un contact à travers son identifiant, etc.) Ici, si vous avez besoin de définir de nouveaux DAO faites-le ! Encore une fois, on compte sur vous pour enrichir votre application de la manière la plus originale qu'il soit (fonctionnalités avancée, nouvelles fonctionnalités, etc.).
- 2- Dans cette étape, il vous ait demandé de pratiquer l'héritage. Pour cela rajouter la classe **Entreprise** au package « **domain** » et tout ce que cela implique (fichier de mapping, génération des getters, setters, constructeurs, etc.). Ensuite optez pour une des stratégies d'héritage fournies par Hibernate et implémentez-là dans le contexte de votre projet. Vérifiez le résultat au niveau de la base de données.
- 3- Modifiez maintenant votre formulaire de création d'un nouveau contact pour y inclure l'information si c'est un contact ou bien une entreprise. Pour les entreprise il y'aura une information de plus qui est le numéro de SIRET.
- 4- Au niveau de votre code de traitement arrangez-vous pour avoir des instances de contact et des instances d'entreprise dans la base de données.

Troisième Partie (HQL)

Dans cette dernière partie, il s'agit de s'exercer à utiliser le langage HQL pour interroger la base de données. Pour cela, vous devez avoir quelque part dans votre code des requêtes HQL (pas de requêtes trop simples SVP).

- Des requêtes de type from.....jointure, etc.
- Des requêtes avec Critères
- Des requêtes avec l'Exemple.

Pour Hibernate

Télécharger la dernière version stable d'Hibernate Tools for Eclipse sur:

http://sourceforge.net/projects/jboss/files/JBossTools/jbosstools4.2.x/jbosstools-4.2.0.CR1_2014-09-15_02-47-21-B157-updatesite-hibernatetools.zip/download

ou bien à travers l'update site

<http://tools.jboss.org/downloads/jbosstools/luna/4.2.0.CR1.html>

Vous aurez le choix entre télécharger un zip ou bien de passer par le « update site » d'Eclipse (conseillé). Attention cela dépend de votre version d'Eclipse, vérifiez la compatibilité.

Pour l'installer, juste dézipper et copie/coller le contenu du package « plugin » à destination du package du même nom d'Eclipse. Même chose pour le répertoire « features »

Attention : si vous passez via update site, ne téléchargez pas tout JBoss Tool, choisissez seulement Hibernate Tool

Récupérez les libs Hibernate, elles se trouvent dans un zip nommé **Libs_Hibernate.zip** dans le zip de votre atelier/TP.

ATTENTION : Si vous êtes dans le contexte d'un projet web i.e. Servlet/JSP (dans votre cas oui), vous devez mettre tous les jars qui se trouvent dans Libs_Hibernate.zip dans le package **webcontent/WEB-INF/lib** de votre projet Web.

Pour l'arborescence du projet dans Eclipse et pour l'emplacement des fichiers, .jar, etc. voir la figure ci-dessous :



