.

**MICROSERVICES HANDS-ON SUBMISSION**

**NAME: Divyansh Tiwari**

**SUPERSET ID: 6394658**

**HANDS ON FROM: 2. Microservices with API gateway**

**Hands On : Create Microservices for Account and Loan**

**Step-by-Step Instructions**

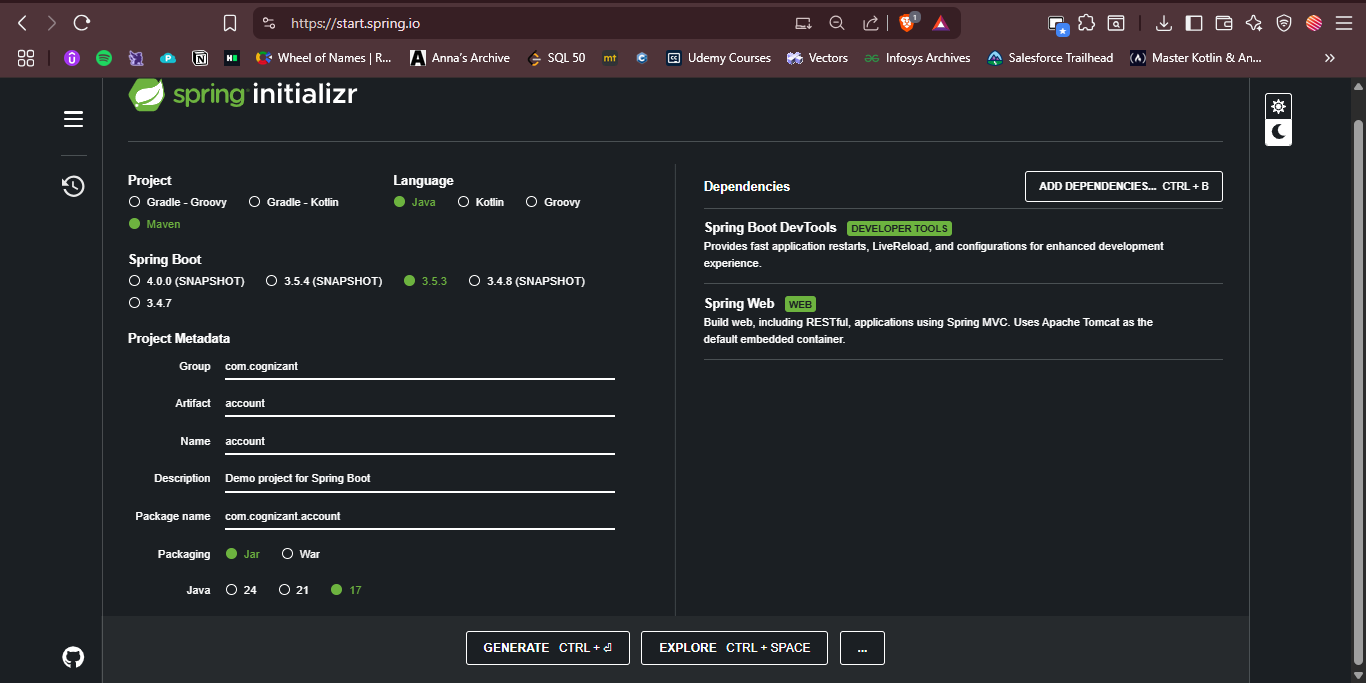
1. **Directory Setup**
   * Created a folder: D:\6394658

*I have used my SuperSet ID here, instead of the Employee ID.*

* + Created sub-folder: D:\6394658\microservices

1. **Generated Account Microservice**
   * Opened <https://start.spring.io>
   * Configured:
     + Group: com.cognizant
     + Artifact: account
     + Dependencies: Spring Web, Spring Boot DevTools
   * Extracted and placed project in D:\6394658\microservices\account

**Screenshot:** Spring Initializr for Account:



1. **Implemented REST Controller for Account**

*@RestController*

*@RequestMapping("/accounts")*

*public class AccountController {*

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

*public Account getAccount(@PathVariable String number) {*

*return new Account(number, "savings", 234343);*

*}*

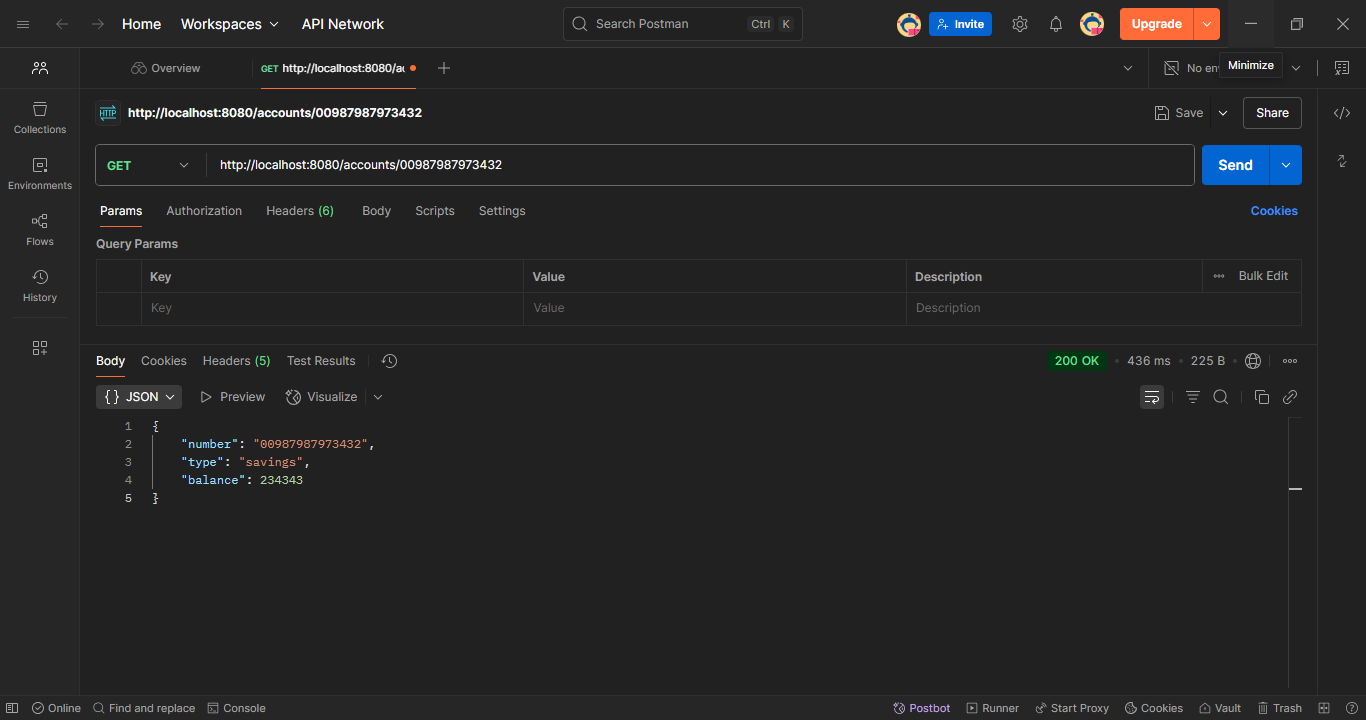
*}*

* + Created Account POJO class with fields: number, type, balance.

1. **Set Port in application.properties**

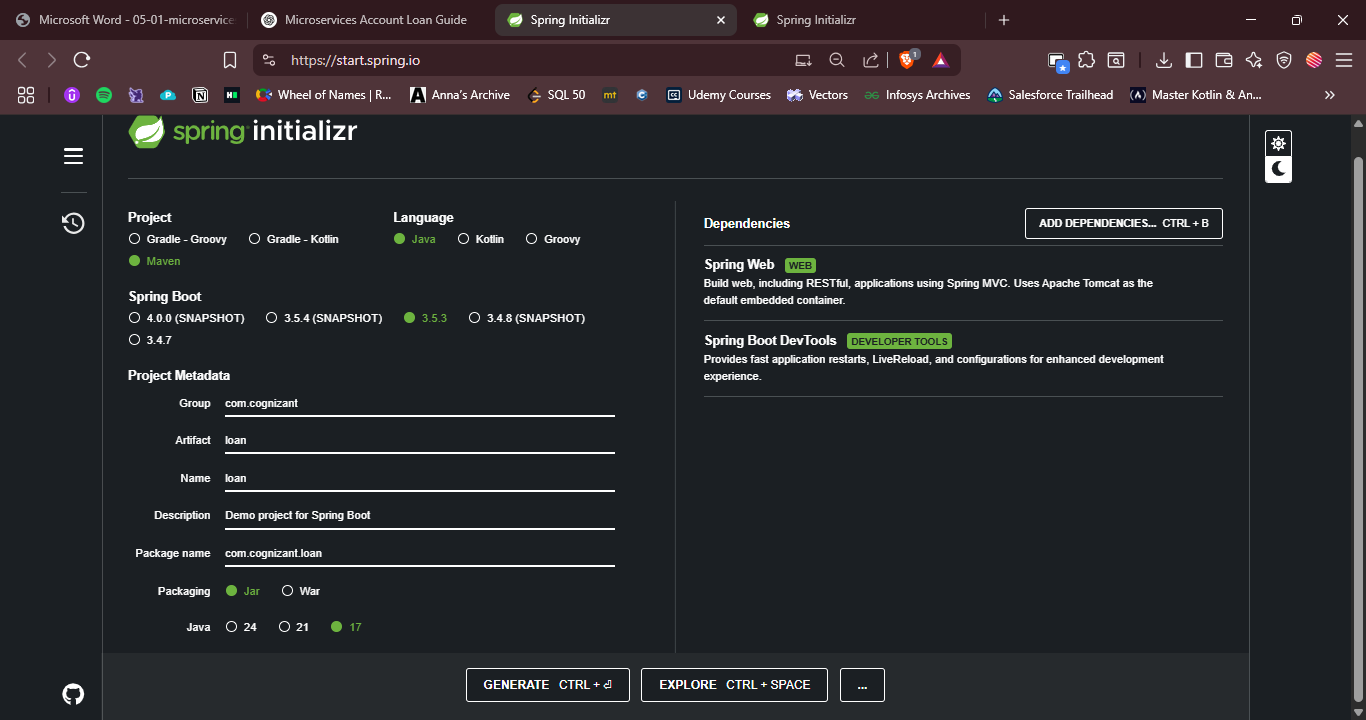
*server.port=8080*

* + **Screenshot**: Account Controller Output in Postman

****

1. **Generated Loan Microservice**
   * Followed same steps as Account microservice
   * Set different port:

*server.port=8081*

**Screenshot: Loan Microservice Spring Intializr****

1. **Implemented REST Controller for Loan**

*@RestController*

*@RequestMapping("/loans")*

*public class LoanController {*

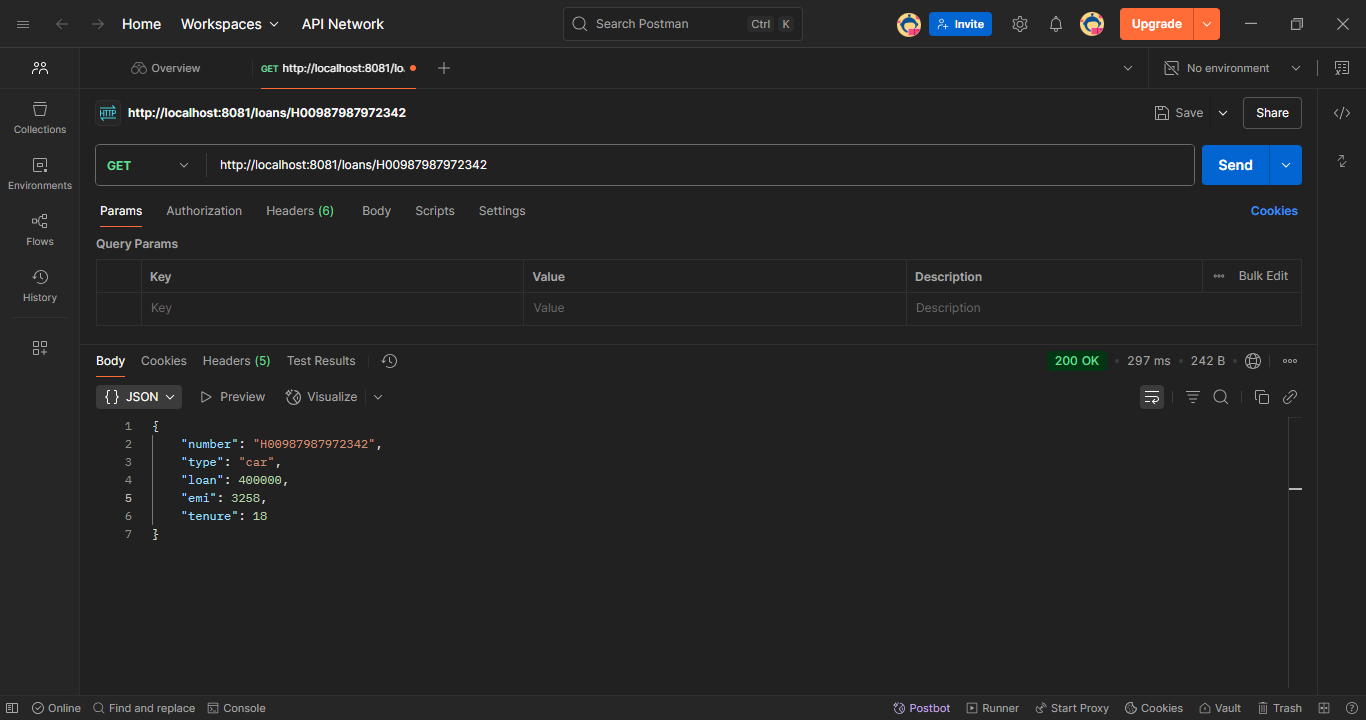
*@GetMapping("/{number}")*

*public Loan getLoan(@PathVariable String number) {*

*return new Loan(number, "car", 400000, 3258, 18);*

*}*

*}*

* + Created Loan POJO class with fields: number, type, loan, emi, tenure.
  + **Screenshot 2**: Loan Controller Output in Postman

1. **Create Eureka Discovery Server**

* Started <https://start.spring.io>
  + Group: com.cognizant
  + Artifact: eureka-discovery-server
  + Dependencies: Eureka Server
* Opened in IntelliJ, and add @EnableEurekaServer to the main class

*@EnableEurekaServer*

*@SpringBootApplication*

*public class EurekaDiscoveryServerApplication {*

*public static void main(String[] args) {*

*SpringApplication.run(EurekaDiscoveryServerApplication.class, args);*

*}*

*}*

* In application.properties:

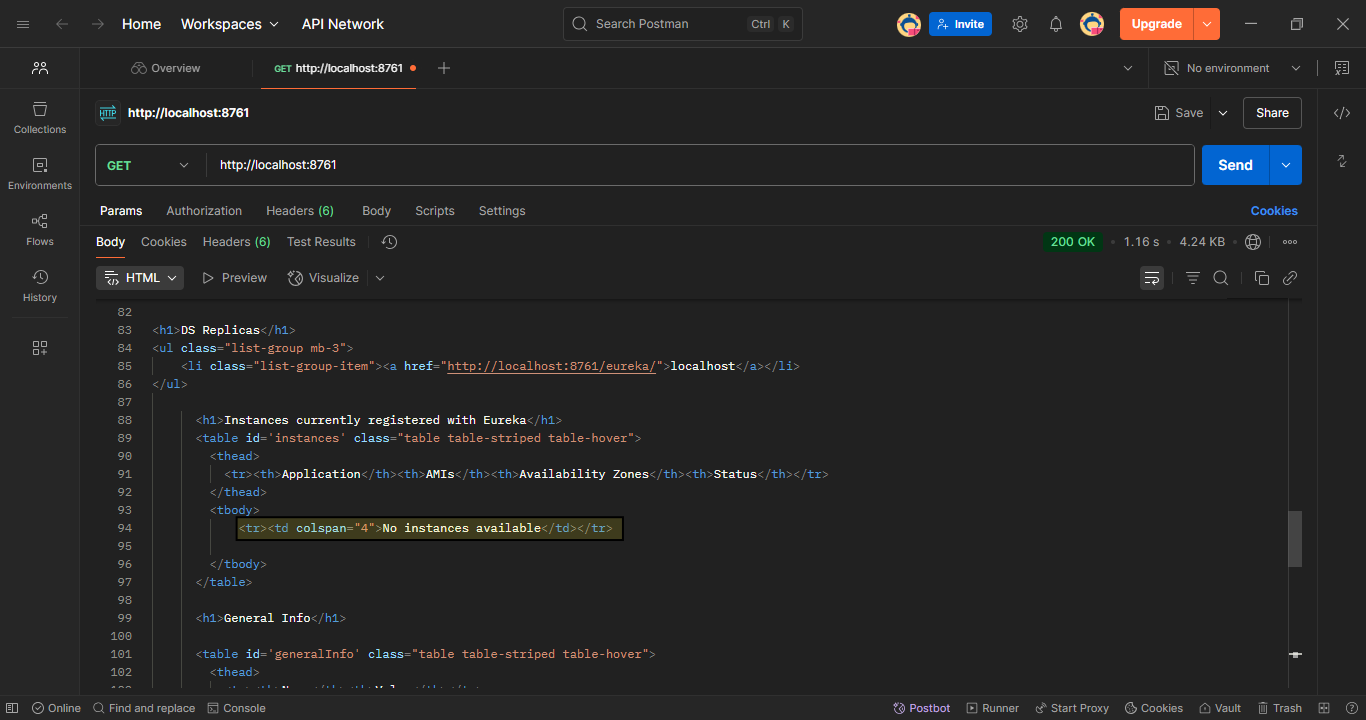
*server.port=8761*

*eureka.client.register-with-eureka=false*

*eureka.client.fetch-registry=false*

* Run and go to:  
  [*http://localhost:8761*](http://localhost:8761)

**Screenshot**: Eureka dashboard showing **no instances**

****

1. **Register Account & Loan Microservices**

**Updated pom.xml for both:**

* Added Eureka Discovery Client dependency.
* Added spring-cloud-dependencies in <dependencyManagement>.

*<dependency>*

*<groupId>org.springframework.cloud</groupId>*

*<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>*

*</dependency>*

<*dependencyManagement>*

*<dependencies>*

*<dependency>*

*<groupId>org.springframework.cloud</groupId>*

*<artifactId>spring-cloud-dependencies</artifactId>*

*<version>2022.0.4</version>*

*<type>pom</type>*

*<scope>import</scope>*

*</dependency>*

*</dependencies>*

*</dependencyManagement>*

**In both microservices:**

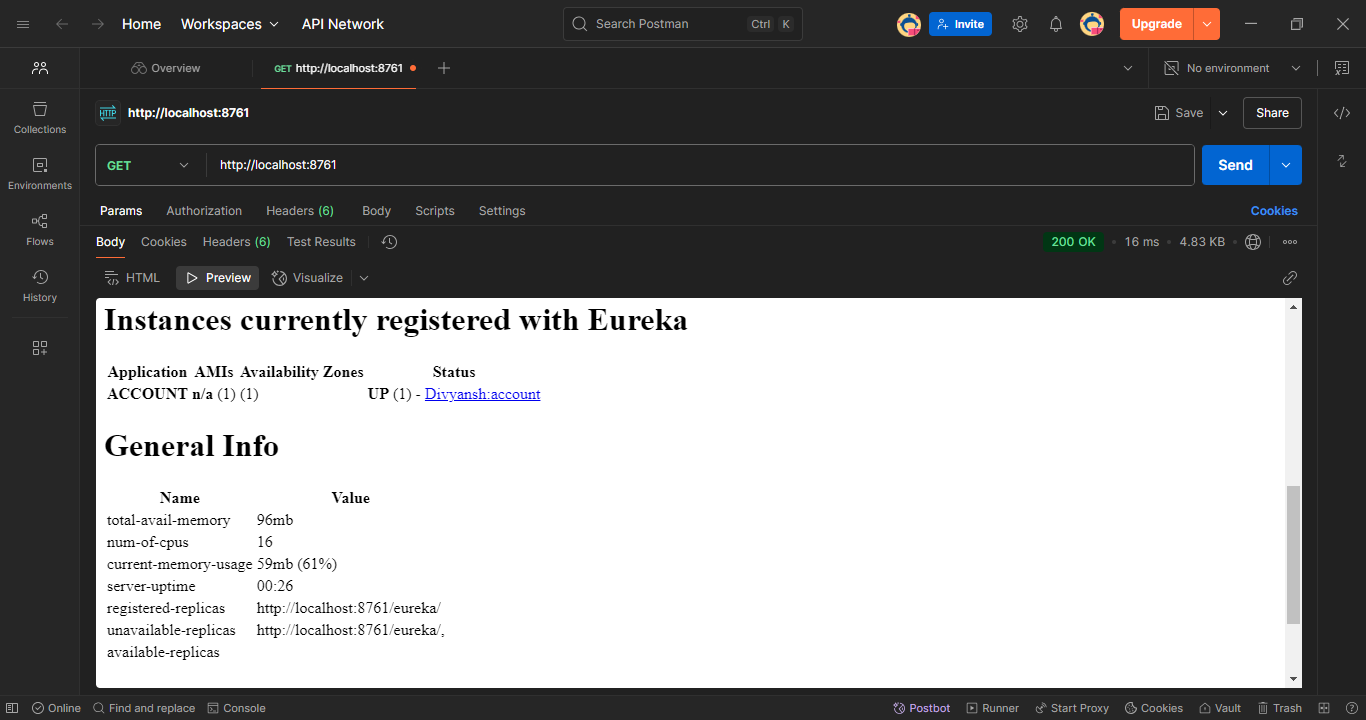
1. Add @EnableDiscoveryClient to the main class.
2. Add to application.properties:

