Projet Application WEB Orientée Services

Réalisé par : Ruben ALBERTINI , Samar RHILANE

Encadré par : M. Mouloud MENCEUR

1. Objectif

Le projet consiste à créer deux micro-services. Le micro-services :

CompteBancaire se caractérise par {"IBAN" : "FR7630004000031234567890143", "type de compte" : "Compte courant", " interet" : 0.0, "frais de tenue de compte" : "gratuit" "}

OperationBancaire se caractérise par {"id operation" : 12345, "type operation" : "VIREMENT", "IBAN source" : "FR7630004000031234567890143","IBAN destination" : "USD", "montant" : 1000.0, "date" : "2021-12-30"}

Les deux microservices communiquent par l'envoie d'une requête HTTP

Le client utilisé est POSTMAN où nous pouvons réaliser des requêtes HTTP (POST, GET, PUT, DELETE)

Key words : JAVA, SpringBoot, Docker, Junit, H2, REST API, Kubernete

Installation et Utilisation

```
$ git clone --single-branch --branch ruben
https://github.com/samarrhilane/MicroService1_CompteBancaire
```

```
$ git clone --single-branch --branch ruben
https://github.com/samarrhilane/MicroService2_OperationBancaire
```

```
mvn clean install package
```

Set up the Spring Boot Application

```
java -jar target/ CompteBancaire-0.0.1-SNAPSHOT.jar
```

```
java -jar target/ OperationBancaire -0.0.1-SNAPSHOT.jar
```

Set up Docker

```
docker build -t CompteBancaire-0.0.1-SNAPSHOT.jar
```

```
docker build -t OperationBancaire -0.0.1-SNAPSHOT.jar
```

Set up Kubernetes

```
CompteBancaire>minikube start --driver=docker
```

```
 Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.
```

```
CompteBancaire>kubectl cluster-info
```

```
Kubernetes control plane is running at https://127.0.0.1:57577
CoreDNS is running at https://127.0.0.1:57577/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

```
CompteBancaire>kubectl run comptebancaire --image=comptebancaire:latest \ --port=8080 --image-p
```

```
ull-policy Never
pod/comptebancaire created
```

```
\CompteBancaire>kubectl create deployment comptebancaire --image=k8s.gcr.io/echoserver:
```

```
deployment.apps/comptebancaire created
```

```
\CompteBancaire>kubectl expose deployment comptebancaire --type=NodePort --port=8080
```

```
service/comptebancaire exposed
```

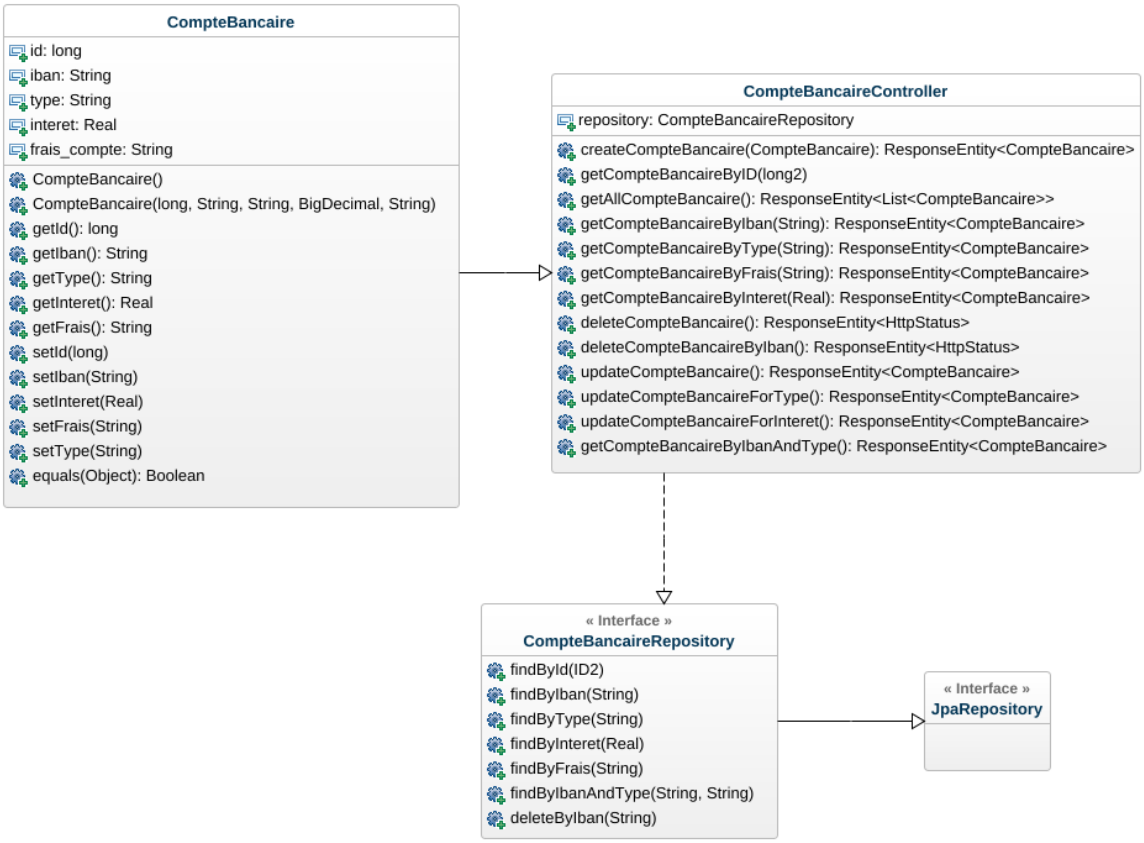
```
CompteBancaire>kubectl get services comptebancaire
```

