

Département Mathématiques et Informatique
« Ingénierie Informatique : Big Data et Cloud Computing »
II-BDCC3

Examen Systèmes distribués

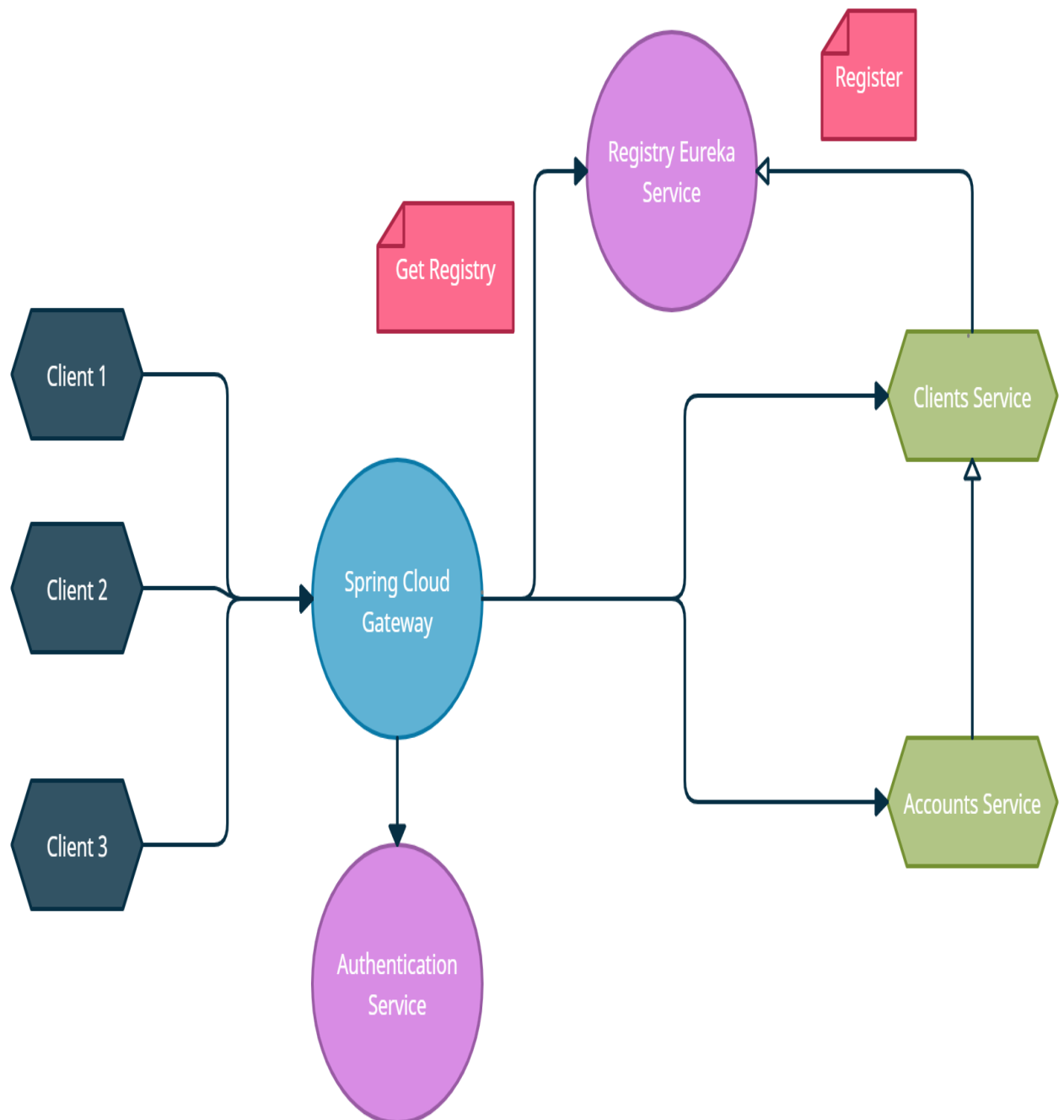
Réalisé par :

