

# Microservices with API gateway

## Creating Microservices for account and loan

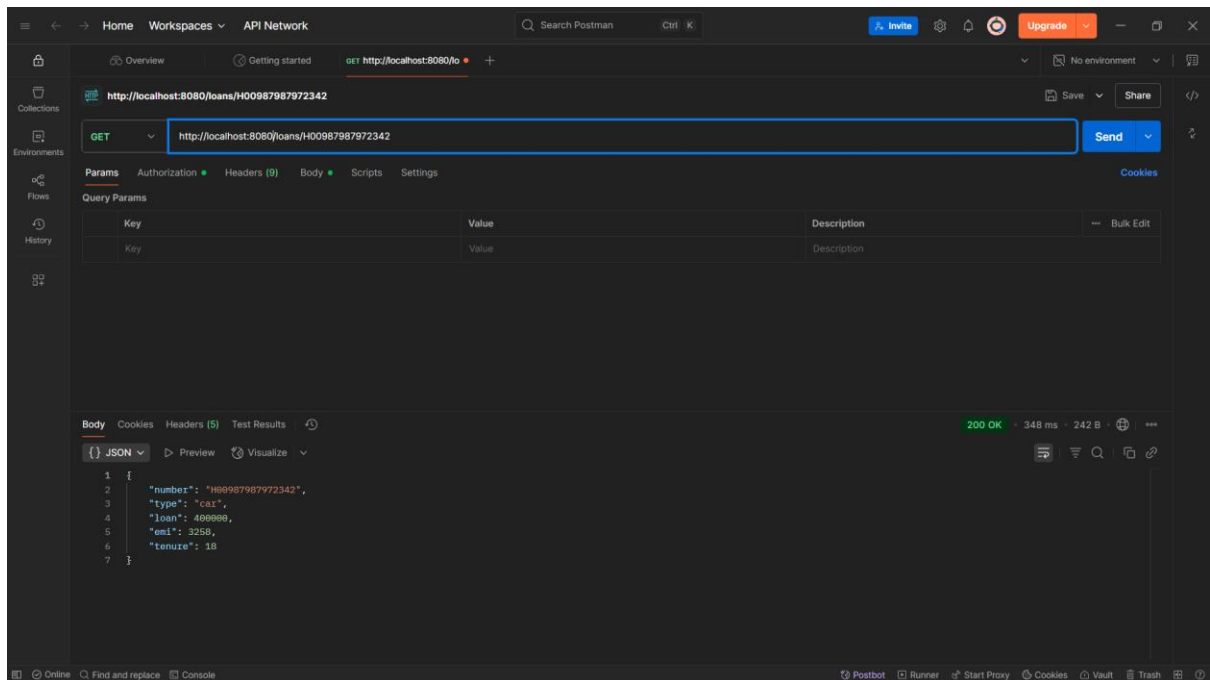
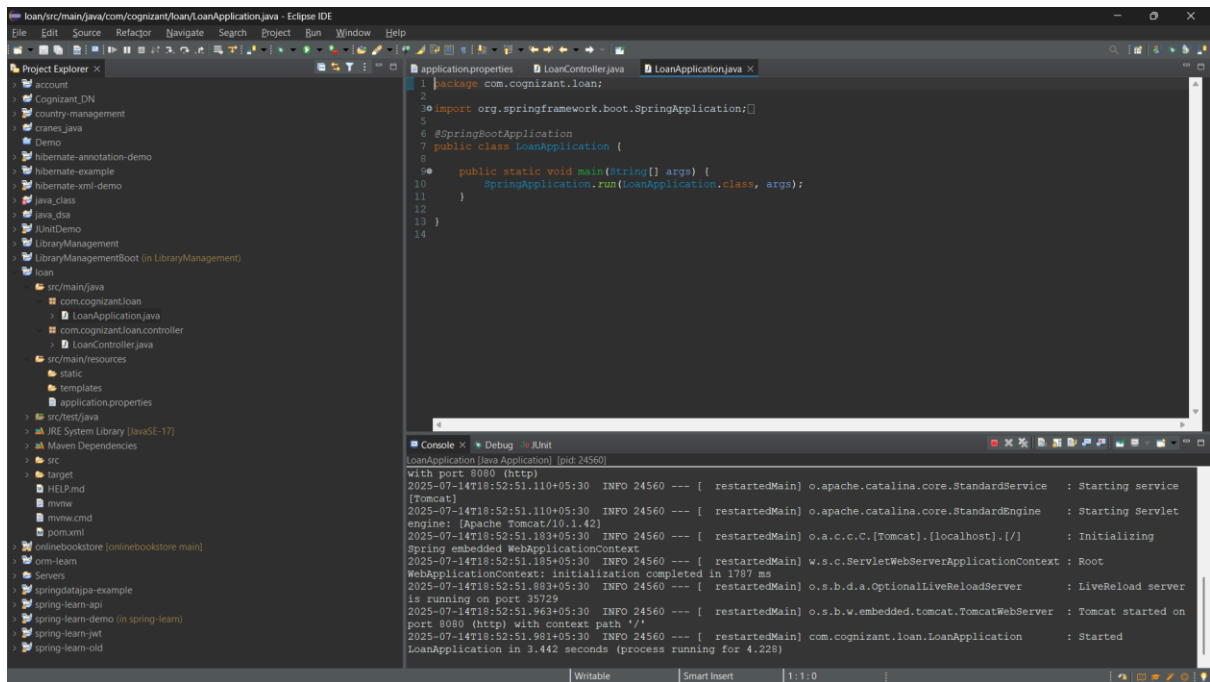
The screenshot displays the development environment for creating a microservice. The top panel shows the Project Explorer with the 'account' project structure. The main editor displays the 'AccountApplication.java' file, which is a Spring Boot application. The code defines a package 'com.cognizant.account', imports 'org.springframework.boot.SpringApplication', and defines a 'main' method to run the application.

```
1 package com.cognizant.account;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class AccountApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(AccountApplication.class, args);
10    }
11
12 }
13
14
```

