ENSET

Département Mathématiques et Informatique

Filière :
« Ingénierie Informatique : Big Data et CloudComputing »
II-BDCC 3

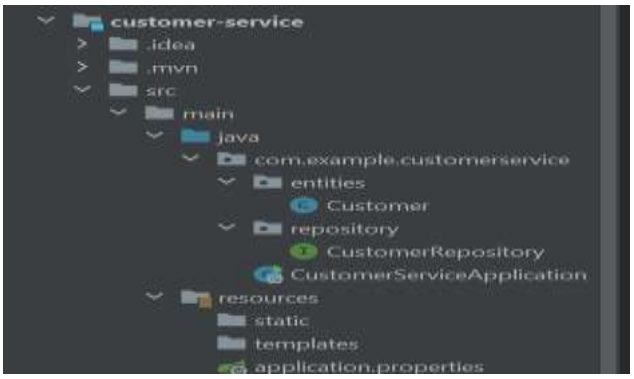
Traitement-parallele-en-Big-Data

Réalisé par :	Professeur :
LADKHAN Tarhla	Mohamed YOUSSFI

Activité2: Architectures Micro-services avec Spring cloud

Partie1 :

Customer-service :



L'entité Customer:

```
package com.example.customerservice.entities;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Customer {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;
}
```

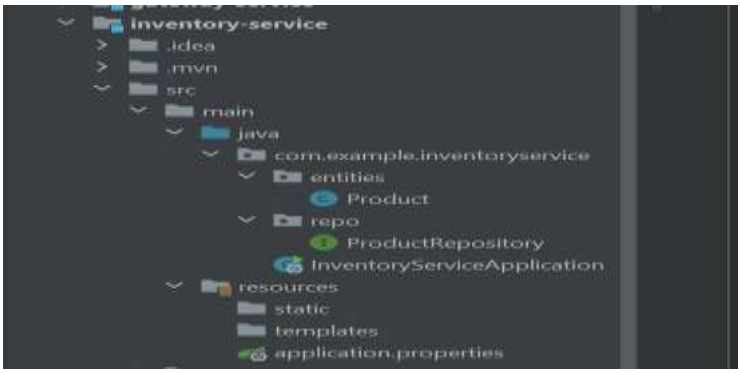
l'application:

```
@SpringBootApplication
public class CustomerServiceApplication {

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

   @Bean
   CommandLineRunner start(CustomerRepository customerRepository) {
      return args -> {
         customerRepository.saveAll(List.of(
               Customer.builder().name("kaoutar").email("k@gmail.com").build(),
               Customer.builder().name("laila").email("l@gmail.com").build(),
               Customer.builder().name("nour").email("n@gmail.com").build()
         ));
         customerRepository.findAll().forEach(System.out::println);
      };
   }
}
```

Les proprieties du classe:

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

Inventory-service :



L'entité Product:

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

L'application:

```
@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(null,"ordi",600,12));
         productRepository.save(new Product(null,"fax",1000,4));
         productRepository.save(new Product(null,"Impre",800,8));

         productRepository.findAll().forEach(p->{
            System.out.println(p.getName());
         });

      };
   }
}
```

Les proprieties du classe:

```
server.port=8082
spring.application.name=inventory-service
spring.datasource.url=jdbc:h2:mem:products-db
spring.cloud.discovery.enabled=true
#management.endpoints.web.exposure.include=*
```

Gateway-service:

L'application:

```
@SpringBootApplication
public class GatewayServiceApplication {

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


   /*RouteLocator routeLocator(RouteLocatorBuilder builder) {
      return builder.routes()
            .route((r) -> r.path("/customers/**").uri("lb://CUSTOMER-SERVICE"))
            .route((r) -> r.path("/products/**").uri("lb://INVENTORY-SERVICE"))
            .build();
   }*/

   @Bean
   public DiscoveryClientRouteDefinitionLocator definitionLocator(
         ReactiveDiscoveryClient rdc,
         DiscoveryLocatorProperties properties) {
      return new DiscoveryClientRouteDefinitionLocator(rdc, properties);
   }
```

App.yml:

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

    - id: r2
      uri: http://localhost:8082/
      predicates:
          - Path=/products/**
```

app.properties:

```
server.port=8083
spring.application.name=gateway-service
spring.cloud.discovery.enabled=true
```

Eureka-discovery:

Application:

```java
@SpringBootApplication
@EnableEurekaServer
public class EurikaDiscoveryApplication {

