1.Creating Microservices for account and loan

Answer:

AccountApplication.java:

```
package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class AccountApplication {

   public static void main(String[] args) {

      SpringApplication.run(AccountApplication.class, args);

   }

}
```

AccountController.java:

```
package com.cognizant.account.controller;

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

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

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


import java.util.Map;


@RestController

public class AccountController {


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

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

      return Map.of(

            "number", number,
                "type", "savings",
                "balance", 234343
        );
    }
}
```

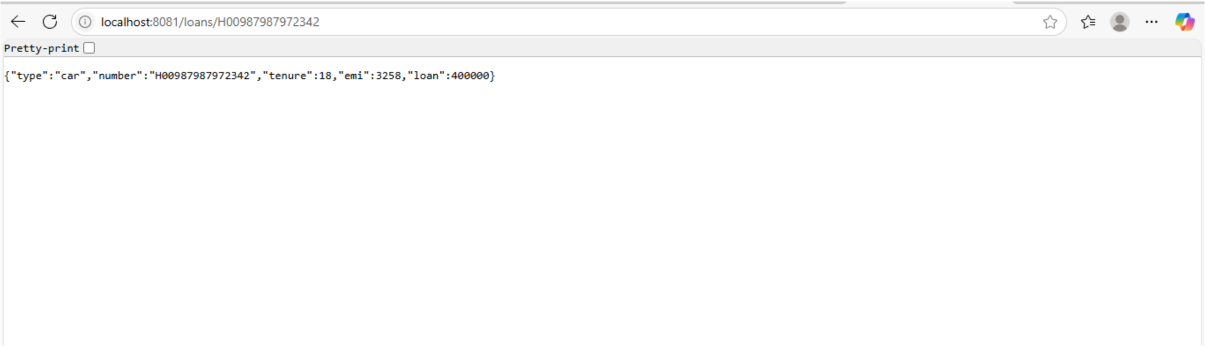
LoanApplication.java:

```
package com.cognizant.loan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class LoanApplication {
    public static void main(String[] args) {
        SpringApplication.run(LoanApplication.class, args);
    }
}
```

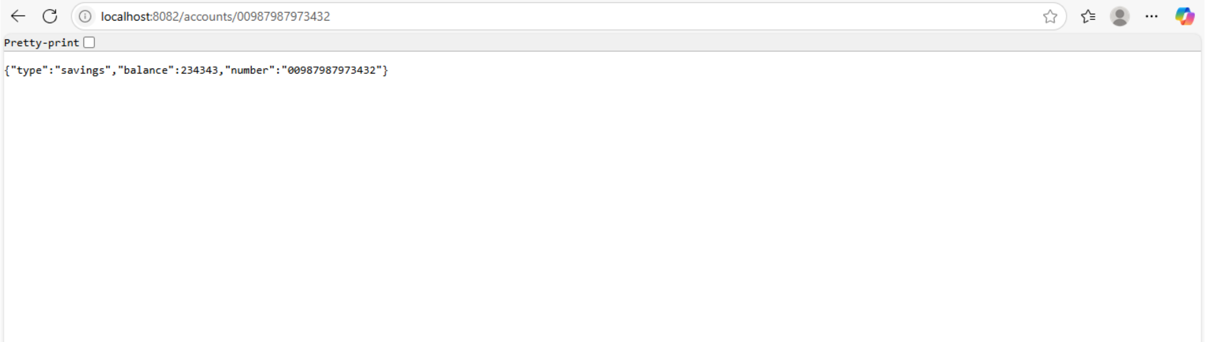
LoanController.java:

```
package com.cognizant.loan.controller;

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

import java.util.Map;

@RestController
public class LoanController {

    @GetMapping("/loans/{number}")
    public Map<String, Object> getLoanDetails(@PathVariable String number) {
        return Map.of(
                "number", "H00987987972342",
                "type", "car",
                "loan", 400000,
                "emi", 3258,
                "tenure", 18
        );
    }
}
```

Output:



```
{"type":"car","number":"H00987987972342","tenure":18,"emi":3258,"loan":400000}
```



```
{"type":"savings","balance":234343,"number":"00987987973432"}
```

2.Create Eureka Discovery Server and register microservices

Answer:

EurekaServerApplication.java:

```
package com.example.eurekaserver;
```

```java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer
public class EurekaServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(EurekaServerApplication.class, args);
    }
}
```

Application.yml:

```yaml
server:
  port: 8761

spring:
  application:
    name: eureka-server

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
  server:
    wait-time-in-ms-when-sync-empty: 0
```

AccountServiceApplication.java:

```java
package com.example.accountservice;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication
@EnableDiscoveryClient
public class AccountServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(AccountServiceApplication.class, args);
    }
}
```

AccountController.java:

```
package com.example.accountservice.controller;

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

@RestController
public class AccountController {

    @GetMapping("/hello")
    public String sayHello() {
        return "Hello from Account Service!";
    }
}
```

Application.yml:

```
server:
  port: 8082
```

```
spring:
  application:
    name: account-service

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

instance:
prefer-ip-address: true
```

ApiGatewayApplication.java:

```
package com.example.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);
    }
}
```

LogFilter.java:

```
package com.example.apigateway;

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

