**Microservices with API Gateway - Assignment**

**MODULE 1:** **Creating Microservices for account and loan**

**Exercise 1: Account Microservice**

**Filename:**

/account/src/main/java/com/cognizant/account/AccountApplication.java

**Code:**

**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);

}

}

**Controller:**

/account/src/main/java/com/cognizant/account/controller/AccountController.java

package com.cognizant.account.controller;

import com.cognizant.account.model.Account;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{accountNumber}")

public Account getAccount(@PathVariable String accountNumber) {

return new Account(accountNumber, "savings", 234343);

}

}

**Model:**

/account/src/main/java/com/cognizant/account/model/Account.java

**package** com.cognizant.account.model;

**public** **class** Account {

**private** String number;

**private** String type;

**private** **int** balance;

**public** Account() {}

**public** Account(String number, String type, **int** balance) {

**this**.number = number;

**this**.type = type;

**this**.balance = balance;

}

**public** String getNumber() {

**return** number;

}

**public** **void** setNumber(String number) {

**this**.number = number;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** **int** getBalance() {

**return** balance;

}

**public** **void** setBalance(**int** balance) {

**this**.balance = balance;

}

}

**application.properties:**

spring.application.name=account

server.port=8081

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

**Sample Output:**

Endpoint: http://localhost:8081/accounts/1234

**Response:**

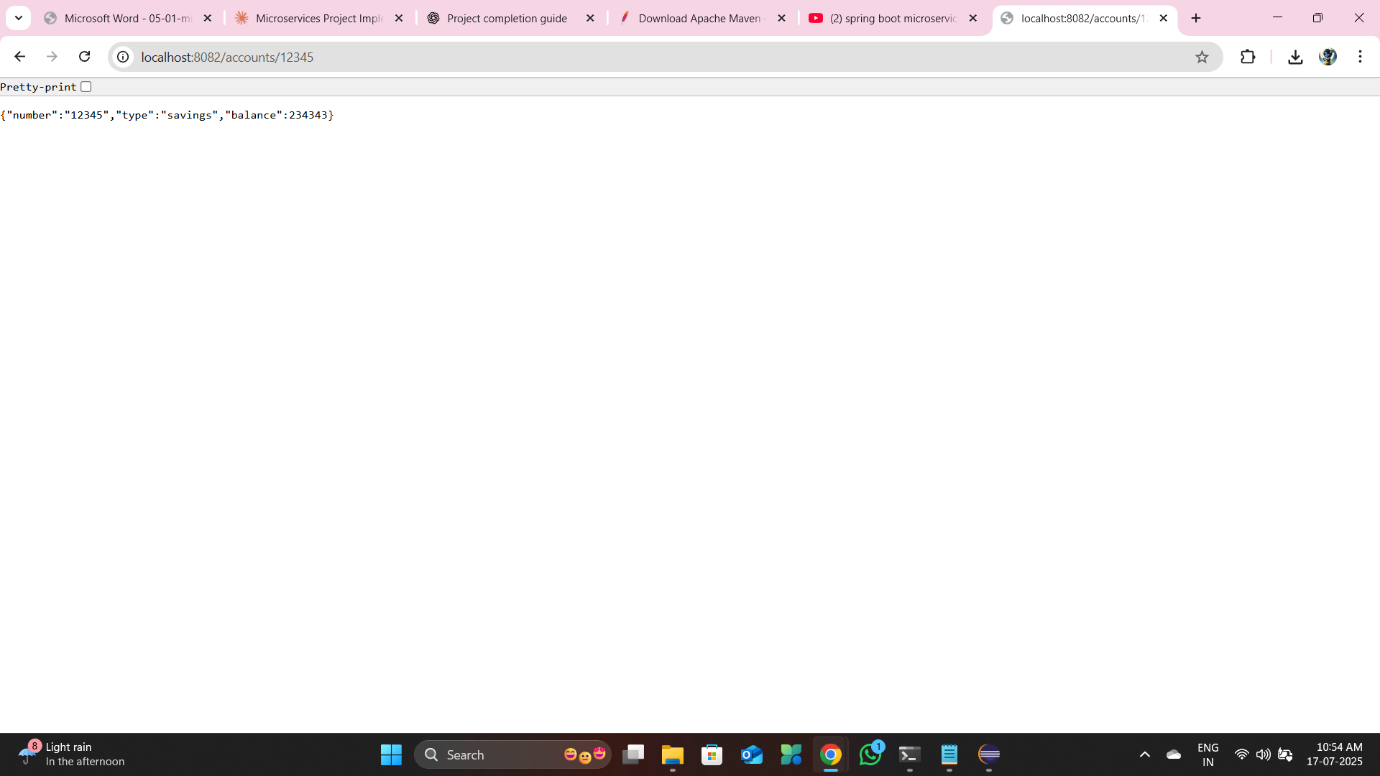
{

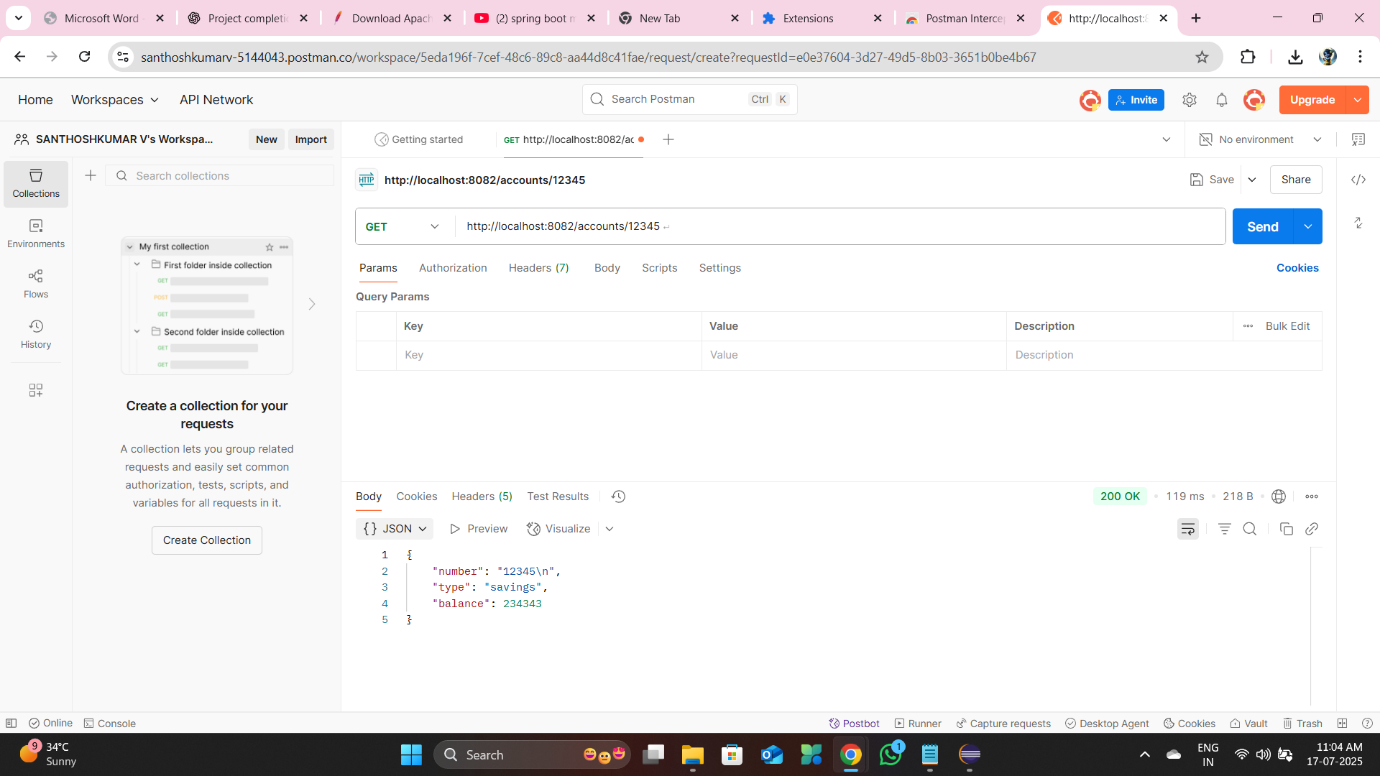
"number": "12345",

"type": "savings",

"balance": 234343

}





**Exercise 2: Loan Microservice**

**Filename:**

/loan/src/main/java/com/cognizant/loan/LoanServiceApplication.java

**Code:**

**package** com.cognizant.loan;

**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);

}

}

**Controller:**

/loan/src/main/java/com/cognizant/loan/controller/LoanController.java

package com.cognizant.loan.controller;

import com.cognizant.loan.model.Loan;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{loanNumber}")

public Loan getLoan(@PathVariable String loanNumber) {

return new Loan(loanNumber, "personal", 50000);

