**Creating Microservices for account and loan**

/microservices-project/

├── /account /

│ ├── src/

│ │ └── main/

│ │ └── java/com/cognizant/account/

│ │ ├── AccountApplication.java

│ │ └── controller/AccountController.java

│ ├── resources/

│ │ └── application.properties

│ └── pom.xml

│

├── /loan /

│ ├── src/

│ │ └── main/

│ │ └── java/com/cognizant/loan/

│ │ ├── LoanApplication.java

│ │ └── controller/LoanController.java

│ ├── resources/

│ │ └── application.properties

│ └── pom.xml

**Code for 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);

    }

}

**Code for 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;

    }

}

**Code for pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

                             http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.cognizant</groupId>

    <artifactId>account</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>account</name>

    <description>Account Microservice</description>

    <packaging>jar</packaging>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.4</version>

        <relativePath/>

    </parent>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-devtools</artifactId>

            <scope>runtime</scope>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**Code for LoanServiceApplication.java:**

package com.example.loans;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LoanServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(LoanServiceApplication.class, args);

    }

}

**Code for LoanController.java:**

package com.example.loans.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", "home loan");

        loan.put("amount", 500000);

        return loan;

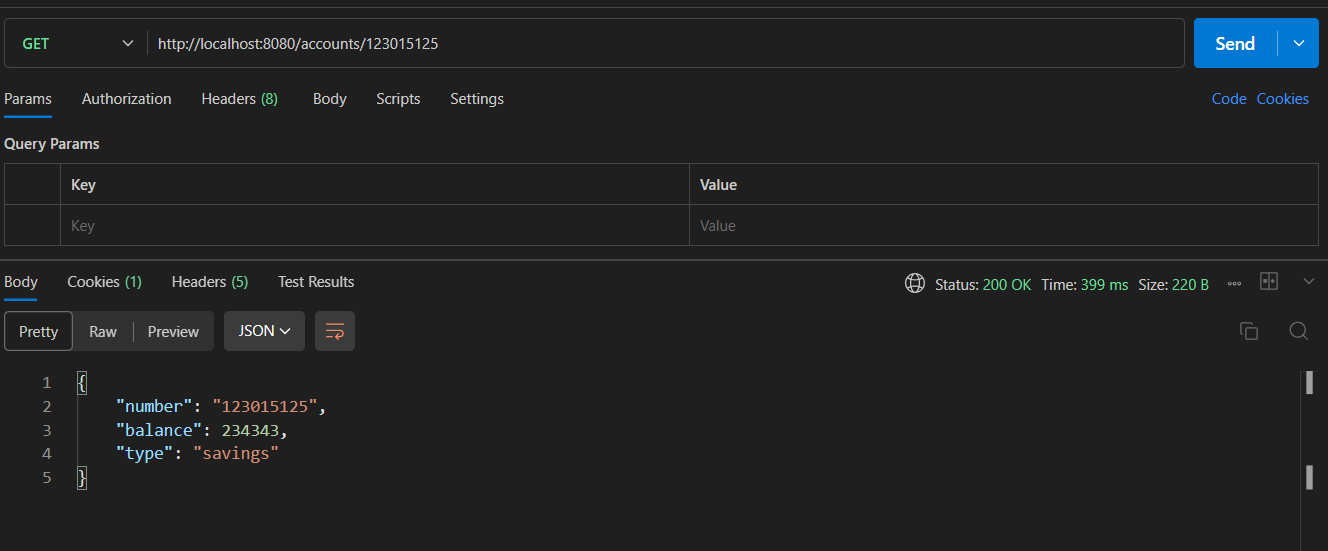
    }

}

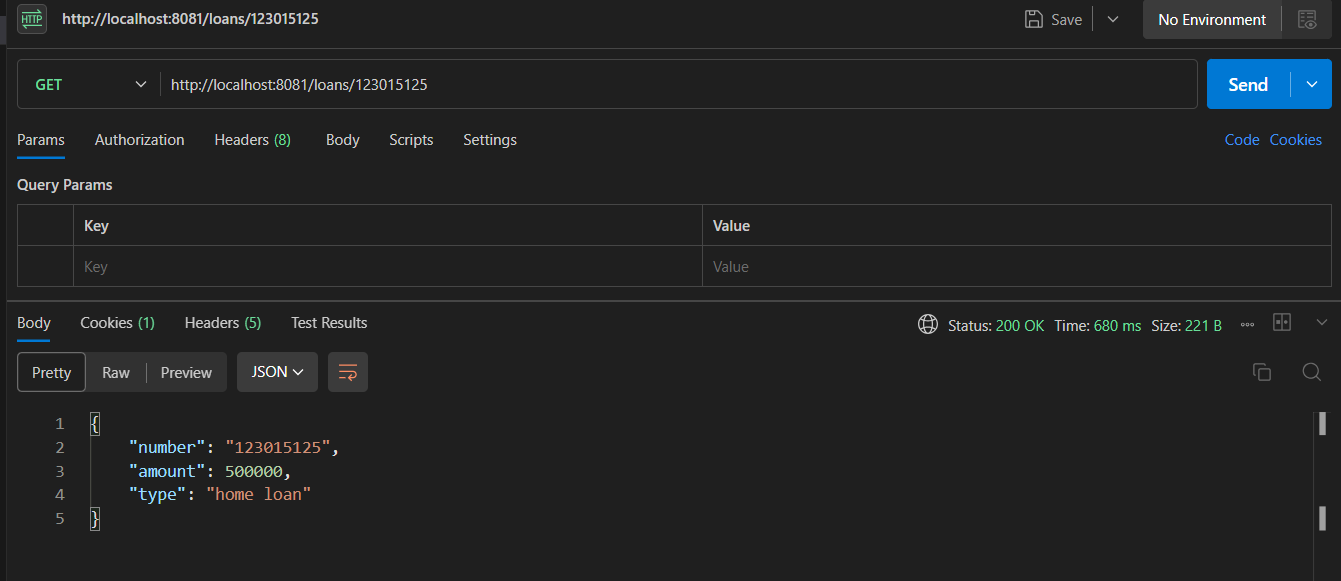
**application.properties:**

server.port=8081

**Output for account:**

****

**Output for Loans:**

****

**Create Eureka Discovery Server and register microservices**

Directory:

microservices/

│ ├── accounts/

│ │ ├── pom.xml

│ │ └── src/

│ │ └── main/

│ │ └── java/

│ │ └── com/

│ │ └── cognizant/

│ │ └── account/

│ │ └── AccountApplication.java

│ ├── apigateway/

│ │ └── src/

│ │ └── main/

│ │ └── resources/

│ │ └── application.properties

│ └── eureka/

│ └── src/

│ └── main/

│ └── java/

│ └── com/

│ └── cognizant/

│ └── eurekaserver/

│ └── EurekaDiscoveryServerApplication.java

**Code for 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);

    }

}

**Code for 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;

    }

}

**Code for LoanServiceApplication.java:**

package com.example.loans;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LoanServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(LoanServiceApplication.class, args);

    }

}

**Code for LoanController.java:**

package com.example.loans.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", "home loan");

        loan.put("amount", 500000);

        return loan;

    }

}

**Code for 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

**Code for ApiGatewayApplication.java:**

package com.cognizant.apigateway;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class ApiGatewayApplication {

    public static void main(String[] args) {

        SpringApplication.run(ApiGatewayApplication.class, args);

    }

}

**Application.properties:**

server.port=8765

spring.application.name=api-gateway

eureka.client.service-url.defaultZone=http://localhost:8080/eureka

eureka.instance.prefer-ip-address=true

spring.cloud.gateway.routes[0].id=account-service

spring.cloud.gateway.routes[0].uri=lb://ACCOUNT

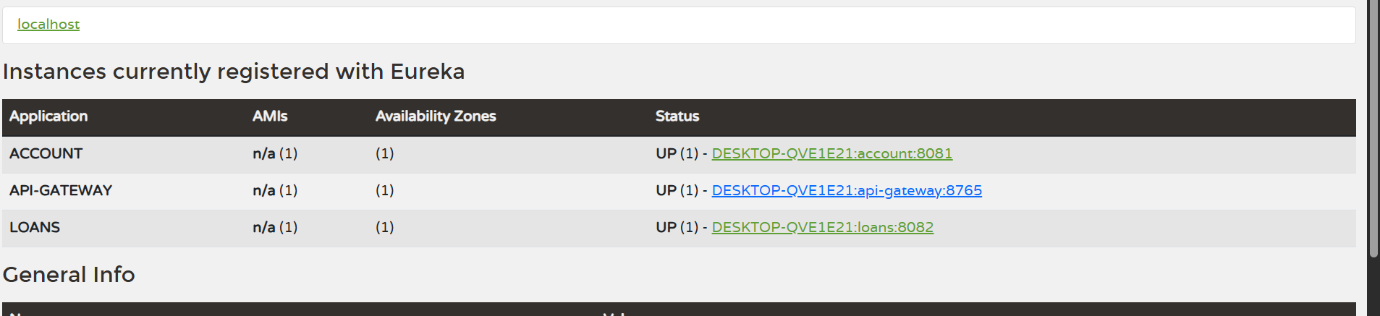
spring.cloud.gateway.routes[0].predicates[0]=Path=/accounts/\*\*

