

## TP 6 Java EE :

### JPA Relation entre entités et CRUD

On se propose dans ce TP, d'ajouter une deuxième entité *Categorie* et de la relier avec l'entité *Produit* existante. Et de créer les interfaces et les traitements nécessaires. Sachant qu'un produit possède une seule catégorie et une catégorie contient plusieurs produits.

#### Objectifs :

- Création de l'entité *Categorie* et sa relation avec l'entité *Produit*,
  - Création de l'interface *ICategorieDao* et son implémentation *CategorieDaoImpl*,
  - Création de la servlet *CategorieServlet*,
  - Création des vues jsp nécessaires pour les opérations CRUD sur l'entité *Categorie*,
  - Ajouter une liste déroulante contenant les catégories aux interfaces *saisieProduit* et *editerProduit*.
1. Ajoutez au projet TP5\_JEE, l'entité ***metier/entities/Categorie***, ses attributs sont :
    - *idCat* (Long, auto increment)
    - *nomCat* (String)
    - *dateCreation* (java.util.Date)

```
package metier.entities;

import java.io.Serializable;
import java.util.Date;
import java.util.List;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

@Entity
public class Categorie implements Serializable {

    @Id
    @GeneratedValue (strategy=GenerationType.IDENTITY)
    private Long idCat;
```

```

private String      nomCat;

@Temporal( TemporalType.DATE )
private Date dateCreation;

@OneToMany (mappedBy="categorie")
private List<Produit> produits;

public Categorie() {
    super();
}
public Categorie(String nomCat, Date dateCreation) {
    super();
    this.nomCat = nomCat;
    this.dateCreation = dateCreation;
}
public Long getIdCat() {
    return idCat;
}
public void setIdCat(Long idCat) {
    this.idCat = idCat;
}
public String getNomCat() {
    return nomCat;
}
public void setNomCat(String nomCat) {
    this.nomCat = nomCat;
}
public Date getDateCreation() {
    return dateCreation;
}
public void setDateCreation(Date dateCreation) {
    this.dateCreation = dateCreation;
}
public List<Produit> getProduits() {
    return produits;
}
public void setProduits(List<Produit> produits) {
    this.produits = produits;
}
}

```

2. Clic droit sur le fichier Persistence.xml puis « Synchronize Class List »
3. Sachant que l'association entre *Produit* et *Categorie* est **bidirectionnelle**, Modifier l'entité Produit pour y ajouter l'attribut categorie :

```

private Categorie categorie;

public Produit(String nomProduit, double prix,Categorie cat) {
    super();
    this.nomProduit = nomProduit;
    this.prix = prix;
    this.setCategorie(cat);
}

```

```

public Kategorie getKategorie() {
    return categorie;
}

public void setKategorie(Kategorie categorie) {
    this.categorie = categorie;
}

```

4. Pour créer la table catégorie et recréer la table Produit : Clic droit sur le projet/JPA Tools/generate tables from entities...

#### 5. Créer l'interface *IKategorieDao*

```

package dao;

import java.util.List;
import metier.entities.Kategorie;

public interface IKategorieDao {
    public Kategorie save(Kategorie cat);
    public Kategorie getKategorie(Long id);
    public Kategorie updateKategorie(Kategorie cat);
    public void deleteKategorie(Long id);
    public List<Kategorie> getAllKategories();
}

```

#### 6. Créer l'implémentation *KategorieDaoImpl* de l'interface *IKategorieDao*,

```

package dao;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityTransaction;
import metier.entities.Kategorie;
import util.JPAUtil;

public class KategorieDaoImpl implements IKategorieDao {
    // TP6_JEE à remplacer par votre persistence unit, consultez votre
    //fichier persistence.xml <persistence-unit name="TP6_JEE">
    private EntityManager entityManager=JPAUtil.getEntityManager("TP6_JEE");

    @Override
    public Kategorie save(Kategorie cat ) {
        EntityTransaction tx = entityManager.getTransaction();
        tx.begin();
        entityManager.persist(cat);
        tx.commit();

        return cat;
    }

    @Override
    public Kategorie getKategorie(Long id) {
        return entityManager.find(Kategorie.class, id);
    }
}