}

}

**Model:**

/loan/src/main/java/com/cognizant/loan/model/Loan.java

**package** com.cognizant.loan.model;

**public** **class** Loan {

**private** String number;

**private** String type;

**private** **int** amount;

**public** Loan() {}

**public** Loan(String number, String type, **int** amount) {

**this**.number = number;

**this**.type = type;

**this**.amount = amount;

}

**public** String getNumber() {

**return** number;

}

**public** **void** setNumber(String number) {

**this**.number = number;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** **int** getAmount() {

**return** amount;

}

**public** **void** setAmount(**int** amount) {

**this**.amount = amount;

}

}

**application.properties:**

spring.application.name=loan

server.port=8083

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

**Sample Output:**

Endpoint: http://localhost:8083/loans/98765

**Response:**

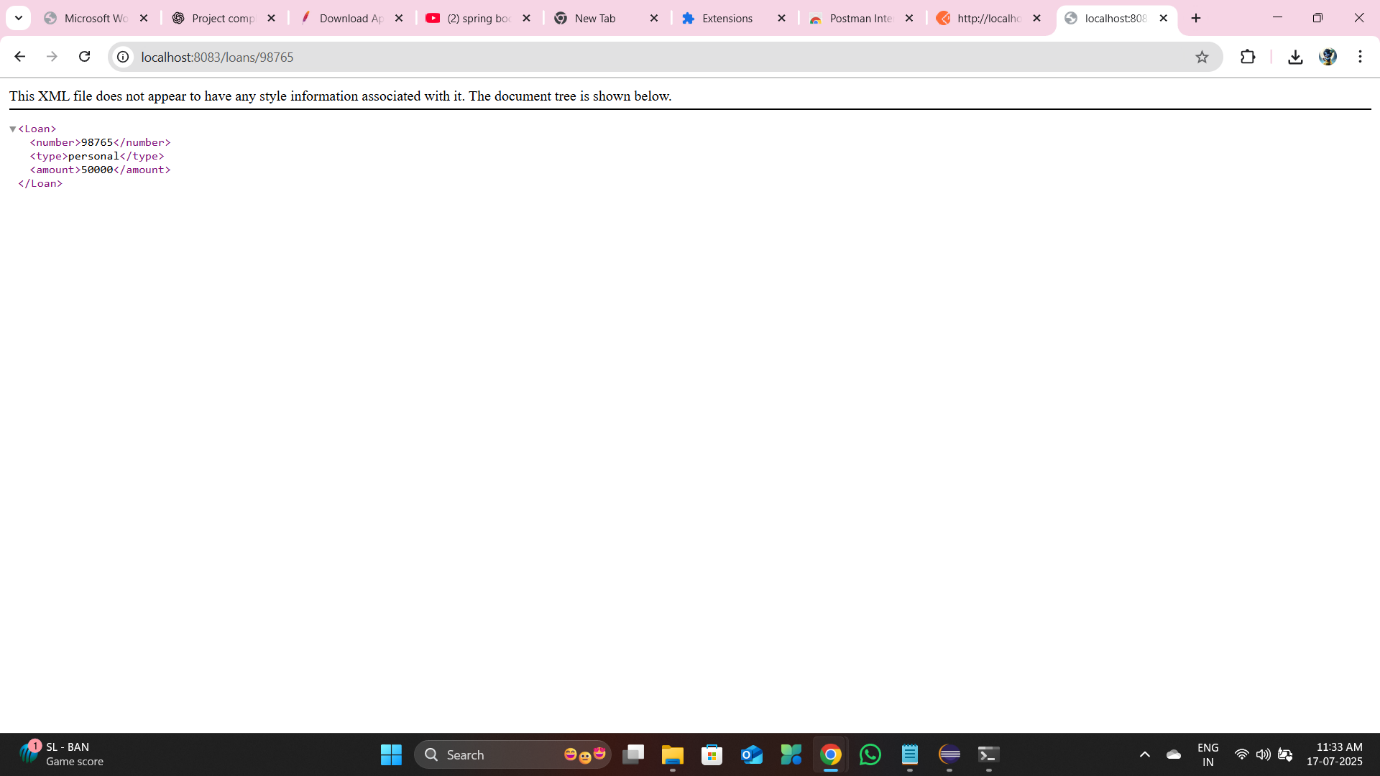
{

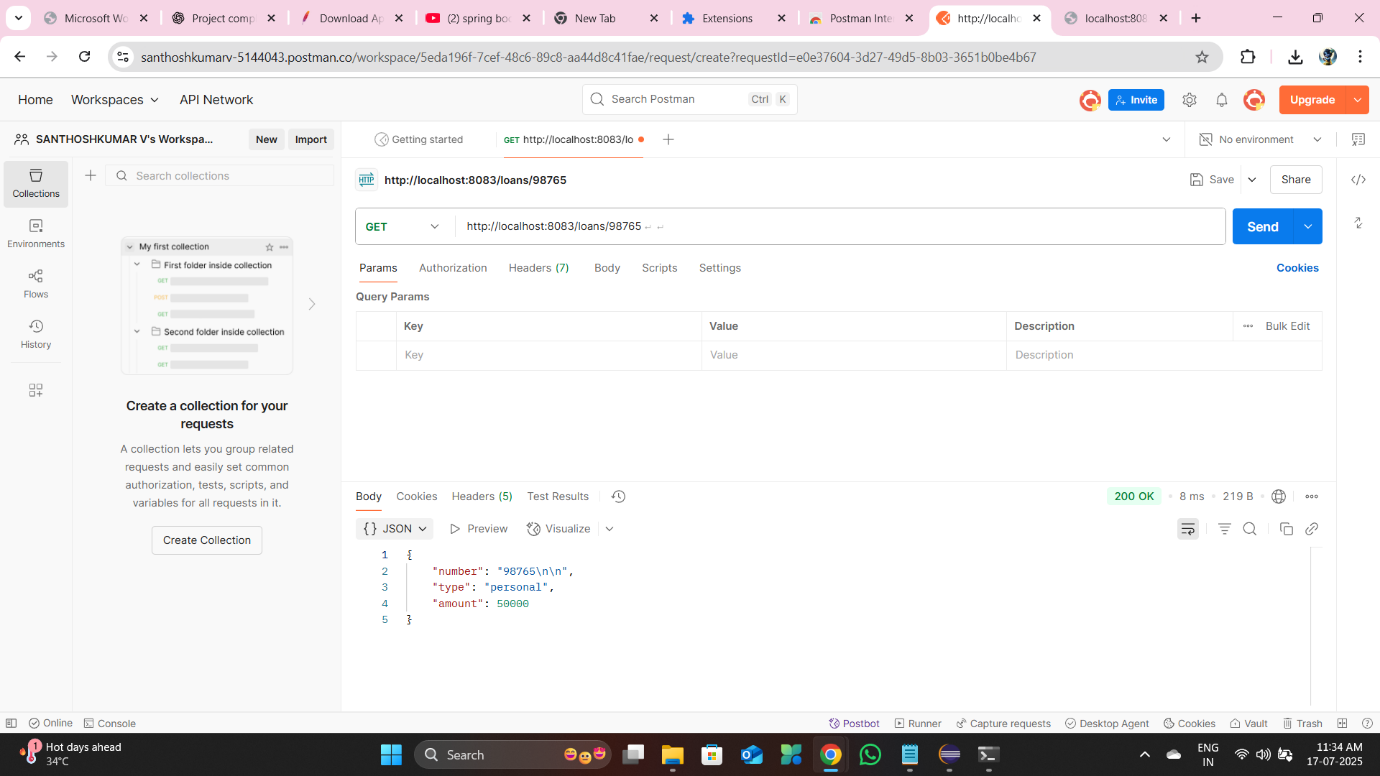
"number": "98765",

"type": "personal",

"amount": 50000

}





**MODULE 2:** **Create Eureka Discovery Server and register**

**Exercise 1: Discovery Server Setup**

**Filename:**

/discovery-server/src/main/java/com/cognizant/discoveryserver/DiscoveryServerApplication.java

**Code:**

package com.cognizant.discoveryserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class DiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(DiscoveryServerApplication.class, args);

}

}

**application.properties:**

spring.application.name=discoveryserver

server.port=8761

eureka.client.register-with-eureka=false

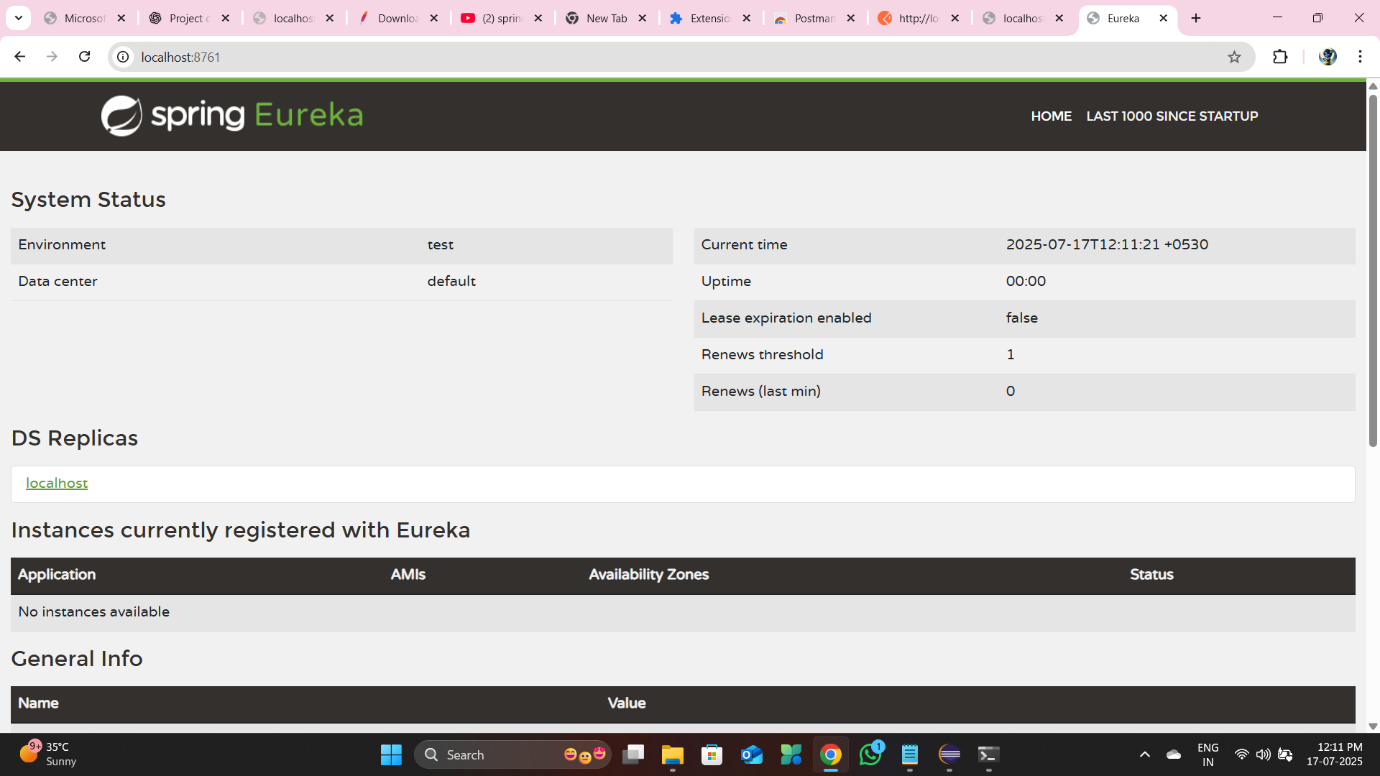
eureka.client.fetch-registry=false

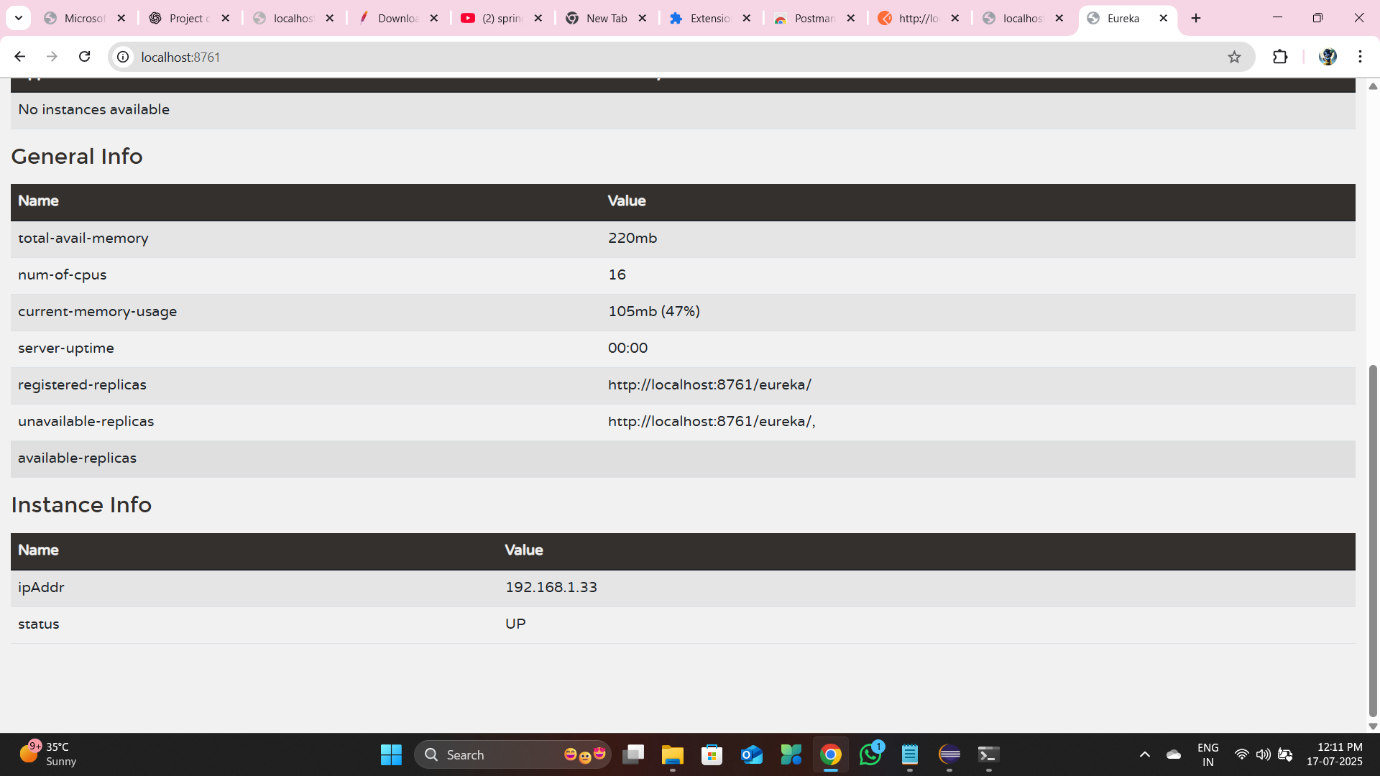
**Sample Output:**

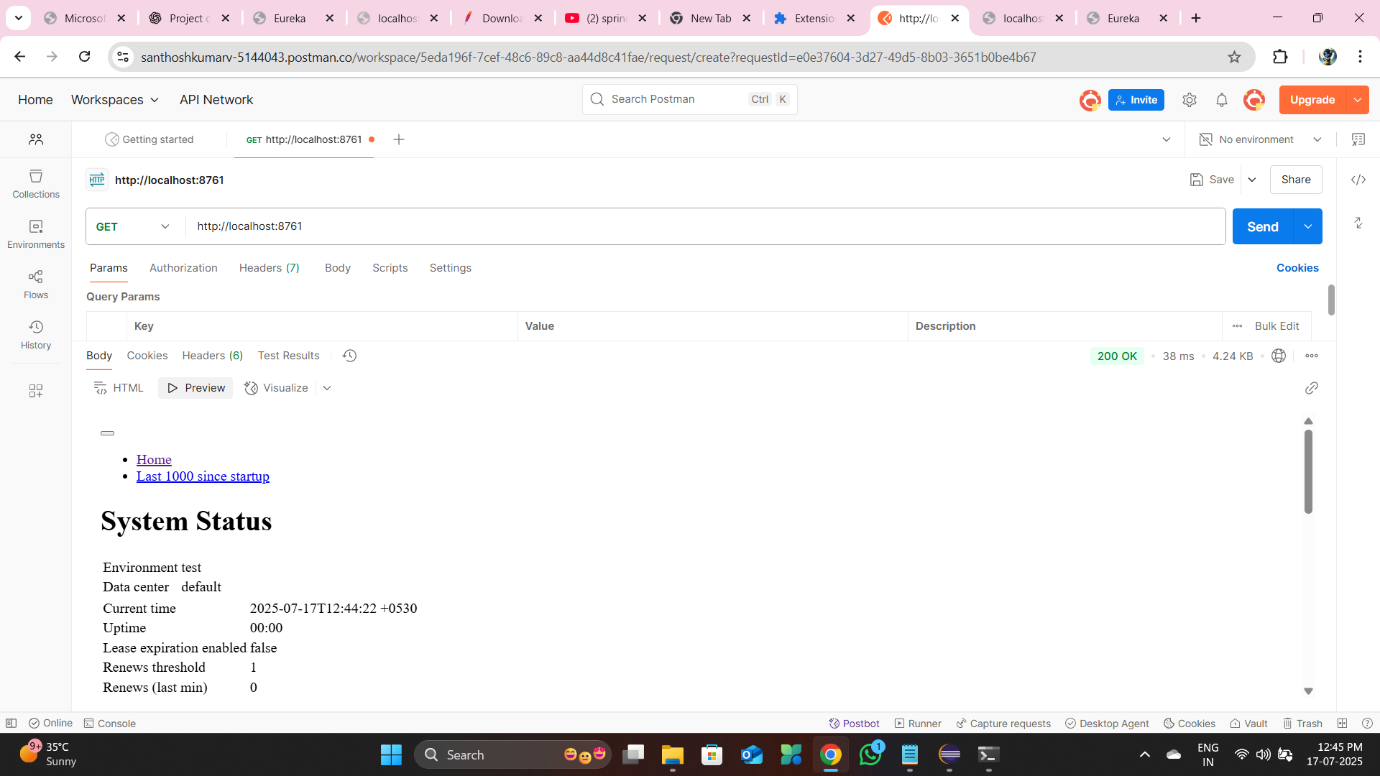
Open [http://localhost:8761](http://localhost:8761/) in browser.

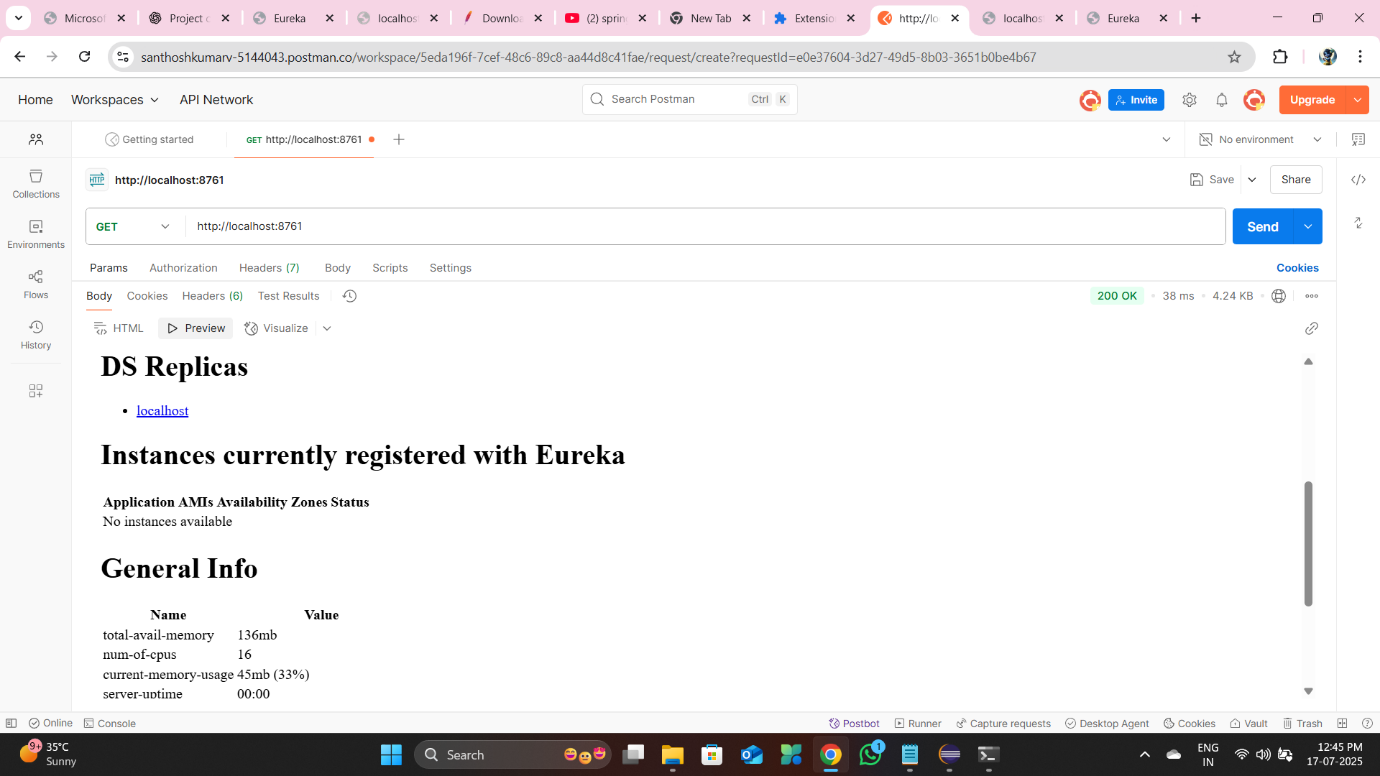
You will see:

Eureka Dashboard - listing registered services









**Exercise 2: Account Microservice(Register)**

**Filename:**

/account/src/main/java/com/cognizant/account/AccountApplication.java

**Code:**

**package** com.cognizant.account;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

**public** **class** AccountApplication {

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

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

}

}

**Controller:**

/account/src/main/java/com/cognizant/account/controller/AccountController.java

package com.cognizant.account.controller;

import com.cognizant.account.model.Account;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{accountNumber}")

public Account getAccount(@PathVariable String accountNumber) {

return new Account(accountNumber, "savings", 234343);

