**SuperSet Id: 6376594**

Microservices(Hands-On)

**Exercise-1: Creating Microservices for account and loan**

**Project Name: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);

}

}

AccountController.java Code:

**package** com.cognizant.account.controller;

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

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

@RestController

**public** **class** AccountController {

@GetMapping("/account")

**public** String getAccount() {

**return** "Hello from Account Service!";

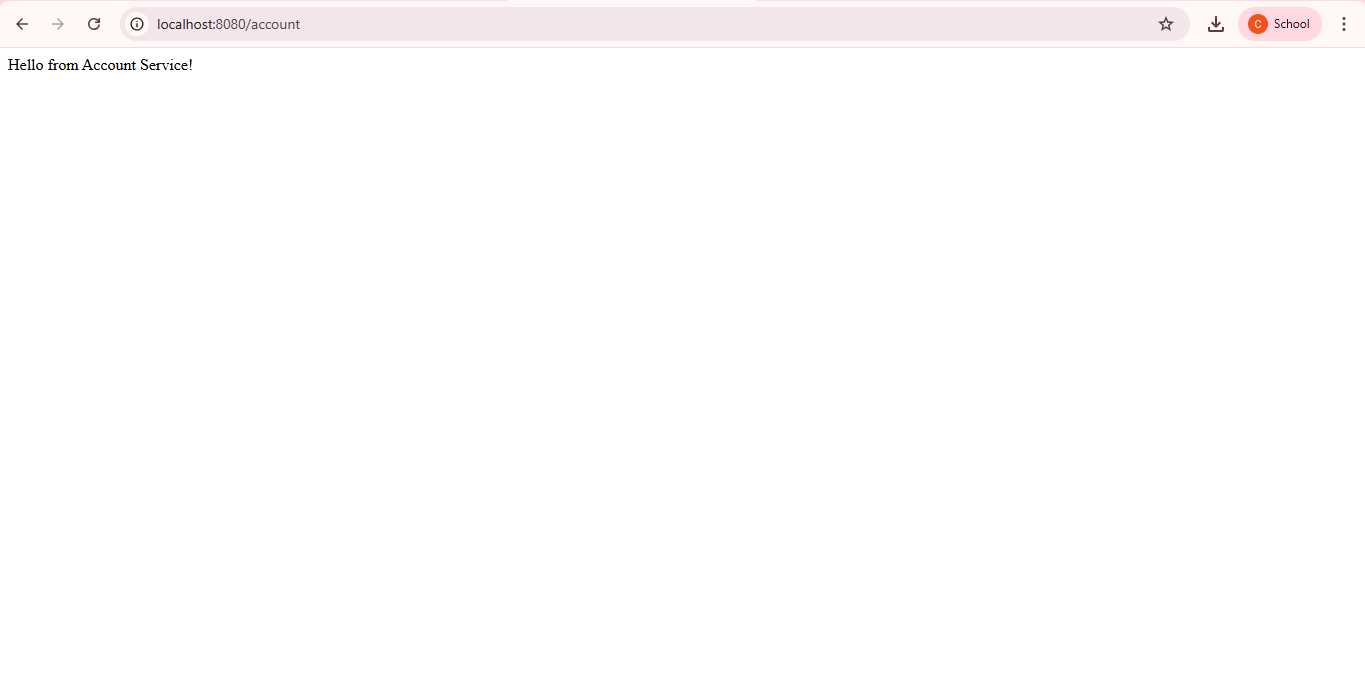
}

}

Excepted Output:

Hello from Account Service!

Output:



**Project Name:loan**

LoanApplication.java Code:

**package** com.cognizant.loan;

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

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

@SpringBootApplication

**public** **class** LoanApplication {

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

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

}

}

LoanController.java Code:

**package** com.cognizant.loan;

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

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

@RestController

**public** **class** LoanController {

@GetMapping("/loan")

**public** String getLoanInfo() {

**return** "Hello from Loan Service!";

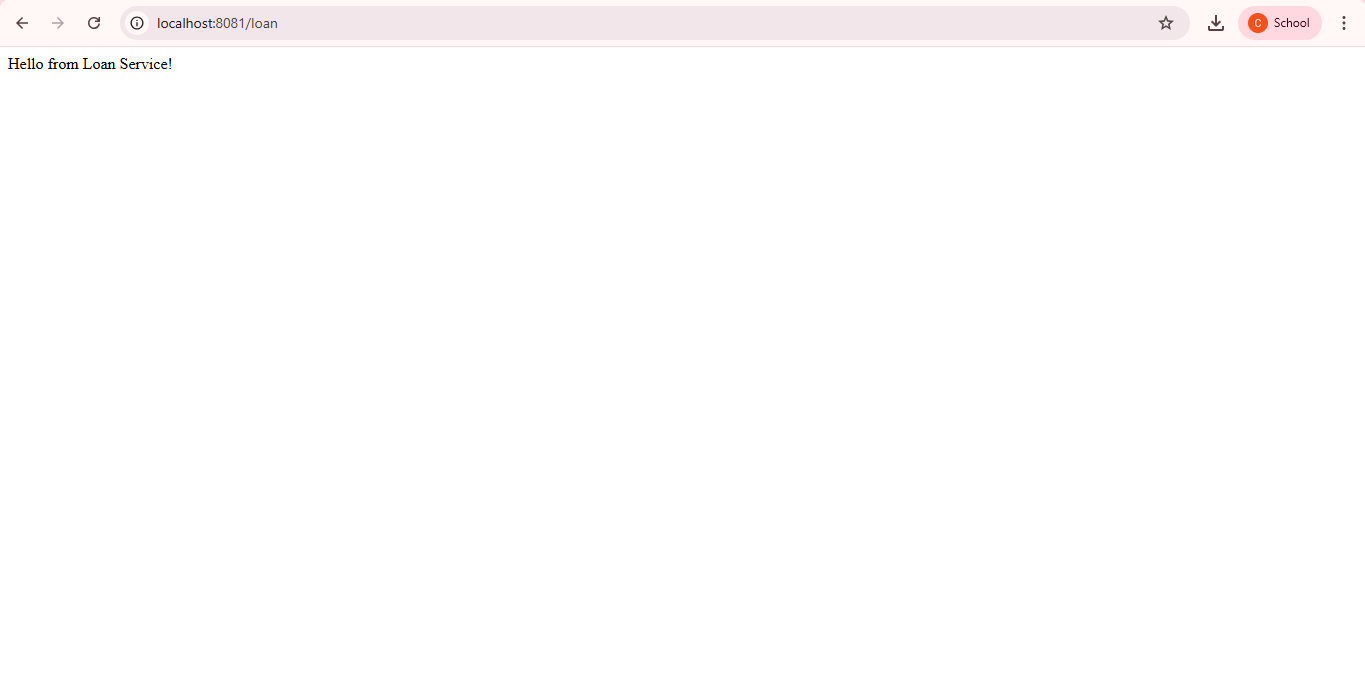
}

}

Expected Output:

Hello from Loan Service!

Output:



Additional Hands-On

**Exercise-1: Create Eureka Discovery Server and register**

**Microservices**

Project Name:eureka-discovery-server

EurekaDiscoveryServerApplication.java Code:

**package** com.cognizant.eureka\_discovery\_server;

**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 Code:

spring.application.name=eureka-server

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

pom.xml Code:

<?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>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</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 for Microservices</description>

<properties>

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

<spring-cloud.version>2025.0.0</spring-cloud.version>

</properties>

<dependencies>

<!-- Eureka Server -->

<dependency>

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

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

</dependency>

<!-- DevTools for hot reload (optional) -->

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Test Dependencies -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<!-- Spring Cloud BOM -->

<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>

<version>3.5.3</version>

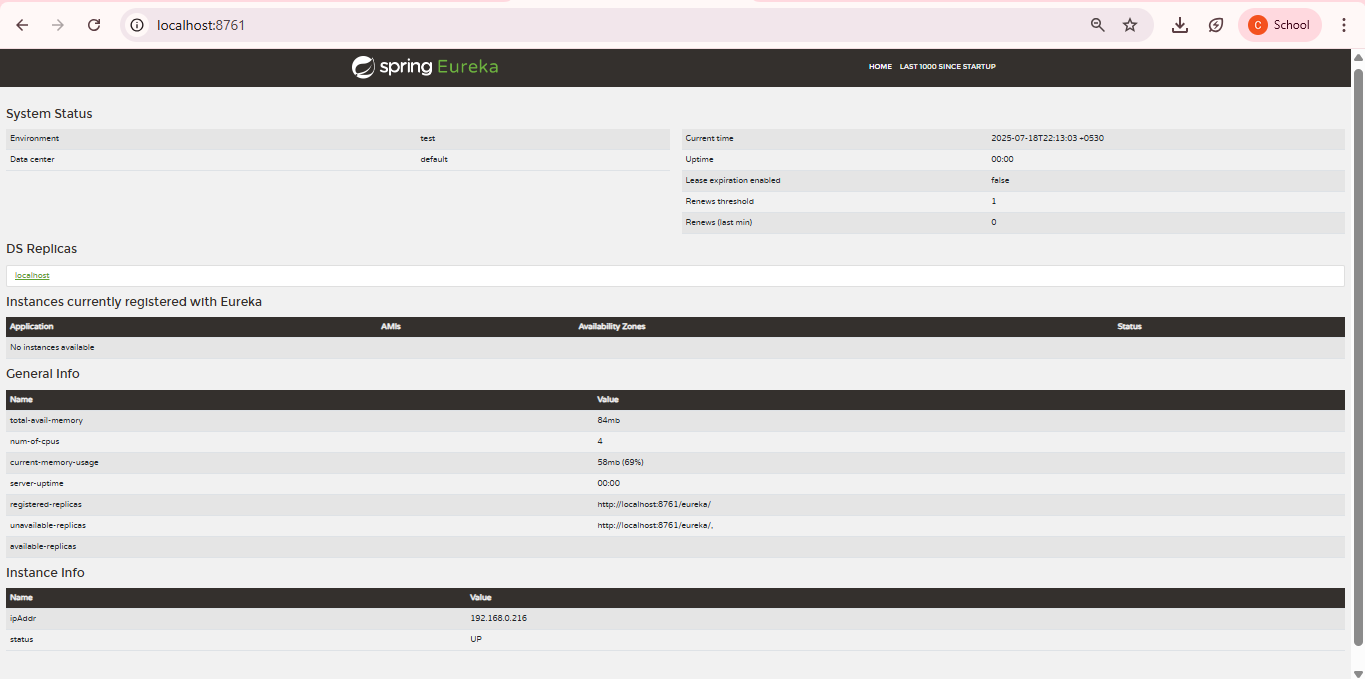
</plugin>

</plugins>

</build>

</project>

Output:



**Project Name: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);

}

}

AccountController.java Code:

**package** com.cognizant.account.controller;

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

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

@RestController

**public** **class** AccountController {

@GetMapping("/account")