Hamza Gueddi

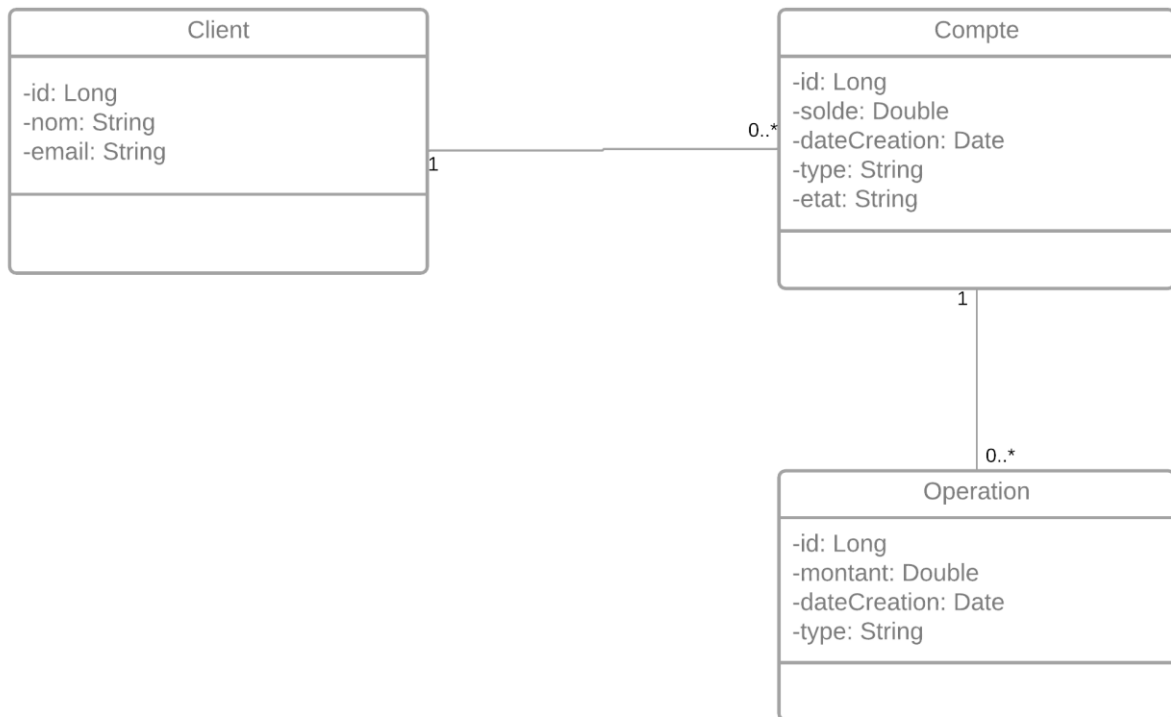
Année Universitaire : 2020-2021

A.Conception

1. Architecture du projet



2. Diagramme de classes



B. Implémentation

1. Implémentation du micro-service de gestion des clients

Selected Dependencies		
Developer Tools		
Spring Boot DevTools		×
Lombok		×
Web		
Spring Web		×
Rest Repositories		×
SQL		
Spring Data JPA		×
H2 Database		×
Spring Cloud Discovery		
Eureka Discovery Client		×

```
server.port=8081
spring.application.name=client-service
spring.datasource.url=jdbc:h2:mem:client-db
spring.cloud.discovery.enabled=true
```

Requête GET



The screenshot shows a web browser window with the address bar displaying `localhost:8081/clients`. The browser's developer tools are open, showing the JSON response to the GET request. The response is a JSON array containing three client objects. Each object has an `id`, `nom`, `email`, and a `_links` object. The `_links` object contains a `self` link and a `client` link, both pointing to the same URL as the request.

```
{
  "_embedded": {
    "clients": [
      {
        "id": 1,
        "nom": "hamza",
        "email": "hamza@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/1"
          },
          "client": {
            "href": "http://localhost:8081/clients/1"
          }
        }
      },
      {
        "id": 2,
        "nom": "john",
        "email": "john@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/2"
          },
          "client": {
            "href": "http://localhost:8081/clients/2"
          }
        }
      },
      {
        "id": 3,
        "nom": "jane",
        "email": "jane@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/3"
          },
          "client": {
            "href": "http://localhost:8081/clients/3"
          }
        }
      }
    ]
  },
  "_links": {
    "self": {
      "href": "http://localhost:8081/clients"
    }
  }
}
```