}

}

**Model:**

/account/src/main/java/com/cognizant/account/model/Account.java

**package** com.cognizant.account.model;

**public** **class** Account {

**private** String number;

**private** String type;

**private** **int** balance;

**public** Account() {}

**public** Account(String number, String type, **int** balance) {

**this**.number = number;

**this**.type = type;

**this**.balance = balance;

}

**public** String getNumber() {

**return** number;

}

**public** **void** setNumber(String number) {

**this**.number = number;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** **int** getBalance() {

**return** balance;

}

**public** **void** setBalance(**int** balance) {

**this**.balance = balance;

}

}

**application.properties:**

spring.application.name=account

server.port=8081

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

**Sample Output:**

Endpoint: http://localhost:8081/accounts/1234

**Response:**

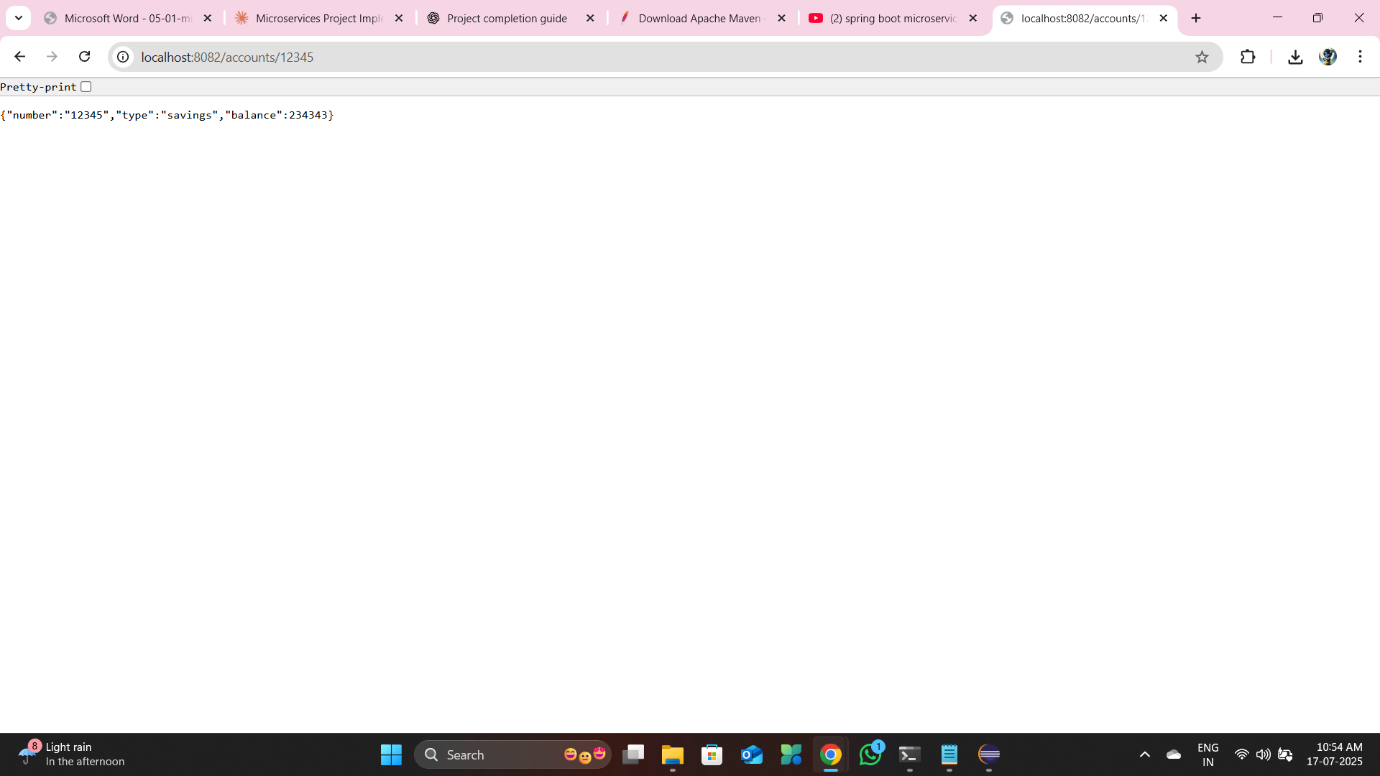
{

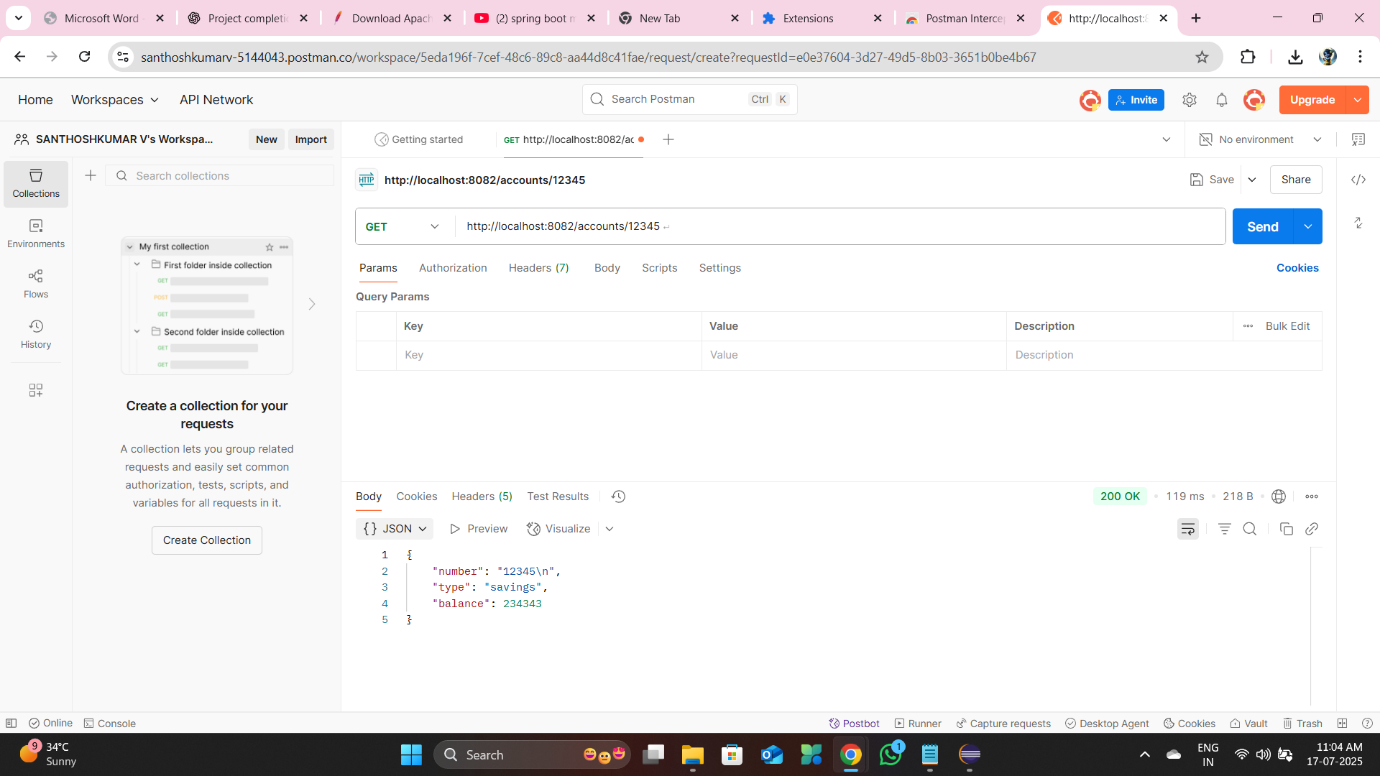
"number": "12345",

"type": "savings",

"balance": 234343

}





**Exercise 3: Loan Microservice (Register)**

**Filename:**

/loan/src/main/java/com/cognizant/loan/LoanServiceApplication.java

**Code:**

**package** com.cognizant.loan;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

**public** **class** LoanServiceApplication {

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

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

}

}

**Controller:**

/loan/src/main/java/com/cognizant/loan/controller/LoanController.java

package com.cognizant.loan.controller;

import com.cognizant.loan.model.Loan;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{loanNumber}")

public Loan getLoan(@PathVariable String loanNumber) {

return new Loan(loanNumber, "personal", 50000);

