**Hands-on:1 Creating Microservices for account and loan**

**a. Account Microservice**

**AccountController.java:**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.\*;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AccountController {

@GetMapping("/accounts/{number}")

public Map<String, Object> getAccountDetails(@PathVariable String number) {

Map<String, Object> response = new HashMap<>();

response.put("number", number);

response.put("type", "savings");

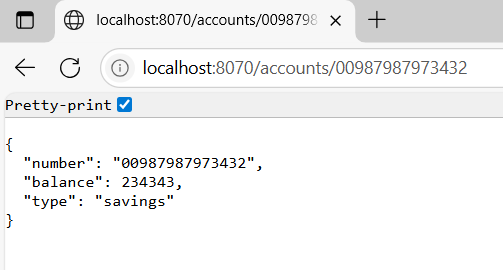
response.put("balance", 234343);

return response;

}

}

**OUTPUT:**

****

**b.** **Loan Microservice:**

**application.properties:**

server.port=8071

**LoanController.java:**

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.\*;

import java.util.HashMap;

import java.util.Map;

@RestController

public class LoanController {

@GetMapping("/loans/{number}")

public Map<String, Object> getLoanDetails(@PathVariable String number) {

Map<String, Object> response = new HashMap<>();

response.put("number", number);

response.put("type", "car");

response.put("loan", 400000);

response.put("emi", 3258);

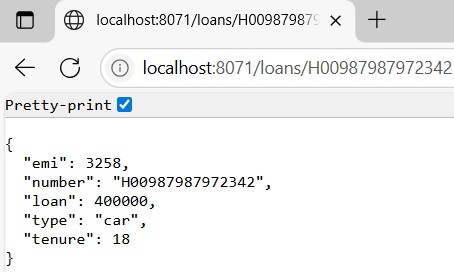
response.put("tenure", 18);

return response;

}

}

**OUTPUT:**

****

**Hands-on 2:**

**Create Eureka Discovery Server and register microservices**

**1. EUREKA DISCOVERY SERVER**

**EurekaDiscoveryServerApplication:.java**

package com.cognizant;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class EurekaDiscoveryServerApplication {

    public static void main(String[] args) {

        SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

    }

}

**Application.properties:**

server.port=8764

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF

logging.level.com.netflix.discovery=OFF

**2. ACCOUNT SERVICE**

**AccountApplication.java:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

    public static void main(String[] args) {

        SpringApplication.run(AccountApplication.class, args);

    }

}

**Application.properties:**

spring.application.name=account

server.port=8081

spring.application.name=account-service

eureka.client.service-url.defaultZone=http://localhost:8764/eureka

**AccountController.java:**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/account")

public class AccountController {

    @GetMapping("/status")

    public String getStatus() {

        return "Account Service is up";

    }

}

**3. GREET SERVICE**

**GreetServiceApplication.java:**

package com.cognizant.greet;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class GreetServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(GreetServiceApplication.class, args);

    }

}

**GreetController.java:**

package com.cognizant.greet.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/greet")

public class GreetController {

    @GetMapping

    public String greet() {

        return "Hello World";

    }

}

**Application.properties:**.

server.port=8083

spring.application.name=greet-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

**5. API GATEWAY**

**ApiGatewayApplication.java**

package com.cognizant.gateway;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class ApiGatewayApplication {

    public static void main(String[] args) {

        SpringApplication.run(ApiGatewayApplication.class, args);

    }

}

**LogFilter.java:**

package com.cognizant.gateway;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.cloud.gateway.filter.GatewayFilterChain;

import org.springframework.cloud.gateway.filter.GlobalFilter;

import org.springframework.stereotype.Component;

import org.springframework.web.server.ServerWebExchange;

import reactor.core.publisher.Mono;

@Component

public class LogFilter implements GlobalFilter {

    private static final Logger logger = LoggerFactory.getLogger(LogFilter.class);

    @Override

    public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

        logger.info("Request Path: {}", exchange.getRequest().getPath());

        return chain.filter(exchange);

    }

}

**application.properties**

server.port=9090

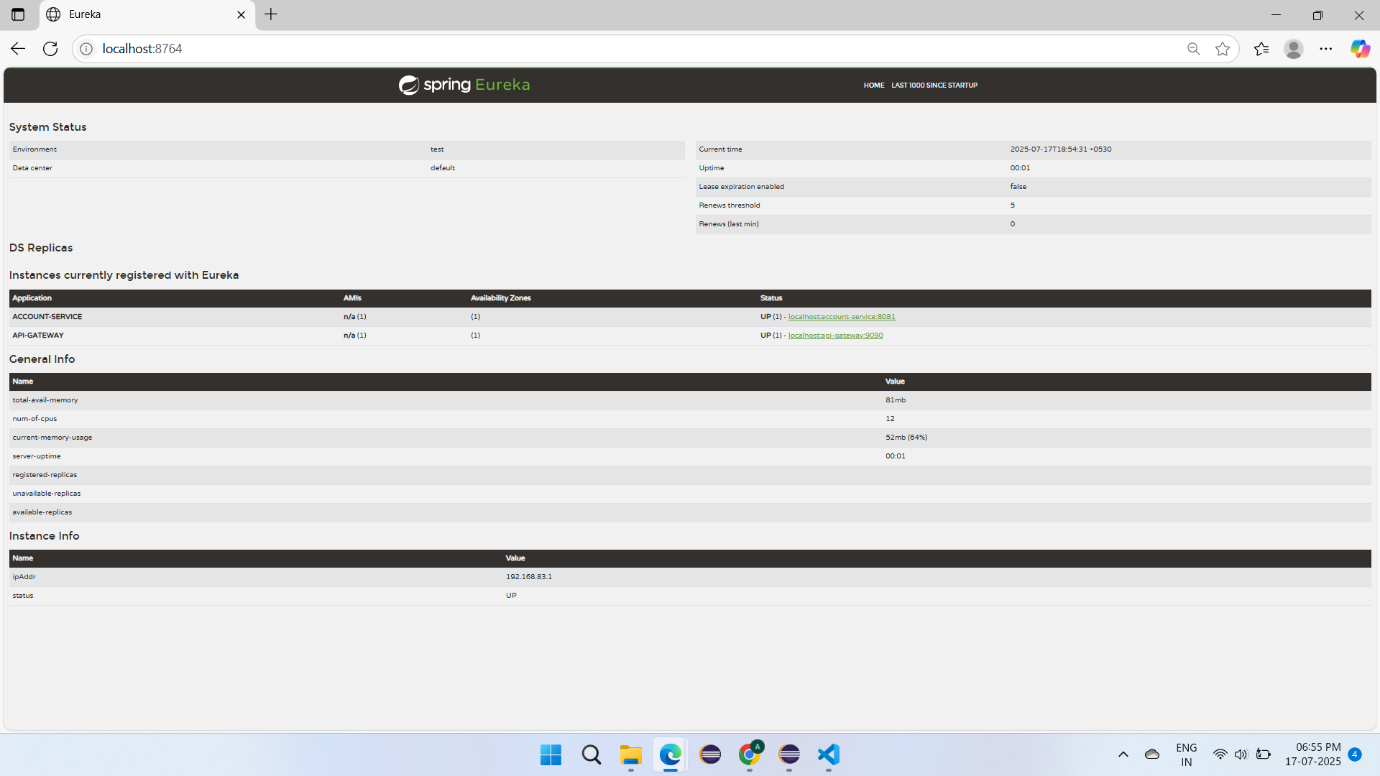
spring.application.name=api-gateway

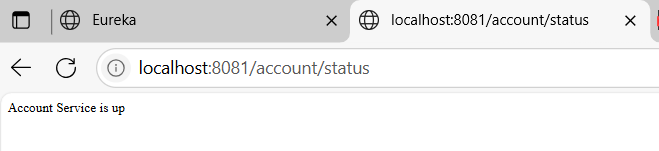
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

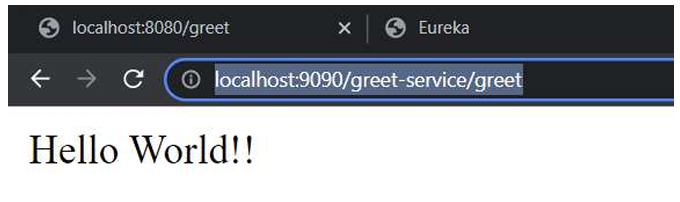
spring.cloud.gateway.discovery.locator.enabled=true

spring.cloud.gateway.discovery.locator.lower-case-service-id=true

**OUPUTS:**

****

****

****