**Etudiant 1** : Nom : Benkaba Prénom : Marwa

**Rapport TP N°01 :**

**A/ Création du projet :**

 Créer un module contenant :

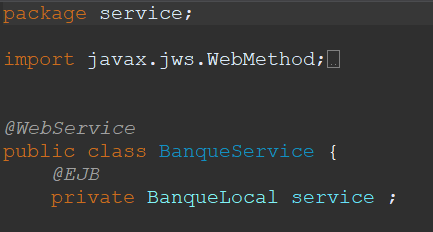
* **BanqueLocale**
* **BanqueRemote**

 Créer un module **entity** contenant une classe **Compte Java**.

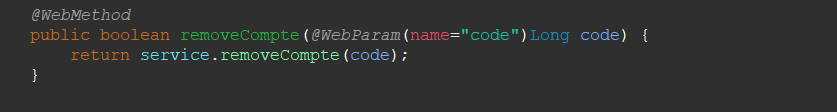
* Préparer la classe pour être mappée sur la base de données avec ORM.

**B/ Modification de la classe "BanqueEJBImpl" :**

1. Remplacer **@Stateful** par **@Stateless**, car le service web ne garde pas l’état du client.

**C/ Création du projet Web :**

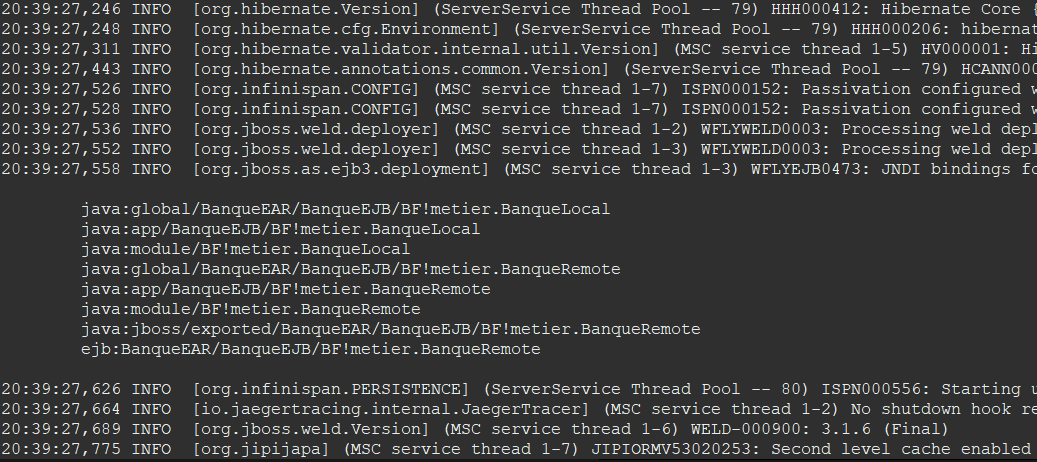
1. Créer un projet **Web Dynamic** nommé **"BanqueWeb"**.
2. Créer le package **"Service"**, puis ajouter une classe **"BanqueService"**.



-nAjouter l’annotation **@WebService** (javax.jws.WebService).

1. Ajouter BanqueEjb a BanqueWeb via : BanqueWeb 🡪 properties 🡪 Add 🡪 BanqueEJB
2. Instancier l’objet : private BanqueLocal service ;
3. Ajouter les methods : addCompte, getCompte, listComptes, verser, retirer, virement, removeCompte.

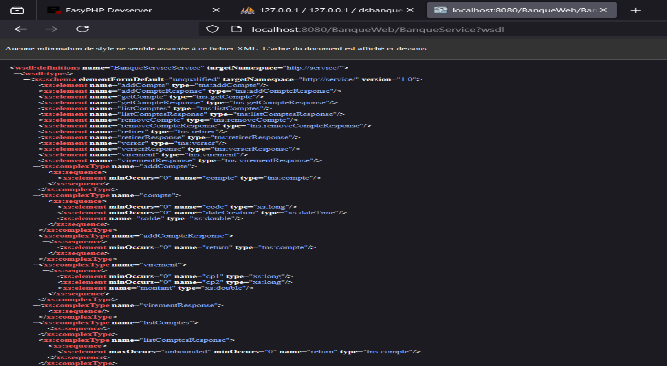
- En utilisant les annotetions *@WebMethod et @WebParam..*

