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

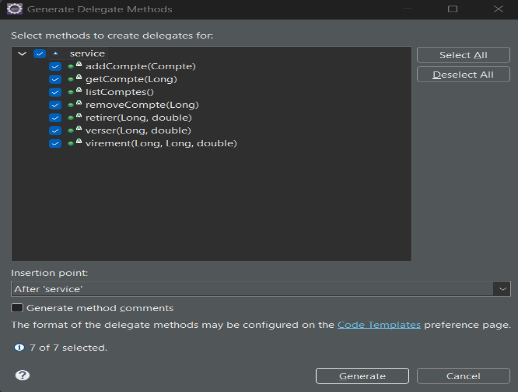
**Rapport TP N°02 :**

**REST:**

**A/ Création du projet :**

 le projet deja exist (projet BanqueWeb).

**B/ Creation du la classe " BanqueServiceRest** **" :**

1. Dans le projet BanqueWeb et sur le package Service on va créé un nouveau class nommé " **BanqueServiceRest** ".
2. Ajouter un annotation **@Path("/")**.
3. Instensé un objet local **BanqueLocal service;** .
4. Ajouter un annotation **@EJB** Pour l'injection de Ejb.
5. Gérer les methodes automatiquement par l'outile " **Generate delegate methods**" et sélectionnez tous les methods nécessaires.
6. Ajouter Les annotations sur tous les methods.

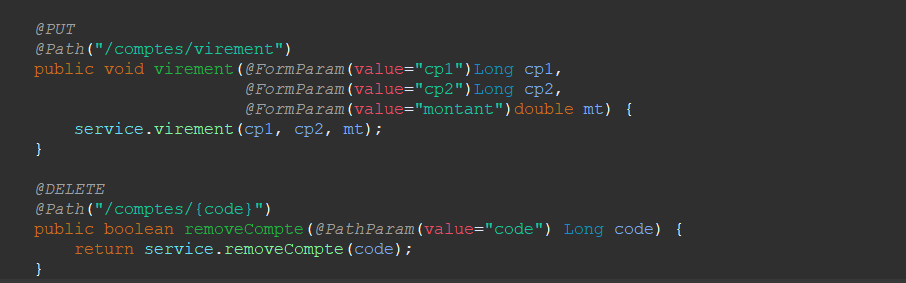
Ex : sur la methode **addCompt**:

\_ ajoutes annotations **@Post** .

\_ **@Path ("/Comptes")** .

\_ **@Produces (MediaType, APPLICATION-JSON)** .

et appliqué set opération sur tous les methode avec ajouter les annotations adéquat.



 les annotations:

-@Stateless: Définit un EJB sans état (REST est une requets sans etat).

-@EJB: Injecte un EJB dans le service.

-*@POST: Ajoute une nouvelle ressource « Create ».*

*-@GET: Récupère une ressource existante « Read ».*

*-@PUT: Met à jour une ressource existante  « Update ».*

*-@DELETE: Supprime une ressource « Delete ».*

*-@Path(..): Définit l'URL du service REST.*

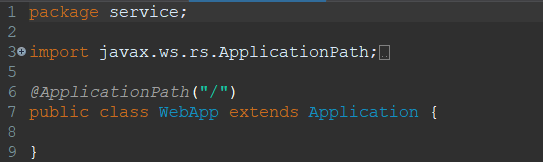
*-@Produces({MediaType.****APPLICATION\_JSON****,MediaType.****APPLICATION\_XML****}): Indique le format de la réponse (JSON, XML).*

*-****@Consumes(MediaType.APPLICATION\_JSON)*** *: Indique que la requête accepte du JSON.*

-*@PathParam*(value="name") Type var : Récupère une valeur depuis l'URL.

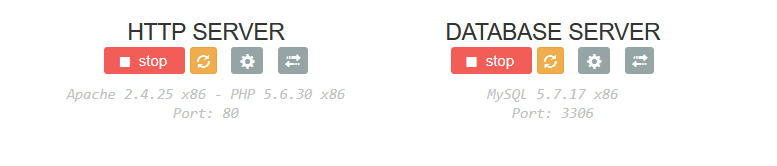
-*@FormParam*(value="name") Type var: Récupère une valeur depuis le corps de la requête.

1. Exposé le serviceRest 🡪 créé un context d'application.

**C/ Création du du class « WebApp » :**

1. Créé un class nommé "**WebApp**" extends **Application**.
2. Ajouter un annotation **@ApplicationPath("/")** Un context d'application que va permétre un client d'accès a service Rest.

**E/ Lancement du database :**



**F/ Lancement du serveur :**

lancé le service Pour obtenir le URL d'accès :

1. Aller dans **Servers**.

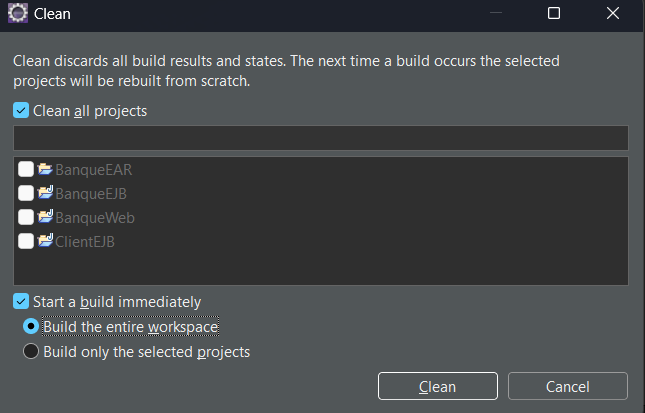
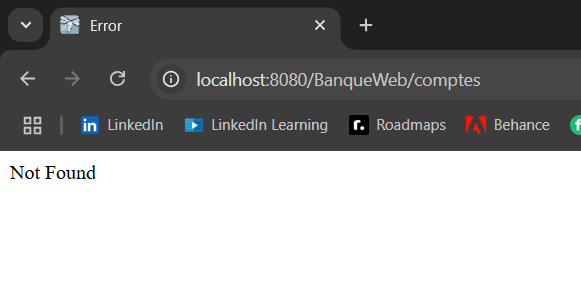
2. Sélectionner **WildFly**.

3. Cliquer sur **Start**.

5. Copier l’URL suivante dans un navigateur :

[**http://localhost:8080/BanqueWeb/comptes**](http://localhost:8080/BanqueWeb/comptes)

⇨**Erreur:** ⇨**Solution:**

**** -Project🡪 Clean.. 🡪 Clean all projects 🡪 Clean .

-Project 🡪 Build Project.

