

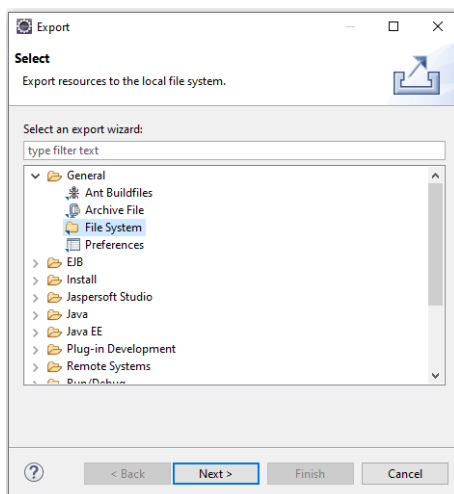
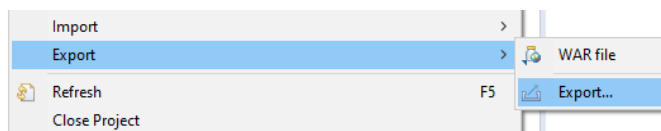
# TP 5 Java EE

## Intégration de l'API JPA à un projet Web dynamique

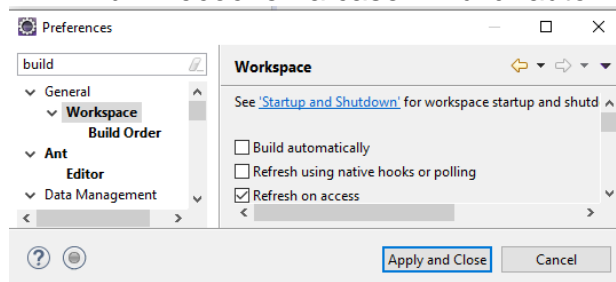
### Objectif :

On se propose dans ce TP, d'ajouter la facet JPA au projet web dynamique du TP4. La couche persistance des entités sera prise en charge par l'api JPA.

1. Copier le projet « *Dynamic Web Project* » *TP4\_JEE* en *TP5\_JEE* :
  - a. Exporter le projet *TP4\_JEE* vers un dossier de votre choix (Clic droit sur le projet/Export/Export...)

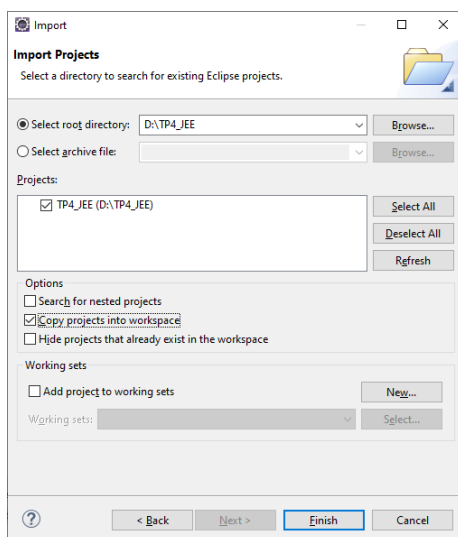
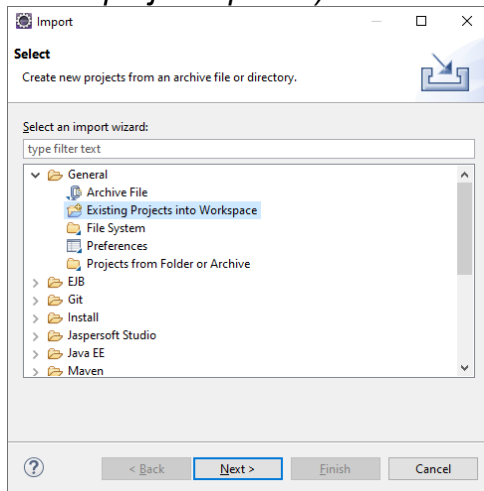


- b. Décocher la case « Build automatically » sans Preferences

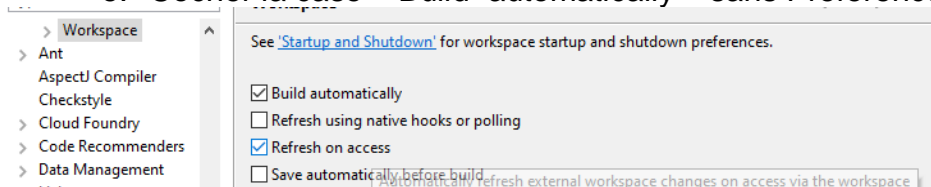


c. Renommer le projet TP4\_JEE en TP5\_JEE (Clic droit sur le projet/Refactor/Rename...)