 **D/ Création du projet EAR:**

1. New project java EE 🡪 BanqueEAR 🡪 webBanque 🡪 BanqueEJB 🡪 Finish.

**E/ Lancement du serveur :**

1. Aller dans **Server**, puis sélectionner **WildFly**.

2. Supprimer **BanqueEJB** et ajouter **BanqueEAR**.

3. Cliquer sur **Finish**.

4. Redémarrer le serveur.

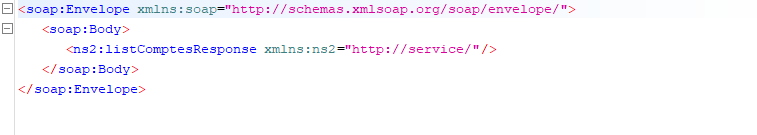
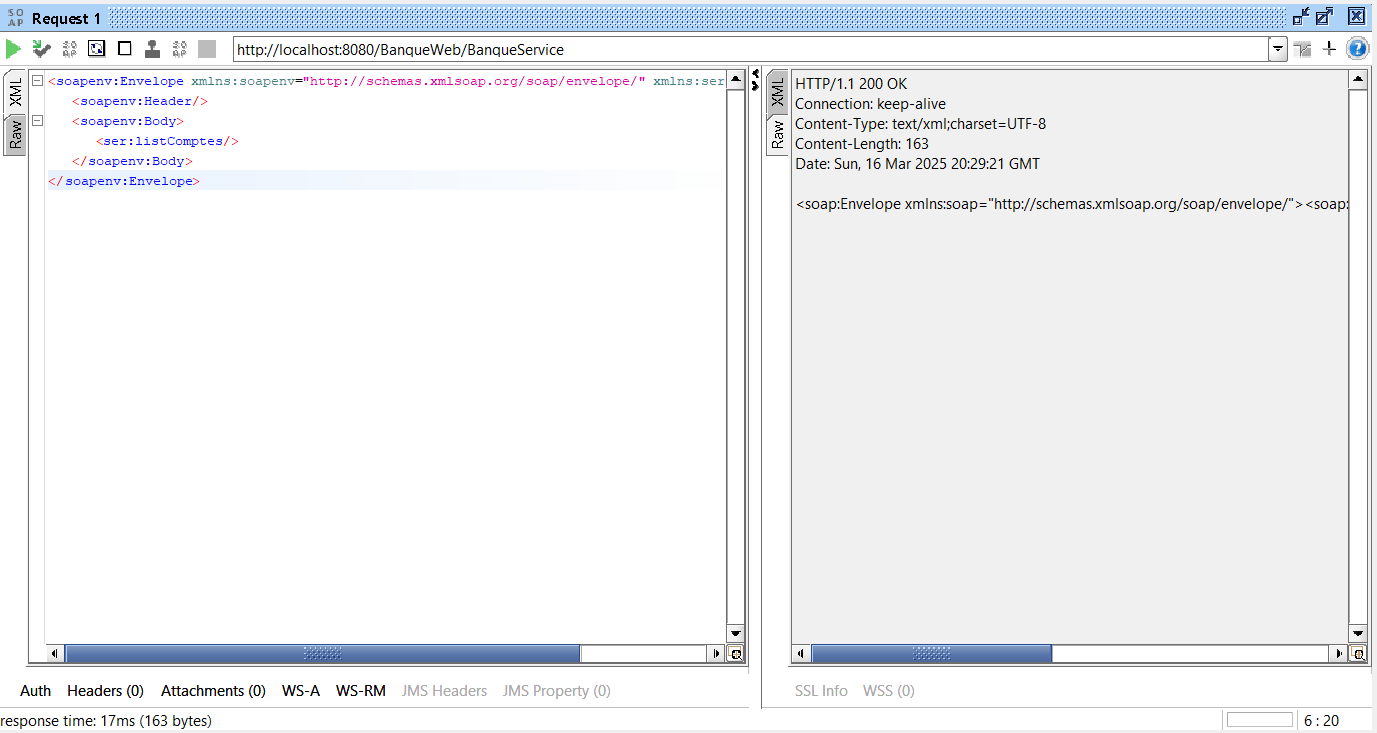
5. Copier l’URL suivante dans un navigateur :

http://localhost:8080/BanqueWeb/BanqueService?wsdl’

