**Creating Microservices for account and loan and Create Eureka Discovery Server and register microservices**

**Code:**

1. **Account-Servie:**

**Controller:**

import org.springframework.web.bind.annotation.\*;  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Account getAccountDetails(@PathVariable String number) {  
 return new Account(number, "savings", 234343);  
 }  
 public record Account(String number, String type, double balance) {}  
}

**Application:**

@SpringBootApplication  
@EnableDiscoveryClient  
public class AccountApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(AccountApplication.class, args);  
 }  
  
}

**Properties:**

spring.application.name=account  
server.port=8081  
eureka.client.register-with-eureka=true  
eureka.client.fetch-registry=true  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

1. **Loan-Service:**

**Code:**

**Controller:**

@RestController  
@RequestMapping("/loans")  
public class LoanController {  
 @GetMapping("/{number}")  
 public Loan getLoanDetails(@PathVariable String number) {  
 return new Loan(number, "car", 400000, 3258, 18);  
 }  
 public record Loan(String number, String type, double loan, double emi, int tenure) {}  
}

**Application:**

@SpringBootApplication  
@EnableDiscoveryClient  
public class LoanApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(LoanApplication.class, args);  
 }  
  
}

**Properties:**

spring.application.name=loan  
server.port=8082  
eureka.client.register-with-eureka=true  
eureka.client.fetch-registry=true  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

1. **Eureka-Server:**

**Code:**

**Application:**

@SpringBootApplication  
@EnableEurekaServer  
public class EurekaDiscoveryServerApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(EurekaDiscoveryServerApplication.class, args);  
 }  
  
}

**Properties:**

spring.application.name=eureka-discovery-server  
server.port=8761  
eureka.client.register-with-eureka=false  
eureka.client.fetch-registry=false

1. **API-Gateway:**

**Code:**

**Application:**

@SpringBootApplication  
public class ApiGatewayApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(ApiGatewayApplication.class, args);  
 }  
  
}

**LogFilter:**

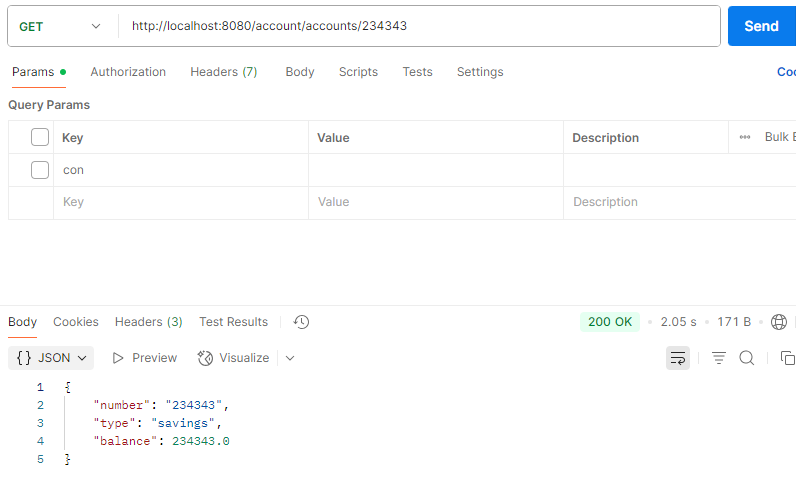
@Component  
public class LogFilter implements GlobalFilter, Ordered {  
  
 @Override  
 public Mono<Void> filter(ServerWebExchange exchange, org.springframework.cloud.gateway.filter.GatewayFilterChain chain) {  
 System.*out*.println("Incoming request URI: " + exchange.getRequest().getURI());  
 return chain.filter(exchange);  
 }  
 @Override  
 public int getOrder() {  
 return -1;  
 }  
}

