Microservices Assignment

**1. Creating Microservices for account and loan**

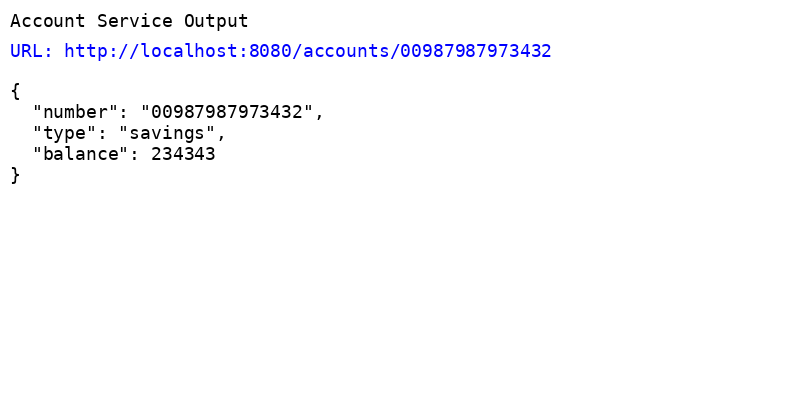
# A. Account Microservice

## Controller Code:

@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{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:

URL: http://localhost:8080/accounts/00987987973432



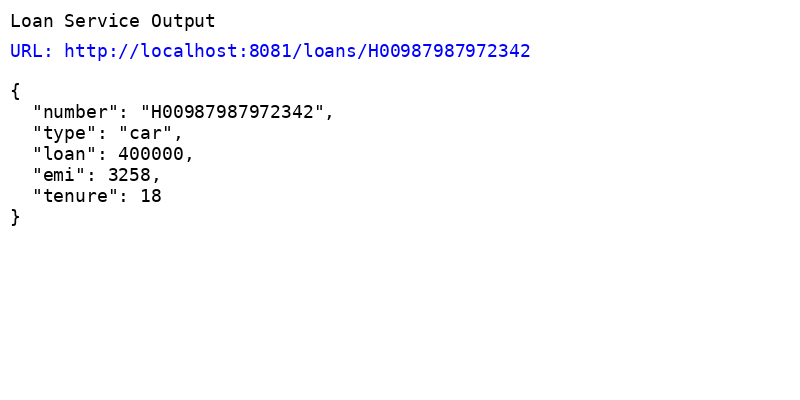
# B. Loan Microservice

## Controller Code:

@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{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:

URL: http://localhost:8081/loans/H00987987972342



# 2. Create Eureka Discovery Server and register microservices

**Step 1: Create Eureka Discovery Server**

* **In main class, add:**

@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

**Step 2: Register Account Microservice**

* **application.properties:**

server.port=8080

spring.application.name=account-service

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

* **Main Class:**

@SpringBootApplication

@EnableDiscoveryClient

public class AccountServiceApplication {

public static void main(String[] args) {

SpringApplication.run(AccountServiceApplication.class, args);

}

}

* **AccountController.java:**

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

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

account.put("number", number);

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

account.put("balance", 234343);

return account;

}

}

**Step 3: Register Loan Microservice**

Same as account-service, but:

* **application.properties:**

server.port=8081

spring.application.name=loan-service

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

* **LoanController.java:**

@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;

}

}

## Output:

URL: http://localhost:8761