```
NAME             TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
comptebancaire   NodePort   10.100.217.129   <none>        8080:30279/TCP   4m
```

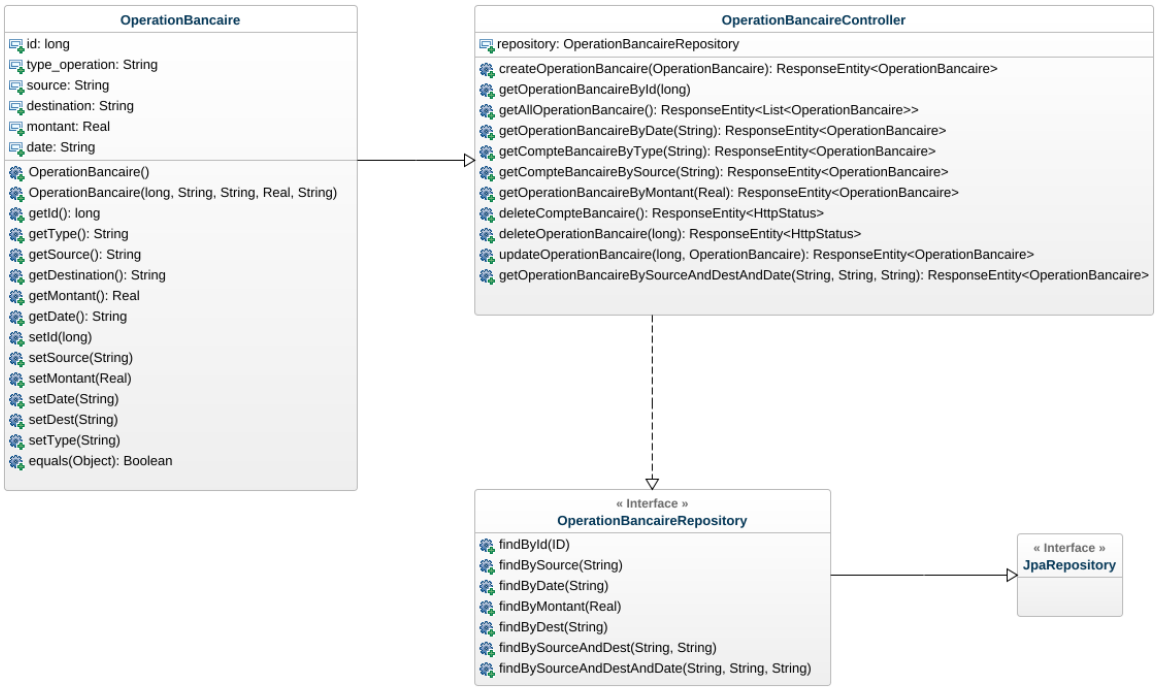
```
\CompteBancaire>minikube service comptebancaire
```

```
|-----------|----------------|-------------|---------------------------|
| NAMESPACE |      NAME      | TARGET PORT |            URL            |
|-----------|----------------|-------------|---------------------------|
| default   | comptebancaire |        8080 | http://192.168.49.2:30279 |
|-----------|----------------|-------------|---------------------------|
```

2. CompteBancaire



3. OperationBancaire



4. REST API

GET

Récupérer tous les comptes bancaires

```
 "http://localhost:8080/COMPTEBANCAIRE"
```

Récupérer une opération bancaire

```
"http://localhost:8088/operation-bancaire"
```

Récupérer le compte bancaire en spécifiant l'IBAN

```
"http://localhost:8080/COMPTEBANCAIRE/iban/FR7630004000031234567891011"
```

Récupérer le compte bancaire en spécifiant le type du compte

```
"http://localhost:8080/COMPTEBANCAIRE/type/Compte_courant"
```

Récupérer le compte bancaire en spécifiant une valeur d'intérêt du compte

```
"http://localhost:8080/COMPTEBANCAIRE/interet/10"
```

Récupérer une opération bancaire en spécifiant le montant

```
"http://localhost:8088/operation-bancaire/montant/1000"
```

Récupérer une opération bancaire en spécifiant une date

```
"http://localhost:8088/operation-bancaire/date/2022-06-21"
```

DELETE

Supprimer un compte bancaire en spécifiant l'identifiant

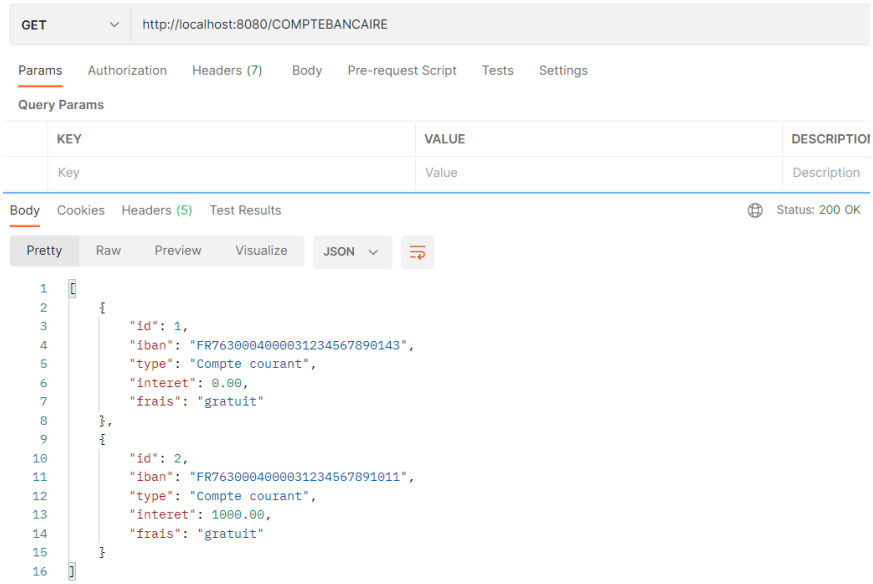
```
"http://localhost:8080/COMPTEBANCAIRE/id/1"
```

Supprimer un compte bancaire en spécifiant l'IBAN

