

The image features a decorative graphic on the right side consisting of three blue circles of varying sizes, each with a lighter blue ring around its center. These circles are connected by thin blue lines that form a triangular shape. The largest circle is at the top right, a smaller one is in the middle, and the largest of all is at the bottom right.

# **Tutoriel d'installation de Hibernate avec Eclipse**

## Table des matières

1.	Création de la base de données .....	4
2.	Installation de Hibernate Synchronizer .....	5
3.	Utilisation de Hibernate .....	6
3.1	Création et préparation du projet .....	6
3.2	Création des éléments de Hibernate.....	9
3.2.1	Fichier de configuration .....	9
3.2.2	Fichier de mapping .....	11
4.	Test des classes .....	13
4.1	Classe HibernateSessionFactory.....	13
4.2	La class de Test .....	14

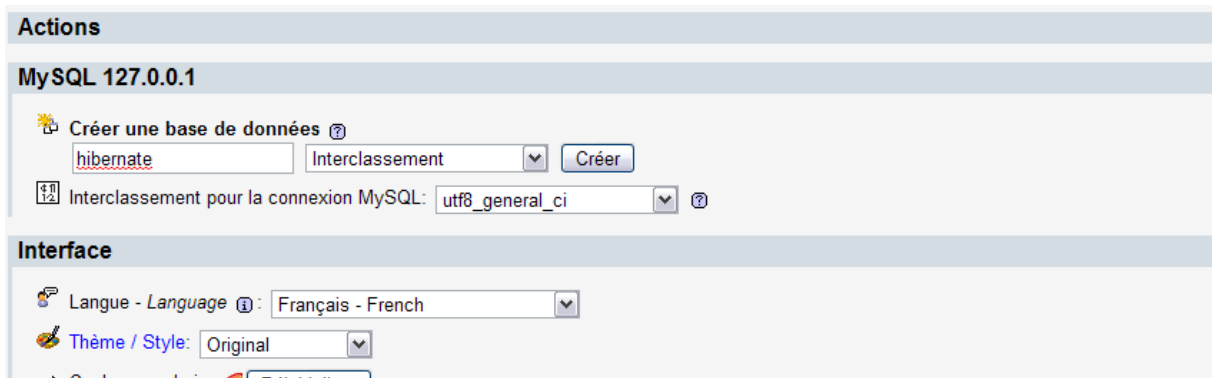


## 1. Création de la base de données

La source de données choisie pour cet article est MySQL. Il est évidemment possible d'en choisir une autre comme Oracle, SQL Server ou même un simple fichier XML pourvu que vous disposiez des drivers JDBC adaptés.

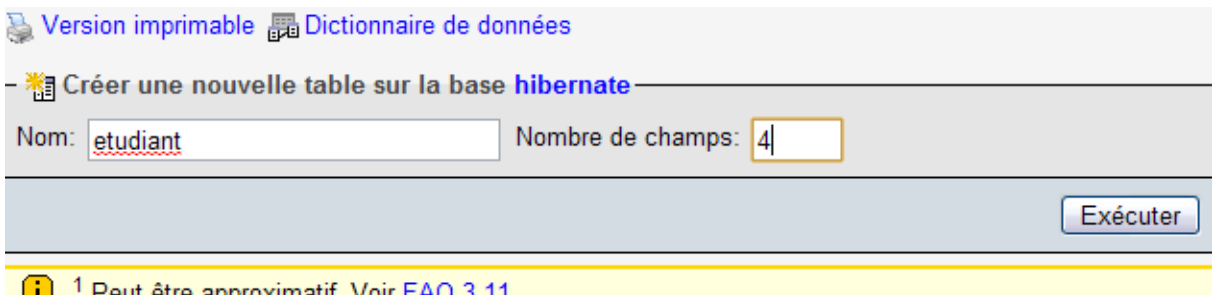
Nous supposons que MySQL est installé et que vous disposez des droits nécessaires pour créer une nouvelle base de données.

*Créez une base de données et donnez-lui le nom « hibernate »*



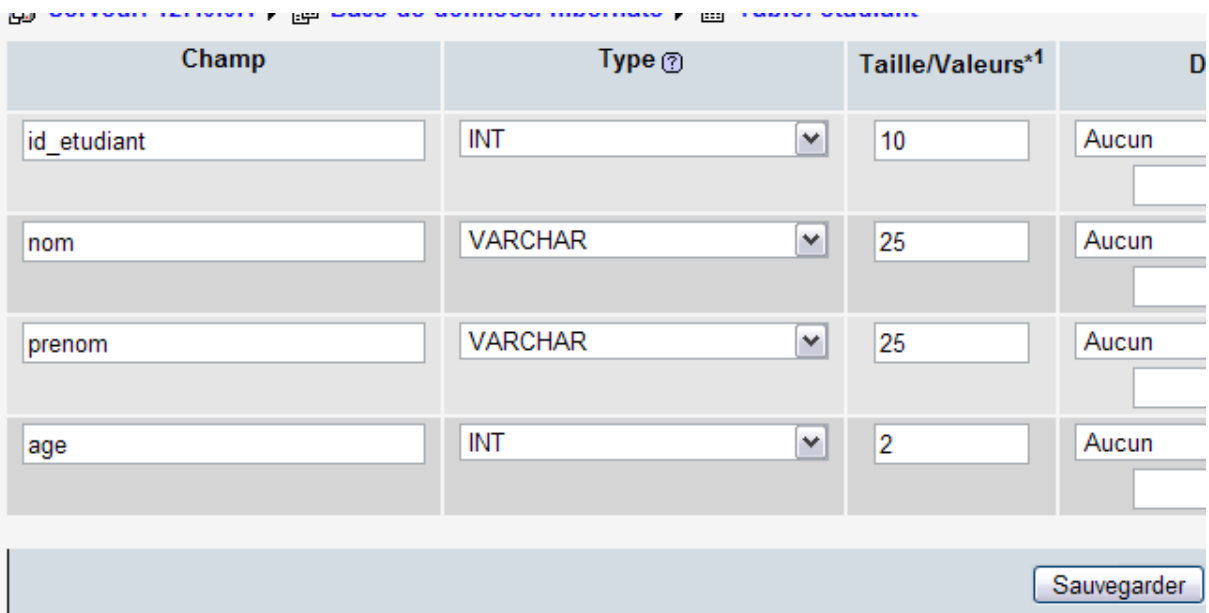
The screenshot shows the MySQL database creation interface. Under the 'Actions' section, there is a 'MySQL 127.0.0.1' header. Below it, there is a 'Créer une base de données' button. A text input field contains 'hibernate', and a dropdown menu is set to 'Interclassement'. A 'Créer' button is next to it. Below that, another dropdown menu is set to 'utf8\_general\_ci' with a label 'Interclassement pour la connexion MySQL:'. Under the 'Interface' section, there is a 'Langue - Language' dropdown set to 'Français - French' and a 'Thème / Style' dropdown set to 'Original'.

*Dans la base de données hibernate créez la table « etudiant »*



The screenshot shows the MySQL table creation interface. At the top, there are links for 'Version imprimable' and 'Dictionnaire de données'. Below that, there is a 'Créer une nouvelle table sur la base hibernate' header. A 'Nom:' label is followed by a text input field containing 'etudiant'. To the right, 'Nombre de champs:' is followed by a text input field containing '4'. An 'Exécuter' button is located at the bottom right. A yellow warning bar at the bottom contains an information icon and the text '1 Peut être approximatif Voir FAQ 3.11'.

*Ajouter les attributs suivants : Id\_etudiant, nom, prenom, age.*



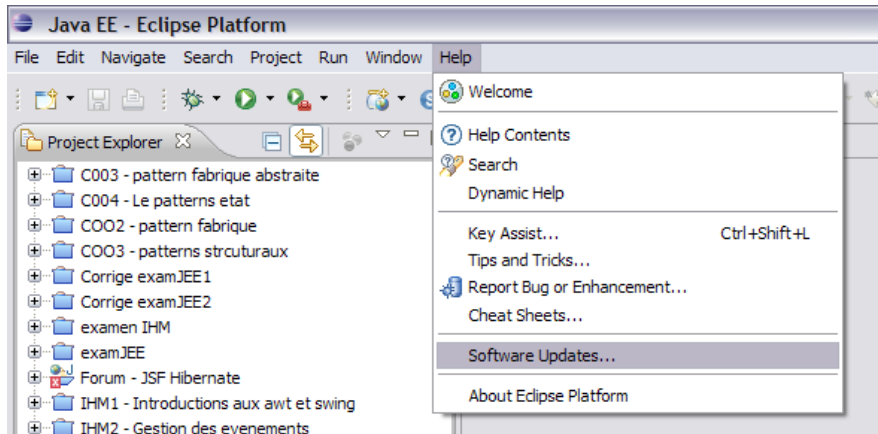
The screenshot shows the MySQL table attribute configuration interface. It features a table with four columns: 'Champ', 'Type', 'Taille/Valeurs\*1', and 'D'. The table contains four rows of data:

Champ	Type	Taille/Valeurs*1	D
id_etudiant	INT	10	Aucun
nom	VARCHAR	25	Aucun
prenom	VARCHAR	25	Aucun
age	INT	2	Aucun

At the bottom right of the interface, there is a 'Sauvegarder' button.