**Account:**

*spring.application.name=account-service*

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

**Eureka with ACCOUNT-SERVICE Registered:**

****

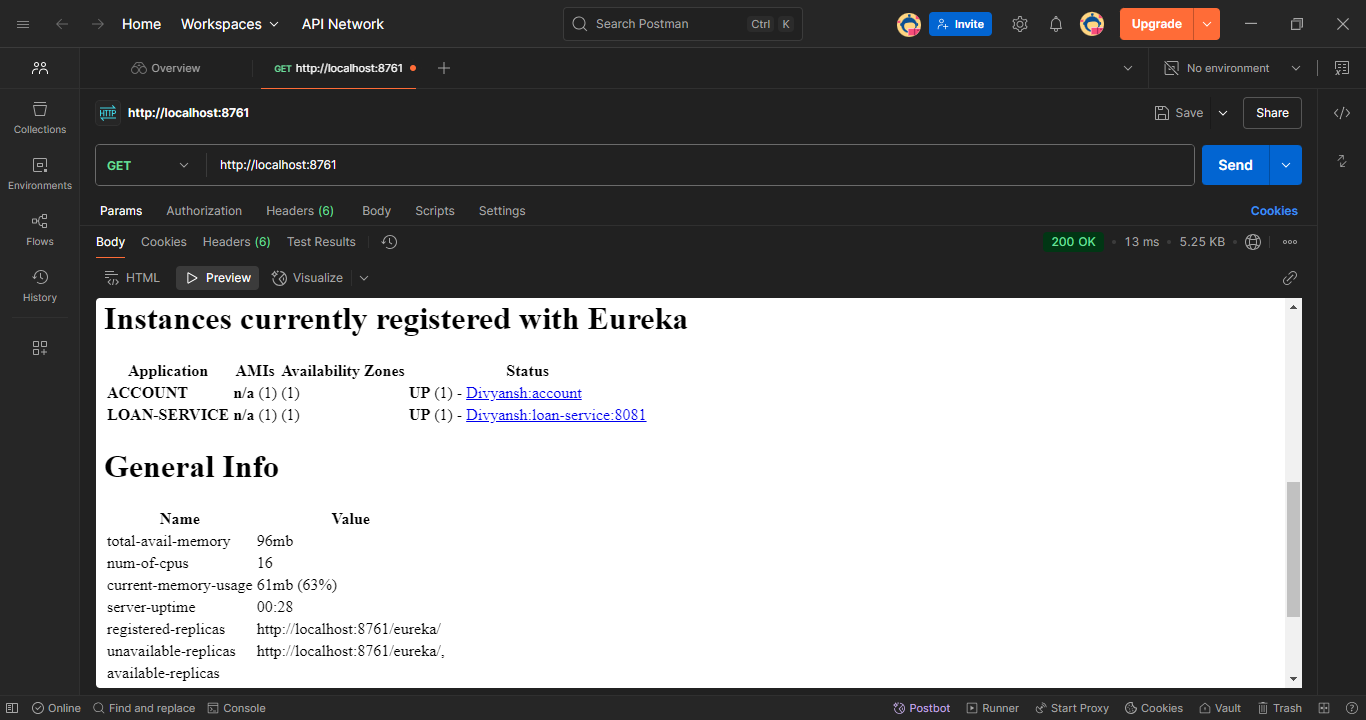
**Loan:**

spring.application.name=loan-service

server.port=8081

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

**Eureka with ACCOUNT-SERVICE Registered:**

****