**Superset ID: 6419073**

**Week 5 – Microservices with Spring Boot3 and Spring Cloud**

**2. Microservices with API**

**Creating Microservices for account and loan**

**Account Microservice**

**Step 1 : Create Project via Spring Initializr**

**Group**: com.cognizant

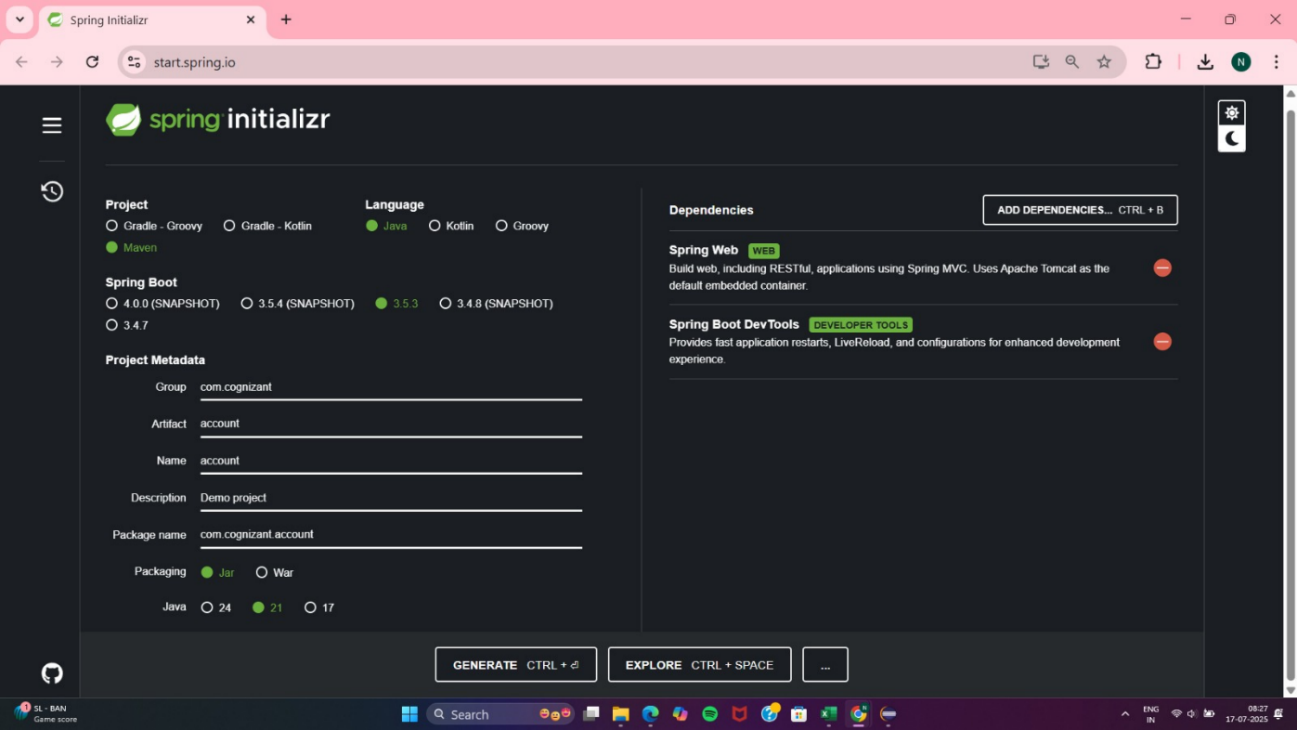
**Artifact**: account

**Name**: account

**Add Dependencies:** Spring Web, Spring Boot DevTools

Click Generate, then extract the zip.

Placed the extracted account folder into this: C:\Users\Nitya\Desktop\microservices\account



**Step 2: Importing project into Eclipse**Eclipse > File > Import > Existing Maven Projects  
Browsed to : C:\Users\Nitya\Desktop\microservices\account

Click finish

**Step 3: Implement Controller**

**AccountController.java**

package com.cognizant.account.controller;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Account getAccountDetails(@PathVariable String number) {

// Simulate dummy response

return new Account(number, "savings", 234343.00);

}

}

**Account.java**

package com.cognizant.account.controller;

public class Account {

private String number;

private String type;

private double balance;

// Constructor

public Account(String number, String type, double balance) {

this.number = number;

this.type = type;

this.balance = balance;

}

// Getters

public String getNumber() {

return number;

}

public String getType() {

return type;

}

public double getBalance() {

return balance;

}

}

**AccountApplication.java**

src/main/java/com/cognizant/account/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);

}

}

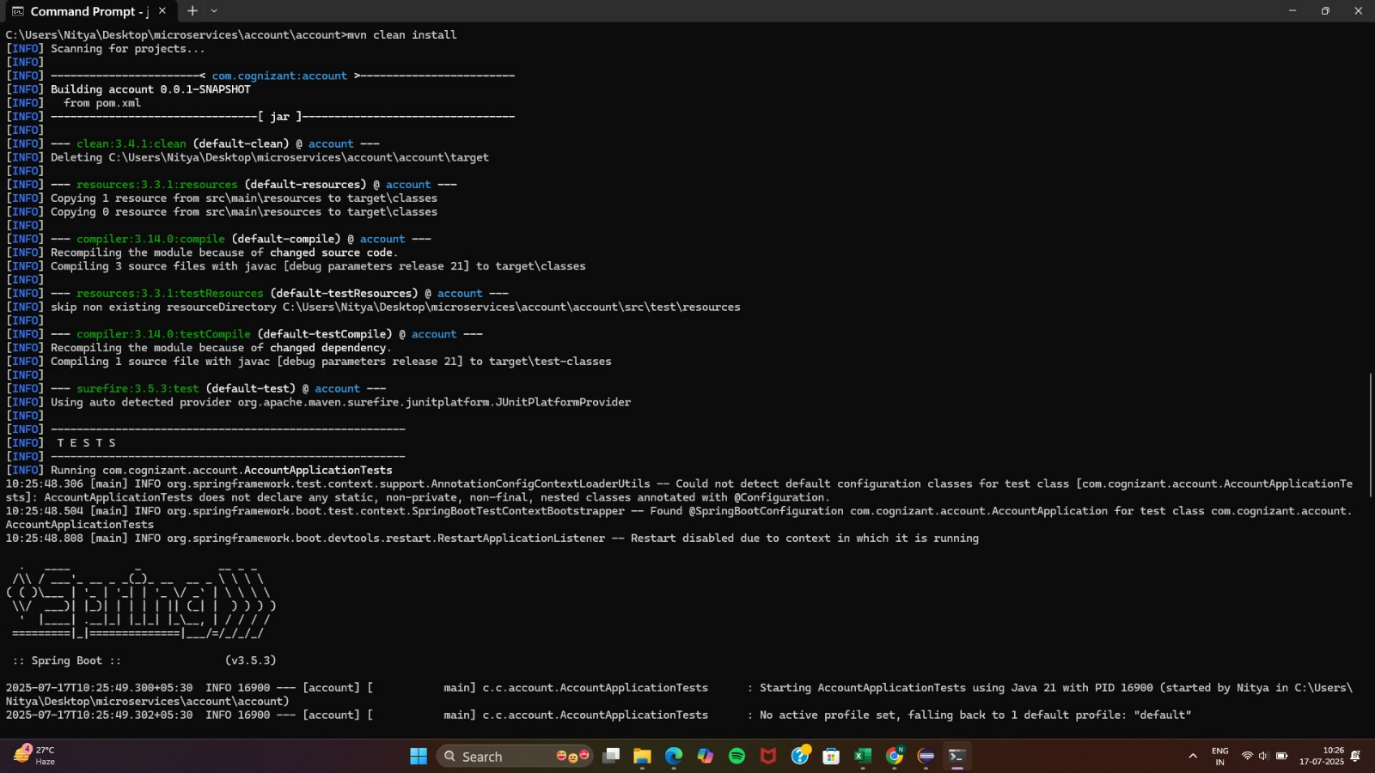
**Output:**

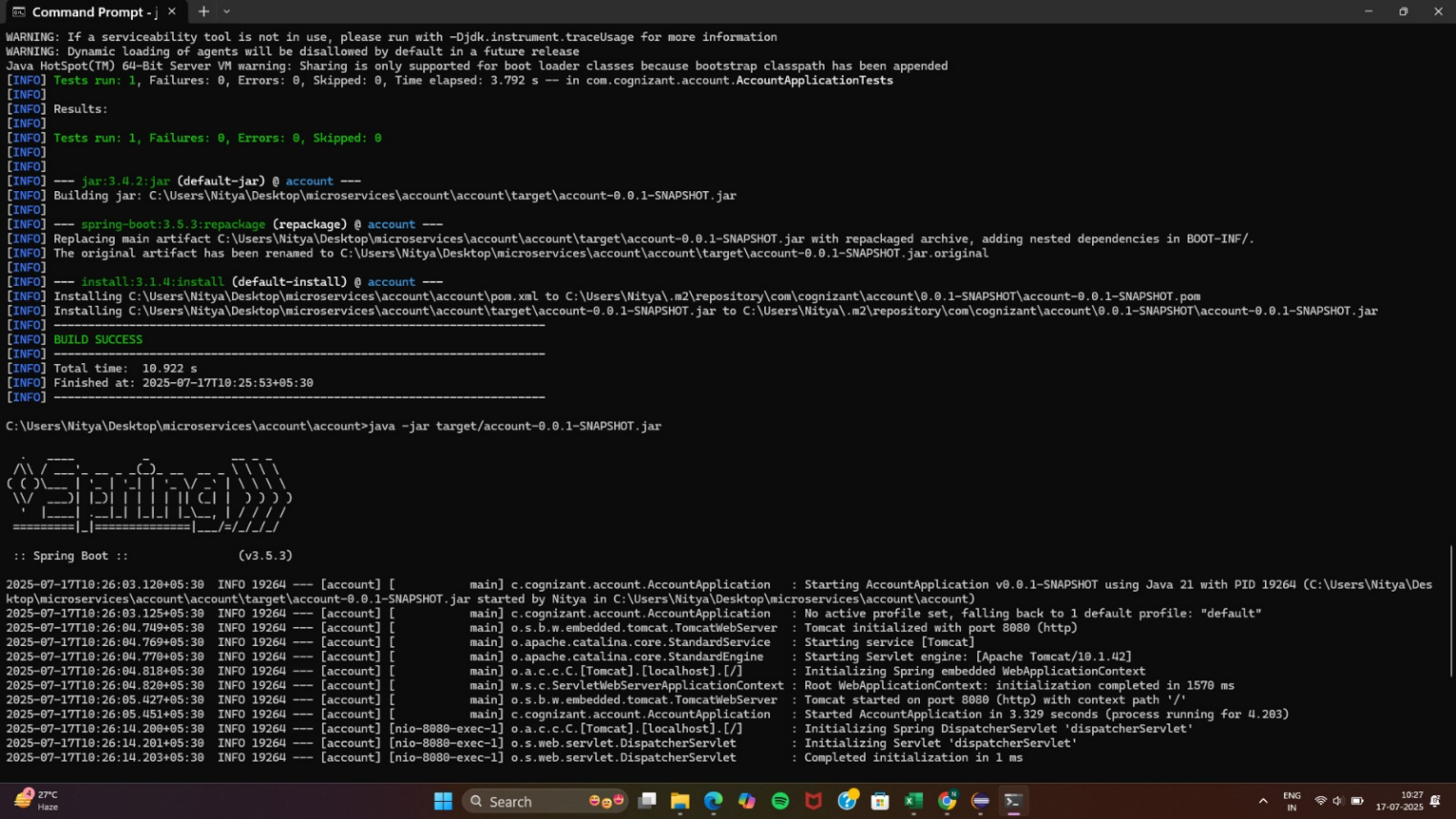
**Command Prompt**

**cd C:\Users\Nitya\Desktop\microservices\account\account**

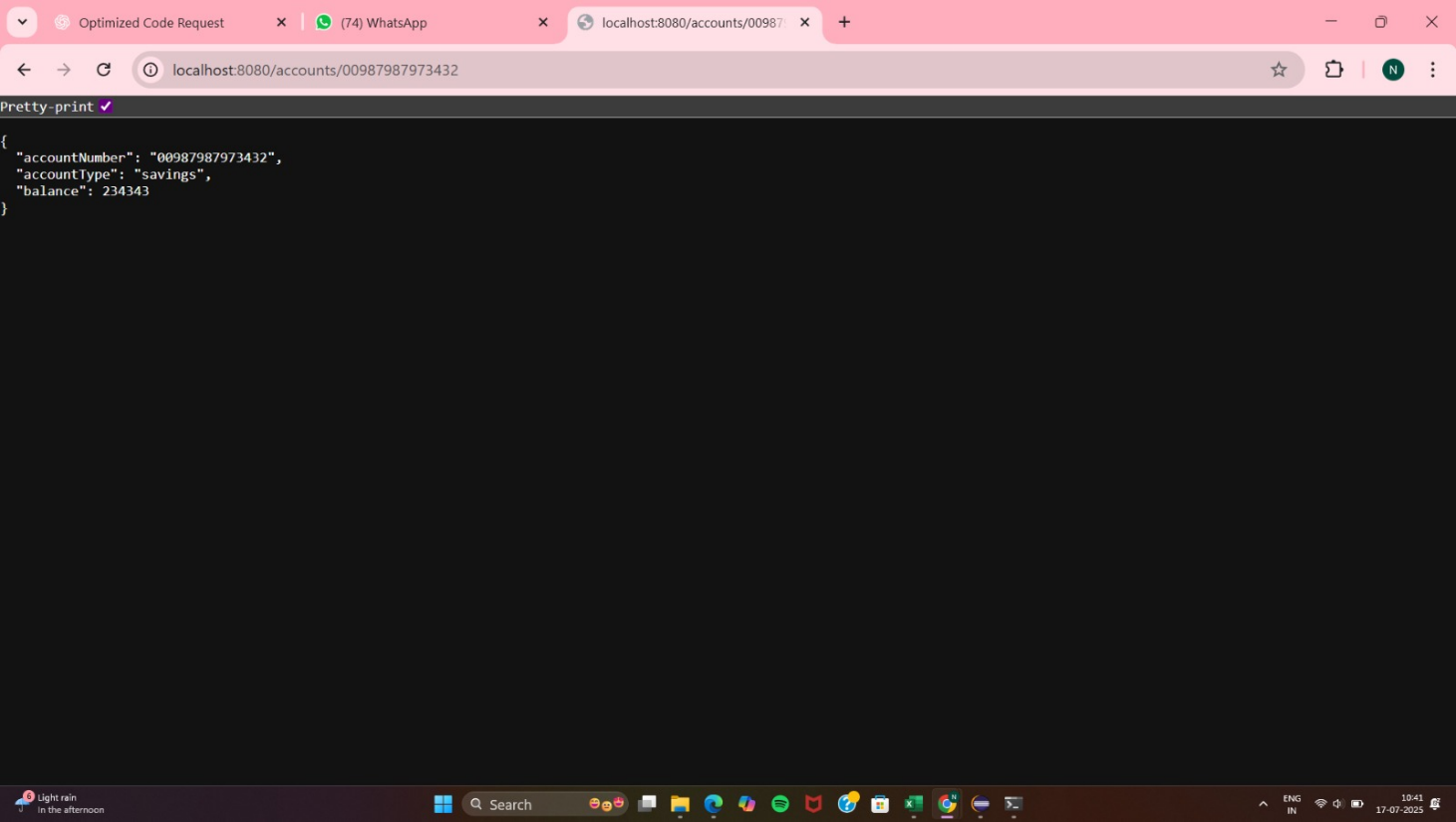
**mvn clean install**

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

****

****

**http://localhost:8080/accounts/00987987973432**

****

**Loan Microservice**

**1. Create Loan Project via Spring Initializr**

**Go to** [**https://start.spring.io**](https://start.spring.io)

* Group: com.cognizant
* Artifact: loan
* Name: loan
* Packaging: Jar
* Java version: 21

**Add Dependencies:**

* Spring Boot DevTools
* Spring Web

Click Generate, then extracted the zip.

Placed the extracted loan folder into: C:\Users\Nitya\Desktop\microservices\loan

**3.** **Import Loan Project into Eclipse**

**Eclipse > File > Import > Existing Maven Projects**

**Selected:** C:\Users\Nitya\Desktop\microservices\loan

**4.Create the Model Class**

**Created the package :** com.cognizant.loan.model

**Loan.java**

package com.cognizant.loan.model;

public class Loan {

private String number;

private String type;

private double loan;

private double emi;

private int tenure;

public Loan() {}

public Loan(String number, String type, double loan, double emi, int tenure) {

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

// Getters and Setters

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 double getLoan() {

return loan;

}

public void setLoan(double loan) {

this.loan = loan;

}

public double getEmi() {

return emi;

}

public void setEmi(double emi) {

this.emi = emi;

}

public int getTenure() {

return tenure;

}

public void setTenure(int tenure) {

this.tenure = tenure;

}

}

**5. Create Contoller**

**com.cognizant.loan.controller**

**LoanController.java**

package com.cognizant.loan.controller;

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

import com.cognizant.loan.model.Loan;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Loan getLoanDetails(@PathVariable String number) {

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

}

}

**7.Application.properties**spring.application.name=loan

server.port=8081

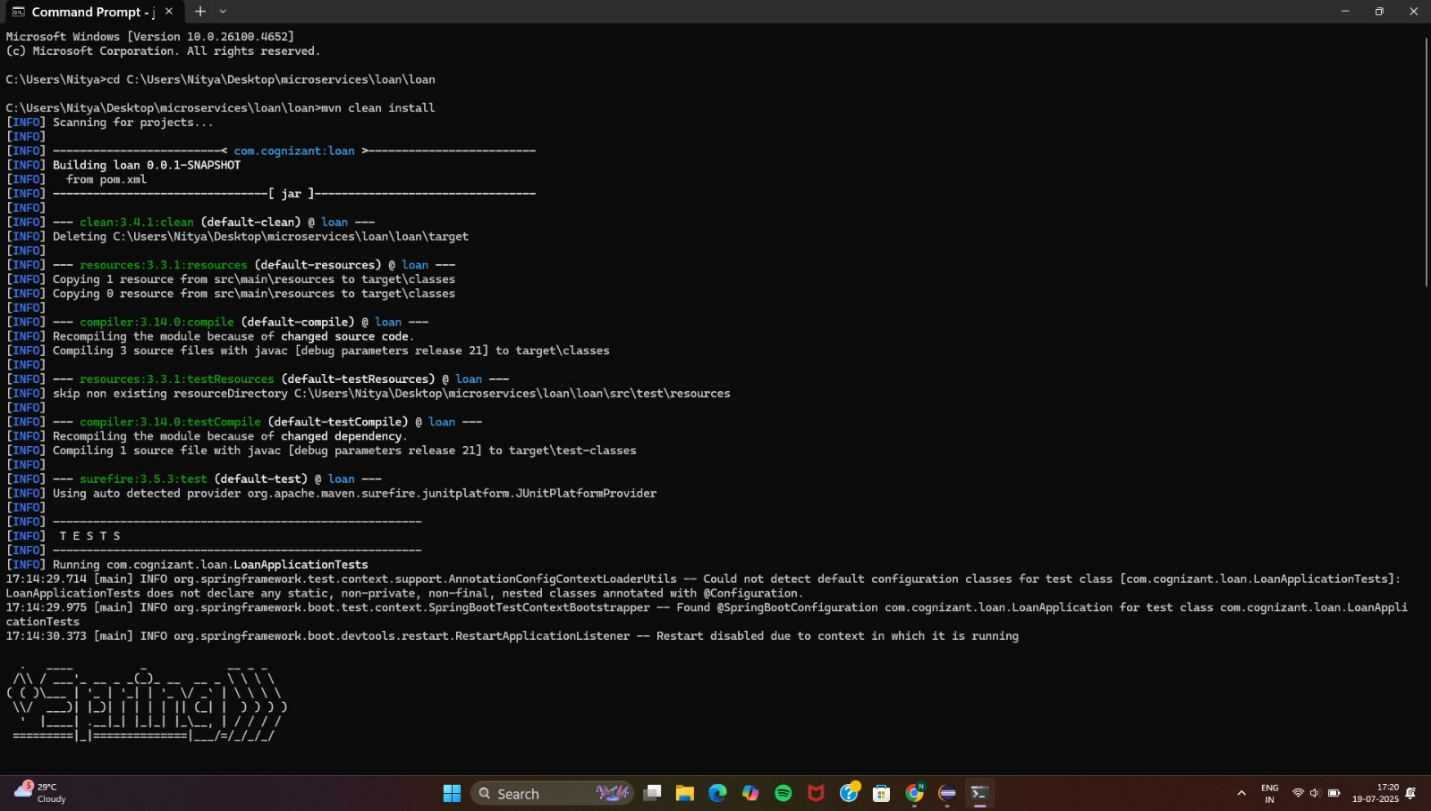
**OUTPUT:**

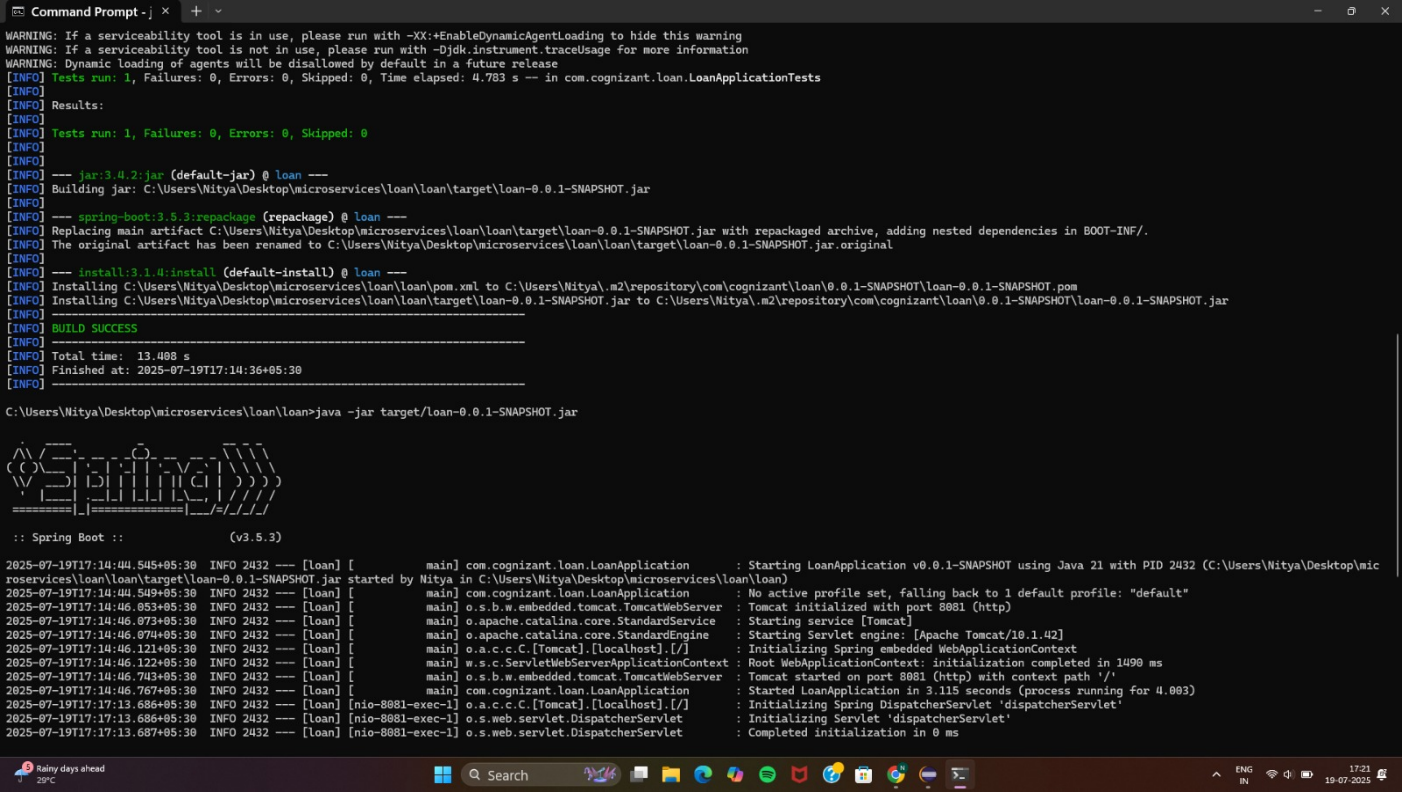
**Command Prompt**

cd C:\Users\Nitya\Desktop\microservices\loan

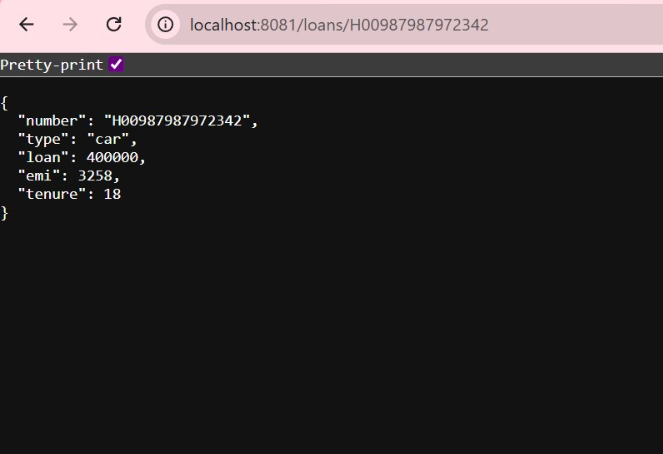
mvn clean install

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





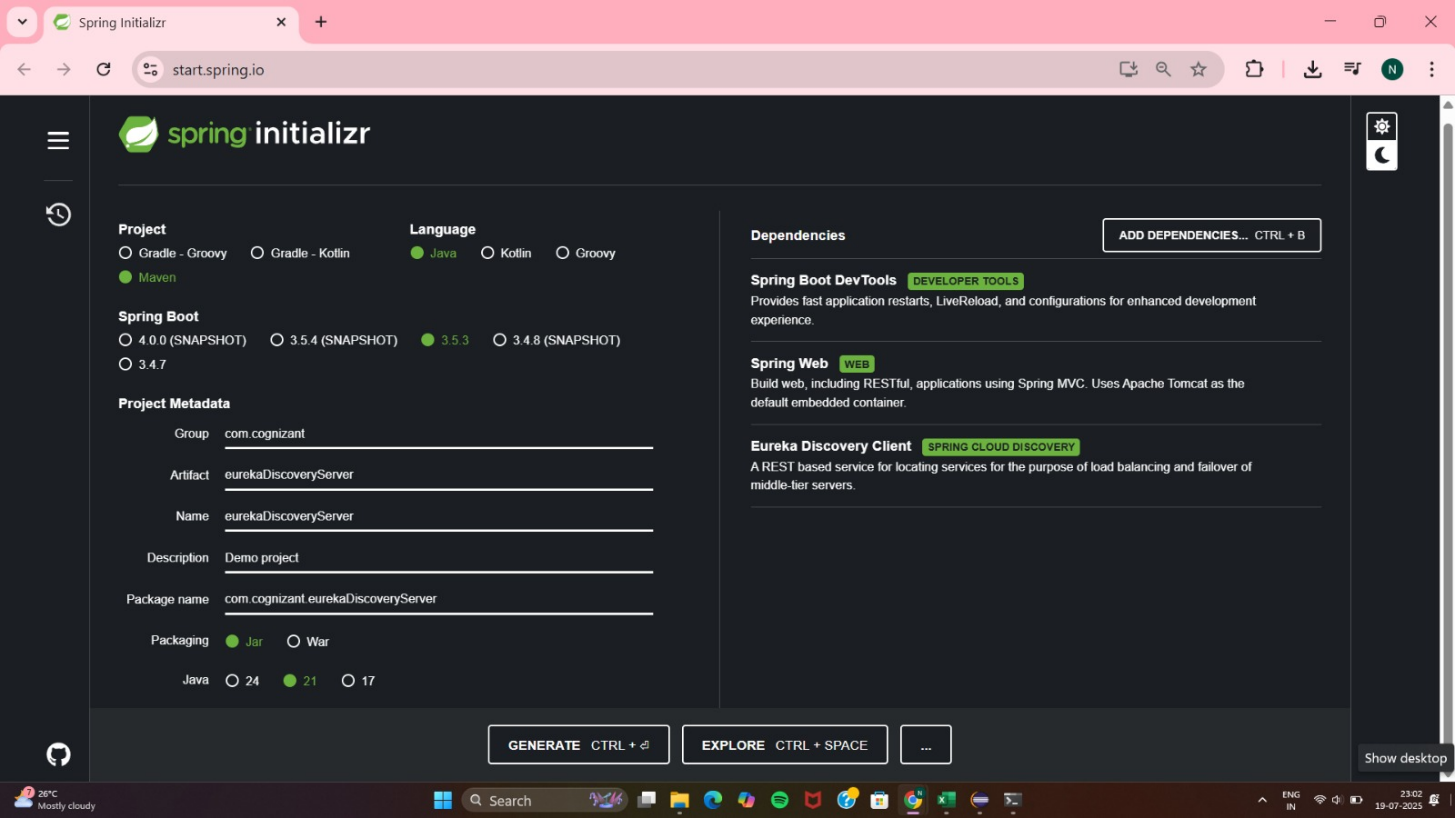
**Browser   
http://localhost:8081/loans/H00987987972342**



**2. Microservices with API gateway**

**Step 1: Eureka Discovery Server**

**1. Generate Eureka Server Project**

****

* Go to <https://start.spring.io>
* **Group:** com.cognizant
* **Artifact:** eurekaDiscoveryServer
* **Dependencies:**
  + Spring Boot DevTools
  + Spring Web
  + Spring Cloud Discovery → Eureka Server
* Click Generate to download the project.