spring.cloud.gateway.routes[1].id=loan-service

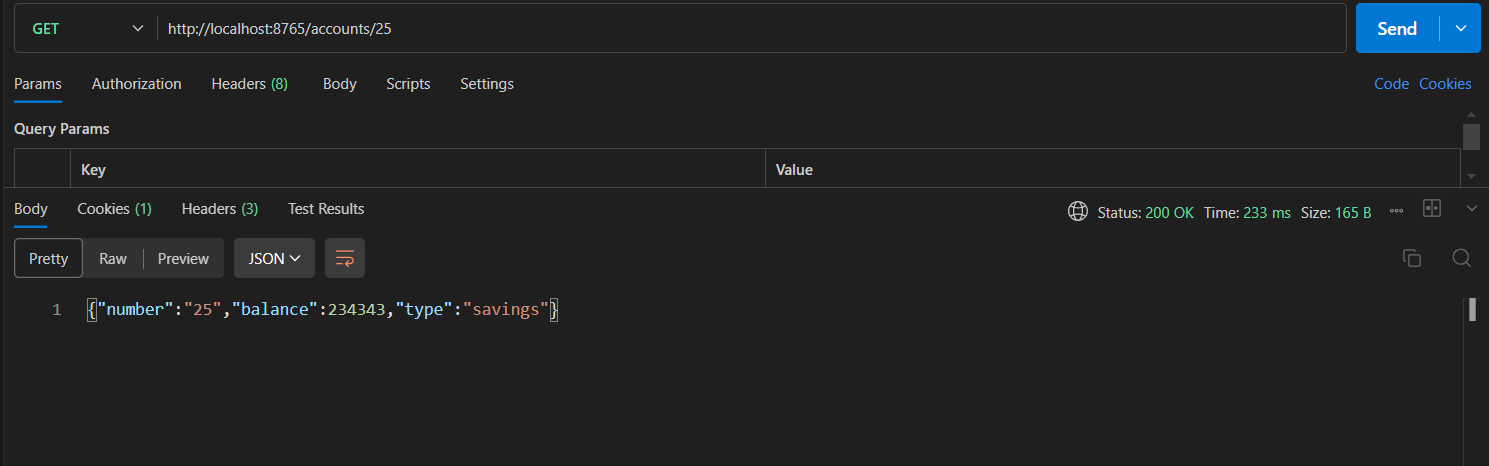
spring.cloud.gateway.routes[1].uri=lb://LOANS

spring.cloud.gateway.routes[1].predicates[0]=Path=/loans/\*\*

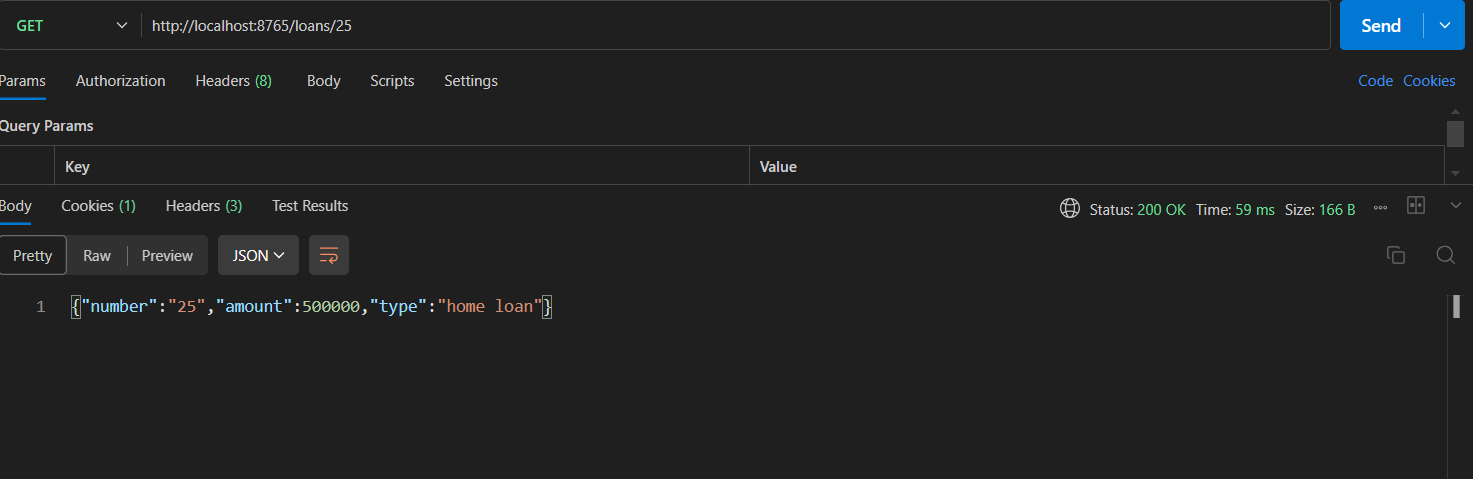
**Output:**



**Output:(Accessing account via api-gateway)**

****

**Output:(Accessing loans via api-gateway)**

****