Microservices with Spring Boot 3 and Spring Cloud – Hands-on Exercises

# Exercise 1: Creating Microservices for Account and Loan

### Code:

// AccountService.java  
@RestController  
@RequestMapping("/account")  
public class AccountService {  
 @GetMapping("/{id}")  
 public String getAccount(@PathVariable int id) {  
 return "Account details for ID: " + id;  
 }  
}  
  
// LoanService.java  
@RestController  
@RequestMapping("/loan")  
public class LoanService {  
 @GetMapping("/{id}")  
 public String getLoan(@PathVariable int id) {  
 return "Loan details for ID: " + id;  
 }  
}

### Expected Output:

# 

# Exercise 2: Create Eureka Discovery Server and Register Microservices

### Code:

// EurekaServerApplication.java  
@EnableEurekaServer  
@SpringBootApplication  
public class EurekaServerApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(EurekaServerApplication.class, args);  
 }  
}  
  
// application.yml (for Eureka server)  
server:  
 port: 8761  
  
eureka:  
 client:  
 register-with-eureka: false  
 fetch-registry: false  
 server:  
 wait-time-in-ms-when-sync-empty: 0  
  
// application.yml (for microservice client)  
eureka:  
 client:  
 service-url:  
 defaultZone: http://localhost:8761/eureka  
spring:  
 application:  
 name: account-service

### Expected Output:

