



## Département Mathématiques et Informatique : II-BDCC 3

### Travail Pratique 2: Systèmes Distribués

---

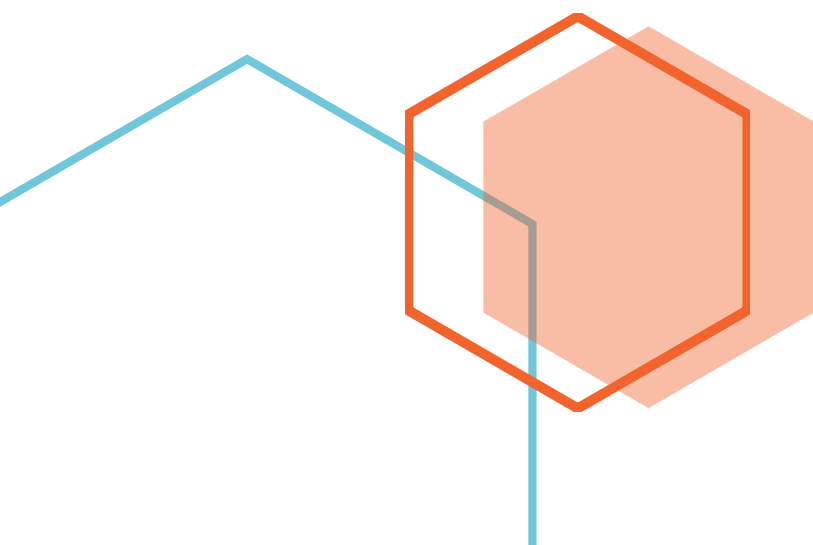
#### Activité pratique N° 2 : Architectures Micro-services avec Spring

-Réalisé par :

**Madani Cherif Oumaima**

-Diriger par :

**Monsieur Mohamed YOUSSEFI**



# Introduction

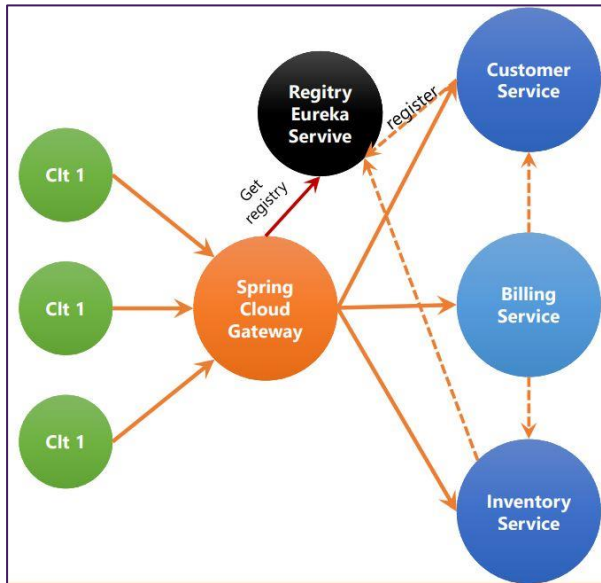
Dans ce TP, nous allons explorer mettre en œuvre plusieurs micro services et services connexes en utilisant des technologies telles que Spring. Les étapes comprennent la création de micro services pour la gestion des clients, le déploiement d'API RESTful à l'aide de Spring Data Rest, la mise en place d'une passerelle avec Spring Cloud Gateway, la création d'un annuaire de services avec Netflix Eureka Server, et la communication entre les micro services à l'aide d'Open Feign. Enfin, l'activité se termine par la création d'un client Angular pour afficher les données des micro services.

## **Sommaire :**

Introduction .....	1
Parties pratiques: Partie 0: Architecture de l'activité pratique .....	3
Partie 1: MICRO-SERVICE CUSTOMER-SERVICE.....	3
Partie 2: MICRO-SERVICE INVENTORY-SERVICE.....	3
Partie 3: GATEWAY SERVICE.....	4
Partie 4: EUREKA SERVER et tests .....	5
Partie 5: BILLING-SERVICE .....	6
Conclusion .....	8

## Parties pratiques:

### Partie 0: Architecture de l'activité pratique



### Partie 1: MICRO-SERVICE CUSTOMER-SERVICE

```

package com.madanicherif.billing.service.model;

import lombok.Data;

6 usages
@Data
public class Customer {
    private Long id;
    private String name;
    private String email;
}
    
```

```

server.port=8081
spring.application.name=customer-service-omc
spring.datasource.url=jdbc:h2:mem:cutsomer-db
spring.cloud.discovery.enabled=true
#management.endpoints.web.exposure.include=*
    
```

```

{
  "name": "Oumaima",
  "email": "oumaimamc@gmail.com",
  "_links": {
    "self": {
      "href": "http://localhost:8081/customers/1"
    },
    "customer": {
      "href": "http://localhost:8081/customers/1"
    }
  }
}
    
```

localhost:8081/h2-console/login.do?sessionId=87c20f59e15c64d6fe6595130aef8653

Auto commit: ☒ Max rows: 1000 Auto complete: ☐ Auto select: ☒

jdbc:h2:mem:cutsomer-db  
 CUSTOMER  
 INFORMATION\_SCHEMA  
 Users  
 H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:  
 SELECT \* FROM CUSTOMER

SELECT \* FROM CUSTOMER;

ID	EMAIL	NAME
1	oumaimamc@gmail.com	Oumaima
2	salma@gmail.com	Salma
3	said@gmail.com	Said
4	khadija@gmail.com	Khadija

(4 rows, 3 ms)

Edit

### Partie 2: MICRO-SERVICE INVENTORY-SERVICE

```

+ usages
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
class Product {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private double price;
    private double quantity;
}

@RepositoryRestResource
interface ProductRepository extends JpaRepository<Product, Long>{}

```

```

@SpringBootApplication
public class InventoryServiceApplication {

    public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }

    @Bean
    CommandLineRunner start(ProductRepository productRepository){
        return args -> {
            productRepository.save(new Product( id: null, name: "Ordinateur", price: 788, quantity: 12));
            productRepository.save(new Product( id: null, name: "Imprimante", price: 88, quantity: 129));
            productRepository.save(new Product( id: null, name: "Smartphone", price: 1288, quantity: 112));
            productRepository.findAll().forEach(p->{
                System.out.println(p.getName());
            });
        };
    }
}

```

```

2023-10-15T11:47:10.474+01:00 INFO 2960 --- [ restartedMain] .s.
2023-10-15T11:47:10.495+01:00 INFO 2960 --- [ restartedMain] .s.
2023-10-15T11:47:10.516+01:00 INFO 2960 --- [nfoReplicator-0] com
2023-10-15T11:47:10.622+01:00 INFO 2960 --- [ restartedMain] c.m
Ordinateur
Imprimante
Smartphone
2023-10-15T11:47:11.101+01:00 INFO 2960 --- []-192.168.254.1] o.a
2023-10-15T11:47:11.101+01:00 INFO 2960 --- []-192.168.254.1] o.s
2023-10-15T11:47:11.104+01:00 INFO 2960 --- []-192.168.254.1] o.s

```

```

server.port=8082
spring.application.name=product-service-omc
spring.datasource.url=jdbc:h2:mem:product-db
spring.cloud.discovery.enabled=false
#management.endpoints.web.exposure.include=*

```

## Partie 3: GATEWAY SERVICE

```
1 server.port=8888
2 spring.application.name=gateway-service
3 spring.cloud.discovery.enabled=false
```

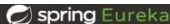
```
@SpringBootApplication
public class GatewayServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(GatewayServiceApplication.class, args);
    }

    @Bean
    public RouteLocator customRouteLocator(RouteLocatorBuilder builder) {
        return builder.routes()
            .route(id: "r1", r -> r
                .path("/customers/**")
                .uri("http://localhost:8081"))
            .route(id: "r2", r -> r
                .path("/products/**")
                .uri("http://localhost:8082"))
            .build();
    }
}
```

```
spring:
  cloud:
    gateway:
      routes:
        - id: r1
          uri: http://localhost:8081/
          predicates:
            - Path=/customers/**
        - id: r2
          uri: http://localhost:8082/
          predicates:
            - Path=/products/**
```

## Partie 4: EUREKA SERVER et tests



[HOME](#) [LAST 1000 SINCE STARTUP](#)

System Status

Environment	test	Current time	2023-10-15T17:31:00 +0100
Data center	default	Uptime	05:51
		Lease expiration enabled	true
		Renews threshold	8
		Renews (last min)	16

DS Replicas


localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
BILLING-SERVICE-OMC	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5.billing-service-omc:8083
CUSTOMER-SERVICE-OMC	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5.customer-service-omc:8081
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5.gateway-service:8888
PRODUCT-SERVICE-OMC	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5.product-service-omc:8082

## Travail Pratique 2: Systèmes Distribués

...

 HOME LAST 1000 SINCE STARTUP

### System Status

Environment	test	Current time	2023-10-15T11:48:22+0100
Data center	default	Uptime	00:09
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	10

### DS Replicas

localhost

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
CUSTOMER-SERVICE-OMC	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5:customer-service-omc:8081
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5:gateway-service:8888
PRODUCT-SERVICE-OMC	n/a (1)	(1)	UP (1) - DESKTOP-36INLL5:product-service-omc:8082

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

← → ↻ 🏠

http://localhost:8888/customers

••• Réseaux, Systè... Développement Inf... debian-12.2.0-amd... Search | Scribd

```
{
  "_embedded": {
    "customers": [
      {
        "name": "Oumaima",
        "email": "oumaimamc@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/customers/1"
          },
          "customer": {
            "href": "http://localhost:8081/customers/1"
          }
        }
      },
      {
        "name": "Salma",
        "email": "salma@gmail.com",
        "_links": {
          "self": {
            "href": "http://localhost:8081/customers/2"
          },
          "customer": {
            "href": "http://localhost:8081/customers/2"
          }
        }
      }
    ]
  },
  "name": "cid"
}
```

```
@SpringBootApplication
@EnableEurekaServer
public class EurekaDiscoveryApplication {

    public static void main(String[] args) { SpringApplication.run(EurekaDiscoveryApplication.class, args);
}
```

## Partie 5: BILLING-SERVICE



## Travail Pratique 2: Systèmes Distribués

...

```
@Entity
public class Bill {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    no usages
    private Date billingDate;
    no usages
    @OneToMany(mappedBy = "bill")
    private Collection<Productitem> productitems;
    no usages
    private long customerID;
    no usages
    @Transient
    private Customer customer;
}
```

```
@Entity
public class Productitem {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    no usages
    private double quantity;
    no usages
    private double price;
    no usages
    private long productID;
    no usages
    @ManyToOne
    private Bill bill;
    no usages
    @Transient
    private Product product;
}
```

```
import com.madanicherif.billingservice.model.Product;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.hateoas.PagedModel;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestParam;

2 usages
@FeignClient(name = "PRODUCT-SERVICE-OMC")
public interface ProductitemRestClient {
    @GetMapping(path = "/products")
    PagedModel<Product> pageProducts(
        // @RequestParam(name = "page") int page,
        // @RequestParam(name = "size") int size
    );
    @GetMapping(path = "/products/{id}")
    Product getProductById(@PathVariable Long id);
}
```

```
package com.madanicherif.billingservice.feign;

import com.madanicherif.billingservice.model.Customer;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

2 usages
@FeignClient(name = "CUSTOMER-SERVICE-OMC")
public interface CustomerRestClient {
    @GetMapping(path = "/customers/{id}")
    Customer getCustomerById(@PathVariable(name = "id") Long id);
}
```

← → ↻ ⌂ ⓘ http://localhost:8888/PRODUCT-SERVICE-OMC/products

••• Réseaux, Systè... Développement Inf... debian-12.2.0-amd... Search |

```
{
  "_embedded": {
    "products": [
      {
        "name": "Ordinateur",
        "price": 788,
        "quantity": 12,
        "_links": {
          "self": {
            "href": "http://DESKTOP-36INLL5:8082/products/1"
          }
        }
      }
    ]
  }
}
```

← → ↻ ⓘ localhost:8888/BILLING-SERVICE-OMC/fullBill/1

```
{
  "id": 1,
  "billingDate": "2023-10-15T15:41:18.794+00:00",
  "productitems": [
    {
      "id": 1,
      "quantity": 34,
      "price": 788,
      "productID": 1,
      "product": null
    },
    {
      "id": 2,
      "quantity": 100,
      "price": 88,
      "productID": 2,
      "product": null
    },
    {
      "id": 3,
      "quantity": 59,
      "price": 1288,
      "productID": 3,
      "product": null
    }
  ],
  "customerID": 1,
  "customer": null
}
```



## Conclusion

En conclusion, cette activité pratique a permis de couvrir un large éventail de concepts et d'outils liés au développement de micro services. De la création des micro services de base à l'utilisation de technologies avancées telles que Spring Cloud Gateway, Netflix Eureka Server et Open Feign, les participants ont acquis une compréhension approfondie de la mise en œuvre pratique de l'architecture de micro services. Cette activité a fourni une base solide pour travailler dans des environnements de développement moderne axés sur les microservices.