**public** String getAccount() {

**return** "Hello from Account Service!";

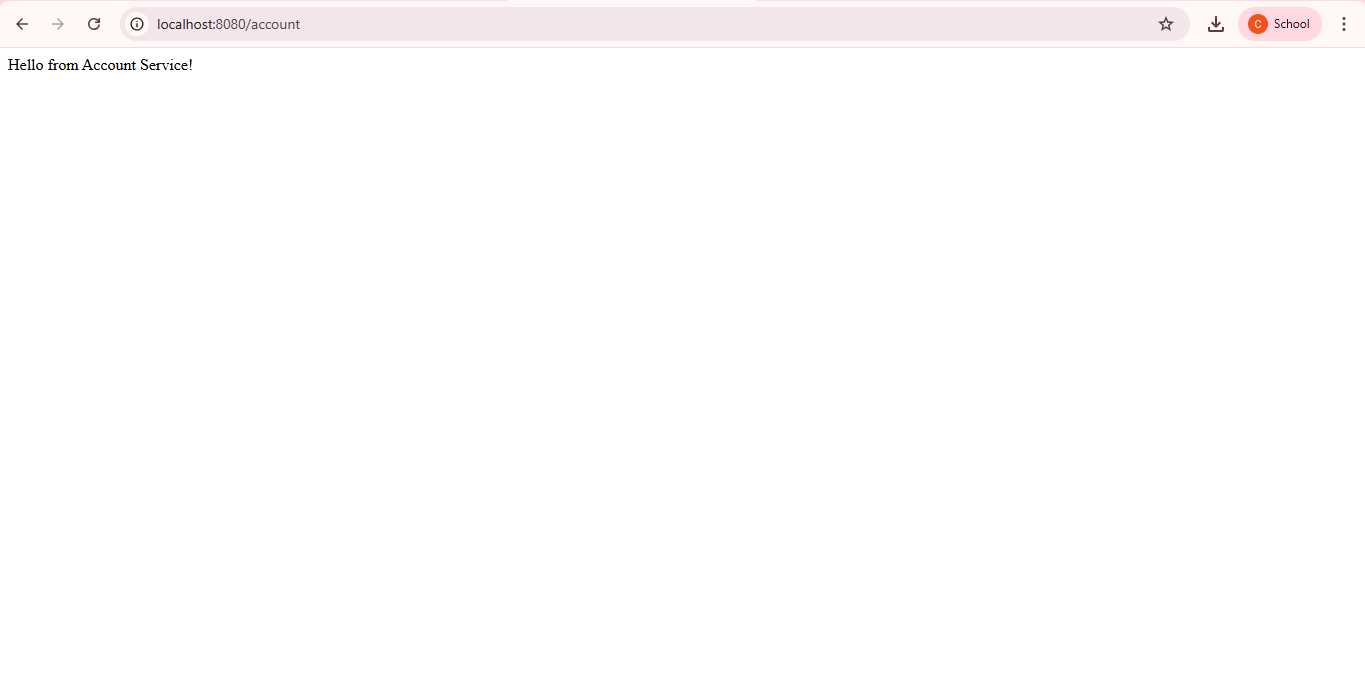
}

}

Excepted Output:

Hello from Account Service!

Output:



**Project Name:loan**

LoanApplication.java Code:

**package** com.cognizant.loan;

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

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

@SpringBootApplication

**public** **class** LoanApplication {

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

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

}

}

LoanController.java Code:

**package** com.cognizant.loan;

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

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

@RestController

**public** **class** LoanController {

@GetMapping("/loan")

**public** String getLoanInfo() {

**return** "Hello from Loan Service!";

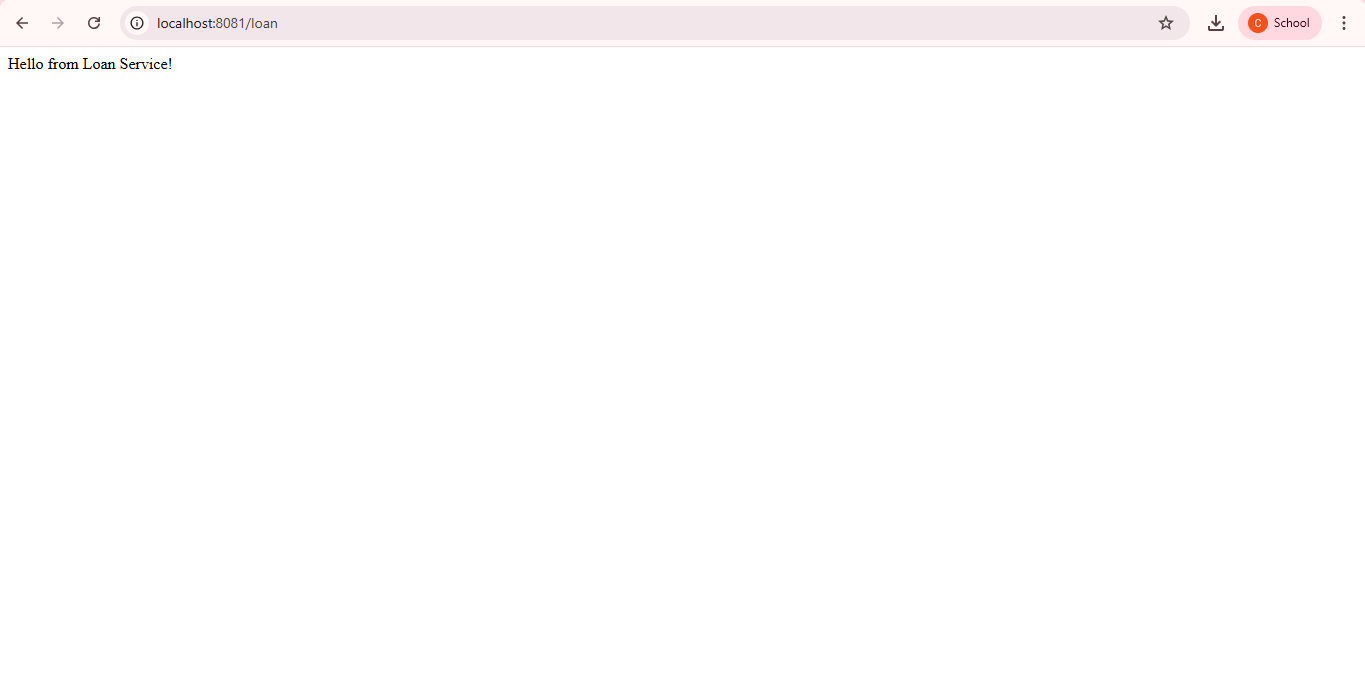
}

}

Expected Output:

Hello from Loan Service!