localhost:8081/h2-console/login.do?jsessionId=745e74b6fbf97829ffa6188eaf6192c4

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

jdbc:h2:mem:client-db

- CLIENT
- INFORMATION_SCHEMA
- Sequences
- Users
- H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM CLIENT

SELECT * FROM CLIENT;

ID	EMAIL	NOM
1	hamza@gmail.com	hamza
2	john@gmail.com	john
3	jane@gmail.com	jane

(3 rows, 11 ms)

Edit

2. Implémentation du Micro-service pour la gestion des comptes et des opérations

Selected Dependencies

Developer Tools

- Spring Boot DevTools
- Lombok

Web

- Spring Web
- Rest Repositories

SQL

- Spring Data JPA
- H2 Database

Spring Cloud Discovery

- Eureka Discovery Client

Spring Cloud Routing

- OpenFeign

```
server.port=8082
spring.application.name=accounts-service
spring.datasource.url=jdbc:h2:mem:accounts-db
spring.cloud.discovery.enabled=true
```

Requête POST

POST <http://localhost:8082/comptes> Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ JSON Beautify

```
1 {
2   "id": null,
3   "solde": 250000,
4   "type": "EPARGNE",
5   "etat": "ACTIVE",
6   "idClient": 1,
7   "dateCreation": "11-02-2021 22:20:00"
8 }
```

Body Cookies Headers (9) Test Results Status: 201 Created Time: 140 ms Size: 733 B Save Response

[←](#) [→](#) [↺](#) localhost:8082

```
{
  "_links": {
    "comptes": {
      "href": "http://localhost:8082/comptes{?page,size,sort}",
      "templated": true
    },
    "operations": {
      "href": "http://localhost:8082/operations{?page,size,sort}",
      "templated": true
    },
    "profile": {
      "href": "http://localhost:8082/profile"
    }
  }
}
```

3. Mise en place des services techniques

a. Spring Cloud Gateway

Selected Dependencies

Ops
Spring Boot Actuator ×

Spring Cloud Discovery
Eureka Discovery Client ×

Spring Cloud Routing
Gateway ×

```
server.port=8888  
spring.application.name=gateway-service  
spring.cloud.discovery.enabled=true
```

b. Eureka Discovery Service

Selected Dependencies

Spring Cloud Discovery
Eureka Server ×

```
server.port=8761  
eureka.client.fetch-registry=false  
eureka.client.register-with-eureka=false
```

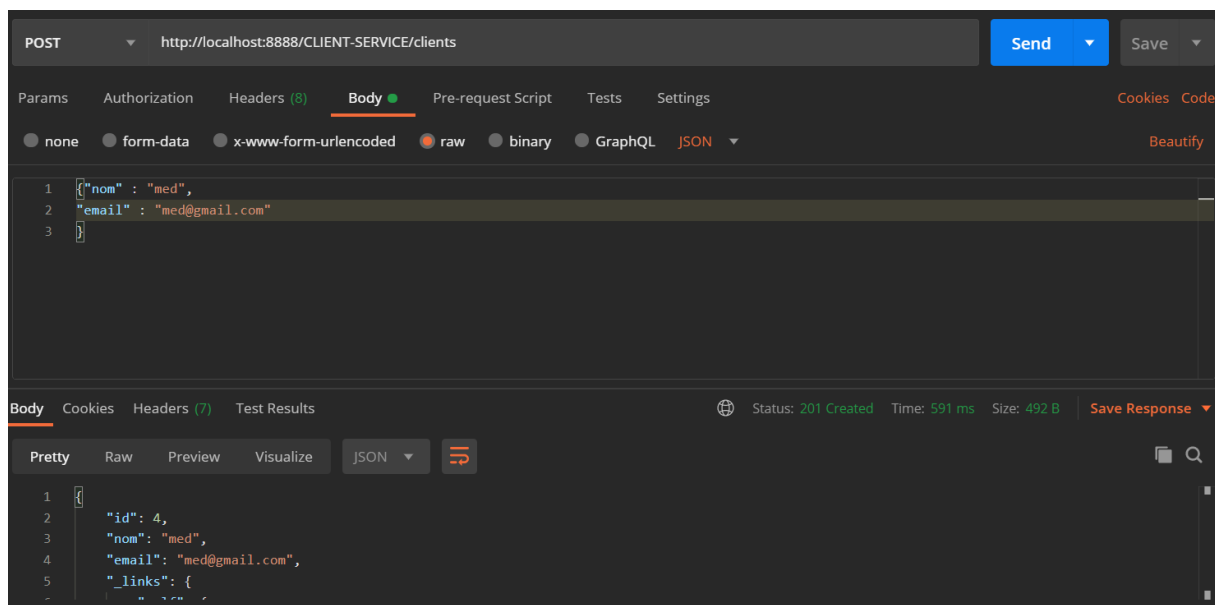
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ACCOUNTS-SERVICE	n/a (1)	(1)	UP (1) - MSI:accounts-service:8082
CLIENT-SERVICE	n/a (1)	(1)	UP (1) - MSI:client-service:8081
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - MSI:gateway-service:8888

4. Test de l'application à partir du gateway

```
localhost:8888/CLIENT-SERVICE/clients

{
  "_embedded": {
    "clients": [
      {
        "id": 1,
        "nom": "hamza",
        "email": "hamza@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/1"
          },
          "client": {
            "href": "http://localhost:8081/clients/1"
          }
        }
      },
      {
        "id": 2,
        "nom": "john",
        "email": "john@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/2"
          },
          "client": {
            "href": "http://localhost:8081/clients/2"
          }
        }
      },
      {
        "id": 3,
        "nom": "jane",
        "email": "jane@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/clients/3"
          },
          "client": {
            "href": "http://localhost:8081/clients/3"
          }
        }
      }
    ]
  }
}
```



5. Broker Kafka

```
C:\kafka_2.13-2.7.0>start bin\windows\zookeeper-server-start.bat config\zookeeper.properties
```

```
C:\kafka_2.13-2.7.0>start bin\windows\kafka-server-start.bat config\server.properties
```

```
CompteOperation(compteid=1, montant=5000.0, date=null, type=CREDIT)
250000.0
operation Montation-->5000.0
2021-02-13 00:25:58.918 INFO 19784 --- [nio-8082-exec-1] e.b.a.service.KafkaProducerService : Operation id: 2, typeDEBIT
2021-02-13 00:25:58.939 INFO 19784 --- [nio-8082-exec-1] o.a.k.clients.producer.ProducerConfig : ProducerConfig values:
acks = 1
batch.size = 16384
bootstrap.servers = [127.0.0.1:9092]
buffer.memory = 33554432
client.dns.lookup = use_all_dns_ips
client.id = producer-1
compression.type = none
connections.max.idle.ms = 540000
delivery.timeout.ms = 120000
enable.idempotence = false
interceptor.classes = []
internal.auto.downgrade.txn.commit = true
key.serializer = class org.apache.kafka.common.serialization.StringSerializer
linger.ms = 0
max.block.ms = 60000
max.in.flight.requests.per.connection = 5
max.request.size = 1048576
metadata.max.age.ms = 300000
metadata.max.idle.ms = 300000
```

```
[2021-02-12 21:15:50,156] WARN [Consumer clientId=consumer-console-consumer-38043-1, groupId=console-consumer-38043] Error while fetching metadata with correlation id 2 : {operationTopic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
[2021-02-12 21:15:50,287] WARN [Consumer clientId=consumer-console-consumer-38043-1, groupId=console-consumer-38043] Error while fetching metadata with correlation id 6 : {operationTopic=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.NetworkClient)
{"id":1,"montant":350.0,"dateOperation":1613163160557,"type":"CREDIT"}
```

```
{"id":2,"date":"12-02-2021 22:42:30","montant":2000.0,"type":"DEBIT"}
{"id":3,"date":"12-02-2021 22:43:36","montant":2000.0,"type":"DEBIT"}
{"id":4,"date":"12-02-2021 22:43:49","montant":1500.0,"type":"CREDIT"}
{"id":5,"date":"12-02-2021 22:43:52","montant":1500.0,"type":"VIREMENT_CREDIT"}
{"id":6,"date":"12-02-2021 22:43:57","montant":1500.0,"type":"VIREMENT_DEBIT"}
```