```

```

@Override
public Kategorie updateCategorie(Kategorie cat) {
    EntityTransaction tx = entityManager.getTransaction();
    tx.begin();
    entityManager.merge(cat);
    tx.commit();
    return cat;
}

@Override
public void deleteCategorie(Long id) {
    Kategorie categorie = entityManager.find(Kategorie.class, id);

    entityManager.getTransaction().begin();
    entityManager.remove(categorie);
    entityManager.getTransaction().commit();
}

@Override
public List<Kategorie> getAllCategories() {
    List<Kategorie> cats =
    entityManager.createQuery("select c from Kategorie c").getResultList();

    return cats;
}
}

```

## 7. Créer la classe web/CategorieModel :

```

package web;

import java.util.ArrayList;
import java.util.List;
import metier.entities.Categorie;

public class CategorieModele {
    List<Categorie> categories = new ArrayList<>();

    public List<Categorie> getCategories() {
        return categories;
    }

    public void setCategories(List<Categorie> categories) {
        this.categories = categories;
    }
}

```

## 8. Créer la servlet web/CategorieServlet (deux servlets **ne peuvent pas avoir** le même urlPatterns),

```

package web;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

```

```

import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.catalina.connector.Response;
import dao.CategorieDaoImpl;
import dao.ICategorieDao;
import metier.entities.Categorie;

@WebServlet (name="catServ",urlPatterns= {"/categories","/saisieCategorie",
    "/saveCategorie","/supprimerCat","/editerCat","/updateCat"})
public class CategorieServlet extends HttpServlet {

    ICategorieDao metier;
    @Override
    public void init() throws ServletException {

        metier = new CategorieDaoImpl();
    }

    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        String path=request.getServletPath();

        System.out.println("PATH "+path);
        if (path.equals("/categories"))
        {

            CategorieModele model= new CategorieModele();

            List<Categorie> cats = metier.getAllCategories();
            model.setCategories(cats);
            request.setAttribute("model", model);

            request.getRequestDispatcher("categories.jsp").forward(request,response);
        }

        else if (path.equals("/saisieCategorie") )
        {

            request.getRequestDispatcher("saisieCategorie.jsp").forward(request,response
);
        }

        else if (path.equals("/saveCategorie") &&
request.getMethod().equals("POST"))
        {
            Date dateCat= new Date();
            String nom=request.getParameter("nom");
            String pattern = "yyyy-MM-dd";
            SimpleDateFormat simpleDateFormat = new
SimpleDateFormat(pattern);

            try {

```

```

        dateCat =
simpleDateFormat.parse(request.getParameter("dateCat"));
    } catch (ParseException e) {

        e.printStackTrace();

    }

    Categorie cat = metier.save(new Categorie(nom,dateCat));
    request.setAttribute("categorie", cat);
    response.sendRedirect("categories");
}
else if (path.equals("/supprimerCat"))
{
    Long id= Long.parseLong(request.getParameter("id"));
    metier.deleteCategorie(id);
    response.sendRedirect("categories");
}
else if (path.equals("/editerCat") )
{
    Long id= Long.parseLong(request.getParameter("id"));
    Categorie cat = metier.getCategorie(id);
    request.setAttribute("categorie", cat);

    request.getRequestDispatcher("editerCategorie.jsp").forward(request,response
);
}
else if (path.equals("/updateCat") )
{
    Date dateCat= new Date();
    Long id = Long.parseLong(request.getParameter("id"));
    String nom=request.getParameter("nom");
    Categorie cat = new Categorie();
    cat.setIdCat(id);
    cat.setNomCat(nom);
    String pattern = "yyyy-MM-dd";
    SimpleDateFormat simpleDateFormat = new
SimpleDateFormat(pattern);

    try {
        dateCat =
simpleDateFormat.parse(request.getParameter("dateCreation"));
    } catch (ParseException e) {

        e.printStackTrace();

    }
    cat.setDateCreation(dateCat);

    metier.updateCategorie(cat);
    response.sendRedirect("categories");

}
else
{
    response.sendError(Response.SC_NOT_FOUND);
}
}
@Override
protected void doPost(HttpServletRequest request,