The bottom panel shows the console output, indicating the application is running on port 8080. The REST client in Postman is configured to send a GET request to 'http://localhost:8080/accounts/00987987973432'. The response is a JSON object representing an account:

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

## Loan Microservice



# Create Eureka Discovery Server and register microservices

The screenshot shows the Eclipse IDE with the 'eureka-discovery-server' project. The Project Explorer on the left shows the project structure, including 'src/main/java' and 'src/main/resources'. The main editor displays the 'application.properties' file with the following configuration:

```
1 spring.application.name=eureka-discovery-server
2 server.port=8761
3
4 # Tell Eureka not to register with itself (very important)
5 eureka.client.register-with-eureka=false
6 eureka.client.fetch-registry=false
7 eureka.client.service-url.defaultZone=
8
9 # Optional: Disable Netflix debug logs
10 logging.level.com.netflix.eureka=OFF
11 logging.level.com.netflix.discovery=OFF
12
```

The Console window at the bottom shows the output of the 'mvn clean install' command, indicating a successful build and installation of the 'eureka-discovery-server' JAR file.

The screenshot shows the Spring Eureka web interface running on localhost:8761. The interface displays the following information:

### System Status

Environment	test	Current time	2025-07-14T19:45:43 +0530
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

### DS Replicas

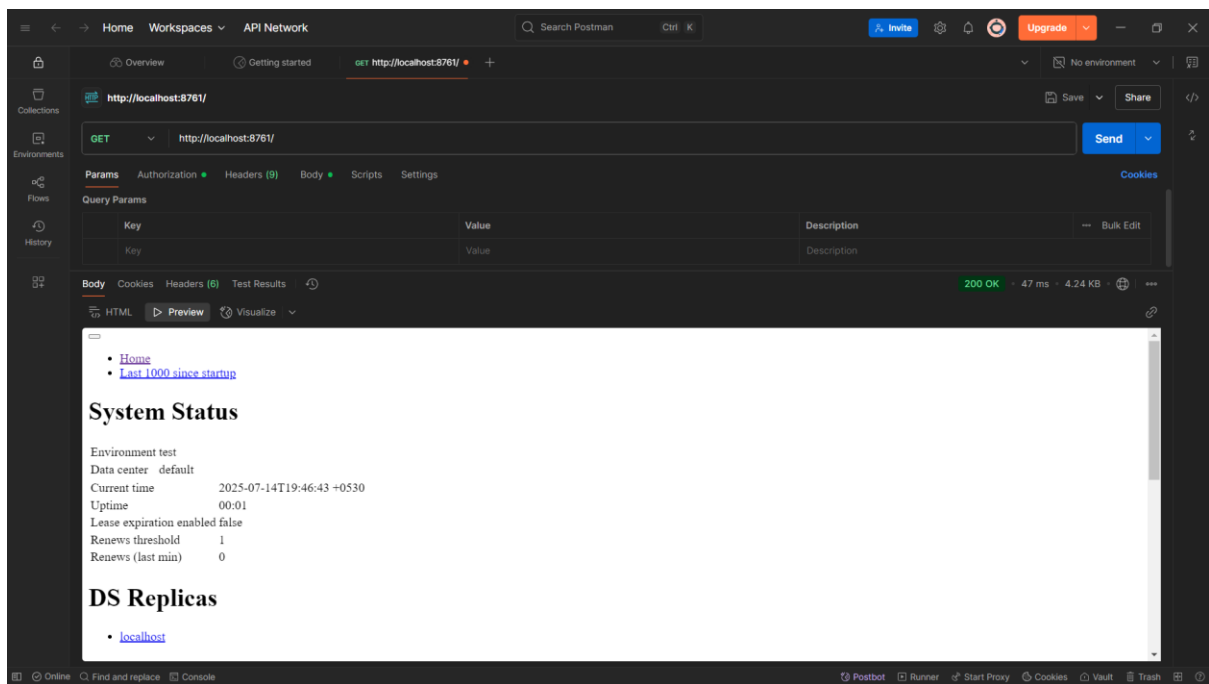
[localhost](#)