7. Sécurité JWT

Selected Dependencies

Developer Tools

Spring Boot DevTools



Lombok



Web

Spring Web



Rest Repositories



Security

Spring Security



SQL

Spring Data JPA



H2 Database



☒ Auto commit Max rows: 1000 SQL statement:

jdbc:h2:mem:users-db

- APP_ROLE
- APP_USER
- APP_USER_APP_ROLES
- INFORMATION_SCHEMA
- Sequences
- Users

 H2 1.4.200 (2019-10-14)

```
SELECT * FROM APP_ROLE;
```

```
SELECT * FROM APP_ROLE;
```

ID	ROLE_NAME
1	USER
2	ADMIN
3	CLIENT_MANAGER
4	ACCOUNTS_MANAGER
5	OPERATIONS_MANAGER

jdbc:h2:mem:users-db
+ APP_ROLE
+ APP_USER
+ APP_USER_APP_ROLES
+ INFORMATION_SCHEMA
+ Sequences
+ Users
i H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM APP_USER|

SELECT * FROM APP_USER;

ID	PASSWORD	USERNAME
1	\$2a\$10\$rrX7N6hGi6GQ.VfWeBHoA.jhrchBIXEdBgDApRaW3S3qnbVgCrW	hamza
2	\$2a\$10\$QoCr.85mnimyCBWTmgM03e3d7du.fFIEsIfh4.FUIGitj0iHY9xiq	admin
3	\$2a\$10\$WEtKeN5MGFmYesDIHBMamuGmRTzx2bS4Ay5B8LecVHsqTE6JQazy.	john
4	\$2a\$10\$wB6SAIb7I23UM8f2yJSzhOQXqwbqThyOqpA2z2NamG7OqS.E3g0Jy	jane

jdbc:h2:mem:users-db
+ APP_ROLE
+ APP_USER
+ APP_USER_APP_ROLES
+ INFORMATION_SCHEMA
+ Sequences
+ Users
i H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM APP_USER_APP_ROLES|

SELECT * FROM APP_USER_APP_ROLES;

APP_USER_ID	APP_ROLES_ID
2	1
2	2
3	3
4	4
1	1
1	5

```
localhost:8080/appUsers
{
  "_embedded": {
    "appUsers": [
      {
        "username": "hamza",
        "_links": {
          "self": {
            "href": "http://localhost:8080/appUsers/1"
          },
          "appUser": {
            "href": "http://localhost:8080/appUsers/1"
          },
          "appRoles": {
            "href": "http://localhost:8080/appUsers/1/appRoles"
          }
        }
      },
      {
        "username": "admin",
        "_links": {
          "self": {
            "href": "http://localhost:8080/appUsers/2"
          },
          "appUser": {
            "href": "http://localhost:8080/appUsers/2"
          },
          "appRoles": {
            "href": "http://localhost:8080/appUsers/2/appRoles"
          }
        }
      },
      {
        "username": "john",
        "_links": {
          "self": {
            "href": "http://localhost:8080/appUsers/3"
          },
          "appUser": {
            "href": "http://localhost:8080/appUsers/3"
          },
          "appRoles": {
            "href": "http://localhost:8080/appUsers/3/appRoles"
          }
        }
      }
    ]
  }
}
```

Lien GITHUB

<https://github.com/gueddihamza/spring-bank>