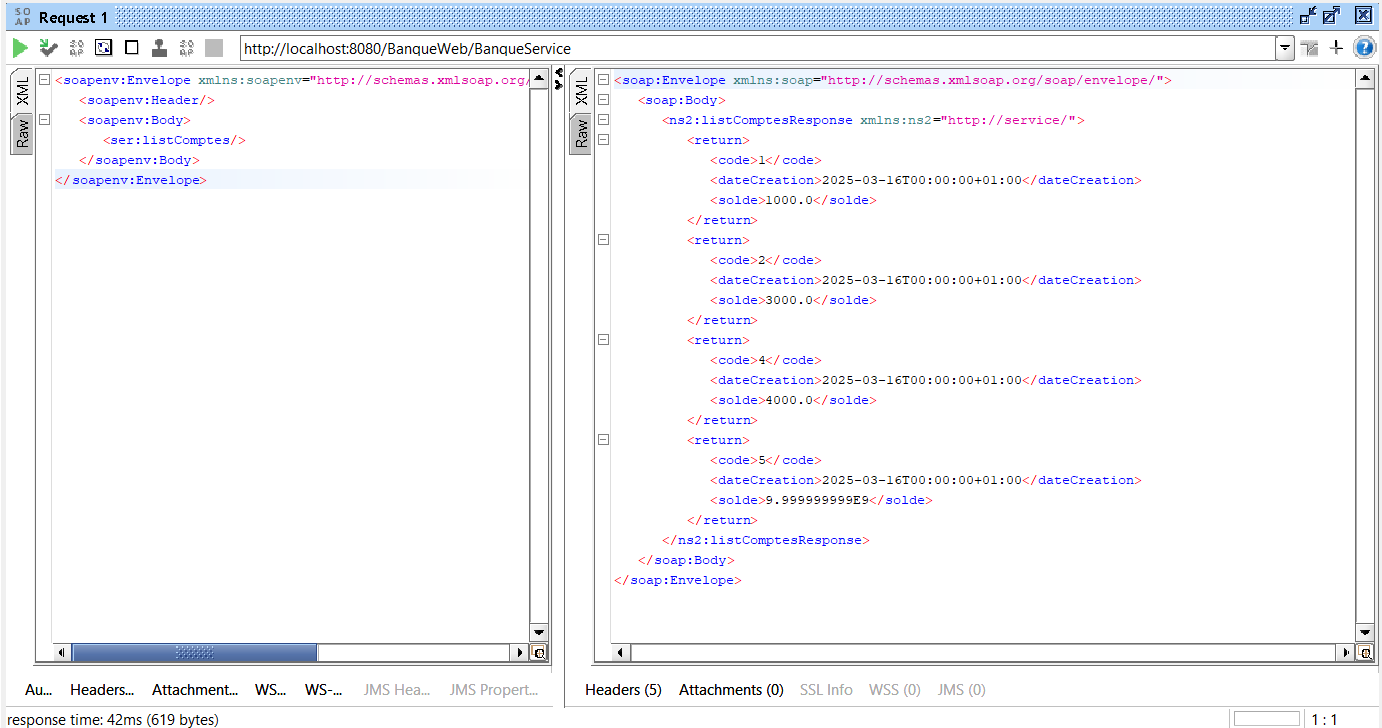
**F/ Test avec SOAP UI:**

1. Importer le fichier WSDL dans SOAP UI.

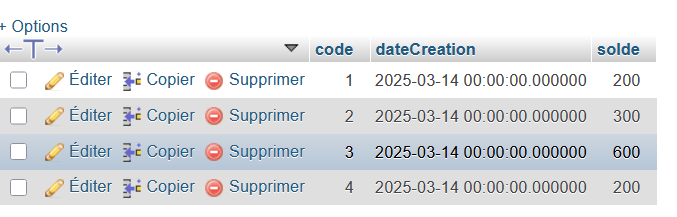
2. Effectuer des tests sur les différentes méthodes du web service.



**G/ Lancement du client:**

**Tester les methodes autre fois dans SOAP UI.**

200 OK

  **La base de données :**