**Creating Microservices for account and loan**

**CONFIGURE EUREKA SERVER**

service-registry/src/main/resources/application.yml

server:

port: 8761

eureka:

client:

register-with-eureka: false

fetch-registry: false

ServiceRegistryApplication.java

@SpringBootApplication

@EnableEurekaServer

public class ServiceRegistryApplication {

public static void main(String[] args) {

SpringApplication.run(ServiceRegistryApplication.class, args);

}

}

**CONFIGURE ACCOUNT & LOAN SERVICE**

**Create Account Microservice**

AccountController.java:

package com.cognizant.account.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

return Map.of(

"number", number,

"type", "savings",

"balance", 234343

);

}

}

**Create Loan Microservice**

LoanController.java:

package com.cognizant.loan.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

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

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

"tenure", 18

);

}

}

**CONFIGURE API GATEWAY**

application.yml

server:

port: 8080

spring:

application:

name: api-gateway

cloud:

gateway:

routes:

- id: account-service

uri: lb://account-service

predicates:

- Path=/account/\*\*

- id: loan-service

uri: lb://loan-service

predicates:

- Path=/loan/\*\*

eureka:

client:

service-url:

defaultZone: <http://localhost:8761/eureka>

**Expected Output :**

**Account Microservice**



**Loan Microservice**