## 2. Installation de Hibernate Synchronizer

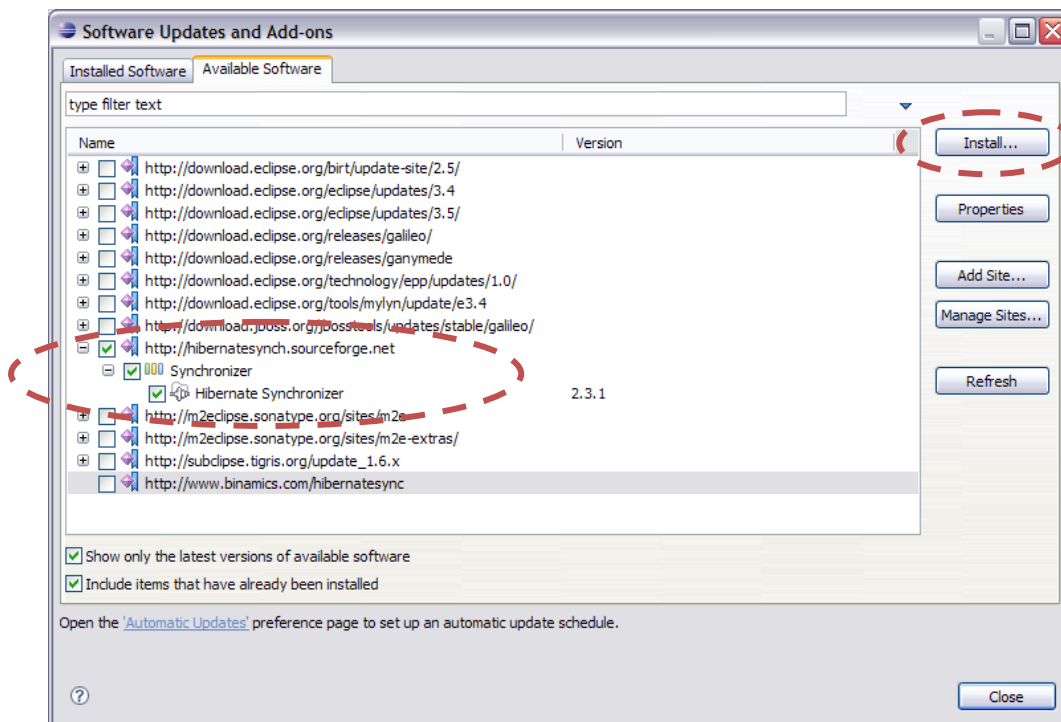
Sous Eclipse, cliquez sur : « Help > Software Updates »



Sous l'onglet Available Software, cliquez sur « Add site »

Entrez l'url suivant : <http://hibernatesynch.sourceforge.net> puis appuyez sur OK

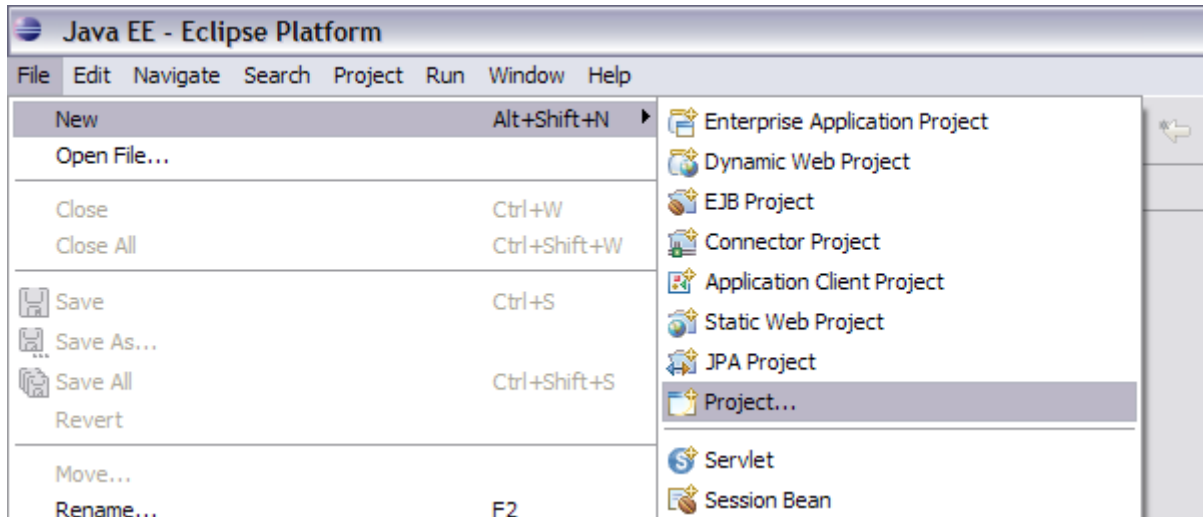
Sélectionnez Hibernate Synchroniser puis install