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

}
```

Properties:

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

Execution:

localhost:8083/CUSTOMER-SERVICE/customers

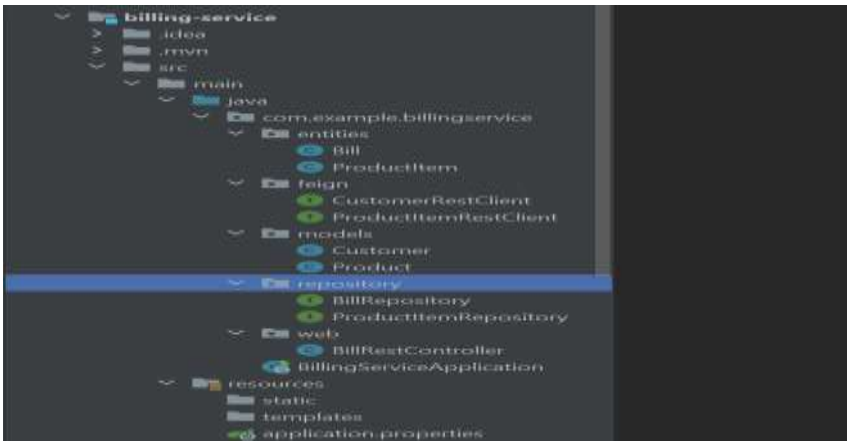

```
{
    "_embedded": {
        "customers": [
            {
                "name": "kaoutar",
                "email": "k@gmail.com",
                "_links": {
                    "self": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/1"
                    },
                    "customer": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/1"
                    }
                }
            },
            {
                "name": "laila",
                "email": "l@gmail.com",
                "_links": {
                    "self": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/2"
                    },
                    "customer": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/2"
                    }
                }
            },
            {
                "name": "nour",
                "email": "n@gmail.com",
                "_links": {
                    "self": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/3"
                    },
                    "customer": {
                        "href": "http://DESKTOP-6CG11AQ.Home:8081/customers/3"
                    }
                }
            }
        ]
    },
    "_links": {
        "self": {
            "href": "http://DESKTOP-6CG11AQ.Home:8081/customers?page=0&size=20"
        },
```


localhost:8083/INVENTORY-SERVICE/products/1


```
{
    "name": "ordi",
    "price": 600,
    "quantity": 12,
    "_links": {
        "self": {
            "href": "http://DESKTOP-6CG11AQ.Home:8082/products/1"
        },
        "product": {
            "href": "http://DESKTOP-6CG11AQ.Home:8082/products/1"
        }
    }
}
```

Partie 2:

BILLING-SERVICE:



Les classes:

```java
@Entity @Data
@NoArgsConstructor
@AllArgsConstructor
public class Bill{
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id; private Date billingDate;
    @OneToMany(mappedBy = "bill")
    private Collection<ProductItem> productItems;
    private Long customerID;
    @Transient
    private Customer customer;
}
```

```java
public class ProductItem{
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private long productID;
    private double price;
    private double quantity;
    @ManyToOne
    private Bill bill;

       @Transient private Product product;
}
```

feign:

```java
@FeignClient(name = "CUSTOMER-SERVICE")

public interface CustomerRestClient {
     @GetMapping(path = "customers/{id}")
     Customer getCustomerById(@PathVariable(name = "id") Long id);
}
```

```java
@FeignClient(name = "INVENTORY-SERVICE")

public interface ProductItemRestClient {
    @GetMapping(path = "/products")
    PagedModel<Product> pageProduct(@RequestParam(name = "page") int page, @RequestParam(name = "size") int size);

    @GetMapping(path="/products/{id}")
    Product getProductById(@PathVariable Long id);
}
```

Web:

```java
@RestController
class BillRestController{

    @Autowired
    private BillRepository billRepository;
    @Autowired private ProductItemRepository productItemRepository;
    @Autowired private CustomerRestClient customerServiceClient;
    @Autowired private ProductItemRestClient inventoryServiceClient;

    @GetMapping(path = "/fullBill/{id}")
    public Bill getBill(@PathVariable(name = "id") Long id){
        Bill bill ;
        bill = billRepository.findById(id).get();
        bill.setCustomer(customerServiceClient.getCustomerById(bill.getCustomerID()));
        bill.getProductItems().forEach(p->{
            p.setProduct(inventoryServiceClient.getProductById(p.getProductID()));
        });
        return bill;
    }}
```

L’application:

```java
@SpringBootApplication
@EnableFeignClients
public class BillingServiceApplication {

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

   @Bean
   CommandLineRunner start(BillRepository billRepository,
                           ProductItemRepository productItemRepository,
                           CustomerRestClient customerRestClient,
                           ProductItemRestClient productItemRestClient) {
      return args -> {
         Customer customer = customerRestClient.getCustomerById(1L);
         Bill bill = billRepository.save(new Bill(null, new Date(), null, customer.getId(), null));
            PagedModel<Product> productPagedModel = productItemRestClient.pageProduct(0,10);
            productPagedModel.forEach(p -> {
               ProductItem productItem = new ProductItem();
               productItem.setPrice(p.getPrice());
               productItem.setBill(bill);
               productItem.setQuantity(1+new Random().nextInt(100));
               productItemRepository.save(productItem);
            });




      };
   }
}
```

Exécution: