



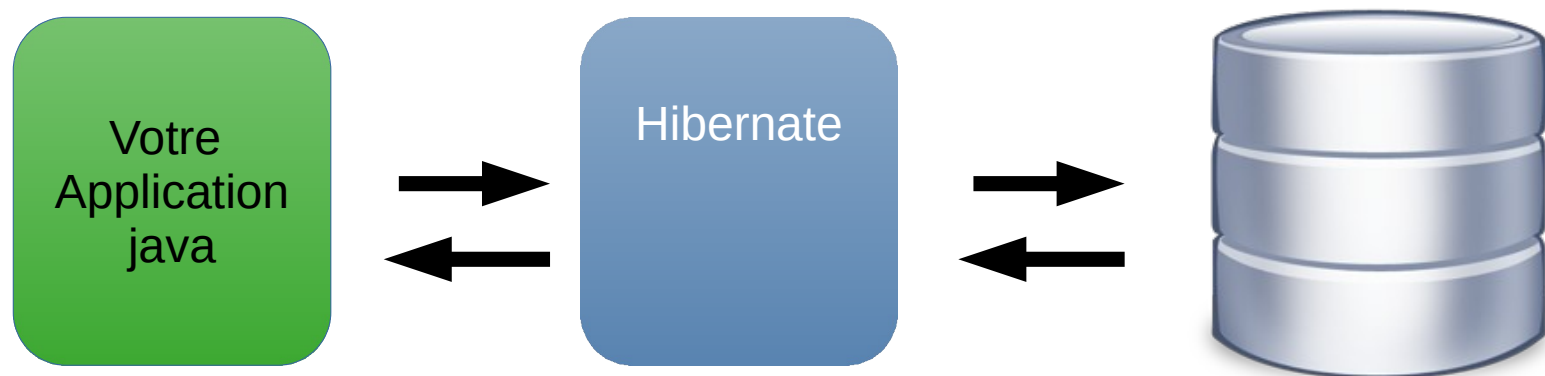
Hibernate

Introduction

Introduction

- Définition : ORM - Object relation mapping

Il s'agit d'un outil qui a pour but de sauvegarder des objets dans une base de données, de les convertir en SQL, ou json, xml



Pourquoi faire ?

- Déléguer le codage en sql
- Minimiser le code jdbc à écrire
- Faciliter par le mapping Objet relationnel la lecture et l'écriture en base de données.

Principe

Classe Java

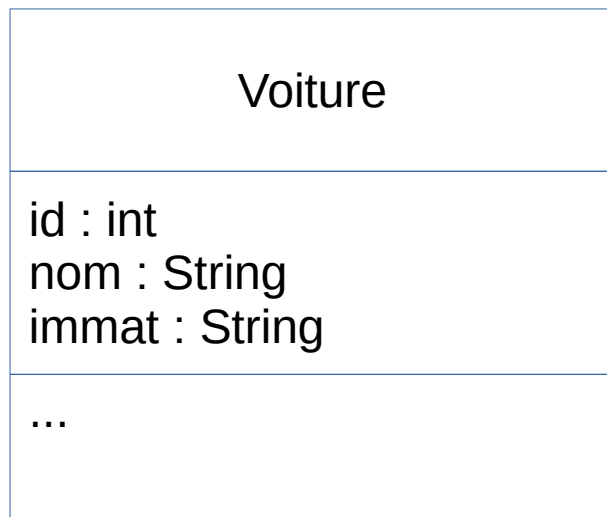
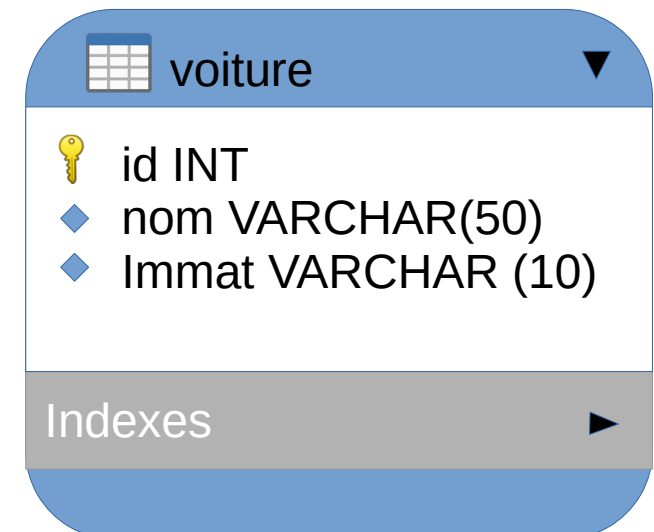


Table Base de données



Concrètement on fera ...

```
Voiture voiture= new Voiture();
```

```
//voiture.setId(1L);  
voiture.setModel("clio");  
voiture.setImmat("AK-725-66");  
// persister un objet voiture  
session.save(voiture);
```

L'objet session est
un objet d'Hibernate

=> SQL INSERT into ...

```
// persister un objet voiture  
Long unId= session.save(voiture);
```

```
// lire un objet spécifié par sa classe d'appartenance et son id  
Voiture voiture = session.get(Voiture.class, unId);
```

List,... update, delete avec HQL

```
// persister un objet voiture
Long unId= session.save(voiture);

// lire un objet spécifié par sa classe d'appartenance et son id
Voiture voiture = session.get(Voiture.class, unId);
```

Hibernate query Language
(HQL)

```
// lire une liste d'objets de type unique
Query query = session.createQuery("from Voiture");
List<Voiture> voitures= query.list();

session.createQuery("update Voiture set immat='123-AB-45'")
.executeUpdate();

session.createQuery("delete from Voiture where id=2" )
.executeUpdate();
```

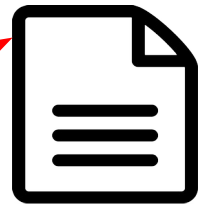
Créer une session hibernate

- SessionFactory un objet sessionFactory attend votre configuration et les classes à mapper
- Session ne va contenir que du code générique d'hibernate , aucune intervention requise

```
// create session factory
SessionFactory factory = new Configuration()
    .configure("hibernate.cfg.xml")
    .addAnnotatedClass(Voiture.class)
    .buildSessionFactory();

// create session
Session session = factory.getCurrentSession();
```

src/hibernate.cfg.xml



src/hibernate-cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

    <session-factory>

        <!-- JDBC Database connection settings -->
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/bdd_hibernate_demo?
useSSL=false&serverTimezone=UTC
        </property>
        <property name="connection.username">padawan</property>
        <property name="connection.password">padawan</property>

        <!-- JDBC connection pool settings ... using built-in test pool -->
        <property name="connection.pool_size">1</property>

        <!-- Select our SQL dialect -->
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>

        <!-- Echo the SQL to stdout -->
        <property name="show_sql">true</property>

        <!-- Set the current session context -->
        <property name="current_session_context_class">thread</property>

    </session-factory>

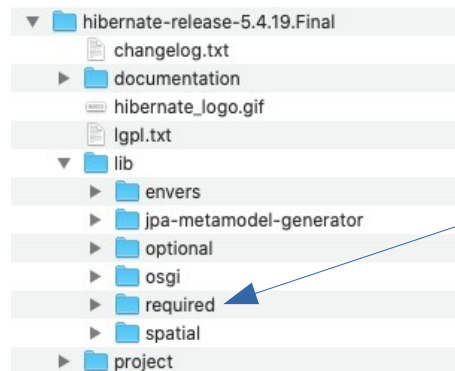
</hibernate-configuration>
```

Hibernate jars

Download hibernate jars ici :

<http://sourceforge.net/projects/hibernate/files/hibernate-orm/5.4.19.Final/hibernate-release-5.4.19.Final.zip/download>

Décompressez le zip, et récupérer les .jars contenus dans hibernate-release-5.4.19.Final/lib/required

A list of 20 jar files that are required for Hibernate 5.4.19.Final. The list is enclosed in a blue bracket on the left, which is connected by an arrow to the 'required' folder in the directory structure image.

- antlr-2.7.7.jar
- byte-buddy-1.10.10.jar
- classmate-1.5.1.jar
- dom4j-2.1.3.jar
- FastInfoset-1.2.15.jar
- hibernate-commons-annotations-5.1.0.Final.jar
- hibernate-core-5.4.19.Final.jar
- istack-commons-runtime-3.0.7.jar
- jandex-2.1.3.Final.jar
- javassist-3.24.0-GA.jar
- javax.activation-api-1.2.0.jar
- javax.persistence-api-2.2.jar
- jaxb-api-2.3.1.jar
- jaxb-runtime-2.3.1.jar
- jboss-logging-3.3.2.Final.jar
- jboss-transaction-spec-1.1.1.Final.jar
- stax-ex-1.8.jar
- txw2-2.3.1.jar

collez -les dans le répertoire des dépendances de votre projet (dans le classpath)

JDBC driver

- Download jdbc driver ici : <https://dev.mysql.com/downloads/connector/j/>

MySQL Community Downloads

Connector/J

Platform Independant

General Availability (GA) Releases Archives

Connector/J 8.0.21

Select Operating System:
Platform Independent

Looking for previous GA versions?

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.21.tar.gz)	8.0.21	3.8M	Download
MD5: 971c74093c2ae2a2b5a8bd65e9e7759 Signature			
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.21.zip)	8.0.21	4.5M	Download
MD5: 5b6ded4a1dd6769274822331d5cafe50 Signature			

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Download Zip

- Dézippez, ouvrez le dossier
- Récupérez directement le mysql-connector-java-8/.0.21.jar
- Ajoutez le au classpath de votre projet



Le projet est complet pour démarrer, mais il manque quelque chose...

- Il nous faut encore une Base de données.
- On s'en occupe dans le prochain chapitre de ce cours.