

DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Compte rendu de contrôle

Filière :
« Génie du Logiciel et des Systèmes Informatiques Distribués »
GLSID

Gestion des transferts

Réalisé par :

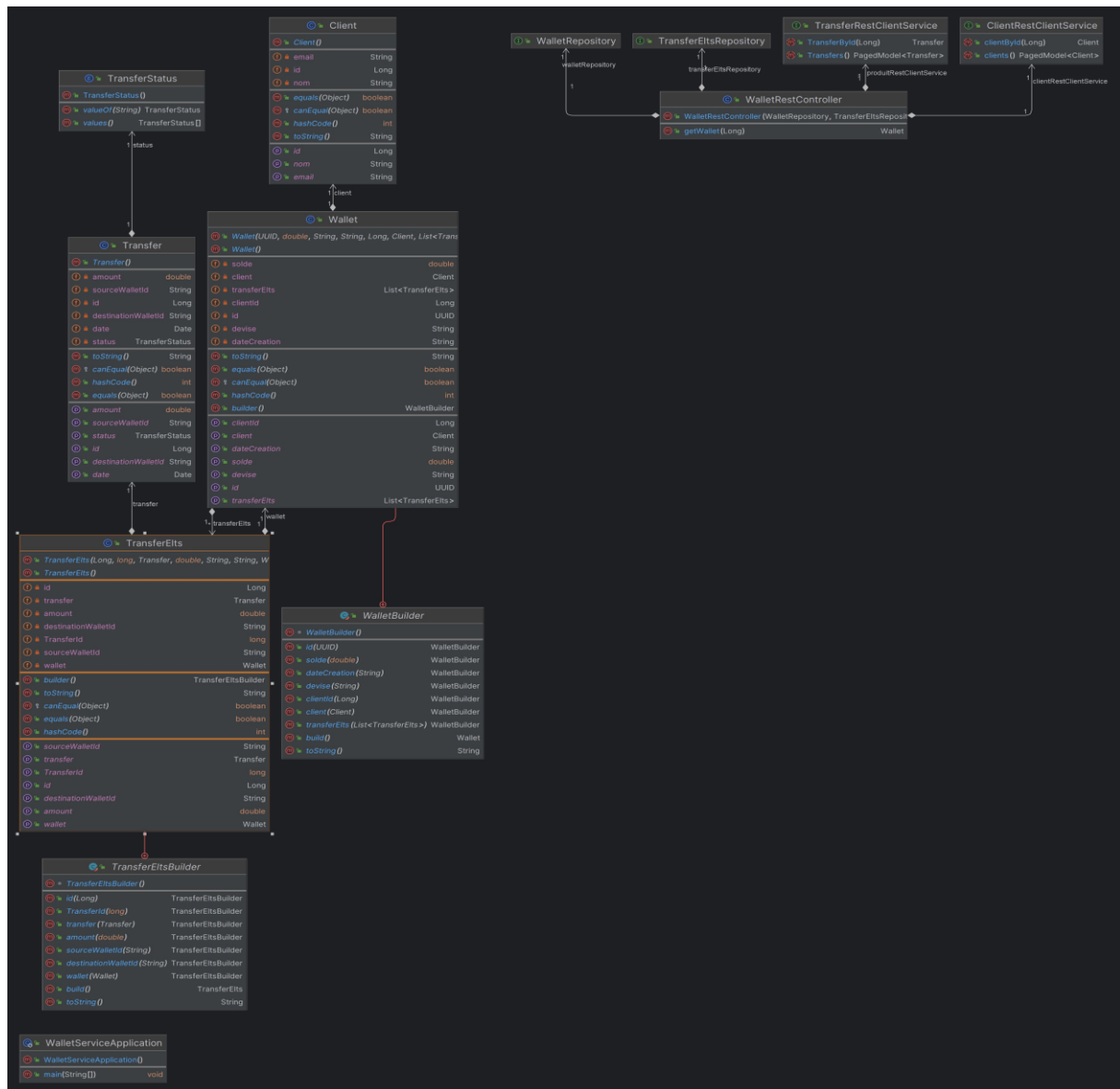
Najwa ZRAIDI

Professeur :

M. Mohamed YOUSSEFI

Année Universitaire : 2023-2024

1- Diagramme de classe :



Mon projet :

```

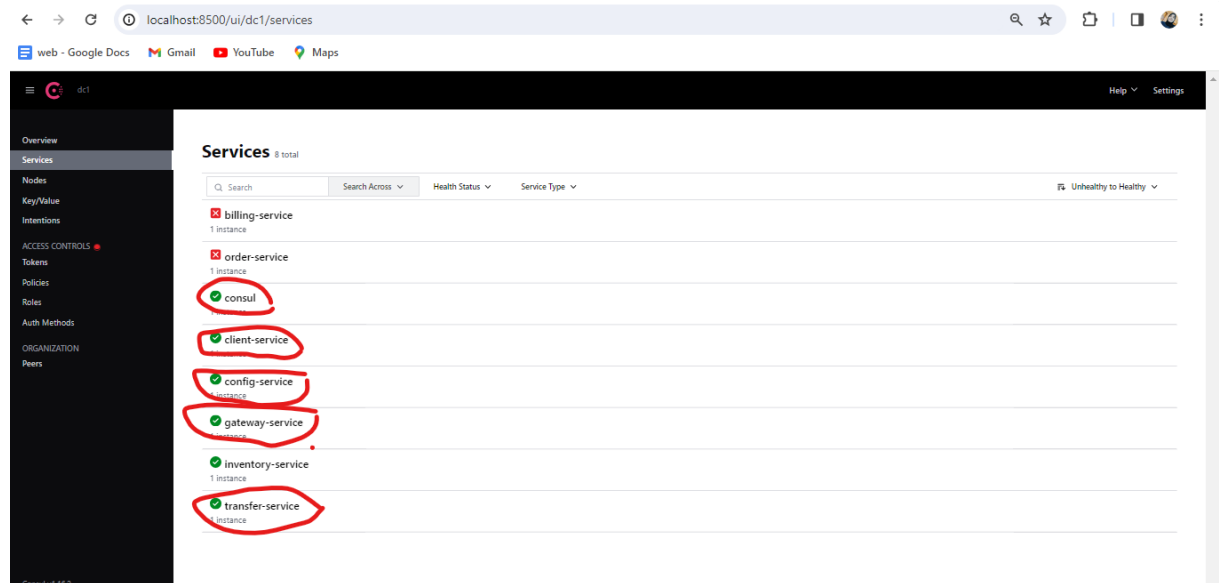
Project
├── client-service
├── config-service
├── e-transfertWallet
├── gateway-service
├── transfer-service
├── wallet-service
├── External Libraries
└── Scratches and Consoles

WalletServiceApplication.java
1  import org.springframework.boot.SpringApplication;
2  import org.springframework.boot.autoconfigure.SpringBootApplication;
3  import org.springframework.cloud.openfeign.EnableFeignClients;
4
5  @SpringBootApplication
6  @EnableFeignClients
7  public class WalletServiceApplication {
8
9      public static void main(String[] args) {
10         SpringApplication.run(WalletServiceApplication.class, args);
11     }
12 }

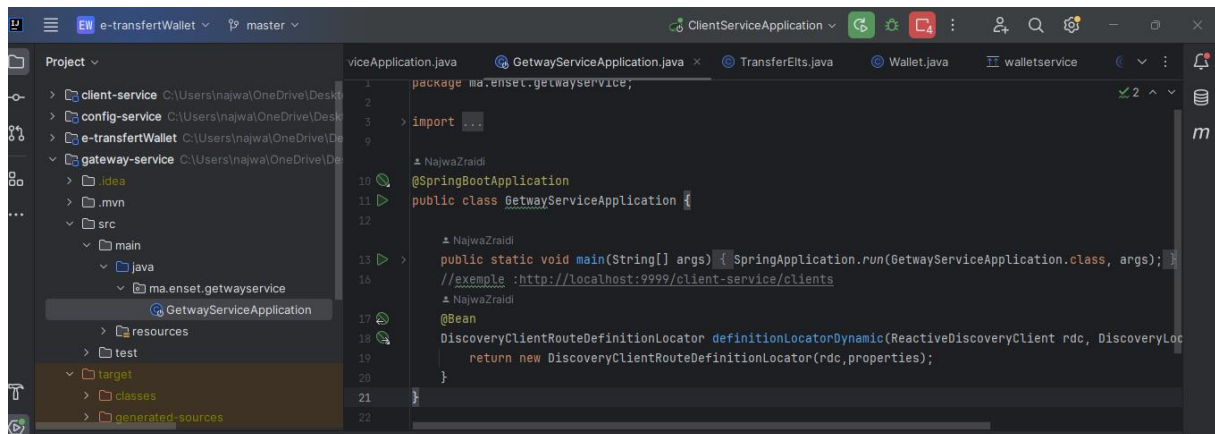
```

⇒ Démarrage de consul avec la commande :

`consul agent -server -bootstrap-expect=1 -data-dir=consul-data -ui -bind=192.168.8.173`



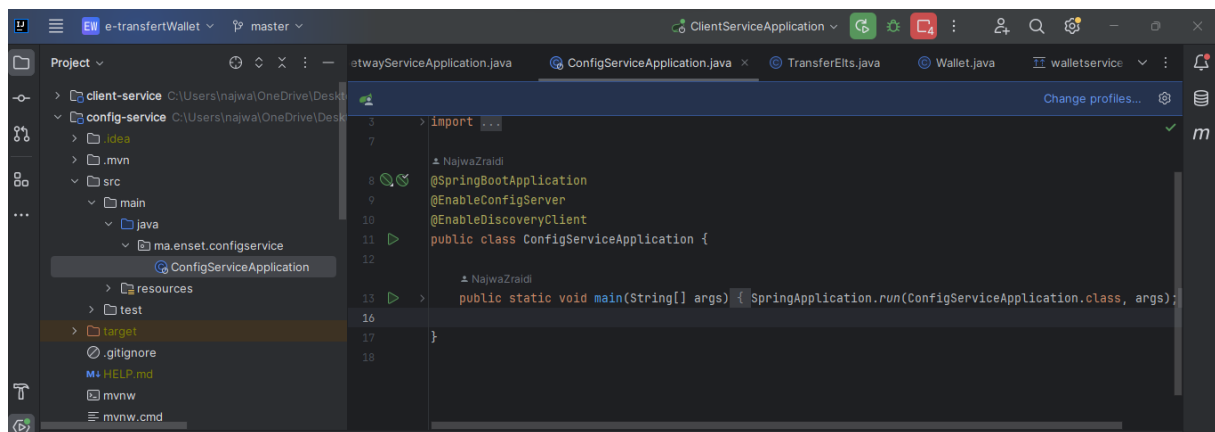
⇒ Micro service Gateway :



```
localhost:9999/transfer-service/transfers

{
  "_embedded": {
    "transfers": [
      {
        "date": null,
        "amount": 100,
        "sourceWalletId": "sourceWallet1",
        "destinationWalletId": "destinationWallet1",
        "status": null,
        "_links": {
          "self": {
            "href": "http://Najwa-ZRAIDI:8082/transfers/1"
          },
          "transfer": {
            "href": "http://Najwa-ZRAIDI:8082/transfers/1"
          }
        }
      },
      {
        "date": null,
        "amount": 200,
        "sourceWalletId": "sourceWallet2",
        "destinationWalletId": "destinationWallet2",
        "status": null,
        "_links": {
          "self": {
            "href": "http://Najwa-ZRAIDI:8082/transfers/2"
          },
          "transfer": {
            "href": "http://Najwa-ZRAIDI:8082/transfers/2"
          }
        }
      }
    ]
  }
}
```

⇒ Micro service config :



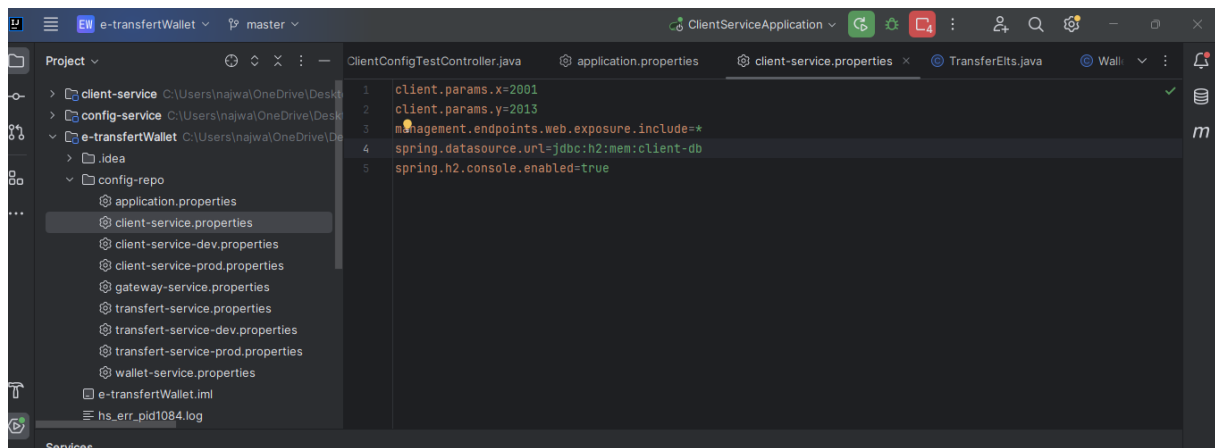
```
atwayServiceApplication.java  ConfigServiceApplication.java  TransferElts.java  Wallet.java  walletservice...

Project
├── client-service
├── config-service
│   ├── idea
│   ├── .mvn
│   ├── src
│   │   ├── main
│   │   │   ├── java
│   │   │   │   └── ma.enset.configservice
│   │   │       └── ConfigServiceApplication
│   │   ├── resources
│   │   ├── test
│   │   ├── target
│   │   ├── .gitignore
│   │   ├── HELP.md
│   │   ├── mvnw
│   │   └── mvnw.cmd
├── .gitignore
├── HELP.md
├── mvnw
└── mvnw.cmd

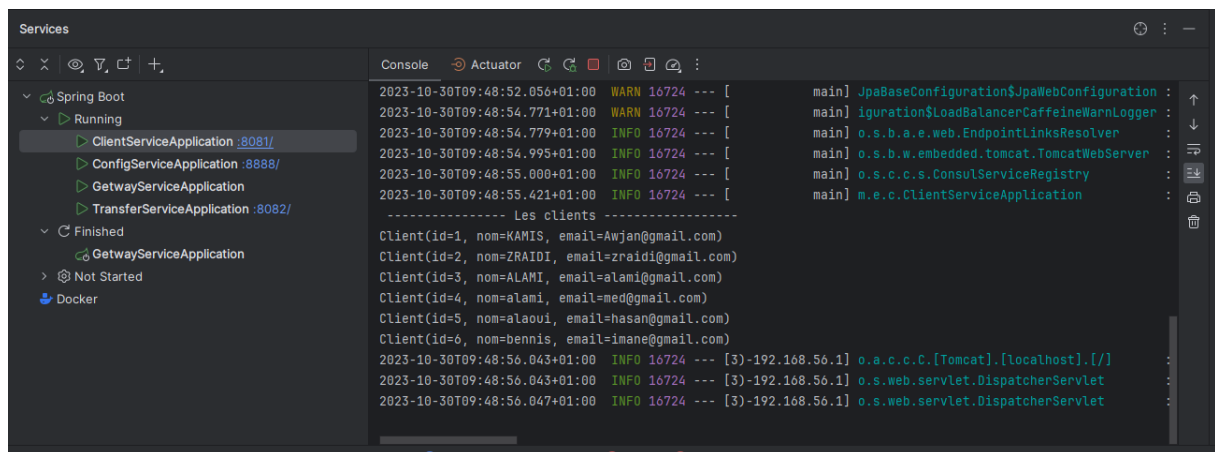
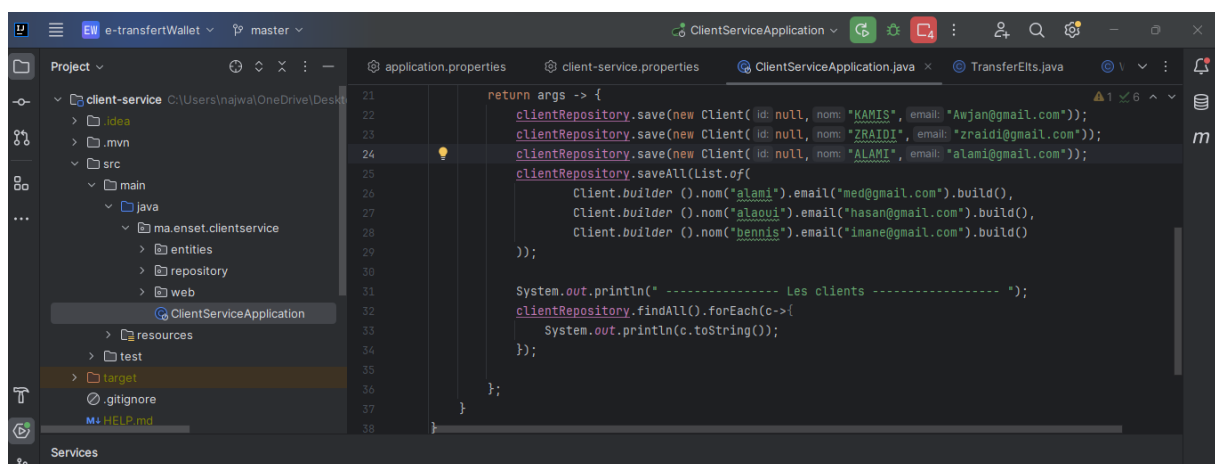
ConfigServiceApplication.java
3  > import ...
7
8  @SpringBootApplication
9  @EnableConfigServer
10 @EnableDiscoveryClient
11 public class ConfigServiceApplication {
12
13     >
14     public static void main(String[] args) { SpringApplication.run(ConfigServiceApplication.class, args); }
15
16
17
18
```

```
localhost:9999/client-service/params

{
  "p1": "435",
  "p2": "12",
  "y": "2013",
  "x": "2001"
}
```