**Properties:**

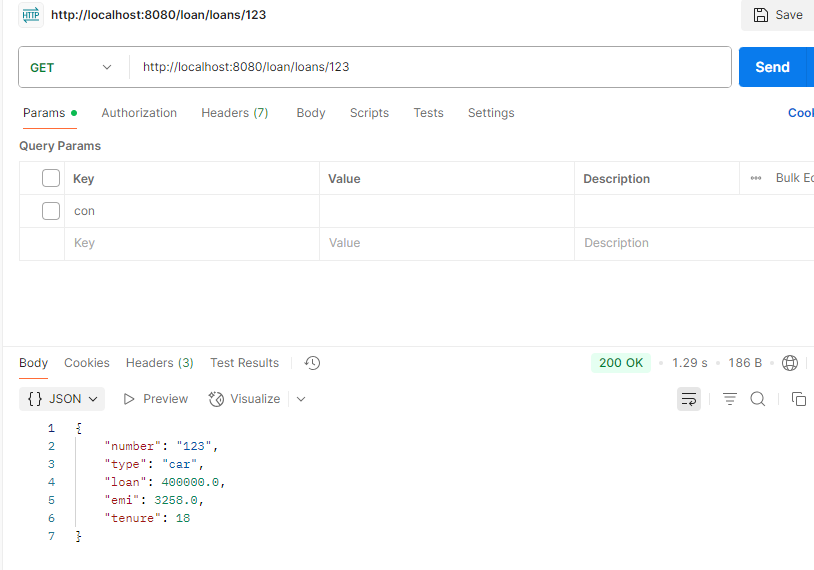
spring.application.name=api-gateway  
server.port=8080  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
  
spring.cloud.gateway.server.webflux.routes[0].id=account  
spring.cloud.gateway.server.webflux.routes[0].uri=lb://account  
spring.cloud.gateway.server.webflux.routes[0].predicates[0]=Path=/account/\*\*  
spring.cloud.gateway.server.webflux.routes[0].filters[0]=RewritePath=/account/(?<segment>.\*), /${segment}  
  
spring.cloud.gateway.server.webflux.routes[1].id=loan  
spring.cloud.gateway.server.webflux.routes[1].uri=lb://loan  
spring.cloud.gateway.server.webflux.routes[1].predicates[0]=Path=/loan/\*\*  
spring.cloud.gateway.server.webflux.routes[1].filters[0]=RewritePath=/loan/(?<segment>.\*), /${segment}

**OUTPUTs:**

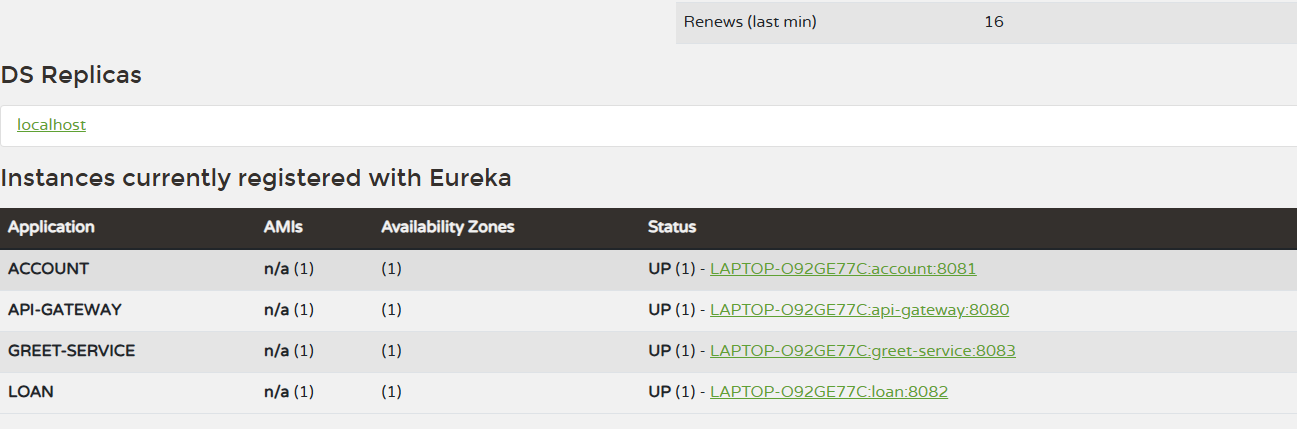
**Account-service:**

****

**Loan-Service:**

****

**Eureka Server Dashboard:**

****