

DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Rapport

Filière :

« Ingénierie Informatique : Big Data et Cloud Computing »

II-BDCC

**Activité pratique N° 3 -
Architectures Microservices
avec (Spring Cloud Config,
Consul Discovery, Consul
Config, Vault)**

Réalisé par:

Soukaina EL KAMOUNI

Année Universitaire : 2022-2023

I. Configuration Service Module:

- Main App:

```
@SpringBootApplication
@EnableConfigServer
@EnableDiscoveryClient
public class ConfigServiceApplication {

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

- Application properties:

```
server.port=8888
spring.application.name=config-service
spring.cloud.config.server.git.uri=file:///D:/S
tudies/Mon_cours/S5/MicroServices/ActivitePrati
queN3/ecom-enset/config-repo
```

- Results:



```
1 // 20221223210127
2 // http://localhost:8888/application/default
3
4 {
5   "name": "application",
6   "profiles": [
7     "default"
8   ],
9   "label": null,
10  "version": "b5908430a9c4ae3a804d591aea2220db07f4d2d5",
11  "state": null,
12  "propertySources": [
13    {
14      "name": "file:///D:/Studies/Mon_cours/S5/MicroServices/ActivitePratiqueN3/ecom-enset/config-repo/application.properties",
15      "source": {
16        "global.params.p1": "435",
17        "global.params.p2": "12"
18      }
19    }
20  ]
21 }
```

II. Customer Service Module:

A. Entities:

- Customer Class:

```

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

```

- Customer Projection Interface:

```

@Projection(name = "fullCustomer", types = Customer.class)
public interface CustomerProjection {
    public Long getId();
    public String getName();
    public String getEmail();
}

```

B. Customer Repository:

```

@RepositoryRestResource
public interface CustomerRepository extends
JpaRepository<Customer, Long> {
}

```

C. Customer Configuration Test Controller:

```

@RestController
@RefreshScope //config annotation pour surveiller les
modifications faites
public class CustomerConfigTestController {
    @Value("${global.params.p1}")
    private String p1;
    @Value("${global.params.p2}")
    private String p2;
    @Value("${customer.params.x}")
    private String x;
    @Value("${customer.params.y}")
    private String y;

    @GetMapping("/params")
    public Map<String, String> params() {
        return Map.of("p1", p1, "p2", p2, "x", x, "y", y);
    }
}

```

```
}  
}
```

D. Main App:

```
@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("soukaina1").email("soukaina1@gmail.com").b  
uild(),  
  
Customer.builder().name("soukaina2").email("soukaina2@gmail.com").b  
uild(),  
  
Customer.builder().name("soukaina3").email("soukaina3@gmail.com").b  
uild()  
            ));  
            customerRepository.findAll().forEach(System.out::println);  
        };  
    }  
}
```

E. Application Properties:

```
server.port=8081  
  
spring.application.name=customer-service  
  
spring.config.import=optional:configserver:http://localhost:8888
```

F. Results:

```
localhost:8888/customer-service/dev
1 // 20221223205919
2 // http://localhost:8888/customer-service/dev
3
4 {
5   "name": "customer-service",
6   "profiles": [
7     "dev"
8   ],
9   "label": null,
10  "version": "b5908430a9c4ae3a804d591aea2220db07f4d2d5",
11  "state": null,
12  "propertySources": [
13    {
14      "name": "file:///D:/Studies/Mon_cours/S5/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/customer-service-dev.properties",
15      "source": {
16        "customer.params.x": "900",
17        "customer.params.y": "500",
18        "management.endpoints.web.exposure.include": "*"
19      }
20    },
21    {
22      "name": "file:///D:/Studies/Mon_cours/S5/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/customer-service.properties",
23      "source": {
24        "customer.params.x": "55",
25        "customer.params.y": "660",
26        "management.endpoints.web.exposure.include": "*",
27        "spring.datasource.url": "jdbc:h2:mem:customer-db",
28        "spring.h2.console.enabled": "true"
29      }
30    },
31    {
32      "name": "file:///D:/Studies/Mon_cours/S5/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/application.properties",
33      "source": {
34        "global.params.p1": "435",
35        "global.params.p2": "12"
36      }
37    }
38  ]
39 }
```

```
localhost:8081/params
1 // 20221223210244
2 // http://localhost:8081/params
3
4 {
5   "y": "660",
6   "x": "55",
7   "p1": "435",
8   "p2": "12"
9 }
```

localhost:8081/h2-console/login.do?jsessionId=b46437b606eb6ea8dc6622a69de93dd0

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

jdbc:h2:mem:customer-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	soukaina1@gmail.com	soukaina1
2	soukaina2@gmail.com	soukaina2
3	soukaina3@gmail.com	soukaina3

(3 rows, 2 ms)

Edit

localhost:8081/customers/1

```
1 // 20221223211713
2 // http://localhost:8081/customers/1
3
4 {
5   "name": "soukaina1",
6   "email": "soukaina1@gmail.com",
7   "_links": {
8     "self": {
9       "href": "http://localhost:8081/customers/1"
10    },
11    "customer": {
12      "href": "http://localhost:8081/customers/1{?projection}",
13      "templated": true
14    }
15  }
16 }
```

```
localhost:9999/customer-service

1 // 20221223211813
2 // http://localhost:9999/customer-service
3
4 {
5   "_links": {
6     "customers": {
7       "href": "http://172.20.1.16:8081/customers/{?page,size,sort,projection}",
8       "templated": true
9     },
10    "profile": {
11      "href": "http://172.20.1.16:8081/profile"
12    }
13  }
14 }
```

```
localhost:9999/customer-service/customers/1

1 // 20221223211846
2 // http://localhost:9999/customer-service/customers/1
3
4 {
5   "name": "soukaina1",
6   "email": "soukaina1@gmail.com",
7   "_links": {
8     "self": {
9       "href": "http://172.20.1.16:8081/customers/1"
10    },
11    "customer": {
12      "href": "http://172.20.1.16:8081/customers/1{?projection}",
13      "templated": true
14    }
15  }
16 }
```

```
localhost:9999/customer-service/customers/1?projection=fullCustomer

1 // 20221223213619
2 // http://localhost:9999/customer-service/customers/1?projection=fullCustomer
3
4 {
5   "name": "soukaina1",
6   "id": 1,
7   "email": "soukaina1@gmail.com",
8   "_links": {
9     "self": {
10      "href": "http://172.20.1.16:8081/customers/1"
11     },
12     "customer": {
13      "href": "http://172.20.1.16:8081/customers/1{?projection}",
14      "templated": true
15     }
16   }
17 }
```

III. Gateway Service Module:

- Main App:

```
@SpringBootApplication

public class GatewayServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(GatewayServiceApplication.class, args);

        System.out.println();

    }

    @Bean

    DiscoveryClientRouteDefinitionLocator
dynamicRoutes(ReactiveDiscoveryClient rdc,
DiscoveryLocatorProperties dlp){

        return new DiscoveryClientRouteDefinitionLocator(rdc,dlp);

    }

}
```

- Application properties:

```
server.port=9999

spring.application.name=gateway-service

spring.config.import=optional:configserver:http://localhost:8888
```

IV. Inventory Service Module:

A. Entities:

- Product class:

```
@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;
}
```

- Product Projection Interface:

```
@Projection(name = "fullProduct", types = Product.class)
public interface ProductProjection {
    public Long getId();
    public String getName();
    public double getPrice();
    public int getQuantity();
}
```

B. Product Repository:

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

C. Main App:

```
@SpringBootApplication
public class InventoryServiceApplication {

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

    @Bean
    CommandLineRunner start(ProductRepository productRepository) {
        return args -> {
            Random random=new Random();
            for (int i = 1; i <10 ; i++) {
                productRepository.saveAll(List.of(
                    Product.builder()
                        .name("Compter "+i)
                        .price(1200+Math.random()*10000)
                        .quantity(1+random.nextInt(200)).build()
                ));
            }
        };
    }
}
```

```

    )) ;

    }

    };

}
}

```

D. Application Properties:

```

server.port=8082
spring.application.name=inventory-service
spring.config.import=optional:configserver:http://localhost:8888

```

E. Results:

```

localhost:8888/inventory-service/dev
1 // 20221223210040
2 // http://localhost:8888/inventory-service/dev
3
4 {
5   "name": "inventory-service",
6   "profiles": [
7     "dev"
8   ],
9   "label": null,
10  "version": "b5908430a9c4ae3a804d591aea2220db07f4d2d5",
11  "state": null,
12  "propertySources": [
13    {
14      "name": "file:///D:/Studies/Mon_cours/SS/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/inventory-service-dev.properties",
15      "source": {
16        "inv.params.a": "988",
17        "inv.params.b": "234"
18      }
19    },
20    {
21      "name": "file:///D:/Studies/Mon_cours/SS/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/inventory-service.properties",
22      "source": {
23        "inv.params.a": "54",
24        "inv.params.b": "44"
25      }
26    },
27    {
28      "name": "file:///D:/Studies/Mon_cours/SS/MicroServices/ActivitePratiqueN3/ecom-enst/config-repo/application.properties",
29      "source": {
30        "global.params.p1": "435",
31        "global.params.p2": "12"
32      }
33    }
34  ]
35 }

```

```
1 // 20221223212023
2 // http://localhost:9999/inventory-service/products/1
3
4 {
5   "name": "Compter 1",
6   "price": 1359.636003902781,
7   "quantity": 198,
8   "_links": {
9     "self": {
10       "href": "http://172.20.1.16:8082/products/1"
11     },
12     "product": {
13       "href": "http://172.20.1.16:8082/products/1{?projection}",
14       "templated": true
15     }
16   }
17 }
```

```
localhost:9999/inventory-service/products?projection=fullProduct

1 // 20221223213503
2 // http://localhost:9999/inventory-service/products?projection=fullProduct
3
4 {
5   "_embedded": {
6     "products": [
7       {
8         "name": "Compter 1",
9         "id": 1,
10        "quantity": 198,
11        "price": 1359.636003902781,
12        "_links": {
13          "self": {
14            "href": "http://172.20.1.16:8082/products/1"
15          },
16          "product": {
17            "href": "http://172.20.1.16:8082/products/1{?projection}",
18            "templated": true
19          }
20        }
21      },
22      {
23        "name": "Compter 2",
24        "id": 2,
25        "quantity": 105,
26        "price": 4075.094087772626,
27        "_links": {
28          "self": {
29            "href": "http://172.20.1.16:8082/products/2"
30          },
31          "product": {
32            "href": "http://172.20.1.16:8082/products/2{?projection}",
33            "templated": true
34          }
35        }
36      },
37      {
38        "name": "Compter 3",
39        "id": 3,
40        "quantity": 142,
```

```
localhost:9999/inventory-service/products/1?projection=fullProduct

1 // 20221223213541
2 // http://localhost:9999/inventory-service/products/1?projection=fullProduct
3
4 {
5     "name": "Compter 1",
6     "id": 1,
7     "quantity": 198,
8     "price": 1359.636003902781,
9     "_links": {
10         "self": {
11             "href": "http://172.20.1.16:8082/products/1"
12         },
13         "product": {
14             "href": "http://172.20.1.16:8082/products/1{?projection}",
15             "templated": true
16         }
17     }
18 }
```

V. Order Service Module:

A. Entities:

- Order Class:

```
@Entity @Table(name = "orders") @Data
@NoArgsConstructor @AllArgsConstructor @Builder
public class Order {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Date createdAt;
    private OrderStatus status;
    private Long customerId;
    @Transient
    private Customer customer;
    @OneToMany(mappedBy = "order")
    private List<ProductItem> productItems;

    public double getTotal() {
        double somme=0;
        for(ProductItem pi:productItems){
            somme+=pi.getAmount();
        }
        return somme;
    }
}
```

- Order Projection Interface:

```
@Projection(name = "fullOrder", types = Order.class)
public interface OrderProjection {
    Long getId();
    Date getCreatedAt();
    Long getCustomerId();
    OrderStatus getStatus();
}
```

- Product Item Class:

```
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class ProductItem {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Long productId;
    @Transient
    private Product product;
    private double price;
    private int quantity;
    private double discount;
    @ManyToOne
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Order order;

    public double getAmount() {
        return price*quantity*(1-discount);
    }
}
```

B. ENUMS:

```
public enum OrderStatus {
    CREATED, PENDING, DELIVERED, CANCELED
}
```

C. Models:

- Customer:

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

- Product:

```
@Data
public class Product {
    private Long id;
    private String name;
    private double price;
    private int quantity;
}
```

D. Repositories:

- Order Repository:

```
@RepositoryRestResource
public interface OrderRepository extends JpaRepository<Order, Long>
{
    @RestResource(path = "/byCustomerId")
    List<Order> findByCustomerId(@Param("customerId") Long
customerId);
}
```

- Product Item Repository:

```
@RepositoryRestResource
public interface ProductItemRepository extends
JpaRepository<ProductItem, Long> {
}
```

E. Order Rest Controller:

```
@RestController
public class OrderRestController {
    private OrderRepository orderRepository;
    private ProductItemRepository productItemRepository;
    private CustomerRestClientService customerRestClientService;
    private InventoryRestClientService inventoryRestClientService;
```



```

    public OrderRestController(OrderRepository orderRepository,
ProductItemRepository productItemRepository,
CustomerRestClientService customerRestClientService,
InventoryRestClientService inventoryRestClientService) {
        this.orderRepository = orderRepository;
        this.productItemRepository = productItemRepository;
        this.customerRestClientService = customerRestClientService;
        this.inventoryRestClientService =
inventoryRestClientService;
    }

    @GetMapping("/fullOrder/{id}")
    public Order getOrder(@PathVariable Long id){
        Order order=orderRepository.findById(id).get();
        Customer
customer=customerRestClientService.customerById(order.getCustomerI
d());
        order.setCustomer(customer);
        order.getProductItems().forEach(pi->{
            Product
product=inventoryRestClientService.productById(pi.getProductId());
            pi.setProduct(product);
        });
        return order;
    }
}

```

F. Services:

- Customer Rest Client Service:

```

@FeignClient(name = "customer-service")
public interface CustomerRestClientService {
    @GetMapping("/customers/{id}?projection=fullCustomer")
    public Customer customerById(@PathVariable Long id);
    @GetMapping("/customers?projection=fullCustomer")
    public PagedModel<Customer> allCustomers();
}

```

- Inventory Rest Client Service:

```

@FeignClient(name = "inventory-service")
public interface InventoryRestClientService {
    @GetMapping("/products/{id}?projection=fullProduct")
    public Product productById(@PathVariable Long id);
    @GetMapping("/products?projection=fullProduct")
    public PagedModel<Product> allProducts();
}

```

G.Main App:

```
@SpringBootApplication
@EnableFeignClients
public class OrderServiceApplication {

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

    @Bean
    CommandLineRunner start(
        OrderRepository orderRepository,
        ProductItemRepository productItemRepository,
        CustomerRestClientService customerRestClientService,
        InventoryRestClientService inventoryRestClientService){
        return args -> {
            List<Customer>
customers=customerRestClientService.allCustomers().getContent().stream().toList();
            List<Product>
products=inventoryRestClientService.allProducts().getContent().stream().toList();
            Long customerId=1L;
            Random random=new Random();
            Customer
customer=customerRestClientService.customerById(customerId);
            for (int i = 0; i <20 ; i++) {
                Order order=Order.builder()

                .customerId(customers.get(random.nextInt(customers.size())).getId())
                .status(Math.random()>0.5?
OrderStatus.PENDING:OrderStatus.CREATED)
                .createdAt(new Date())
                .build();
                Order savedOrder = orderRepository.save(order);
                for (int j = 0; j < products.size() ; j++) {
                    if(Math.random()>0.70){
                        ProductItem productItem=ProductItem.builder()
                            .order(savedOrder)
                            .productId(products.get(j).getId())
                            .price(products.get(j).getPrice())
                            .quantity(1+random.nextInt(10))
                            .discount(Math.random())
                            .build();
```

```
        productItemRepository.save(productItem);
    }
}
};
}
```

H. Application Properties:

```
server.port=8083
spring.application.name=order-service
spring.config.import=optional:configserver:http://localhost:8888
logging.level.org.sid.orderservice.services.CustomerRestClientService=debug
logging.level.org.sid.orderservice.services.InventoryRestClientService=debug
feign.client.config.default.loggerLevel=full
```

I. Results:

```
localhost:9999/order-service/orders/1

1 // 20221223214421
2 // http://localhost:9999/order-service/orders/1
3
4 {
5   "createdAt": "2022-12-23T20:43:13.611+00:00",
6   "status": "PENDING",
7   "customerId": 3,
8   "customer": null,
9   "total": 182351.543303379,
10  "_links": {
11    "self": {
12      "href": "http://172.20.1.16:8083/orders/1"
13    },
14    "order": {
15      "href": "http://172.20.1.16:8083/orders/1{?projection}",
16      "templated": true
17    },
18    "productItems": {
19      "href": "http://172.20.1.16:8083/orders/1/productItems"
20    }
21  }
22 }
```



localhost:9999/order-service/fullOrder/1

```
1 // 20221223214604
2 // http://localhost:9999/order-service/fullOrder/1
3
4 {
5   "id": 1,
6   "createdAt": "2022-12-23T20:43:13.611+00:00",
7   "status": "PENDING",
8   "customerId": 3,
9   "customer": {
10     "id": 3,
11     "name": "soukaina3",
12     "email": "soukaina3@gmail.com"
13   },
14   "productItems": [
15     {
16       "id": 1,
17       "productId": 2,
18       "product": {
19         "id": 2,
20         "name": "Compter 2",
21         "price": 3835.1379922815545,
22         "quantity": 172
23       },
24       "price": 3835.1379922815545,
25       "quantity": 9,
26       "discount": 0.68510039199607,
27       "amount": 10869.151053693968
28     },
29     {
30       "id": 2,
31       "productId": 5,
32       "product": {
33         "id": 5,
34         "name": "Compter 5",
35         "price": 4033.3235285075943,
36         "quantity": 3
37       },
```

VI. Order Service Module:

- Consul Configuration Rest Controller:

```
@RestController

public class ConsulConfigRestController {

    @Autowired

    private MyConsulConfig myConsulConfig;

    @Autowired

    private MyVaultConfig myVaultConfig;

    @GetMapping("/myConfig")

    public Map<String, Object> myConfig() {

        return Map.of("consulConfig", myConsulConfig,
"vaultConfig", myVaultConfig);

    }

}
```

- My Consul Configuration:

```
@Component

@ConfigurationProperties(prefix = "token")

@Data

public class MyConsulConfig {

    private long accessTokenTimeout;

    private long refreshTokenTimeout;

}
```

- My Vault Configuration:

```
@Component

@ConfigurationProperties(prefix = "user")

@Data

public class MyVaultConfig {

    private String username;

    private String password;

    private String otp;

}
```

- Main App:

```
@SpringBootApplication

public class BillingApplication {

    @Autowired

    private VaultTemplate vaultTemplate;

    public static void main(String[] args) {

        SpringApplication.run(BillingApplication.class, args);

    }

    @Bean

    CommandLineRunner commandLineRunner() {

        return args -> {

            vaultTemplate.opsForVersionedKeyValue("secret")

        }

    }

}
```

```

        .put("keypair",
Map.of("privKey", "54321", "pubKey", "8999"));

    };

};

}

```

- Application Properties:

```

spring.application.name=billing-service

server.port=8084

spring.cloud.vault.token=hvs.JZezjhs29y6zHrPwgnltWBo
F

spring.cloud.vault.scheme=http

spring.cloud.vault.kv.enabled=true

spring.config.import=optional:consul:, vault://

management.endpoints.web.exposure.include=*

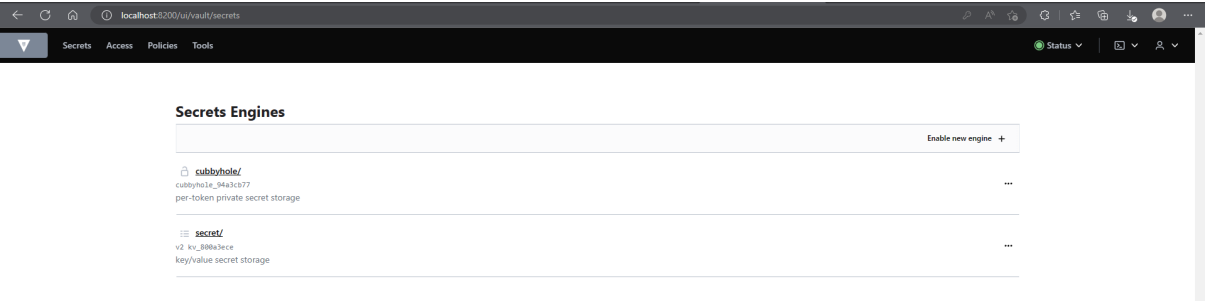
```

- Results:

The screenshot shows the HashiCorp Vault web interface. A green success message at the top states: "Success! Your key has been saved." The left sidebar contains navigation links: Overview, Services, Nodes, Key/Value (selected), Intentions, ACCESS CONTROLS, Tokens, Policies, Roles, and Auth Methods. The main content area is titled "Key / Values / config" and shows the configuration for a key named "billing-service". Below the title is a search bar and a "Type" dropdown. A table lists the key's values:

Name	Type	Actions
token.accessTokenTimeout		...
token.refreshTokenTimeout		...

At the top right of the configuration area is a "Create" button. At the bottom right, there is a link "Folders to Keys".



```
Administrator: Invite de commandes
Microsoft Windows [version 10.0.17763.3770]
(c) 2018 Microsoft Corporation. Tous droits réservés.

C:\Users\user>cd /d D:\Studies\Mon_cours\S5\MicroService plus

D:\Studies\Mon_cours\S5\MicroService plus>set VAULT_ADDR=http://127.0.0.1:8200

D:\Studies\Mon_cours\S5\MicroService plus>vault kv put secret/billing-service user.username=soukaina user.password=123456
===== Secret Path =====
secret/data/billing-service

===== Metadata =====
Key          Value
---          -
created_time  2022-12-23T21:40:50.4754448Z
custom_metadata  <nil>
deletion_time  n/a
destroyed     false
version       1

D:\Studies\Mon_cours\S5\MicroService plus>
```

```
D:\Studies\Mon_cours\S5\MicroService plus>vault kv get secret/billing-service
===== Secret Path =====
secret/data/billing-service

===== Metadata =====
Key          Value
---          -
created_time  2022-12-23T21:40:50.4754448Z
custom_metadata  <nil>
deletion_time  n/a
destroyed     false
version       1

===== Data =====
Key          Value
---          -
user.password 123456
user.username  soukaina

D:\Studies\Mon_cours\S5\MicroService plus>
```



```
localhost:8084/myConfig
1 // 20221223224535
2 // http://localhost:8084/myConfig
3
4 {
5   "vaultConfig": {
6     "username": "soukaina",
7     "password": "123456",
8     "otp": "5432"
9   },
10  "consulConfig": {
11    "accessTokenTimeout": 54000,
12    "refreshTokenTimeout": 540000
13  }
14 }
```

```
Run with: No Environment
POST http://localhost:8084/actuator/refresh
Accept: application/json
<> 2022-12-23T224758.200.json
###

Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 23 Dec 2022 21:47:58 GMT
Keep-Alive: timeout=60
Connection: keep-alive

[]
Response file saved.
> 2022-12-23T224758.200.json

Response code: 200; Time: 1073ms (1 s 73 ms); Content length: 2 bytes (2 B)
```

< secret < keypair

keypair

Secret Metadata

JSON		Delete	Copy	Version 1	Create new version
Key	Value	Version created Dec 23, 2022 10:45 PM			
privKey	54321				
pubKey	8999				

```
localhost:8889/config
1 // 20221223225539
2 // http://localhost:8889/config
3
4 {
5   "consul.token": {
6     "timeout": 0,
7     "authHeader": null
8   },
9   "vault.user": {
10    "username": null,
11    "password": null
12  }
13 }
```

GitHub Link:

<https://github.com/SoukainaElkm/Syst-Distrib-et-Big-Data/tree/main/ActivitePratiqueN3>