```
"http://localhost:8080/COMPTEBANCAIRE/iban/FR7630004000031234567891011"
```

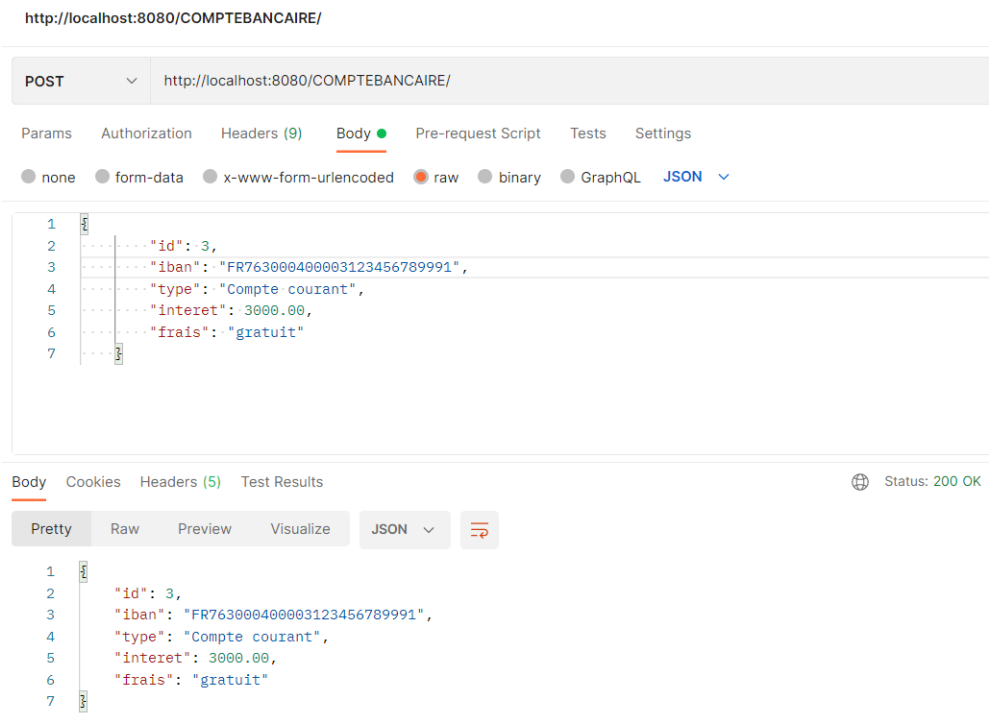
Exemple de GET :

Afin d'afficher tous les comptes bancaires



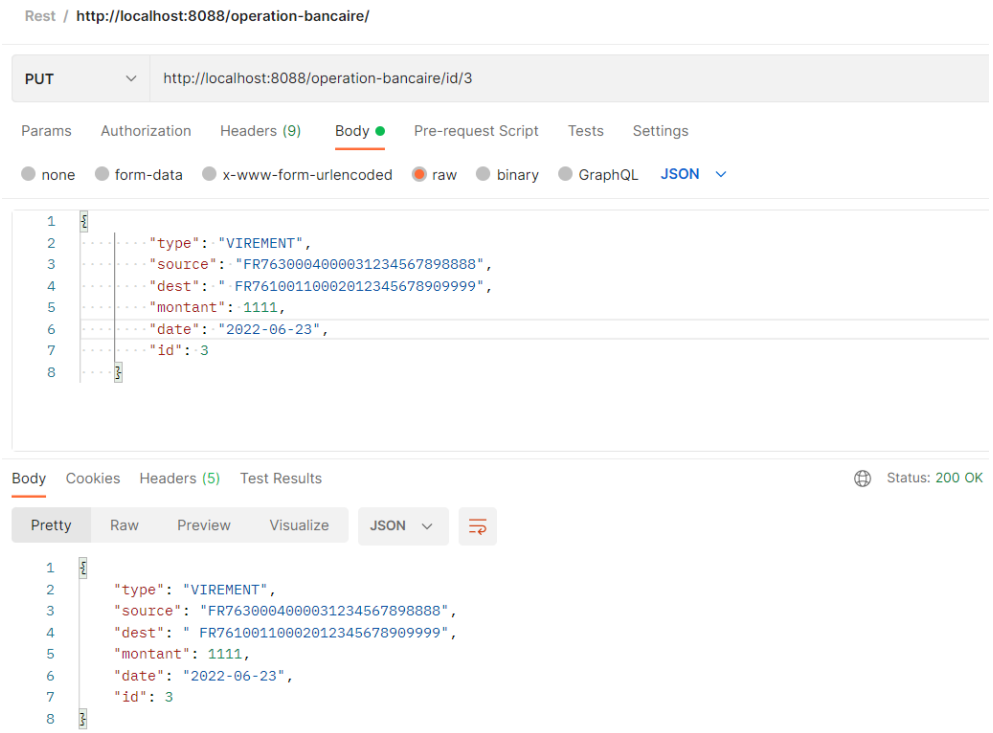
Exemple de POST :

Afin de rajouter un nouveau compte bancaire



Exemple de PUT :

Afin de modifier une opération bancaire prenant l'identifiant 3 (modification du montant)



Exemple GET : operation **Exemple DELETE : operation**