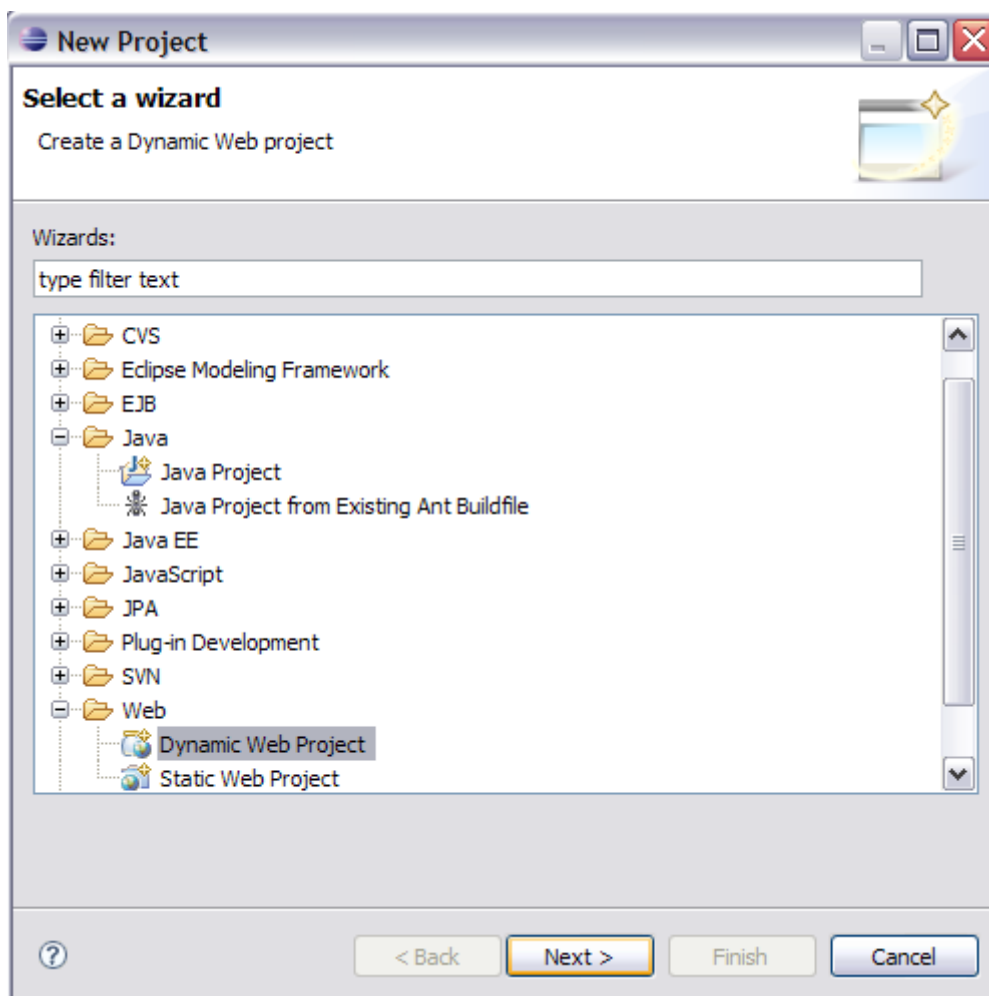
## 3. Utilisation de Hibernate

### 3.1 Création et préparation du projet

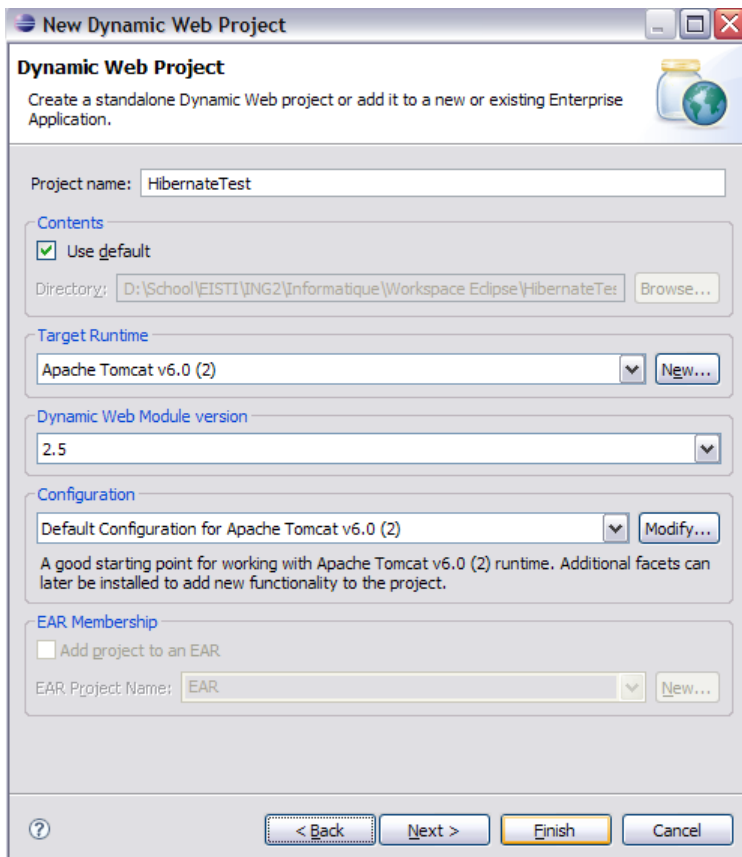
Sous Eclipse faites « File>New>Project »,