⇒ Micro service client



localhost:8081/h2-console/login.do?jsessionId=82873c975a228f84ee8f68d2d7743a5e

web - Google Docs Gmail YouTube Maps

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:Client-db

- CLIENT
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM CLIENT CLIENT

SELECT * FROM CLIENT CLIENT;

ID	EMAIL	NOM
1	Awjan@gmail.com	KAMIS
2	zraidi@gmail.com	ZRAIDI
3	alami@gmail.com	ALAMI
4	med@gmail.com	alami
5	hasan@gmail.com	alaoui
6	imane@gmail.com	bennis

(6 rows, 2 ms)

Edit

⇒ Micro service transfer

e-transferWallet master

Project

- transfer-service
 - idea
 - .mvn
 - src
 - main
 - java
 - ma.enset.transferservice
 - entities
 - enums
 - repositories
 - TransferServiceApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
- wallet-service
- External Libraries

application.properties

client-service.properties

ClientServiceApplication.java

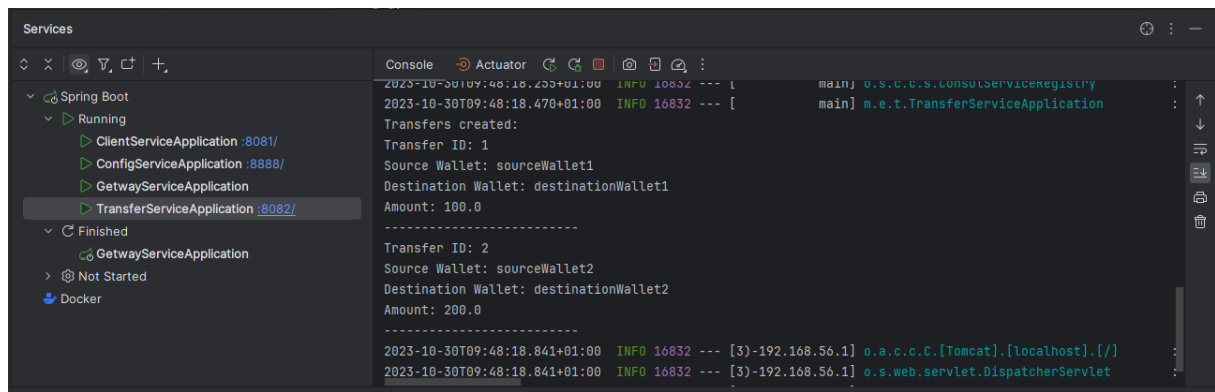
TransferServiceApplication.java

```
transfer1.setSourceWalletId("sourceWallet1");
transfer1.setDestinationWalletId("destinationWallet1");
transfer1.setAmount(100.0);
transferRepository.save(transfer1);

Transfer transfer2 = new Transfer();
transfer2.setSourceWalletId("sourceWallet2");
transfer2.setDestinationWalletId("destinationWallet2");
transfer2.setAmount(200.0);
transferRepository.save(transfer2);

// Affichage les transferts
System.out.println("Transfers created:");

Iterable<Transfer> transfers = transferRepository.findAll();
for (Transfer transfer : transfers) {
    System.out.println("Transfer ID: " + transfer.getId());
    System.out.println("Source Wallet: " + transfer.getSourceWalletId());
    System.out.println("Destination Wallet: " + transfer.getDestinationWalletId());
    System.out.println("Amount: " + transfer.getAmount());
    System.out.println("-----");
}
```



⇒ Micro service wallet