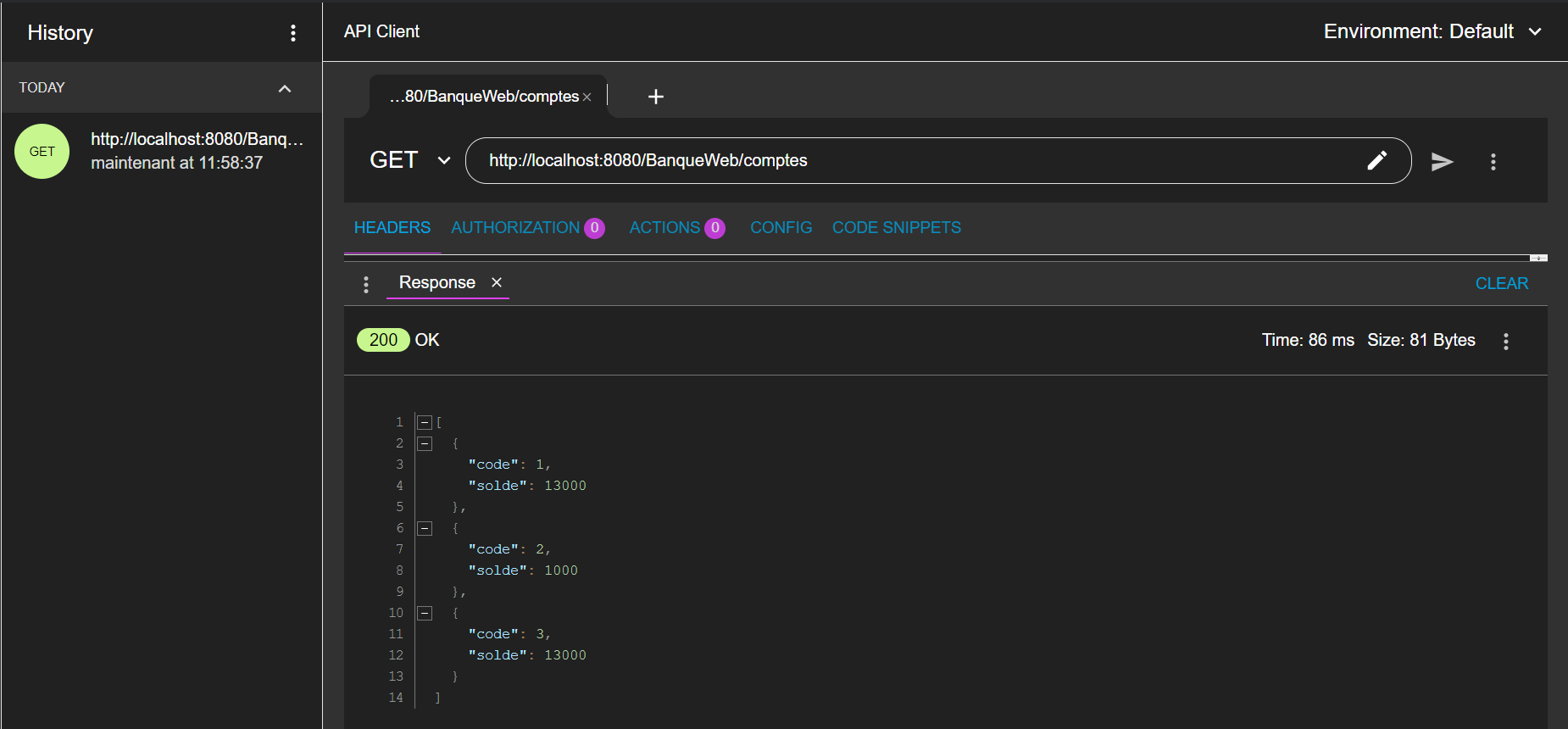
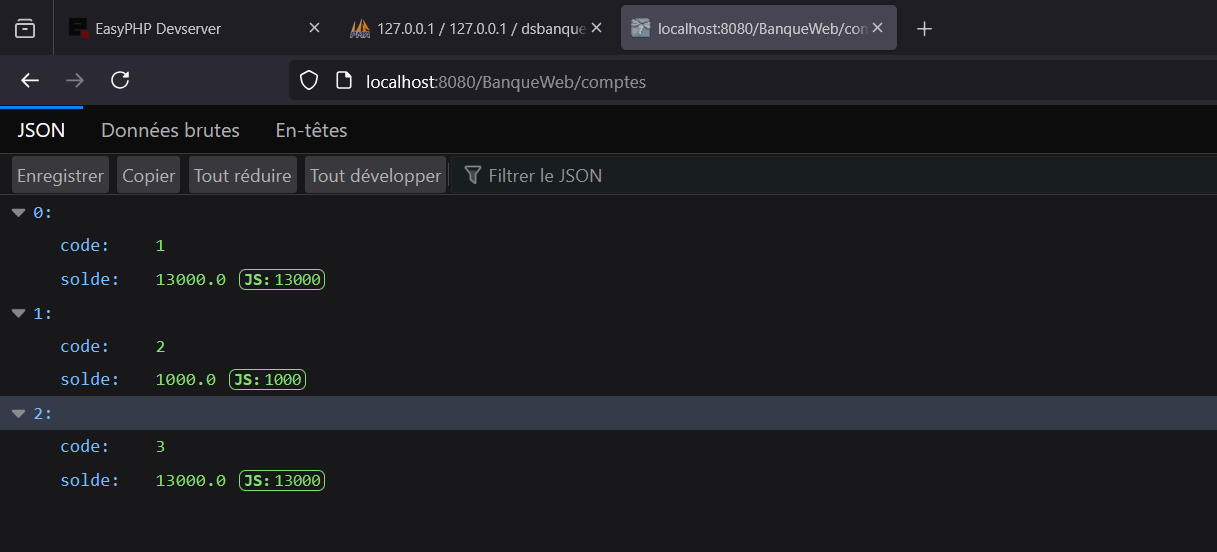
**F/ Test avec ARC :**