Choisissez Dynamic Web Project et cliquez sur next>

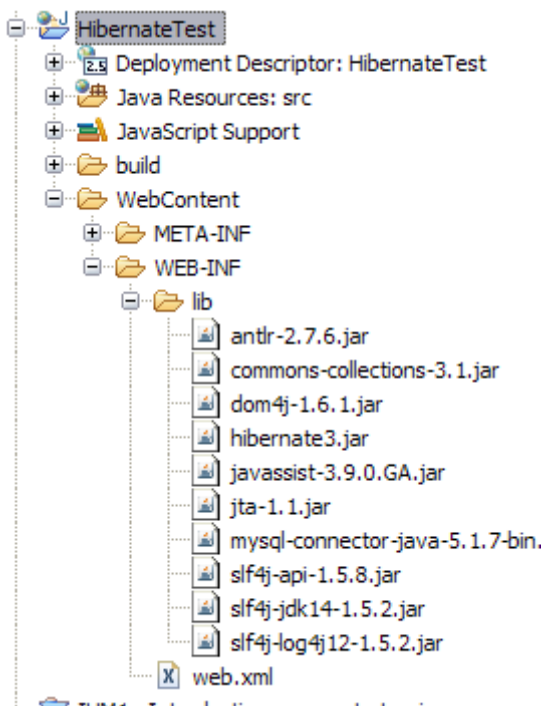


Donnez-lui le nom *HibernateTest*, choisissez le serveur puis cliquez sur *finish*



Téléchargez l'archive hibernate et décompressez le contenu dans le répertoire  
« .../workspace/HibernateTest/WebContent/WEB-INF/lib »

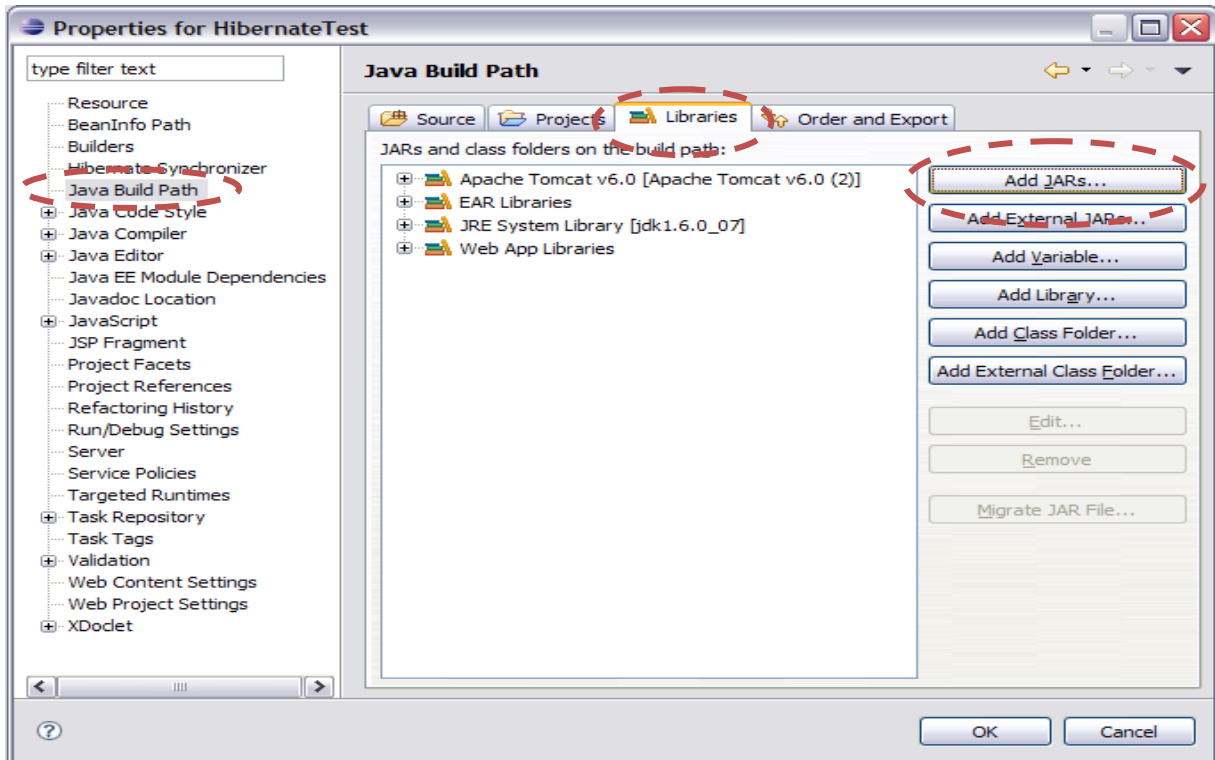
Le projet doit ressembler à ça :



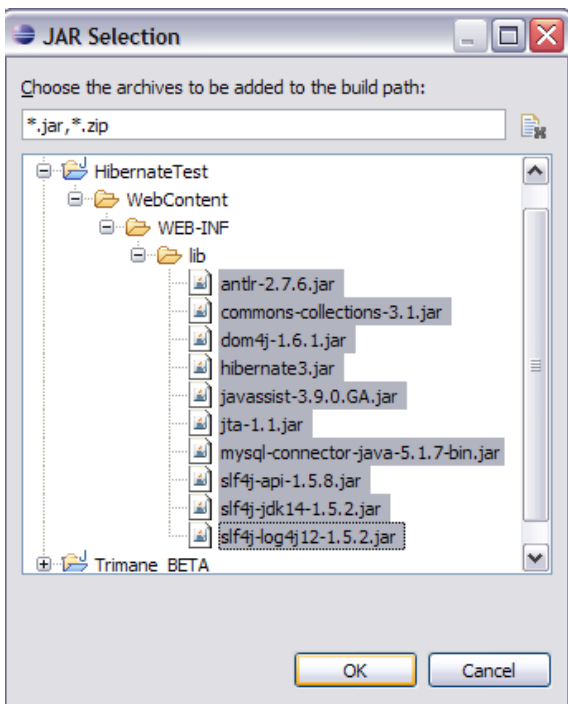
Nous allons ajouter au classpath du projet les librairies.

