

28/09/2025

# Jakarta EE

## TP 1 :

Implémentation de CDI :

Mini Bibliothèque

**Réalisé par :**

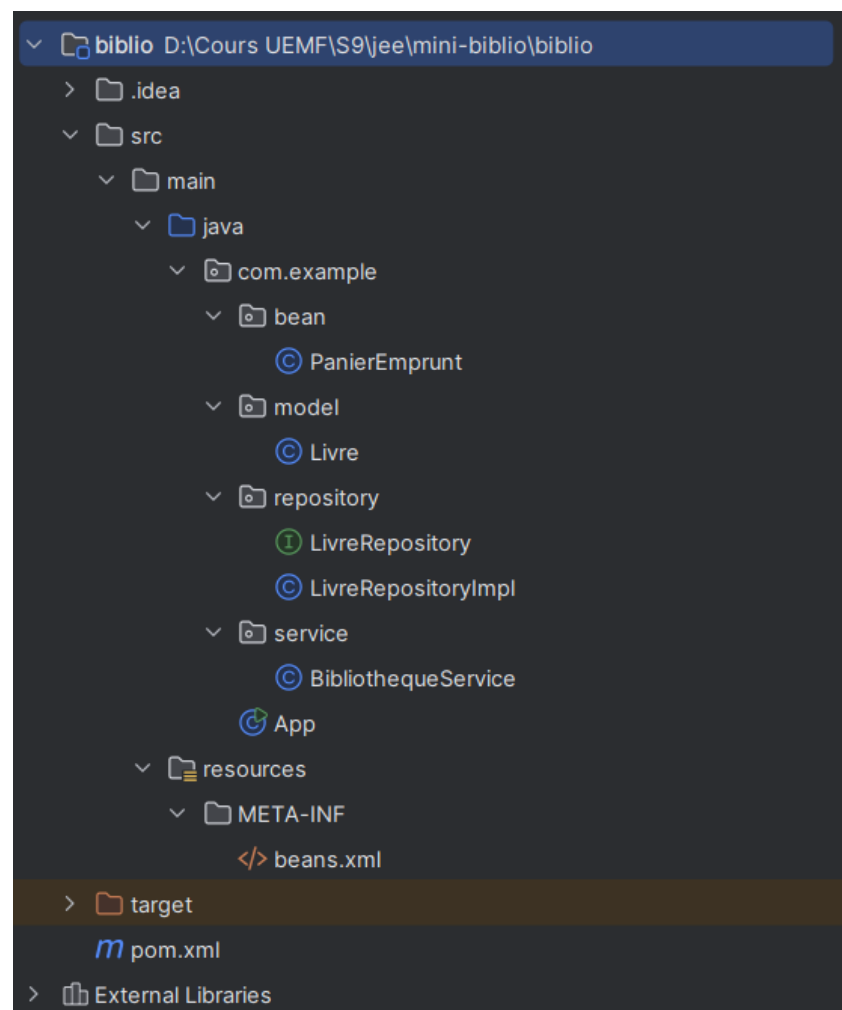
✓ Hamza EL YOUSOUFI

## I. Introduction

Le projet “mini-biblio” consiste en une application console en Java permettant de gérer des livres et des emprunts à l’aide de la spécification CDI (Contexts and Dependency Injection). L’objectif principal est de montrer comment l’injection de dépendances facilite la modularité et la maintenabilité du code.

## II. Structure du projet :

Le projet est organisé comme suit :



- App.java : classe principale pour tester les fonctionnalités :

```
package com.example;

import ...

public class App {
    public static void main(String[] args) {
        Weld weld = new Weld();
        try (WeldContainer container = weld.initialize()) {
            BibliothequeService service = container.select(BibliothequeService.class).get();
            PanierEmprunt panier = container.select(PanierEmprunt.class).get();

            service.afficherLivre(id: 1);
            panier.ajouterLivre(new Livre(id: 2, titre: "Design Patterns"));
            panier.afficherPanier();
        }
    }
}
```

- PanierEmprunt.java : bean CDI représentant le panier d'emprunts :

```
package com.example.bean;

import ...

@ApplicationScoped 3 usages
public class PanierEmprunt implements Serializable {
    private List<Livre> livres = new ArrayList<>(); 2 usages

    public void ajouterLivre(Livre livre) { livres.add(livre); }

    public void afficherPanier() { System.out.println("📖 Panier : " + livres); }
}
```

- LivreRepositoryImpl.java : repository pour accéder aux livres :

```
package com.example.repository;

import ...

@ApplicationScoped no usages
public class LivreRepositoryImpl implements LivreRepository {
    @Override 1 usage
    @Inject
    public Livre trouverParId(long id) { return new Livre(id, titre: "Introduction à CDI"); }
}
```

- BibliothequeService.java : Service CDI (qui permet d'interagir avec le dépôt de livres :

```

package com.example.service;

import com.example.model.Livre;
import com.example.repository.LivreRepository;
import jakarta.inject.Inject;
import jakarta.enterprise.context.Dependent;

@Dependent 3 usages
public class BibliothequeService {

    @Inject 1 usage
    private LivreRepository livreRepo;

    public void afficherLivre(long id) { 1 usage
        Livre livre = livreRepo.trouverParId(id);
        System.out.println("📖 Livre trouvé : " + livre);
    }
}
    
```

### III. Résultat :

```

ept. 28, 2025 3:52:27 PM org.jboss.weld.bootstrap.WeldStartup startContainer
INFO: WELD-000101: Transactional services not available. Injection of @Inject UserTransaction
ept. 28, 2025 3:52:28 PM org.jboss.weld.environment.se.WeldContainer fireContainerInitializ
INFO: WELD-ENV-002003: Weld SE container be0401ba-3130-4981-8a48-42f4cb3eea99 initialized
📖 Livre trouvé : [1] Introduction à CDI
🛒 Panier : [[2] Design Patterns]
ept. 28, 2025 3:52:28 PM org.jboss.weld.environment.se.WeldContainer shutdown
INFO: WELD-ENV-002001: Weld SE container be0401ba-3130-4981-8a48-42f4cb3eea99 shut down
    
```