### Instances currently registered with Eureka

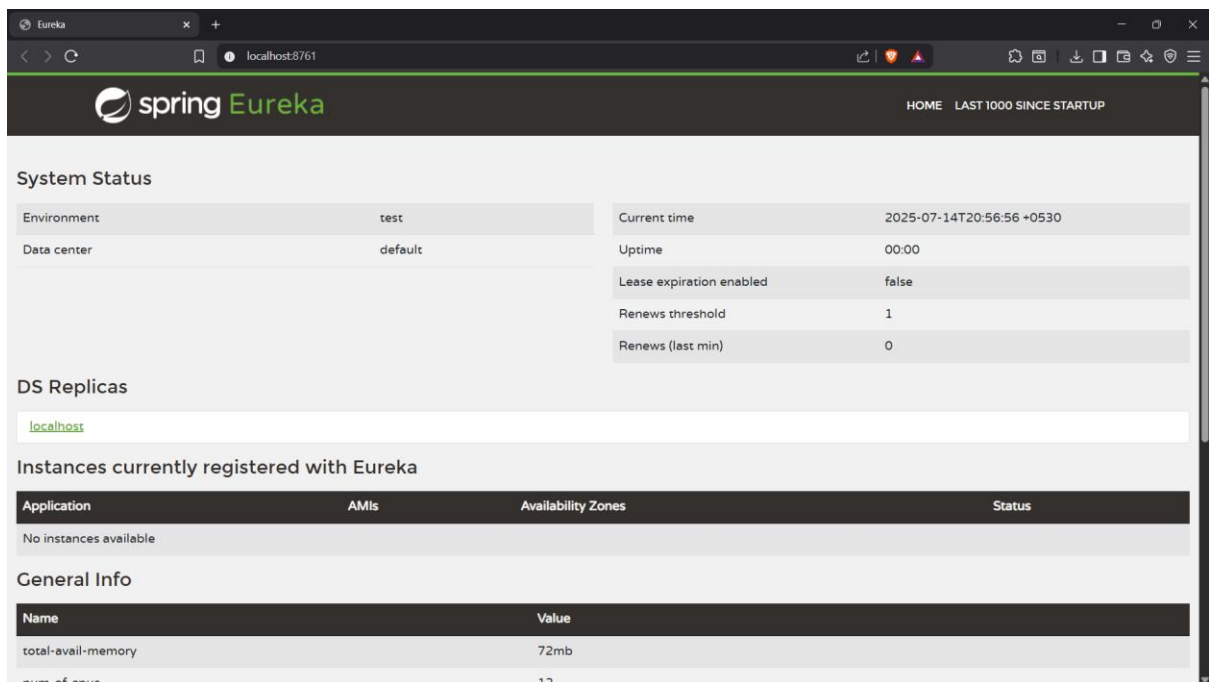
Application	AMIs	Availability Zones	Status
No instances available			