```java
import org.springframework.core.Ordered;
import org.springframework.stereotype.Component;
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, org.springframework.cloud.gateway.filter.GatewayFilterChain chain) {
        logger.info("Request path: {}", exchange.getRequest().getPath());
        return chain.filter(exchange);
    }

    @Override
    public int getOrder() {
        return -1; // Highest priority
    }
}
```

Application.yml:

```yaml
server:
  port: 9090

spring:
  application:
    name: api-gateway

  cloud:
    gateway:
      routes:
        - id: greet-service
          uri: lb://greet-service
          predicates:
            - Path=/greet-service/**
          filters:
            - StripPrefix=1
      discovery:
        locator:
          enabled: true
          lower-case-service-id: true

eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka
```

GreetingController.java:

```
package com.example.greetservice;

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

@RestController
public class GreetingController {
    @GetMapping("/hello")
    public String sayHello() {
        return "Hello World";
    }
}
```

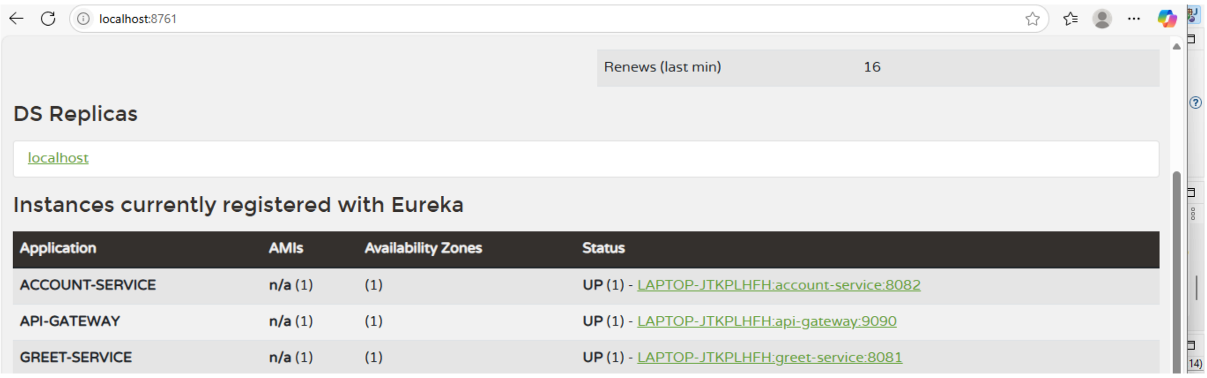
GreetServiceApplication.java:

```
package com.example.greetservice;

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);
    }
}
```

Application.yml:

```
server:
  port: 8081

spring:
  application:
    name: greet-service

eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka
instance:
prefer-ip-address: true
```

Output:



localhost:8761

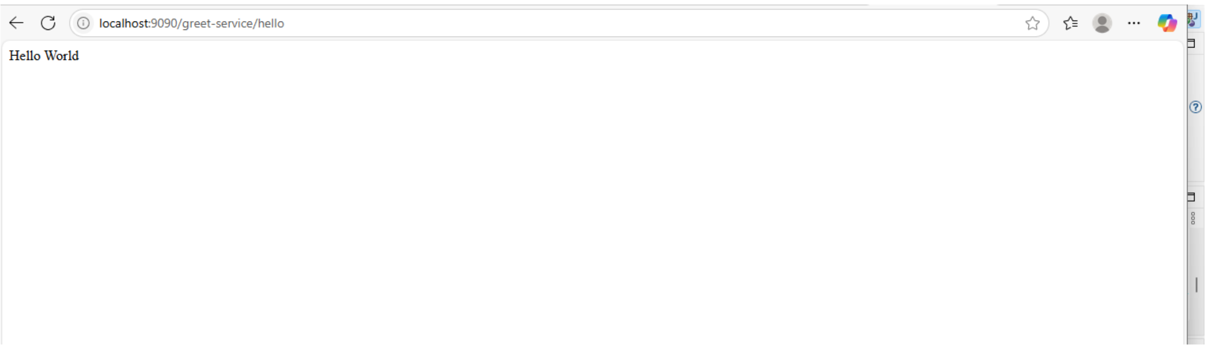
Renews (last min) 16

DS Replicas

localhost

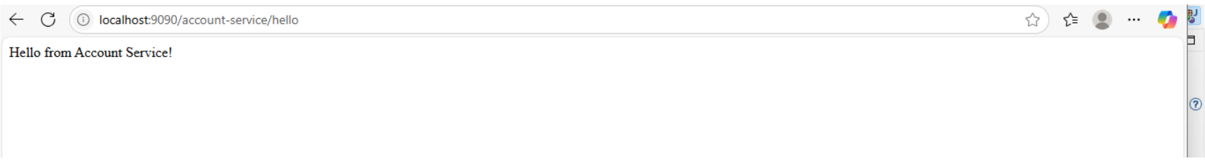
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ACCOUNT-SERVICE	n/a (1)	(1)	UP (1) - LAPTOP-JTKPLHFH:account-service:8082
API-GATEWAY	n/a (1)	(1)	UP (1) - LAPTOP-JTKPLHFH:api-gateway:9090
GREET-SERVICE	n/a (1)	(1)	UP (1) - LAPTOP-JTKPLHFH:greet-service:8081



localhost:9090/greet-service/hello

Hello World



localhost:9090/account-service/hello

Hello from Account Service!