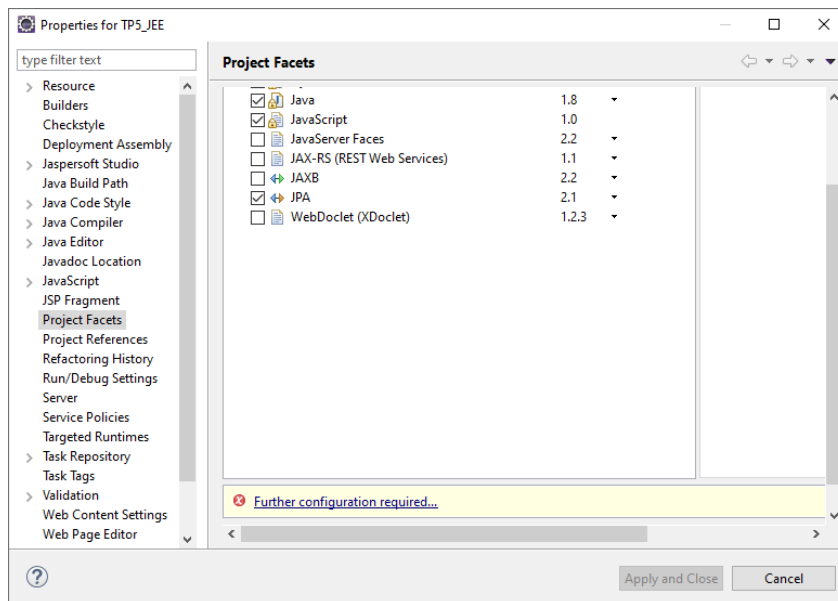
d. Importer le projet TP4\_JEE du dossier de votre choix (Clic droit sur le projet/Import...)



e. Cocher la case « Build automatically » sans Preferences



2. Ajouter la facet JPA au projet : (Clic droit sur le projet/Properties/Project Facets...)



3. Configurer JPA pour exploiter la BD MySQL *DB\_MVC\_CAT* (utilisée au TP4), voir la vidéo youtube intitulé : « Tutoriel : L'API JPA utilisant L'ORM EclipseLink et le SGBD MySQL » url : <https://youtu.be/YpSEpQm890s>
4. Ouvrez votre fichier Persistence.xml et modifier le comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="TP5_JEE">
    <properties>
      <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/DB_MVC_CAT"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="javax.persistence.jdbc.driver"
value="com.mysql.jdbc.Driver"/>
      <property name="eclipseLink.logging.level" value="FINE"/>
      <property name="eclipseLink.ddl-generation" value="none"/>
    </properties>
  </persistence-unit>
</persistence>
```

5. Puisqu'on va utiliser l'api JPA pour la persistance des entités et non pas JDBC, Supprimez la classe *SingletonConnection*,

6. Ajouter les annotations nécessaires à l'entité (la JavaBean Produit) comme suit :

```
package metier.entities;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "PRODUITS")
public class Produit implements Serializable{

    @Id
    @Column (name="ID_PRODUIT")
    @GeneratedValue (strategy=GenerationType.IDENTITY)
    private Long idProduit;

    @Column (name="NOM_PRODUIT")
    private String nomProduit;
    private double prix;

    public Produit() {
        super();
    }
    public Produit(String nomProduit, double prix) {
        super();
        this.nomProduit = nomProduit;
        this.prix = prix;
    }
    public Long getIdProduit() {
        return idProduit;
    }
    public void setIdProduit(Long idProduit) {
        this.idProduit = idProduit;
    }
    public String getNomProduit() {
        return nomProduit;
    }
    public void setNomProduit(String nomProduit) {
        this.nomProduit = nomProduit;
    }
    public double getPrix() {
        return prix;
    }
    public void setPrix(double prix) {
        this.prix = prix;
    }
    @Override
    public String toString() {
        return "Produit [idProduit=" + idProduit + ", nomProduit=" +
nomProduit + ", prix=" + prix + "]\n";
    }
}
```

7. Clic droit sur le fichier Persistence.xml puis « Synchronize Class List »
8. Créez le package util, puis créez dans ce package la classe JpaUtil :

```
package util;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class JPAUtil {
    private static EntityManagerFactory factory;
    private static EntityManager entityManager;

    public static EntityManager getEntityManager(String persistUnit)
    {
        if (entityManager==null)
        {
            factory =
Persistence.createEntityManagerFactory(persistUnit);

            entityManager = factory.createEntityManager();
        }
        return entityManager;
    }
}
```

9. Modifier la classe ProduitDaoImpl pour qu'elle utilise JPA comme suit :

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

public class ProduitDaoImpl implements IProduitDao {
    private EntityManager entityManager=JPAUtil.getEntityManager("TP5_JEE");

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

        return p;
    }

    @Override
    public List<Produit> produitsParMC(String mc) {
        List<Produit> prods =
```

```

        entityManager.createQuery("select p from Produit p where
p.nomProduit like :mc").setParameter("mc", "%" + mc + "%")
        .getResultList();
    return prods;
}

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

@Override
public Produit updateProduit(Produit p) {
    EntityTransaction tx = entityManager.getTransaction();
    tx.begin();
    entityManager.merge(p);
    tx.commit();
    return p;
}

@Override
public void deleteProduit(Long id) {
    Produit produit = entityManager.find(Produit.class, id);

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

```

10. Tester votre travail : [http://localhost:8080/TP5\\_JEE/produits.jsp](http://localhost:8080/TP5_JEE/produits.jsp)