### General Info

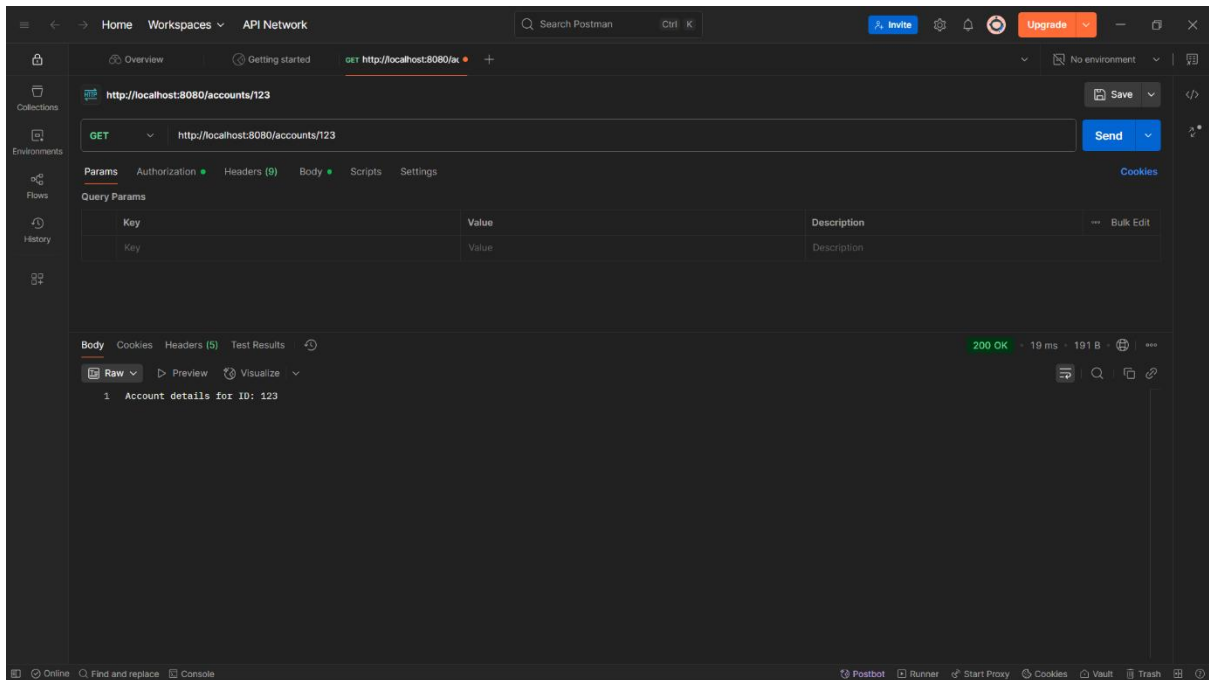
Name	Value
total-avail-memory	134mb
num-of-cpus	12



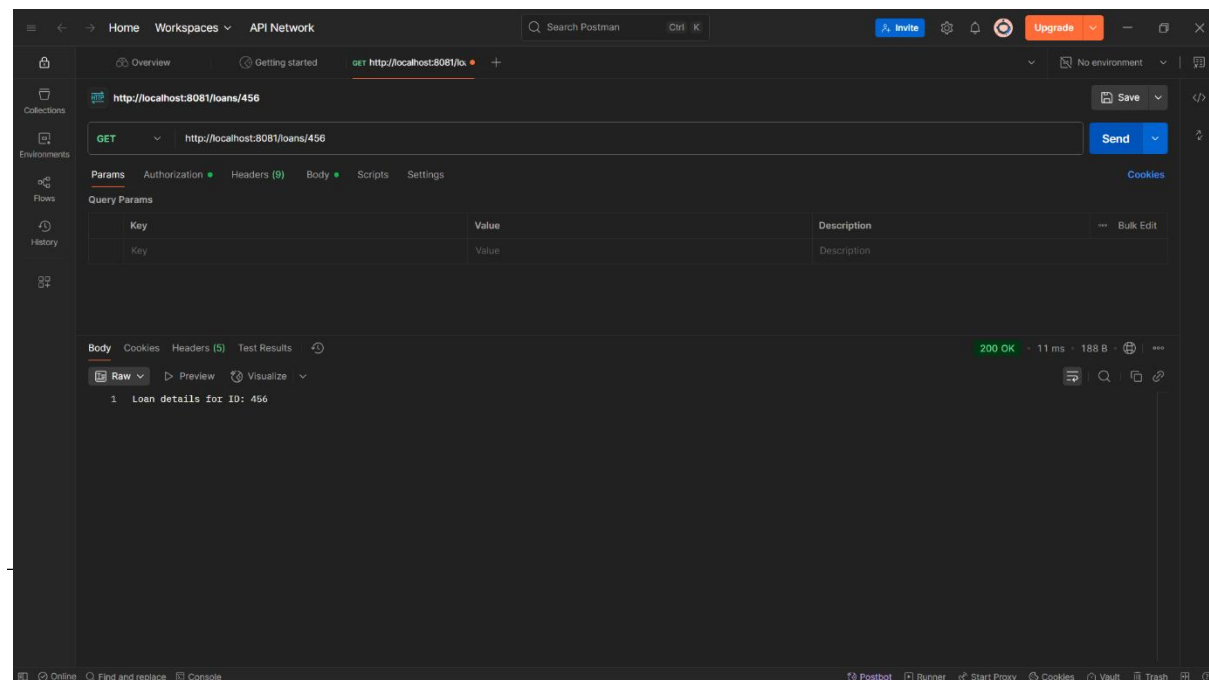
## Create and Launch Eureka Discovery Server



## Accounts:



## Loans:



**Submitted By:**

Name: Lingaraj Nayak

Superset ID: 6387607