```

```

                                HttpServletResponse response) throws
ServletException, IOException {
    doGet(request, response);
}}

```

9. Créer la vue saisieCategorie.jsp pour ajouter une nouvelle catégorie,

```

<%@ page language="java" contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>Saisie Catégorie</title>
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />
</head>
<body>
<%@include file="header.jsp" %>
<p></p>
<div class="container">
<div class="card">
    <div class="card-header">
        Saisie des Catégories
    </div>
    <div class="card-body">
        <form action="saveCategorie" method="post">
            <div class="form-group">
                <label class="control-label">Nom Catégorie :</label>
                <input type="text" name="nom" class="form-control"/>
            </div>
            <div class="form-group">
                <label class="control-label">Date Catégorie : </label>
                <input type="date" name="dateCat" class="form-control" />
            </div>
            <div>
                <button type="submit" class="btn btn-primary">Ajouter</button>
            </div>
        </form>
    </div>
</div>
</div>
</body>
</html>

```

10. Créer la vue categories.jsp pour afficher la liste des catégories :

```

<%@ page language="java" contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>Liste des catégories</title>
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />

```

```

</head>
<body>
<%@include file="header.jsp" %>
<p></p>
<div class="container">
<div class="card">
  <div class="card-header">
    Liste des catégories
  </div>
  <div class="card-body">

    <table class="table table-striped">
      <tr>
        <th>ID</th><th>Catégorie</th><th>Date
Création</th><th>Suppression</th><th>Edition</th>
      </tr>
      <c:forEach items="${model.categories}" var="cat">
        <tr>
          <td>${cat.idCat }</td>
          <td>${cat.nomCat }</td>
          <td><fmt:formatDate pattern="dd/MM/yyyy" value="${cat.dateCreation}"
/></td>
          <td><a onclick="return confirm('Etes-vous sûr ?')"
href="supprimerCat?id=${cat.idCat }">Supprimer</a></td>
          <td><a href="editerCat?id=${cat.idCat }">Edit</a></td>
        </tr>
      </c:forEach>
    </table>
  </div>
</div>
</div>
</body>
</html>

```

11. Ajouter au menu de votre application une commande pour accéder à la gestion des catégories. Pour ce faire ajouter au fichier header.jsp les lignes suivantes :

```

<li class="nav-item active">
  <a class="nav-link" href="categories">Catégories</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="saisieCategorie">Saisie Catégorie</a>
</li>

```

12. Ajouter 3 catégories

13. Créer la vue editerCategorie.jsp pour modifier une catégorie :

```

<%@ page language="java" contentType="text/html; charset=windows-1256"
pageEncoding="windows-1256"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256">

```



```

<title> Modification des catégories</title>
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />
</head>
<body>
<%@include file="header.jsp" %>
<p></p>
<div class="container">
<div class="card">
    <div class="card-header">
        Modification des catégories
    </div>
    <div class="card-body">
        <form action="updateCat" method="post" >
            <div hidden class="form-group" >
                <label class="control-label">ID Catégorie :</label>
                <input type="text" name="id" class="form-control"
value="${categorie.idCat}"/>
            </div>
            <div class="form-group">
                <label class="control-label">Nom Catégorie :</label>
                <input type="text" name="nom" class="form-control"
value="${categorie.nomCat}"/>
            </div>
            <div class="form-group">
                <label class="control-label">Date Catégorie : </label>
                <fmt:formatDate pattern="yyyy-MM-dd" value="${categorie.dateCreation}"
var="formatDate" />
                <input type="date" name="dateCreation" class="form-control"
value="${formatDate}"/></input>
            </div>
            <div>
                <button type="submit" class="btn btn-primary">Modifier</button>
            </div>
        </form>
    </div>
</div>
</div>
</body>
</html>