}

}

**Model:**

/loan/src/main/java/com/cognizant/loan/model/Loan.java

**package** com.cognizant.loan.model;

**public** **class** Loan {

**private** String number;

**private** String type;

**private** **int** amount;

**public** Loan() {}

**public** Loan(String number, String type, **int** amount) {

**this**.number = number;

**this**.type = type;

**this**.amount = amount;

}

**public** String getNumber() {

**return** number;

}

**public** **void** setNumber(String number) {

**this**.number = number;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** **int** getAmount() {

**return** amount;

}

**public** **void** setAmount(**int** amount) {

**this**.amount = amount;

}

}

**application.properties:**

spring.application.name=loan

server.port=8083

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

**Sample Output:**

Endpoint: http://localhost:8083/loans/98765

**Response:**

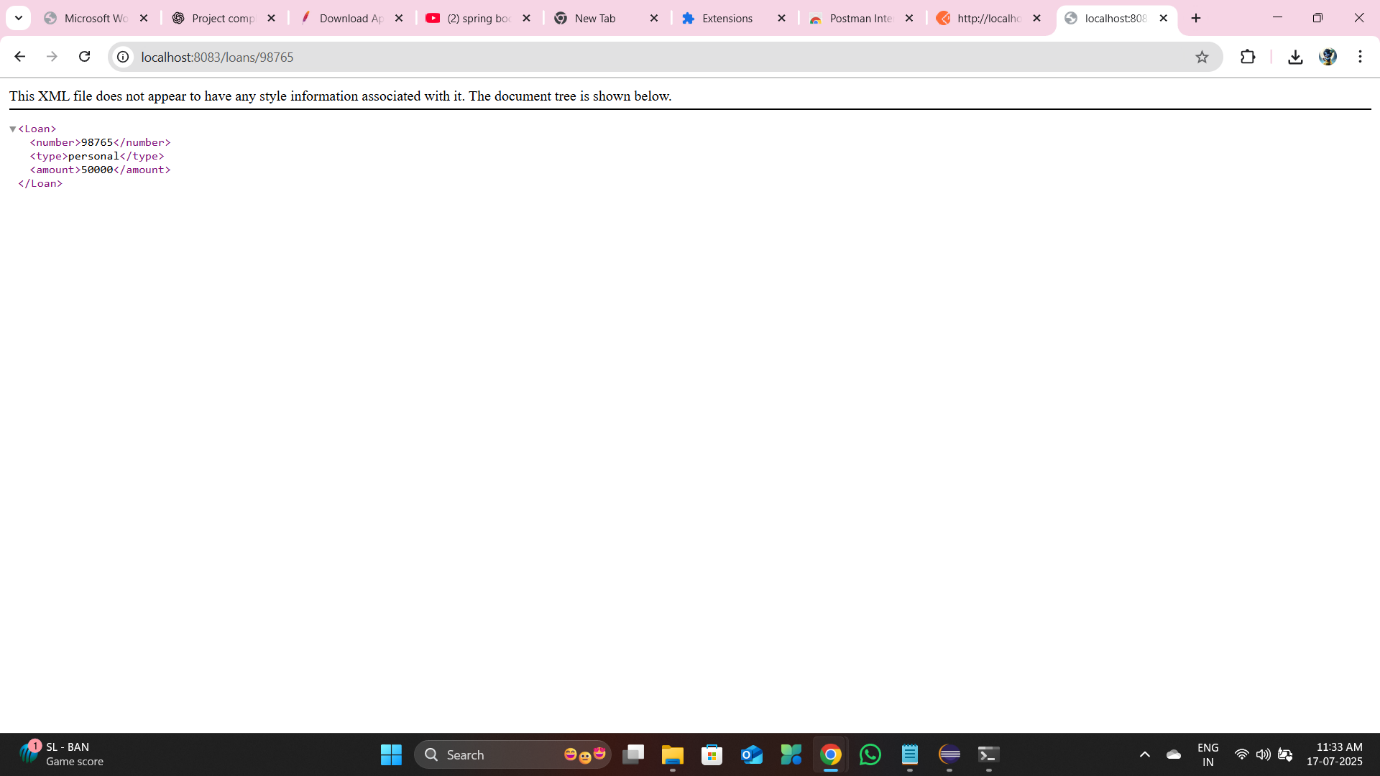
{

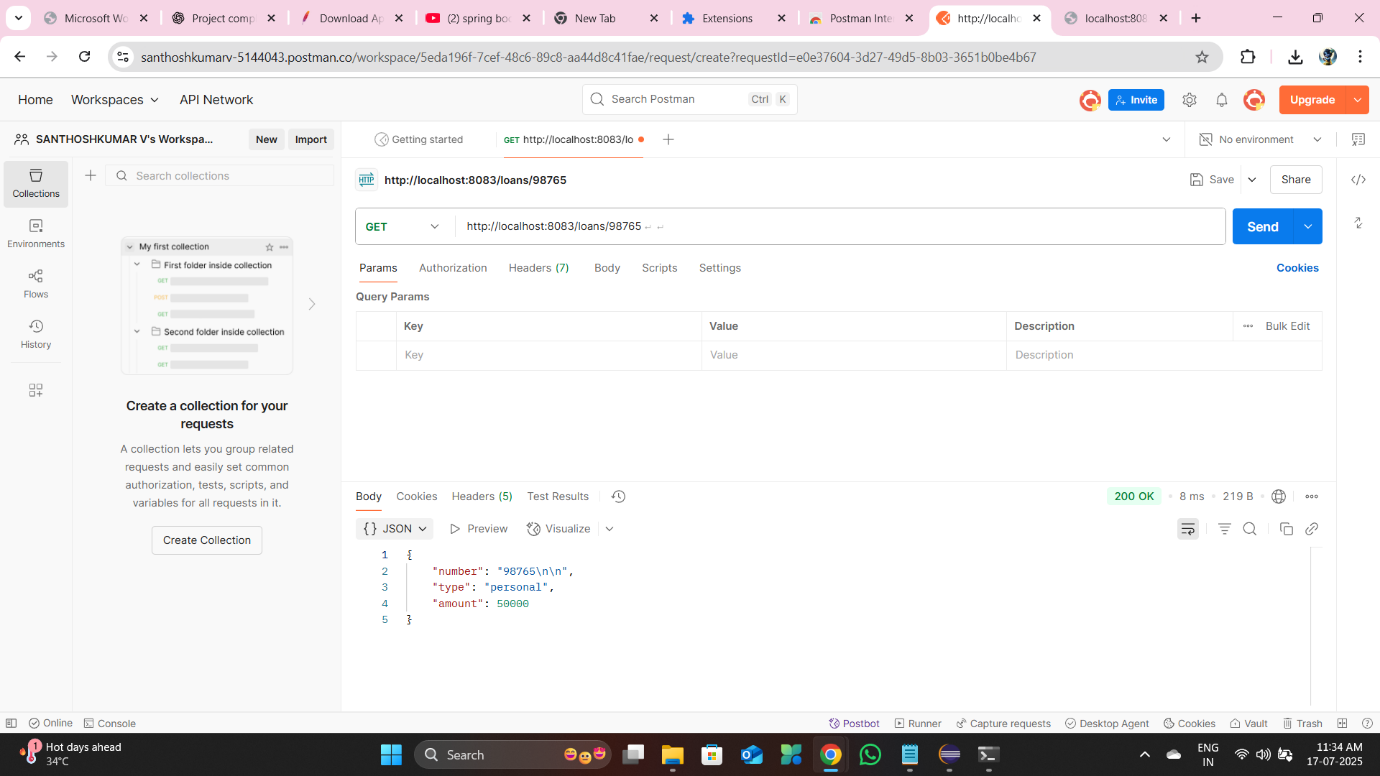
"number": "98765",

"type": "personal",

"amount": 50000

}





**How to Run Each Module**

**1. Start Eureka Discovery Server**

cd discovery-server

mvn clean install

java -jar target/discovery-server-0.0.1-SNAPSHOT.jar

**2. Start Account Service**

cd account

mvn clean install

java -jar target/account-0.0.1-SNAPSHOT.jar

**3. Start Loan Service**

cd loan

mvn clean install

java -jar target/loan-0.0.1-SNAPSHOT.jar

**Note:**

Ensure ports 8761, 8081, and 8083 are free.  
if not kill the task with the help this command  
D:\microservices\account>**netstat -ano | findstr :8082**

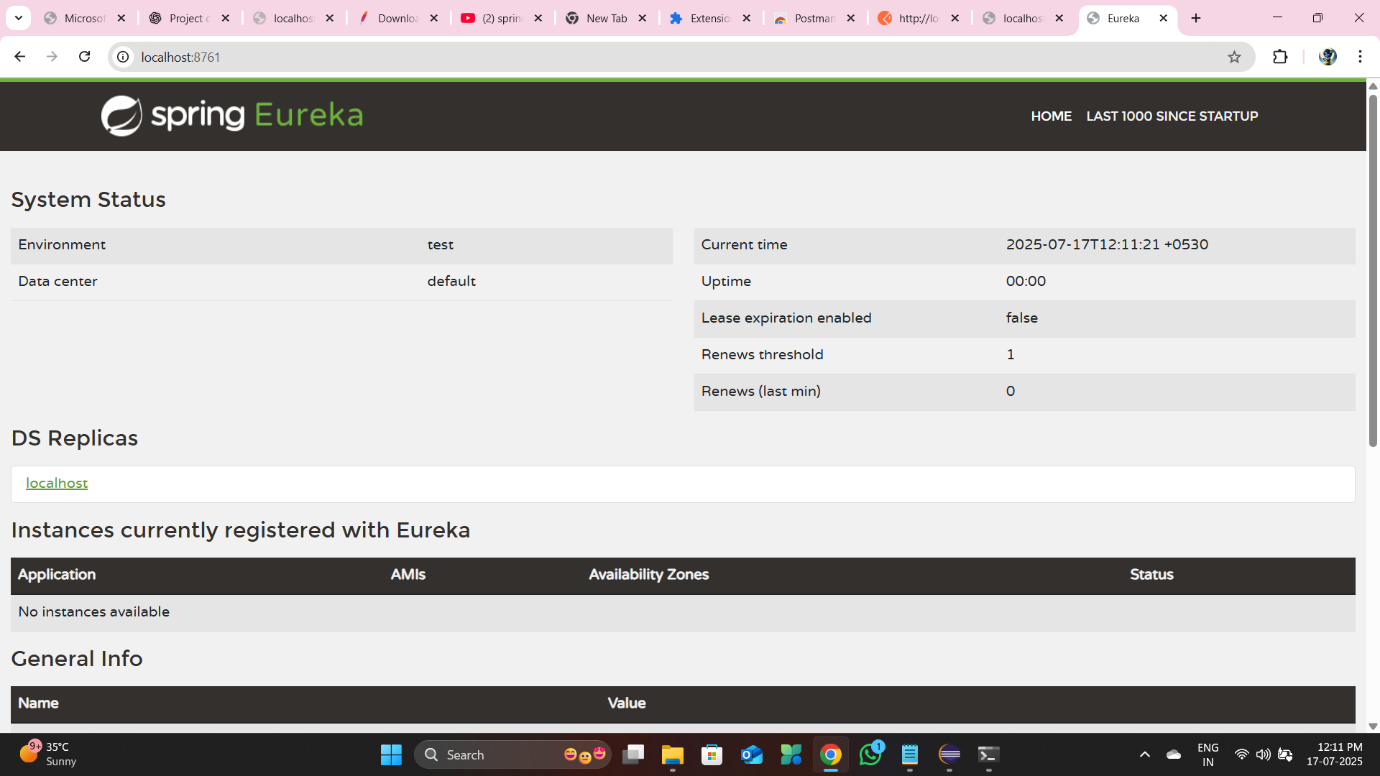
TCP 0.0.0.0:8082 0.0.0.0:0 LISTENING **11332**

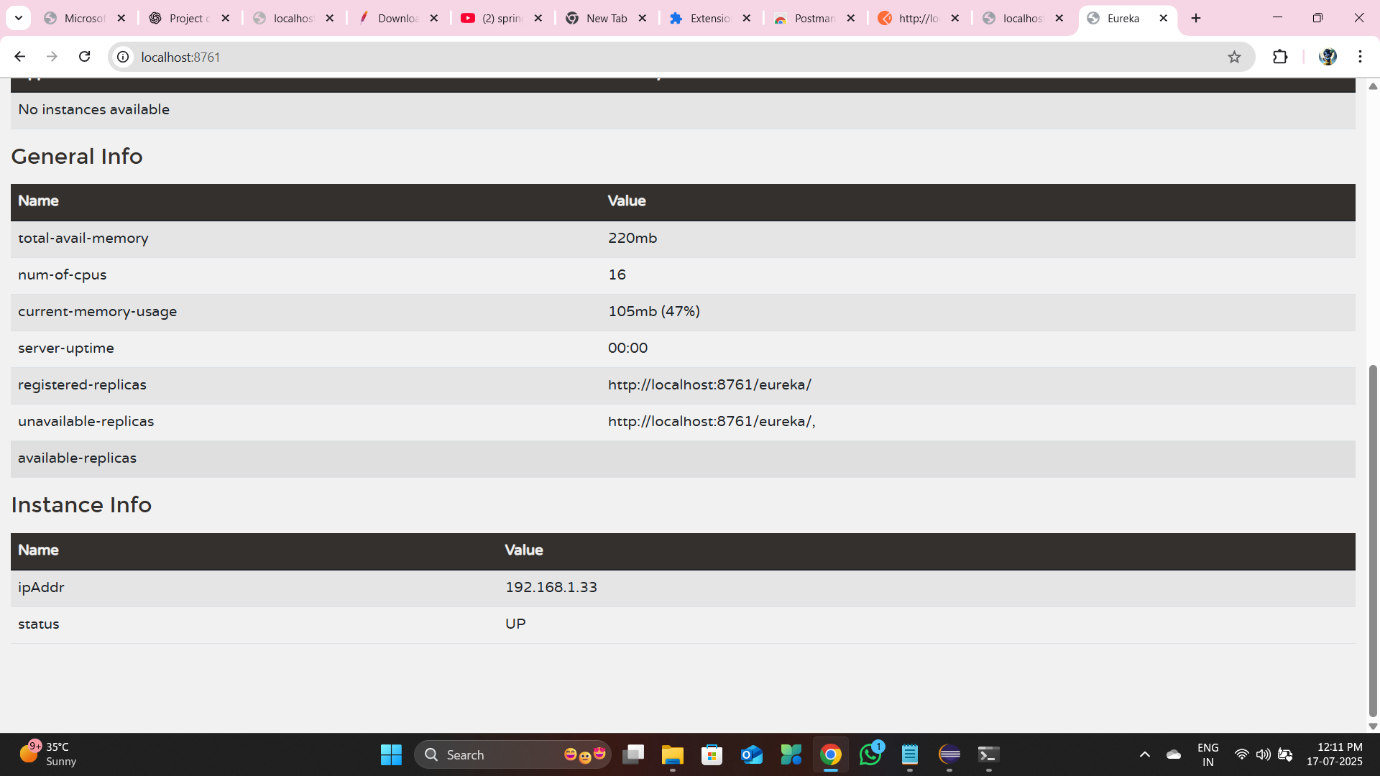
TCP [::]:8082 [::]:0 LISTENING **11332**

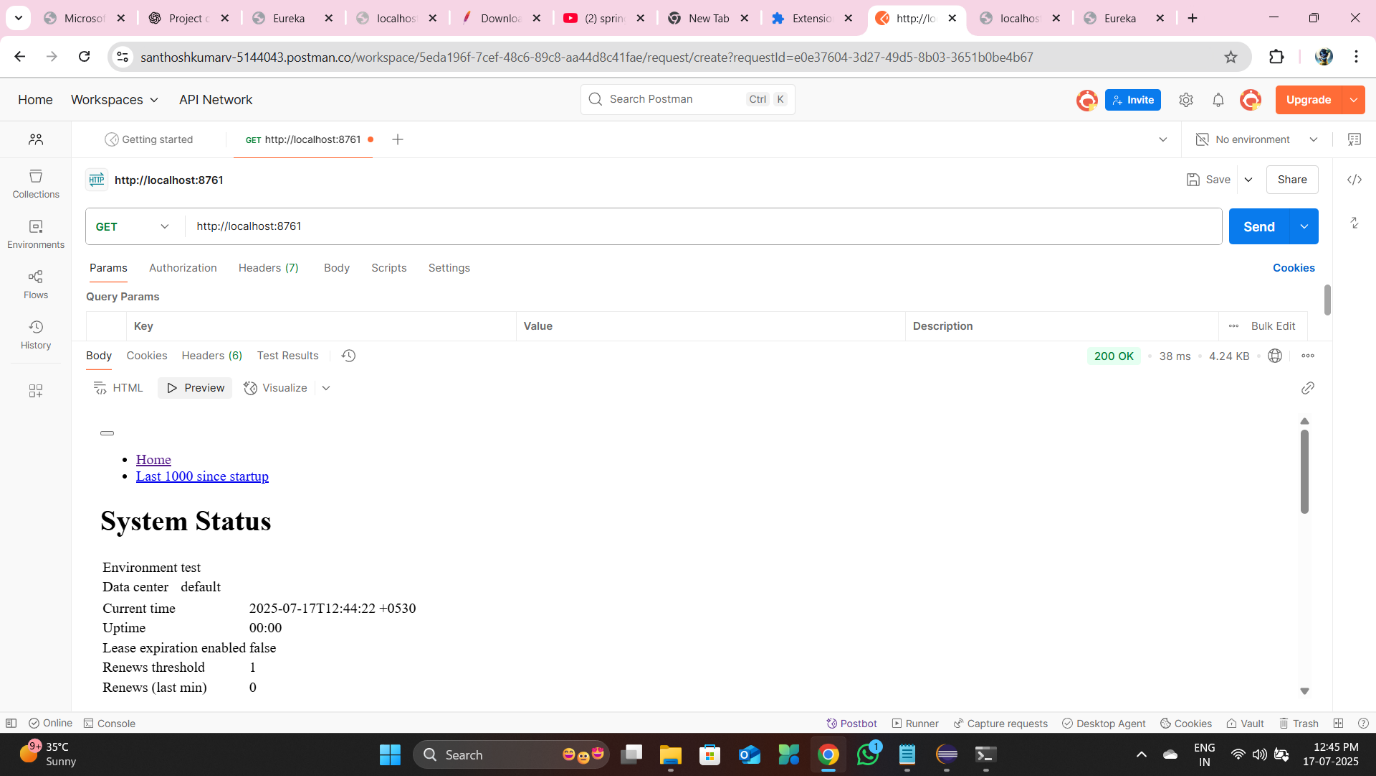
D:\microservices\account>**taskkill /PID 11332 /F**

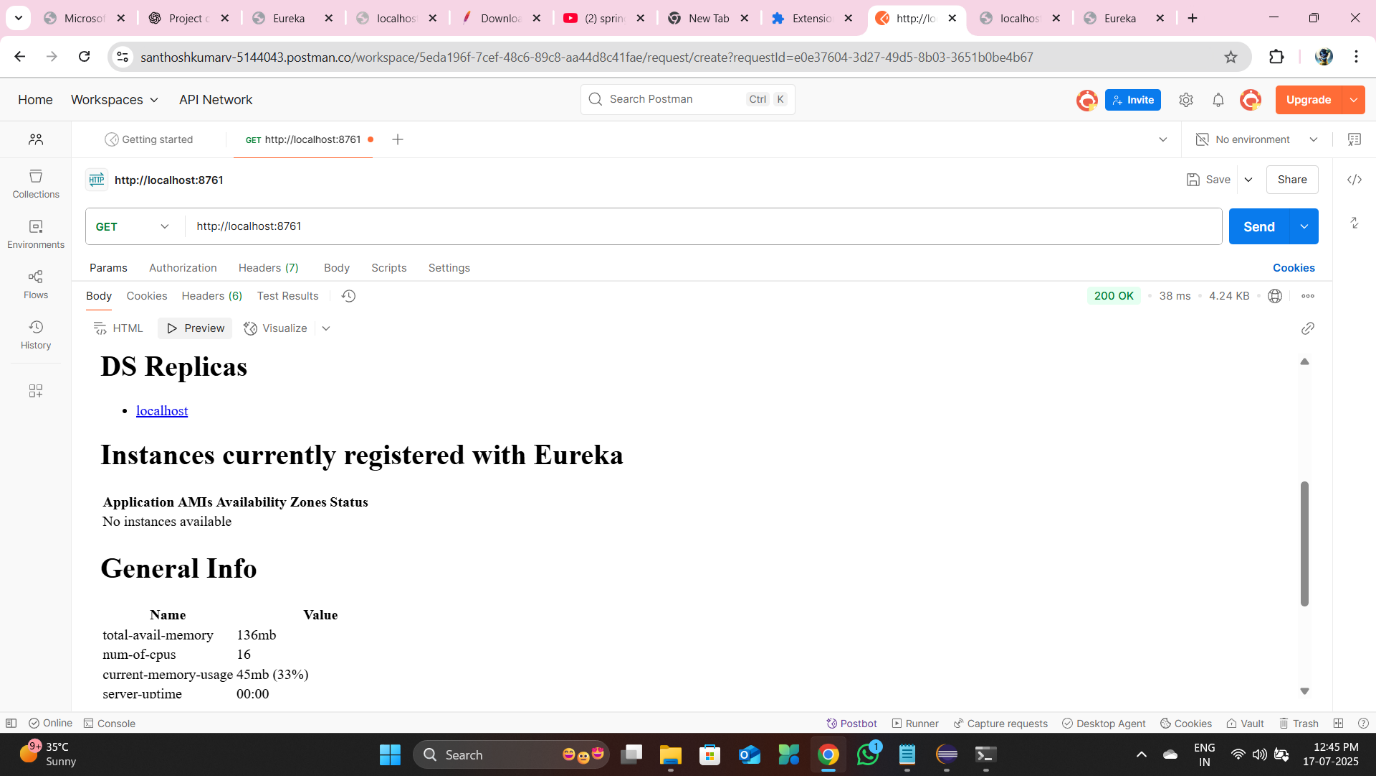
**Initial State**

**Eureka Dashboard: No services registered.**

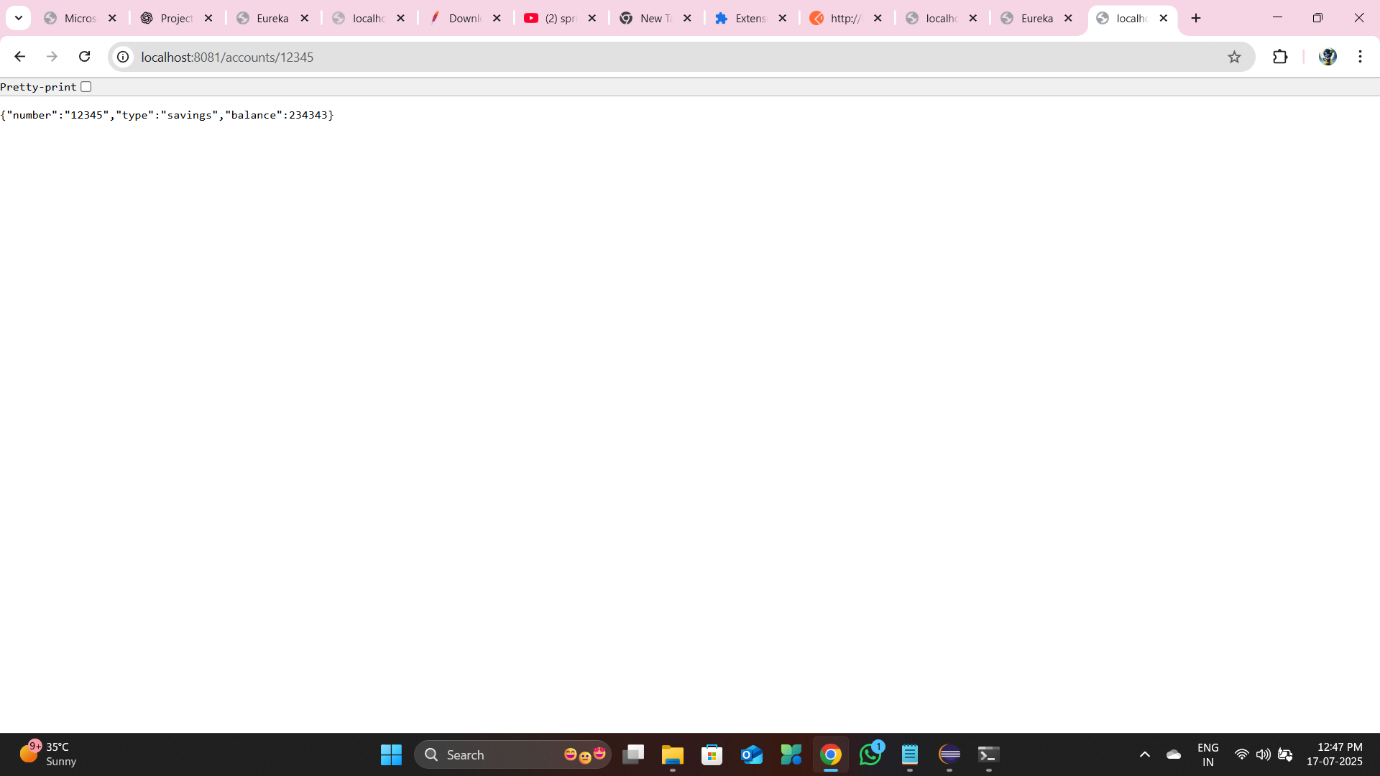




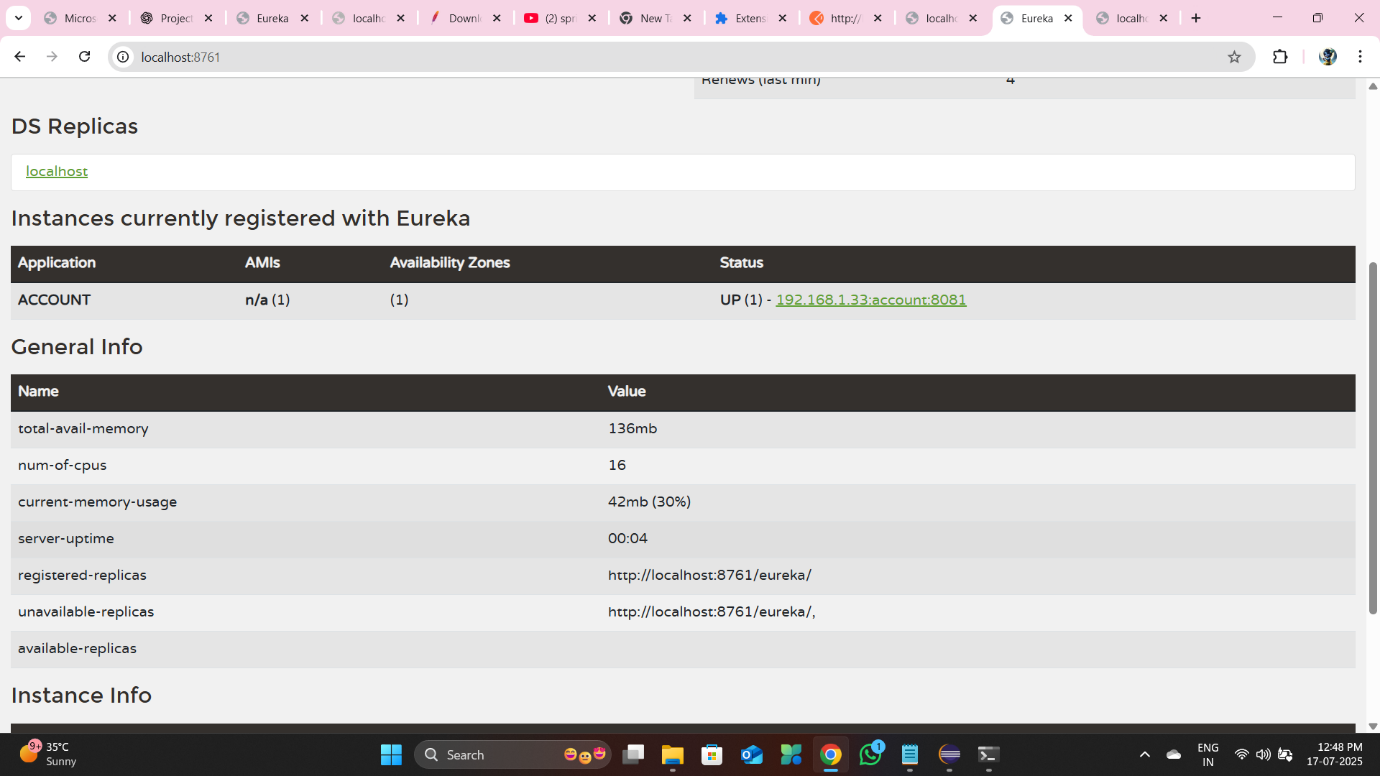
**Postman Dashboard: No endpoints tested.**