Faites sur le projet "HibernateTest" : "Clic Droit > Properties" puis dans la fenêtre qui s'ouvre Cliquez sur "Java Build Path" puis sur l'onglet "Librairies".

Cliquez sur "Add jars".



Une fenêtre apparaît. Déroulez l'arborescence : HibernateTest>WebContent>WEB-INF>lib " et sélectionnez tous les jars présents.



Puis cliquez sur "OK". Les jars sont maintenant ajoutés au projet

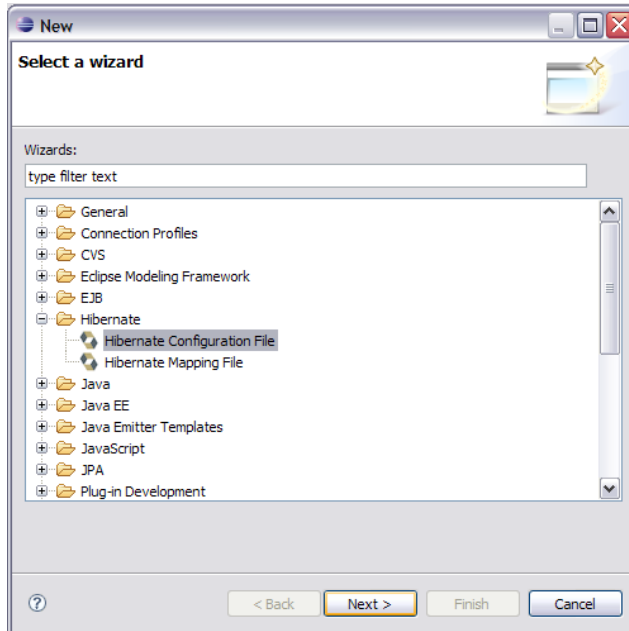


## 3.2 Création des éléments de Hibernate

### 3.2.1 Fichier de configuration

Ce fichier nous permet d'indiquer les paramètres de connexion à la BDD à Hibernate

Faites un clic droit sur le dossier src du projet « New>Other » et dans le fenêtre qui s'ouvre sélectionnez Hibernate>Hibernate Configuration file puis cliquez sur Next



Indiquez les informations suivantes puis cliquez sur « finish »:

**Container** : /HibernateTest/src

**File name** : hibernate.cfg.xml

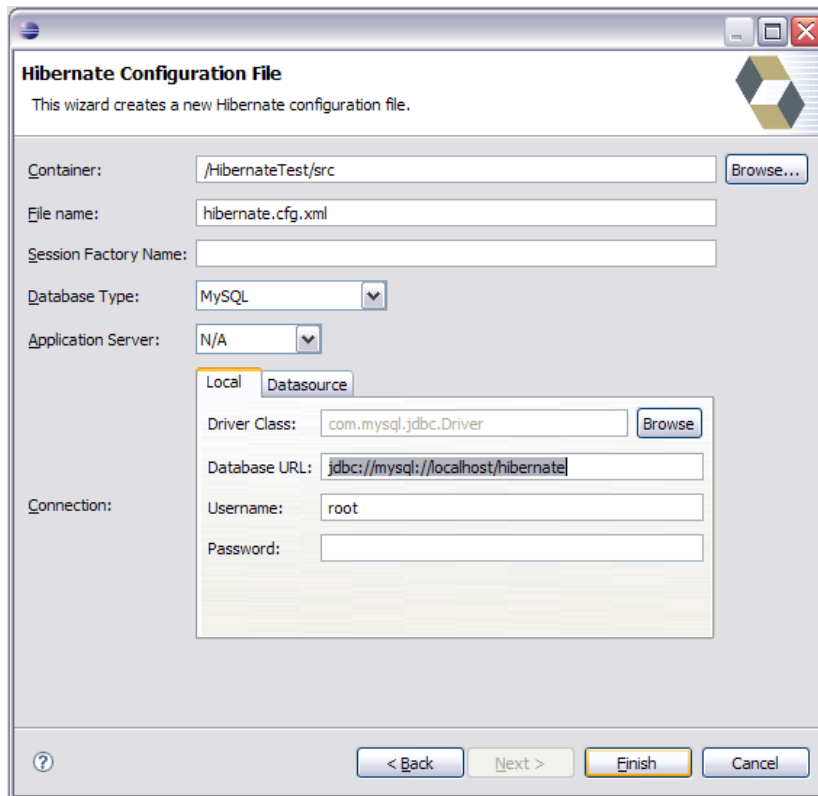
**Database Type** : MySQL

**Driver Class** : com.mysql.jdbc.Driver // Driver à utiliser

**Database URL** : jdbc://mysql://localhost/hibernate // URL de la BDD

**Username** : root

**Password** : laisser la case vide



Le fichier hibernate.cfg.xml est généré. Ajoutez la ligne suivante avant la fermeture de la balise `</sessions-factory>` : `<mapping resource="Etudiant.hbm.xml"/>`

