

TP3 (Hibernate)

On désire développer une application Web de gestion d'un hôtel. Cette application gère deux classes de base : Hôtel et Chambre.

- Un hôtel est caractérisé par son id, un titre et un lieu.
- Une chambre est caractérisée par un id, un titre, un nombre de lits et qui appartient à un hôtel.

L'application devrait permettre les opérations suivantes :

- Consulter un hôtel (par son id) ;
- Consulter tous les hôtels ;
- Consulter les chambres d'un hôtel donné ;
- Ajouter une chambre ;

L'application Web se compose de trois couches :

- La couche **model** : définie par les classes *Hotel*, *Chambre*.
- La couche **dao** : définie par l'interface *IGestionDao* et son implémentation *GestionImp*.
- La couche **Controller** : représenté par quatre Servlets.
 - **Principale** : permet de récupérer la liste des hôtels et de rediriger vers la page principale *home.jsp*
 - **ListeChambres** : permet de récupérer la liste des chambres d'un hôtel sélectionné (dans la page *home.jsp*) et de rediriger vers la page *chambres.jsp*
 - **AvantAjouterChambre**: permet de récupérer la liste des hôtels et de rediriger vers la page *formAjout.jsp*
 - **AjouterChambre**: permet d'ajouter une chambre saisie (*récupérer via la page formAjout.jsp*) et de rediriger vers la servlet *Principale*.

Liste des hôtels

ID	Titre	Lieu	
1	ElMouradi Tunis	Tunis	liste des chambres
2	Chames	Gabès	liste des chambres
3	Hannibal Palace	Djerba	liste des chambres

Page [home.jsp](#)

Liste des Chambres

Hôtel Choisi : ElMouradi Tunis

[Ajouter une chambre](#)

ID	Titre	Nbre_Lits
1	CH207	3
2	CH208	2
3	CH209	2
4	CH202	4

Page [chambres.jsp](#)

Ajouter une Chambre

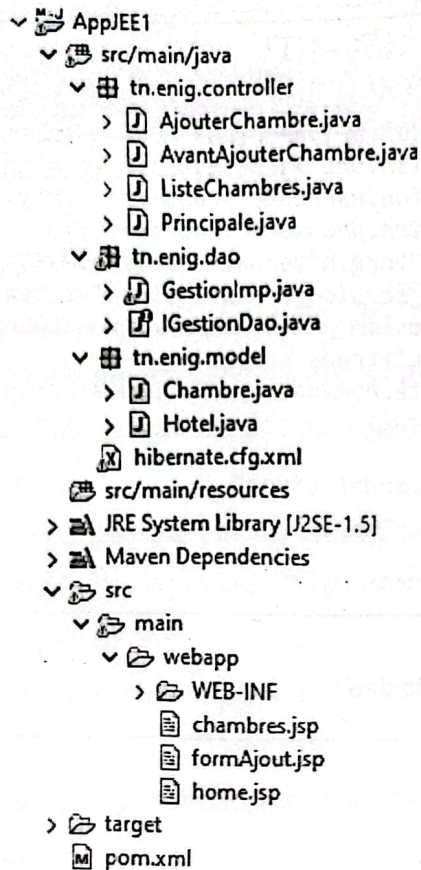
Titre

Nbre des lits

Hotel ▼

Page [formAjout.jsp](#)

Architecture du projet maven



Les dependences nécessaires pour le projet

```
<!-- Servlet Spec -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>javax.servlet.jsp-api</artifactId>
  <version>2.3.1</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>jstl</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
<!-- accès base de données -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.8</version>
</dependency>
<!-- pour hibernate -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.4.8.Final</version>
</dependency>
```

Exemple du fichier hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<hibernate-configuration>
  <session-factory>
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/bdhotel</property>
    <property name="connection.username">root</property>
    <property name="connection.password"></property>
    <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>
    <property name="current_session_context_class">thread</property>
    <property name="cache.provider_class">org.hibernate.cache.internal.NoCacheProvider</property>
    <property name="show_sql">true</property>
    <property name="hibernate.hbm2ddl.auto">update</property>

    <mapping class="tn.enig.model.Livre" />

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

Exemple d'integration du hibernate dans le dao

```
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.service.ServiceRegistry;

public class GestionImp implements IGestion {

  Session session;

  public GestionImp() {
    SessionFactory sessionFactory;
    ServiceRegistry reg1 = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
    Metadata metadata = new MetadataSources(reg1).getMetadataBuilder().build();
    sessionFactory = metadata.getSessionFactoryBuilder().build();
    this.session = sessionFactory.openSession();
  }
}
```