```

14. Pour attribuer une catégorie à un produit, modifier la servlet ControleurServlet comme suit :

```
public class ControleurServlet extends HttpServlet {

    IProduitDao metier;
    ICategoryDao metierCat;
    @Override
    public void init() throws ServletException {
        metier = new ProduitDaoImpl();
        metierCat = new CategoryDaoImpl();
    }
    ...

    else if (path.equals("/saisie.do") )
    {
        List<Category> cats = metierCat.getAllCategories();
        CategoryModele model= new CategoryModele();
        model.setCategories(cats);
        request.setAttribute("catModel", model);

        request.getRequestDispatcher("saisieProduit.jsp").forward(request,response);
    }
    ...

    else if (path.equals("/editer.do") )
    {
        Long id= Long.parseLong(request.getParameter("id"));
        Produit p = metier.getProduit(id);
        request.setAttribute("produit", p);

        List<Category> cats = metierCat.getAllCategories();
        CategoryModele model= new CategoryModele();
        model.setCategories(cats);
        request.setAttribute("catModel", model);

        request.getRequestDispatcher("editerProduit.jsp").forward(request,response);
    }
    ...

    else if (path.equals("/save.do") && request.getMethod().equals("POST"))
    {
        String nom=request.getParameter("nom");
        Long categorieId=Long.parseLong(request.getParameter("categorie"));
        double prix = Double.parseDouble(request.getParameter("prix"));
        Category cat = metierCat.getCategory(categorieId);
        Produit p = metier.save(new Produit(nom,prix,cat));
        request.setAttribute("produit", p);
        response.sendRedirect("chercher.do?motCle=");
    }
    ...

    else if (path.equals("/update.do") )
    {
        Long id = Long.parseLong(request.getParameter("id"));
        String nom=request.getParameter("nom");
        double prix = Double.parseDouble(request.getParameter("prix"));
        Long categorieId=Long.parseLong(request.getParameter("categorie"));
        Produit p = new Produit();
        p.setIdProduit(id);
        p.setNomProduit(nom);
    }
}
```

```

        p.setPrix(prix);
        Categorie cat = metierCat.getCategorie(categorieId);
        p.setCategorie(cat);
        metier.updateProduit(p);
        response.sendRedirect("chercher.do?motCle=");
    }

```

...

#### 15. Ajouter la colonne catégorie à la vue produits.jsp

```

<tr>
    <th>ID</th><th>Nom Produit</th><th>Prix</th><th>Catégorie</th>
</tr>
...
<td>${p.categorie.nomCat}</td>

```

#### 16. Ajouter à la vue saisieProduit.jsp une liste de catégories pour attribuer une catégorie à un produit.

```

<div class="form-group">
    <select name="categorie" class="form-control">
        <c:forEach items="${catModel.categories}" var="cat">
            <option value="${cat.idCat}">${cat.nomCat}</option>
        </c:forEach>
    </select>
</div>

```

#### 17. Ajouter à la vue editierProduit.jsp une liste de catégories pour modifier la catégorie d'un produit.

```

<div class="form-group">
    <select name="categorie" class="form-control">
        <option value="${produit.categorie.idCat}"
selected>${produit.categorie.nomCat}</option>
        <c:forEach items="${catModel.categories}" var="cat">
            <c:if test="${cat.idCat != produit.categorie.idCat}">
                <option value="${cat.idCat}">${cat.nomCat}</option>
            </c:if>
        </c:forEach>
    </select>
</div>

```