1. File🡪New🡪Http request🡪ajouter l’URL “http://localhost:8080/BanqueWeb/comptes”

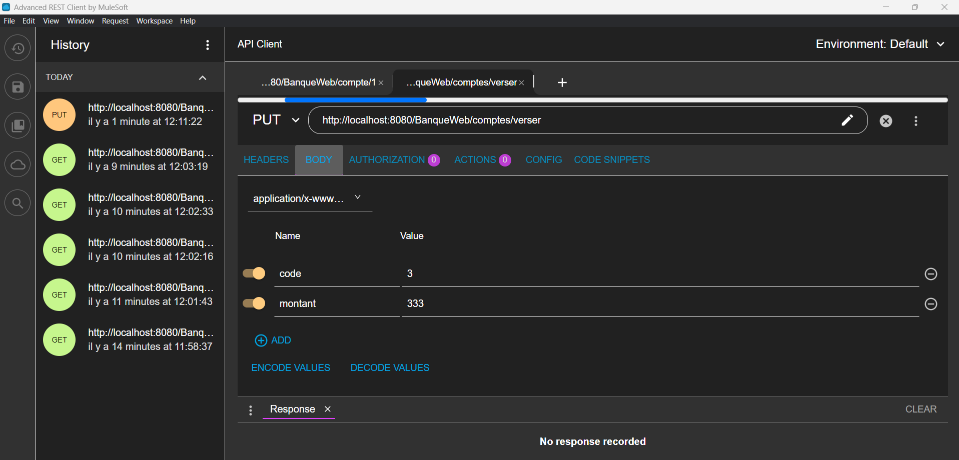
**G/ Lancement du client:**

**Tester les methodes autre fois dans ARC.**

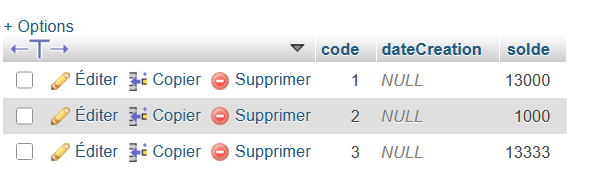
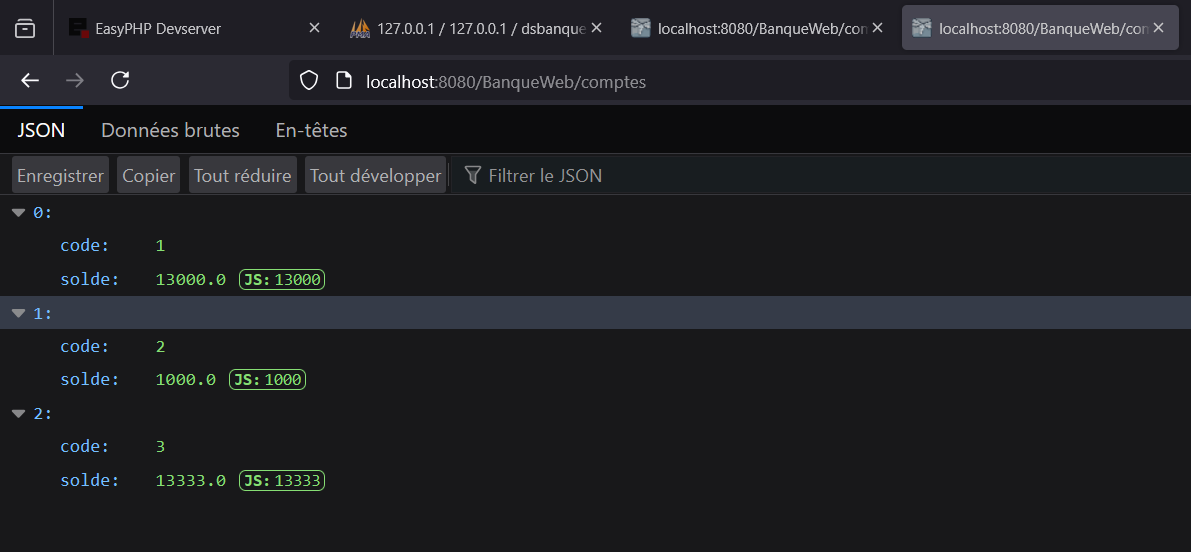
1/listComptes() :

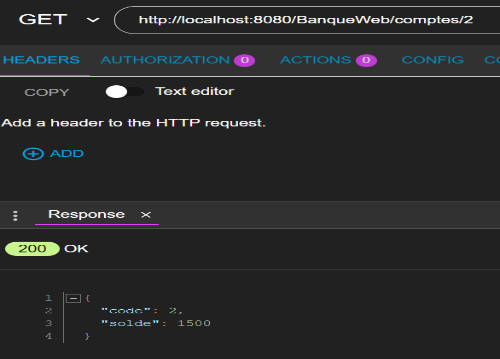


2/ verser(code, mt) :



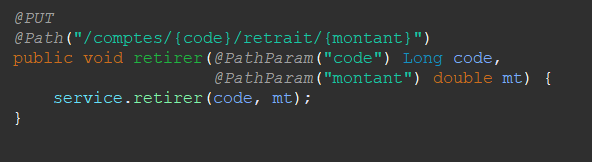
🡪

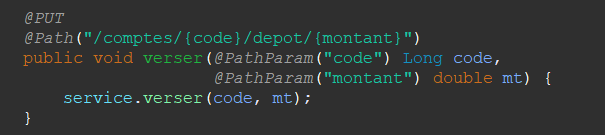


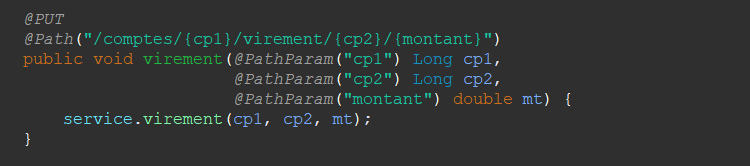
****3/getCompte(code) :

**Remarque :** Le problème Dans la mathod " getCompt " que va return un variable complex " objet de type compte , http n'eccept pas un Type complex Le solutions ici utilise l'annotation **@*Consumes***.

**Optimiser le code selon les principes RESTful :**

****1/retirer(code,mt):

****2/verser(code,mt) :

****

3/virement(cp1,cp2,mt) :