## **1. Eureka Discovery Server**

### **EurekaDiscoveryServerApplication.java**

package com.cognizant.eurekaserver;

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=8761

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 Microservice**

### **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**

server.port=8080

spring.application.name=account-service

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

### **AccountController.java**

package com.cognizant.account.controller;

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

import java.util.\*;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

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

acc.put("number", number);

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

acc.put("balance", 234343);

return acc;

}

}

## **3. Loan Microservice**

### **application.properties**

server.port=8081

spring.application.name=loan-service

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

### **LoanController.java**

package com.cognizant.loan.controller;

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

import java.util.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

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

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

loan.put("number", number);

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

loan.put("loan", 400000);

loan.put("emi", 3258);

loan.put("tenure", 18);

return loan;

}

}

**4. Greet Microservice**

### **GreetController.java**

package com.cognizant.greet;

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

@RestController

public class GreetController {

@GetMapping("/greet")

public String greet() {

return "Hello World";

}

}

### **application.properties**

server.port=8082

spring.application.name=greet-service

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

## **5. API Gateway**

### **ApiGatewayApplication.java**

package com.cognizant.apigateway;

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);

}

}

### **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

### **LogFilter.java**

package com.cognizant.apigateway;

import org.springframework.stereotype.Component;

import org.springframework.cloud.gateway.filter.\*;

import org.springframework.web.server.ServerWebExchange;

import reactor.core.publisher.Mono;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Component

public class LogFilter implements GlobalFilter, Ordered {

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

@Override

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

logger.info("Gateway intercepted request: {} {}",

exchange.getRequest().getMethod(),

exchange.getRequest().getURI());

return chain.filter(exchange);

}

@Override

public int getOrder() {

return -1;

}

}

## 

## 

## **Output**

### **Eureka Dashboard -** [**http://localhost:8761**](http://localhost:8761/)

Applications registered with Eureka:

APPLICATION INSTANCES

ACCOUNT-SERVICE 1

LOAN-SERVICE 1

GREET-SERVICE 1

API-GATEWAY 1

**Greet API via Gateway -** [**http://localhost:9090/greet-service/greet**](http://localhost:9090/greet-service/greet)

Hello World

### **Account API via Gateway -** [**http://localhost:9090/account-service/accounts/00987987973432**](http://localhost:9090/account-service/accounts/00987987973432)

{

"number": "00987987973432",

"type": "savings",

"balance": 234343

}

**Loan API via Gateway -** [**http://localhost:9090/loan-service/loans/H00987987972342**](http://localhost:9090/loan-service/loans/H00987987972342)

{

"number": "H00987987972342",

"type": "car",

"loan": 400000,

"emi": 3258,

"tenure": 18

}

### **Console Log from API Gateway**

Gateway intercepted request: GET /greet-service/greet

Gateway intercepted request: GET /account-service/accounts/00987987973432

Gateway intercepted request: GET /loan-service/loans/H00987987972342