

TP N°1 : Gestion de livres avec les Enterprise JavaBeans (EJB)

Description de l'application

On voudrait créer une application qui permettra de gérer des livres :

- Ajouter un nouveau livre ;
- Obtenir la liste de tous les livres existants.

Cette gestion doit respecter les règles suivantes :

1. Un auteur est défini par un identifiant et un nom ;
2. Un livre est défini par un identifiant et un intitulé ;
3. Un livre peut être écrit par plusieurs auteurs.

Etape 1 : Implémentation des beans entité

Pour réaliser votre application, vous allez créer deux beans entité comme suit :

Auteur.java :

```
@Entity
public class Auteur implements Serializable{

    @Id
    private String id;
    private String nom;
    @ManyToMany(mappedBy="auteurs")
    private Set<Livre> livres;

    public Auteur() {
        super();
    }
    //Constructeurs avec paramètres
    //Getters et setters
}
```

Livre.java:

```
@Entity
public class Livre implements Serializable{

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int id;
    private String intitule;
    @ManyToMany(cascade = {CascadeType.PERSIST, CascadeType.MERGE}, fetch
    = FetchType.EAGER)
    private Set<Auteur> auteurs;
    public Livre() {
        super();
    }
    //Constructeurs avec paramètres
    //Getters et setters
}
```

Etape 2 : Création du fichier persistence.xml

Sous le répertoire **META-INF**, créez un nouveau fichier que vous nommez **persistence.xml**, puis ajoutez le code suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
version="1.0">

<persistence-unit name="MonEntiteEJB3">
<jta-data-source>java:jboss/datasources/ExampleDS</jta-data-source>
<properties>
<property name="hibernate.hbm2ddl.auto" value="update"/>
</properties>
</persistence-unit>
</persistence>
```

Etape 3 : Création de l'interface de l'EJB session stateless : GestionLivresRemote.java

Cette interface propose deux services :

```
@Remote
public interface GestionLivresRemote {
    void ajouterLivre(Livre livre);
    List<Livre> listerTousLesLivres();
}
```

Etape 4 : Implémentation de l'EJB session stateless : GestionLivres.java

Cette classe implémente l'interface précédente comme suit:

```
@Stateless
public class GestionLivres implements GestionLivresRemote {

    @PersistenceContext(unitName="MonEntiteEJB3")
    private EntityManager em;

    public GestionLivres() {
    }
    @Override
    public void ajouterLivre(Livre livre) {
        em.persist(livre);
    }
    @Override
    public List<Livre> listerTousLesLivres() {
        return em.createQuery("select l from Livre l")
            .getResultList();
    }
}
```

Etape 5 : Implémentation du Client : ClientGestionLivres.java

Avant d'écrire le client, n'oubliez pas de déployer le projet EJB et d'ajouter, au projet du client, la librairie **jboss-client.jar** et le fichier **jboss-ejb-client.properties**.

jboss-ejb-client.properties :

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLE
D=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port =8080
remote.connection.default.connect.options.org.xnio.Options.SASL_POLI
CY_NOANONYMOUS=false
```

ClientGestionLivres.java :

```
public class ClientGestionLivres {

    public static void main(String[] args) {

        try {
            Properties p= new Properties();
            p.put(Context.URL_PKG_PREFIXES, "org.jboss.ejb.client.naming");
            InitialContext context = new InitialContext(p);
            GestionLivresRemote gestion = (GestionLivresRemote)
            context.lookup("ejb:/GestionLivres/GestionLivres!metier.GestionLivresRemote");

            //L'interface invoquée appartient au package appelé « metier » dans
            //un projet appelé « GestionLivres »

            //Complétez par le code remplissant la base de données

            //Complétez par le code listant tous les livres ajoutés
```

Test de l'application

Lancez le programme client après avoir vérifié que le serveur Wildfly est en état de service et que le projet EJB est déployé dedans avec succès.