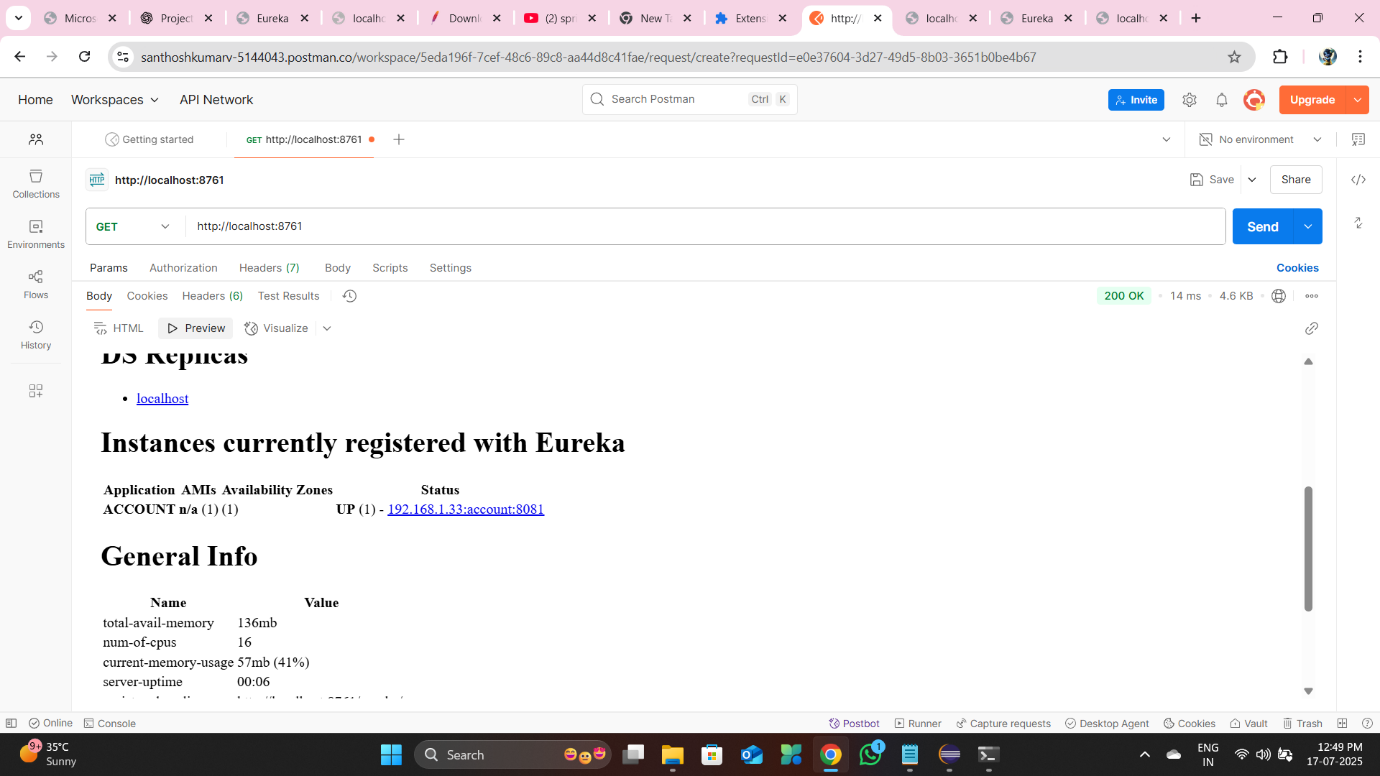
**After Starting Account Service**



**Eureka Dashboard: The Account Service is registered and visible.**



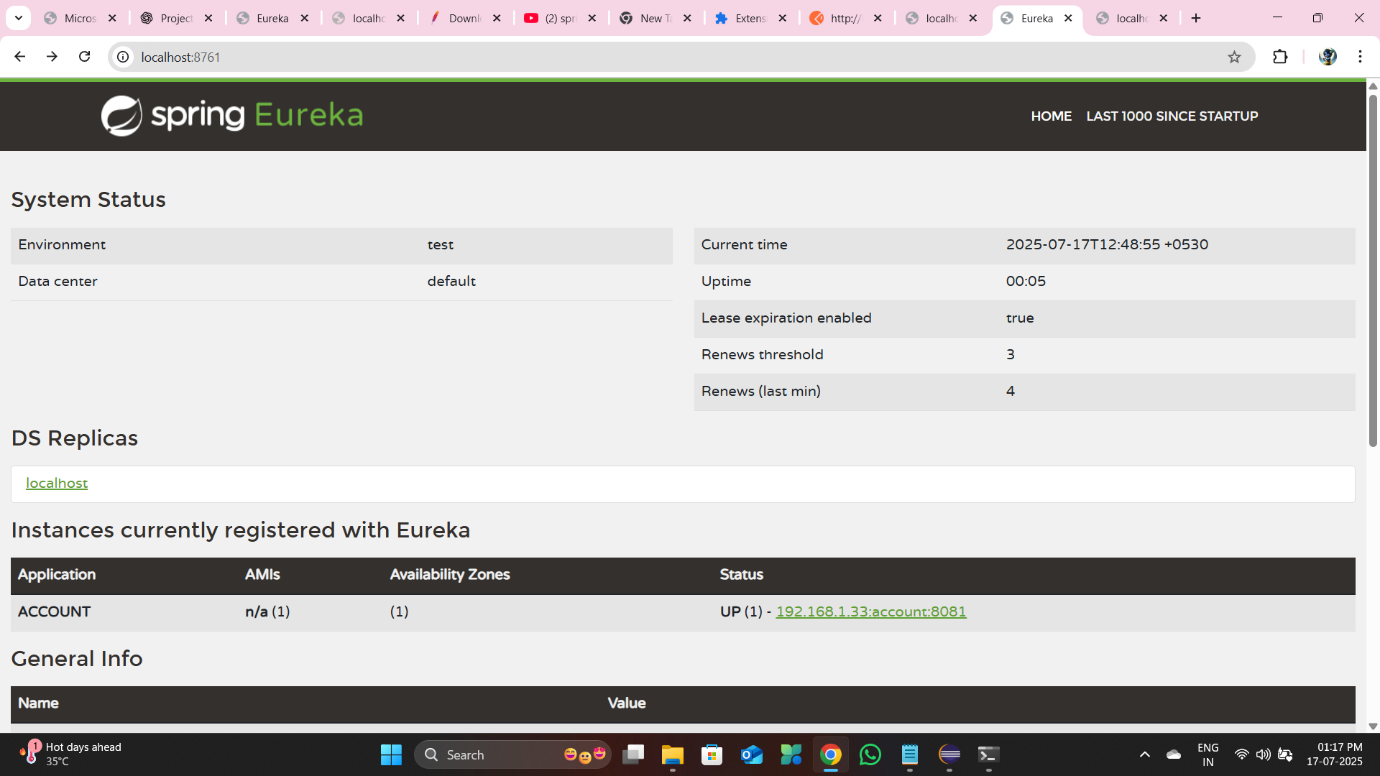
**Postman: The /accounts/{accountNumber} endpoint is functional and returns a successful response.**