Le fichier doit ressembler à ça :

### **hibernate.cfg.xml**

```

<hibernate-configuration>
  <session-factory>
    <!-- local connection properties -->
    <property
name="hibernate.connection.url">jdbc:mysql://localhost/forum_jsf_hibernate<
/property>
    <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password"></property>
    <!-- property name="hibernate.connection.pool_size"></property
-->

    <!-- dialect for MySQL -->
    <property
name="dialect">org.hibernate.dialect.MySQLDialect</property>

    <property name="hibernate.show_sql">>false</property>
    <property
name="hibernate.transaction.factory_class">org.hibernate.transaction.JDBCTr
ansactionFactory</property>

    <mapping resource="Etudiant.hbm.xml"/>

  </session-factory>
</hibernate-configuration>

```

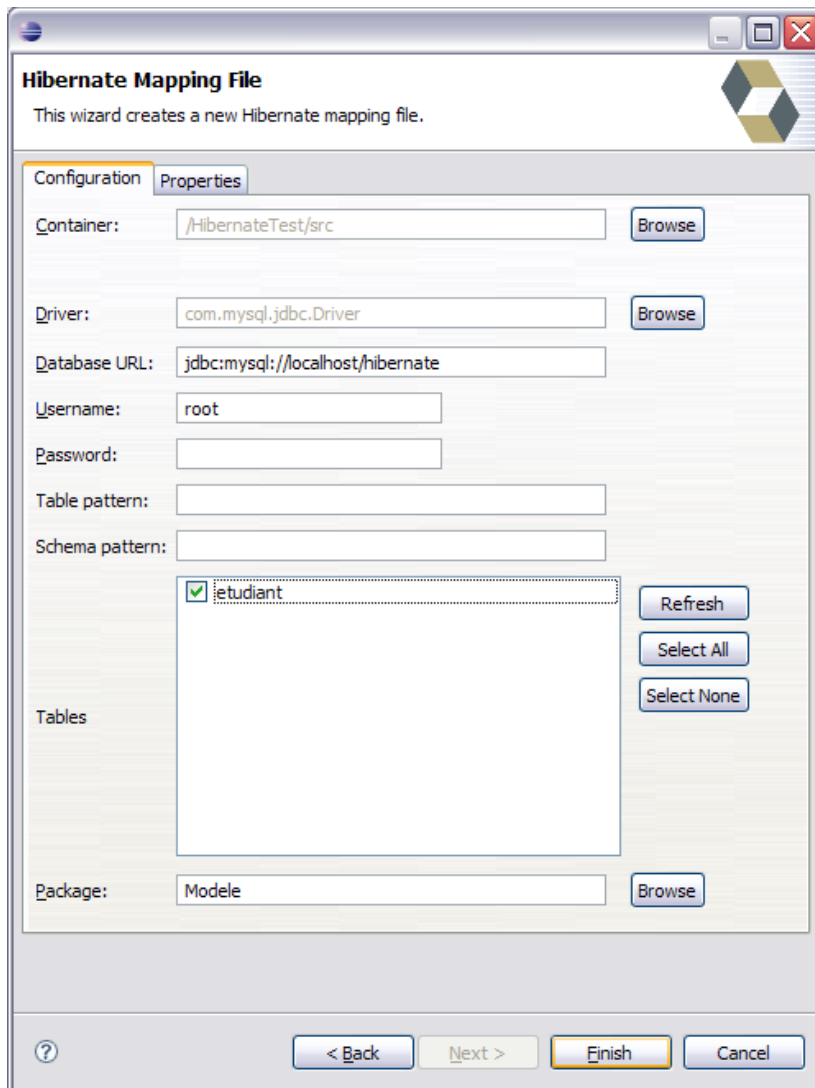
### 3.2.2 Fichier de mapping

Faites un clic droit sur src : New > Package" et donnez comme nom : Modele.

Faites un clic droit sur src : New > Other" et choisissez "Hibernate > Hibernate Mapping File".

Indiquez les informations requises puis cliquez sur refresh pour voir la liste des tables.

Si vous avez l'écran suivant c'est que la Base de données est correctement liée notre application. Sélectionner alors la table etudiant et cliquez sur finish.



