

# TP6 : JEE

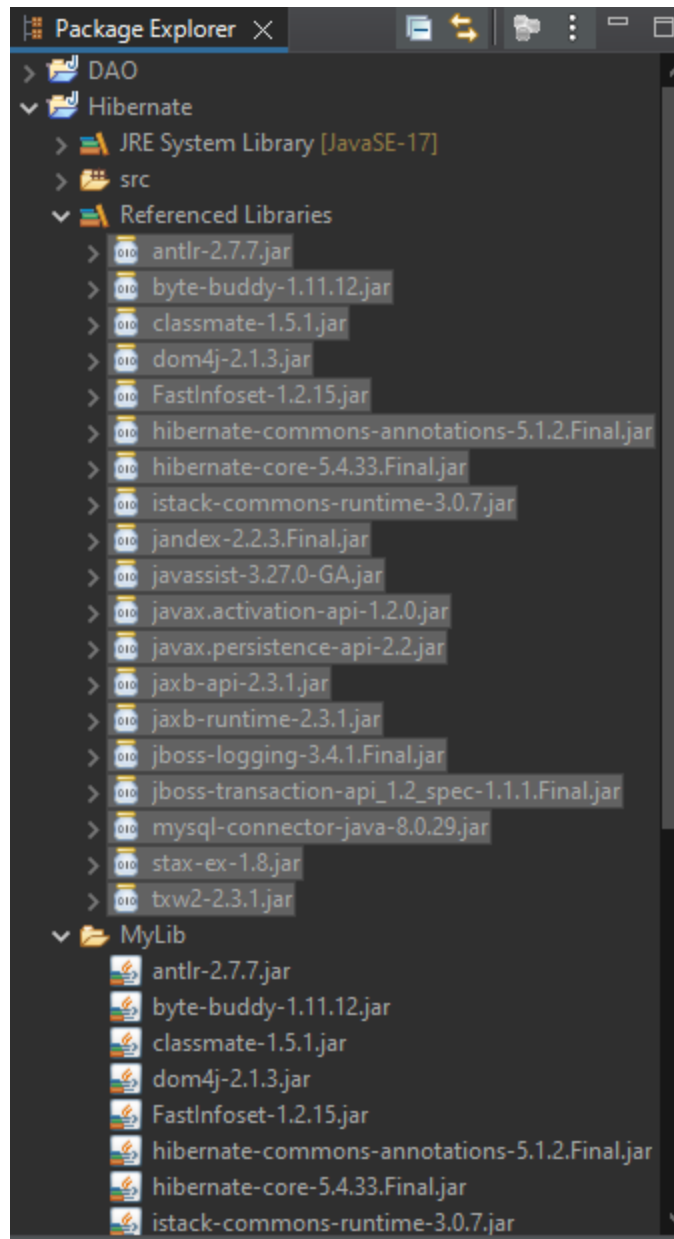
☰ Tags	
▼ status	sent

**Par : IBTISSAM MASSA - GINFO2**

L'objectif de ce TP est de se familiariser avec les mécanismes d'accès aux bases de données en utilisant le JPA- Hibernate,

## JPA-Hibernate

---



```

Book.java X
1 package jpaPack;
2 import javax.persistence.Entity;
3 import javax.persistence.GeneratedValue;|
4 import javax.persistence.Id;
5 import javax.persistence.Table;
6
7 @Entity
8 @Table(name = "book")
9 public class Book {
10     @Id
11     @GeneratedValue
12     private long book_id;
13     private String title;
14     private String author;
15     private float price;
16
17     public Book(long book_id, String title, String author, float price) {
18         this.book_id = book_id;
19         this.title = title;
20         this.author = author;
21         this.price = price;
22     }
23
24     public long getBook_id() {
25         return book_id;
26     }
27     public void setBook_id(long book_id) {
28         this.book_id = book_id;
29     }
30     public String getTitle() {
31         return title;
32     }
33     public void setTitle(String title) {
34         this.title = title;

```

```

Book.java  module-info.java  persistence.xml X
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
5   <persistence-unit name="Bookstore">
6     <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
7     <class>jpaPack.Book</class>
8     <properties>
9       <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />
10      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/biblio" />
11      <property name="javax.persistence.jdbc.user" value="MyUser" />
12      <property name="javax.persistence.jdbc.password" value="1234" />
13
14      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
15      <property name="hibernate.hbm2ddl.auto" value="update" />|
16      <property name="hibernate.show_sql" value="true" />
17    </properties>
18  </persistence-unit>
19 </persistence>
20
21

```

```
Book.java  persistence.xml  BookEm.java  >>_1
1 package jpaPack;
2
3 import java.util.List;
4
5 public interface BookEm {
6     void bookAdd(Book book);
7     void updateBook(Book book);
8     void deleteBook(Long bookId);
9     List<Book> bookDisplayAll();
10    List<Book> bookFindByKw(String kw);
11 }
12
```

```
Book.java  module-info.java  persistence.xml  BookEm.java  BookEmImpl.java  X
1 package jpaPack;
2
3 import java.util.List;
4
5 import javax.persistence.EntityManager;
6 import javax.persistence.EntityManagerFactory;
7 import javax.persistence.TypedQuery;
8
9 public class BookEmImpl implements BookEm{
10
11     private EntityManagerFactory emf;
12     private EntityManager em;
13
14     public BookEmImpl(EntityManagerFactory emf) {
15         this.emf = emf;
16         this.em = emf.createEntityManager();
17     }
18
19     @Override
20     public void bookAdd(Book book) {
21         em.getTransaction().begin();
22         em.persist(book);
23         em.getTransaction().commit();
24     }
25
26     @Override
27     public void updateBook(Book book) {
28         em.getTransaction().begin();
29         em.merge(book);
30         em.getTransaction().commit();
31     }
32
33     @Override
34     public void deleteBook(Long bookId) {
```

```

30     em.getTransaction().commit();
31 }
32
33 ● @Override
34 public void deleteBook(Long bookId) {
35     em.getTransaction().begin();
36     Book book = em.find(Book.class, bookId);
37     if (book != null) {
38         em.remove(book);
39     }
40     em.getTransaction().commit();
41 }
42
43 ● @Override
44 public List<Book> bookDisplayAll() {
45     TypedQuery<Book> query = em.createQuery("SELECT b FROM Book b", Book.class);
46     return query.getResultList();
47 }
48
49 ● @Override
50 public List<Book> bookFindByKw(String kw) {
51     TypedQuery<Book> query = em.createQuery("SELECT b FROM Book b WHERE b.title LIKE :kw", Book.class);
52     query.setParameter("kw", "%" + kw + "%");
53     return query.getResultList();
54 }
55
56 ● public void close() {
57     if (em != null) {
58         em.close();
59     }
60     if (emf != null) {
61         emf.close();
62     }
63 }

```

<