**2. 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 https://maven.apache.org/xsd/ maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.2.4</version> <!-- Use compatible Spring Boot version -->

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>eureka-discovery-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eureka-discovery-server</name>

<description>Eureka Discovery Server</description>

<properties>

<java.version>21</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version> <!-- Compatible with Spring Boot 3.2.4 ->

</properties>

<dependencies>

<!-- Eureka Server -->

<dependency>

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

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

<!-- Web (optional, for error/info endpoints) -->

<dependency>

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

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

</dependency>

<!-- Dev tools -->

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Tests -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

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

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

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**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

**EurekaDiscoveryServerApplication.java**

package com.cognizant.eurekaDiscoveryServer;

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);

}

}

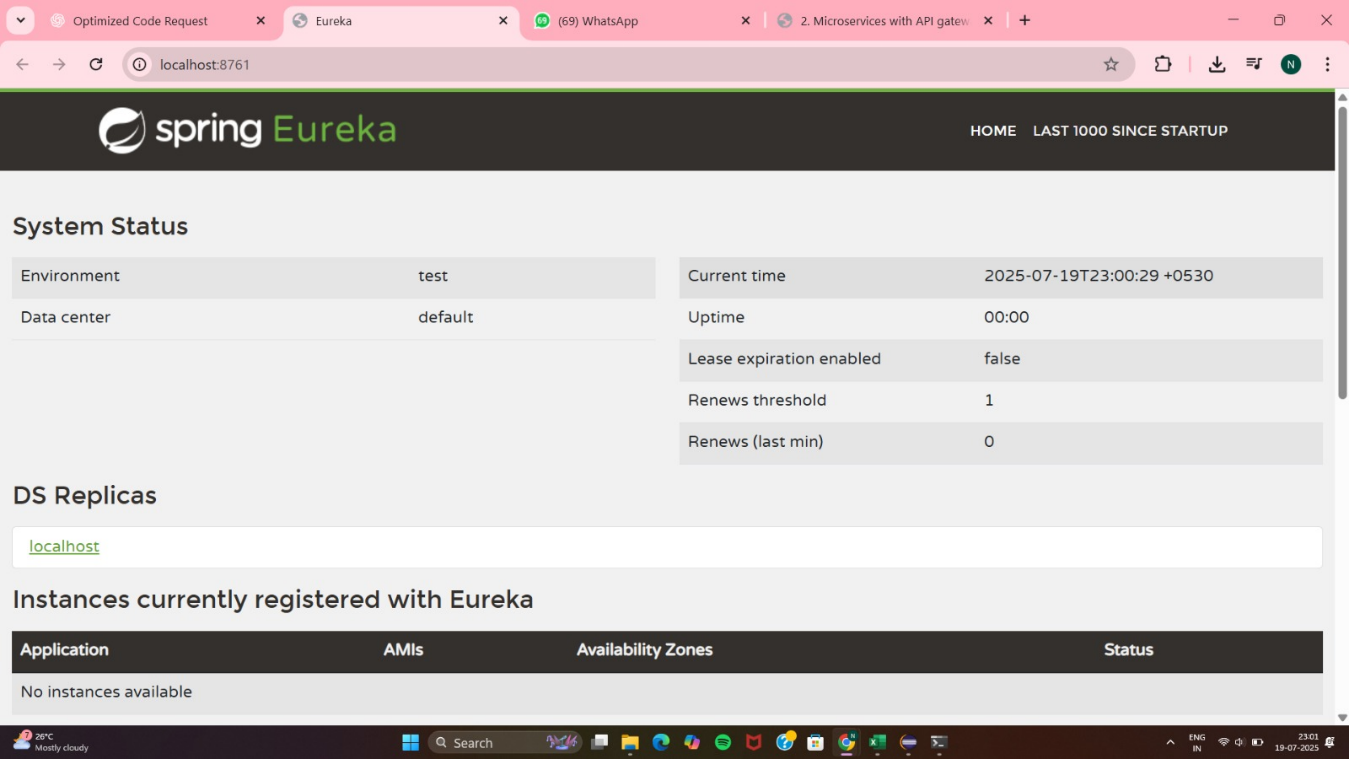
**OUTPUT:**

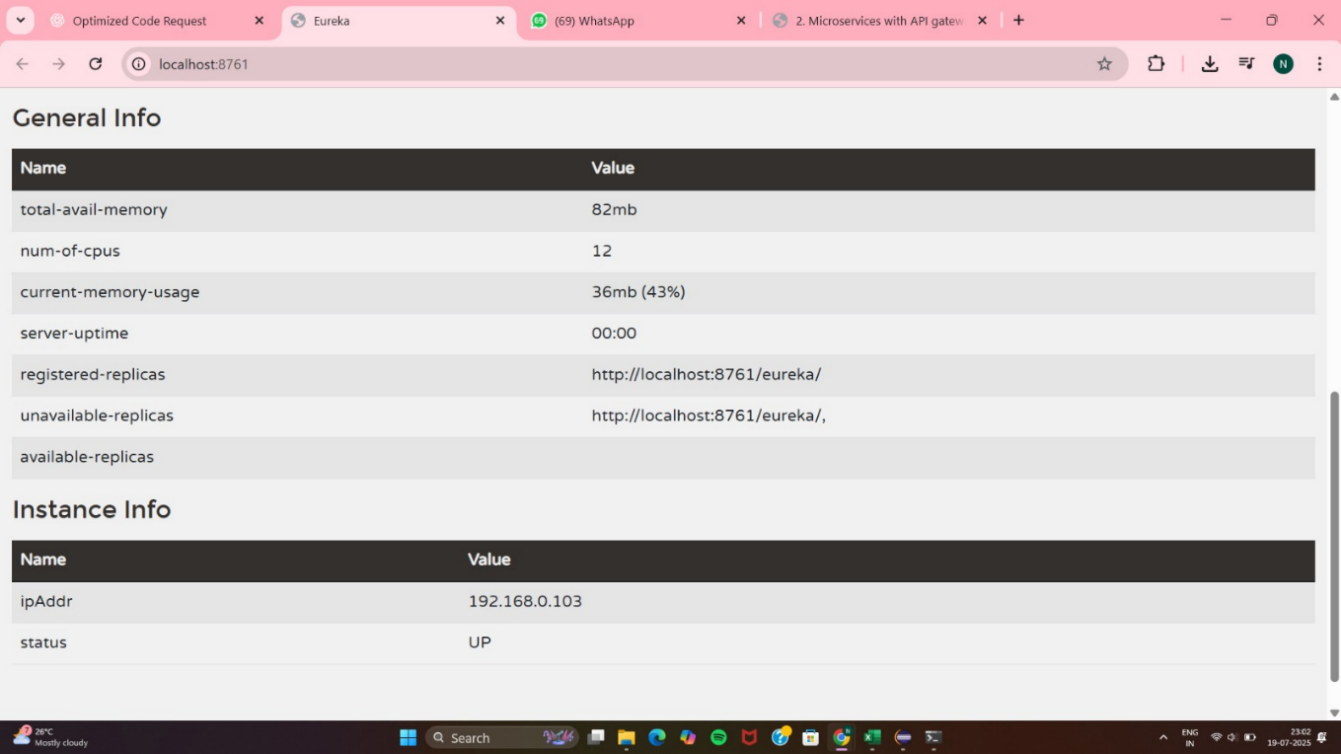
**Build and Run**

mvn clean install

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

[**http://localhost:8761**](http://localhost:8761)

****

****

**Step 2 : Account & Loan Microservices**

**1. Generate Project from Spring Initializr**

Group: com.cognizant

Artifact: account or loan

Dependencies:

* Spring Boot DevTools
* Spring Web
* Eureka Discovery Client

**2.1 pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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

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

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.2.5</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>loan</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>loan</name>

<description>Loan microservice registered with Eureka</description>

<properties>

<java.version>21</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<!-- Spring Web -->

<dependency>

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

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

</dependency>

<!-- Eureka Discovery Client -->

<dependency>

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

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

</dependency>

<!-- DevTools for hot reload -->

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Testing support -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<!-- Dependency management for Spring Cloud -->

<dependencyManagement>

<dependencies>

<dependency>

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

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

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<!-- Build plugin -->

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**2.2 Application class**

**Account**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

**Loan**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;  
@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

**2.3 application.properties**

Account:

spring.application.name=account-service

server.port=8081

eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
  
Loan  
spring.application.name=loan-service

server.port=8082

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

**STEP 3: Greet Service Microservice**

**3.1 Generate greet-service Microservice**

**Project Configuration**

* **Project**: Maven
* **Language**: Java
* **Spring Boot Version**: **3.2.5**
* **Group**: com.cognizant
* **Artifact**: greet
* **Name**: greet
* **Package name**: com.cognizant.greet

**Dependencies :**

* **Spring Web**
* **Eureka Discovery Client**

**3.2 pom.xml**  
<?xml version="1.0" encoding="UTF-8"?>

<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

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

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.2.5</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>greet</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>greet</name>

<description>Greet Microservice with Eureka</description>

<properties>

<java.version>21</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<!-- Web API support -->

<dependency>

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

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

</dependency>

<!-- Eureka client -->

<dependency>

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

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

</dependency>

<!-- Dev tools -->

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Unit testing -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<!-- Manage spring-cloud dependencies -->

<dependencyManagement>

<dependencies>

<dependency>

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

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

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**3.3 Applications.properties**

server.port=8083

spring.application.name=greet-service

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

**3.4 Main Application Class**

package com.cognizant.greet;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class GreetApplication {

public static void main(String[] args) {

SpringApplication.run(GreetApplication.class, args);

}

}

**3.5 Add a REST Controller  
GreetController.java**

package com.cognizant.greet;

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

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

@RestController

public class GreetController {

@GetMapping("/greet")

public String sayHello() {

return "Hello from Greet Service!";

}

}

When we run this and see <http://localhost:8083/greet> this works. And greet-service appears in eureka.

**STEP 4: API Gateway with Global Filter**

**4.1** Generate api-gateway from start.spring.io with dependencies.

* Spring Cloud Gateway
* Eureka Discovery Client
* Spring Boot DevTools

**4.2 pom.xml same**

**4.3 application.properties**

server.port=9090

spring.application.name=api-gateway

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

spring.cloud.gateway.discovery.locator.enabled=true

spring.cloud.gateway.discovery.locator.lowerCaseServiceId=true

**4.4 LogFilter.java**

@Component

public class LogFilter implements GlobalFilter {

@Override

public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

System.out.println("Incoming Request: " + exchange.getRequest().getURI());

return chain.filter(exchange);

}

}

<http://localhost:8083/greet> -> Hello World

**OUTPUT:**

Eureka Dashboard



Logging filter