Contrairement à ce que l'on peut croire le fichier Etudiant.hbm n'est pas conforme à la dtd. Je vous propose donc de remplacer son contenu par celui-ci :

## Etudiant.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd" >

<hibernate-mapping package="Modele">
    <class name="Etudiant" table="etudiant">
        <meta attribute="sync-DAO">false</meta>

        <id name="idUtilisateur" type="integer">
            <column name="id_etudiant" sql-type="int(4)" />
            <generator class="increment" />
        </id>

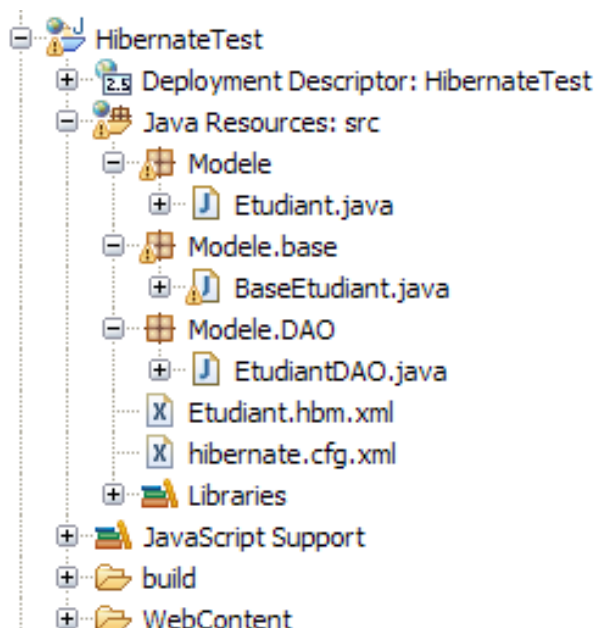
        <property name="nom" type="string">
            <column name="nom" sql-type="varchar(25)" not-null="true"
/>
        </property>

        <property name="prenom" type="string">
            <column name="prenom" sql-type="varchar(25)" not-
null="true" />
        </property>
        <property name="age" type="integer">
            <column name="age" sql-type="int(2)" not-null="true" />
        </property>
    </class>

</hibernate-mapping>
```

*Le fichier précédent n'est pas totalement faux, mais celui-ci ne présente pas d'erreurs. La balise <id></id> définit la clé primaire et <generator class="increment" /> montre que le champ est incrémenté automatiquement. Les balises <property></property> définissent les autres attributs.*

*Le projet doit ressembler à ça :*



Les fichiers sont générés automatiquement. Si ce n'est pas le cas, faites un clic droit sur *Etudiant.hbm* puis:  
Hibernate Synchroniser > Synchronize File

Maintenant que des classes ont été générées grâce à Hibernate et Hibernate Synchronizer, nous pouvons les tester.

## 4. Test des classes

### 4.1 Classe HibernateSessionFactory

Une classe Hibernate appelée *SessionFactory* permet à partir du fichier de configuration (*hibernate.cfg.xml*) d'être associé à la source de données. Elle fournit des objets *Session* pour manipuler les données. Nous allons, pour cela, écrire une classe trouvée dans l'aide en ligne d'Hibernate.org.

Dans votre projet *HibernateTest*, faites sur *Modele.DAO*: Clic droit > New > class. Donnez-lui comme nom *HibernateSessionFactory*

Copiez dans ce fichier vide, le contenu suivant :

#### HibernateSessionFactory.java

```
package Modele.DAO;
```

```
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;
```

```
public class HibernateSessionFactory {  
    private static final SessionFactory sessionFactory;  
    static  
    {  
        try  
        {  
            // Crée l'objet SessionFactory à partir de hibernate.cfg.xml  
            sessionFactory = new  
Configuration().configure().buildSessionFactory();  
        }  
        catch (Throwable ex)  
        {  
            System.err.println("Initial SessionFactory creation failed." + ex);  
            throw new ExceptionInInitializerError(ex);  
        }  
    }  
    public static SessionFactory getSessionFactory()  
    {  
        return sessionFactory;  
    }  
}
```

## 4.2 La class de Test

*Nous allons maintenant créer une classe pour les tests.*

*Nous allons d'abord créer 2 étudiants, l'enregistrer dans la base de données puis les afficher*

### EtudiantDAO.java

```
package Modele.DAO;

import java.util.*;
import org.hibernate.Session;
import org.hibernate.Transaction;
import Modele.Etudiant;
import Modele.DAO.HibernateSessionFactory;

public class EtudiantDAO {

    public static void main(String [] args){

        Session session = HibernateSessionFactory.currentSession();

        Transaction tx = session.beginTransaction();

        //On insère 2 etudiant dans la BDD
        Etudiant eleve = new Etudiant();
        eleve.setNom("Durand");
        eleve.setPrenom("Jacques");
        eleve.setAge(new Integer(20));
        session.save(eleve);

        eleve = new Etudiant();
        eleve.setNom("Petit");
        eleve.setPrenom("Lucie");
        eleve.setAge(new Integer(19));
        session.save(eleve);

        tx.commit();

        List list = session.createQuery("from Etudiant").list();
        Iterator it = list.iterator();
        while(it.hasNext())
        {
            Etudiant e = (Etudiant)it.next();
            System.out.println(e.getPrenom()+" "+ e.getNom()+": "+
e.getAge()+" ans");
        }

        HibernateSessionFactory.closeSession();

    }

}
```