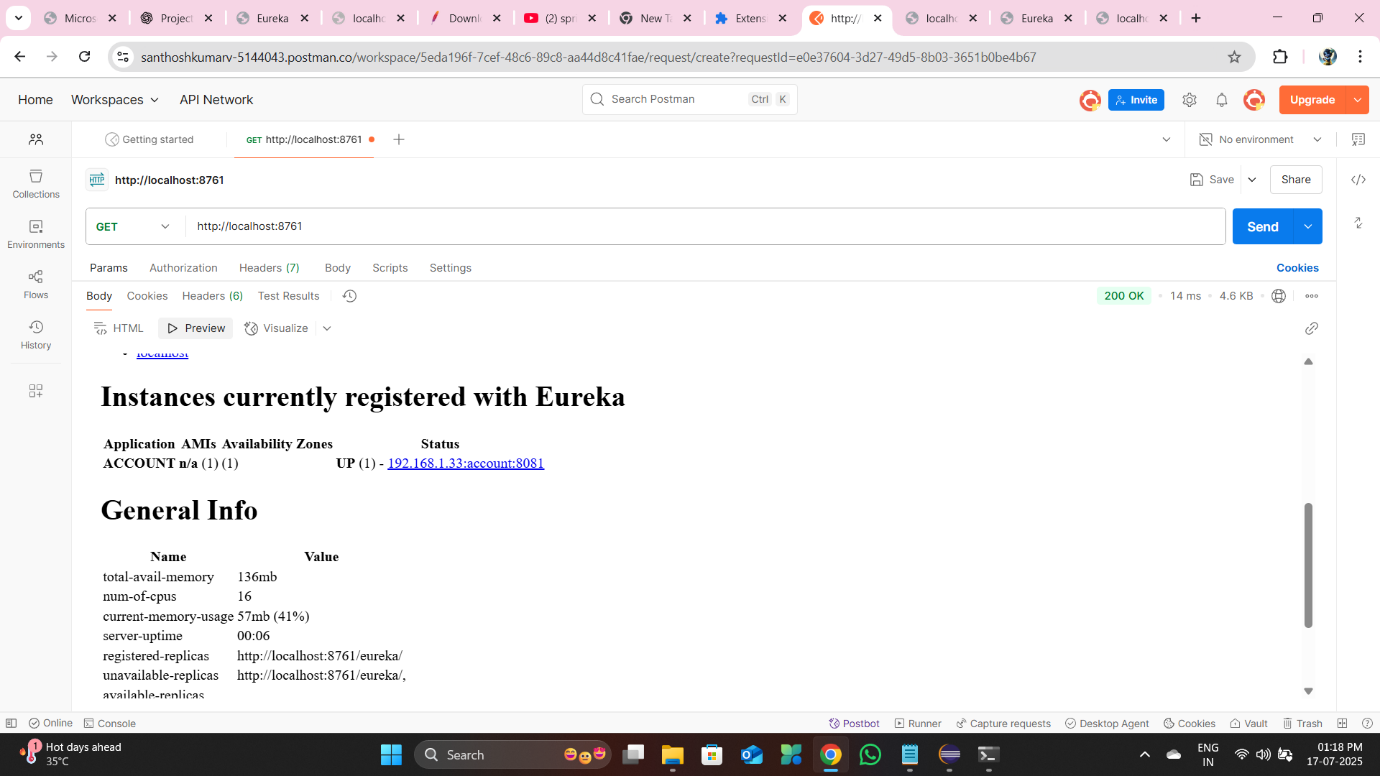
**Account Service Running Only**

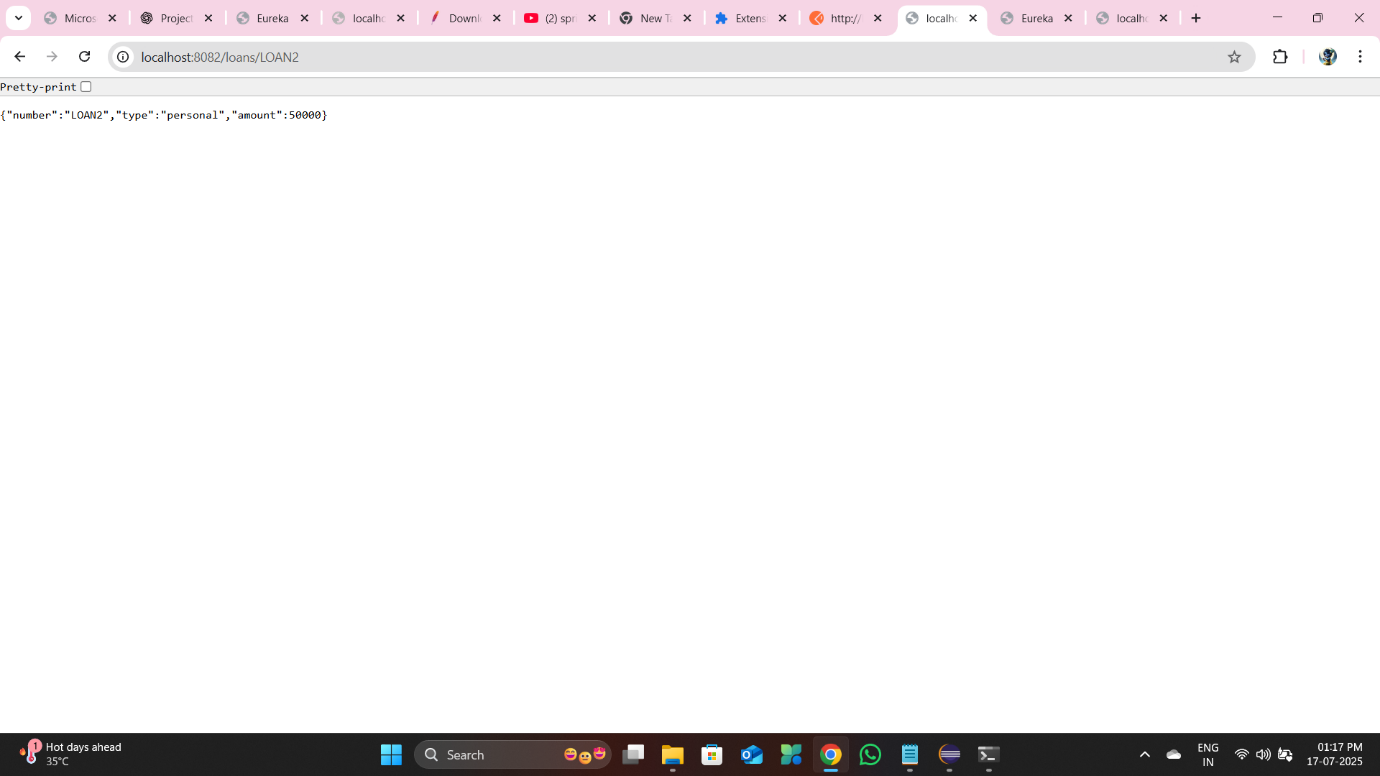
**Confirmed the isolation of the Account Service by stopping the Loan Service.**

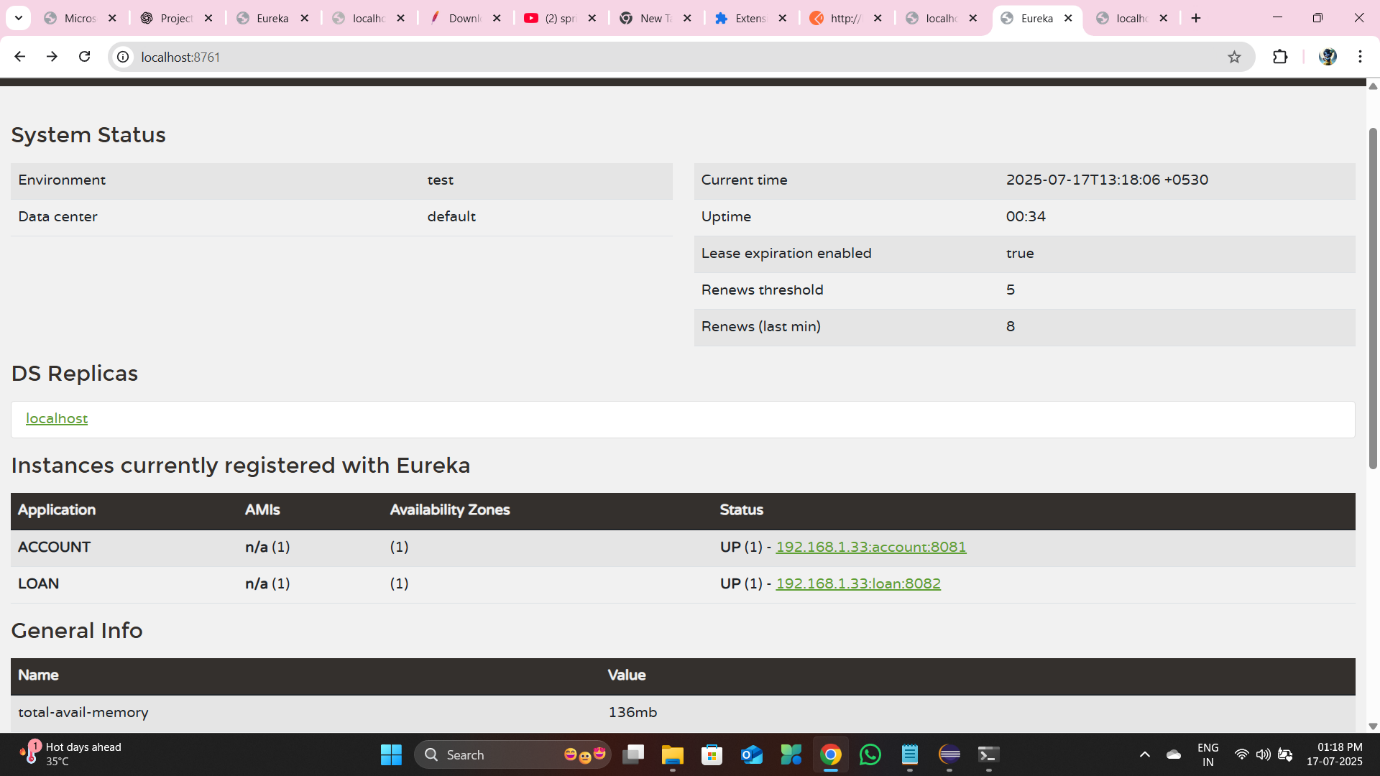
**Eureka Dashboard: Only the Account Service remains registered.**

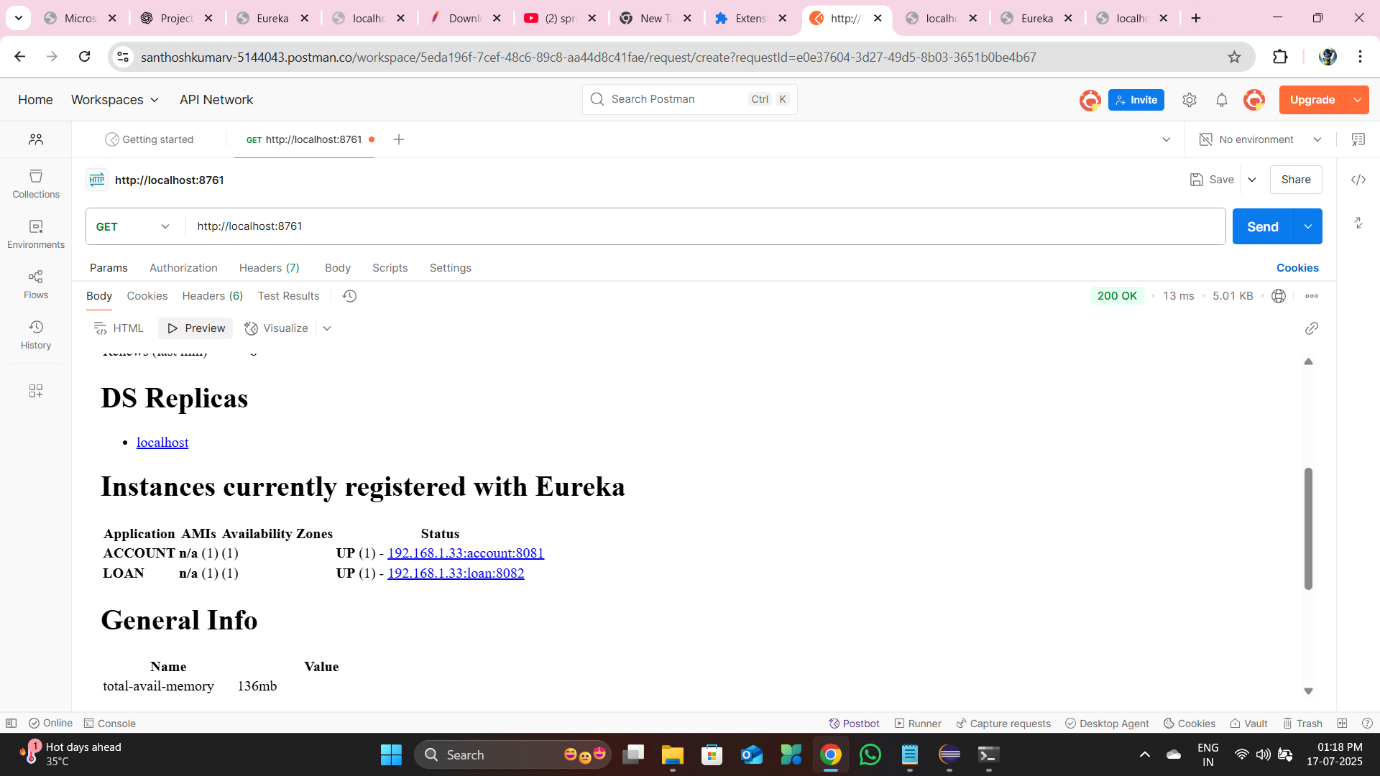


**Postman: The /accounts/{accountNumber} endpoint continues to function.**



**After Starting Loan Service**  
  
**Eureka Dashboard: Both the Loan Service and Account Service are now registered.**



**Postman: Both the /accounts/{accountNumber} and the Loan Service's corresponding endpoint are fully functional.**