Output:



Project Name:greet-service

GreetController.java Code:

**package** com.microservices.greet;

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

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

@RestController

**public** **class** GreetController {

@GetMapping("/greet")

**public** String greet() {

**return** "Welcome to Greet Service!";

}

}

GreetServiceApplication.java Code:

**package** com.microservices.greet;

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

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

@SpringBootApplication

**public** **class** GreetServiceApplication {

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

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

}

}

application.properties Code:

# Greet service runs on port 8083

server.port=8083

# Application name for Eureka registration

spring.application.name=greet-service

# Eureka client configuration

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

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

# Optional: Debug Spring Web requests

logging.level.org.springframework.web=DEBUG

pom.xml Code:

<?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>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.microservices</groupId>

<artifactId>greet-service</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>greet-service</name>

<description>Greet microservice</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

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

<spring-cloud.version>2025.0.0</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<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>

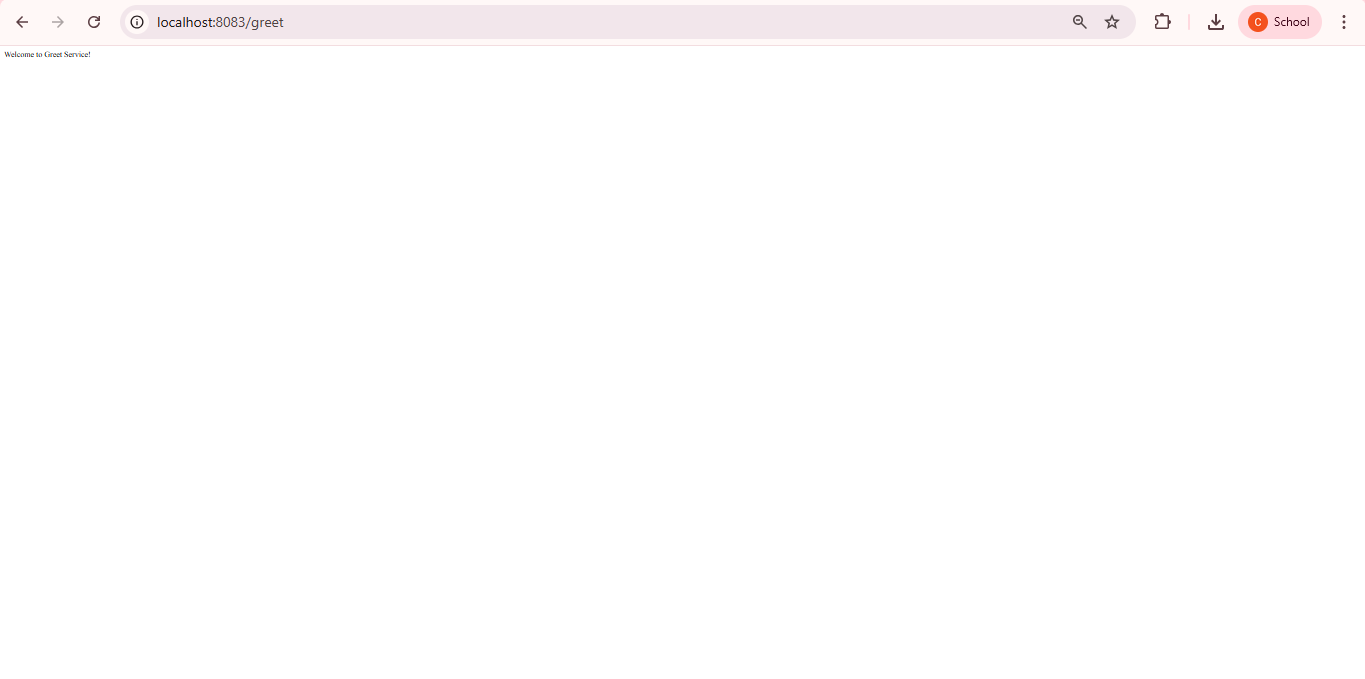
</build>

</project>

Excepted Output:

Welcome to Greet Service!

Output:



Project Name:api-gateway

ApiGatewayApplication.java Code:

**package** com.microservices.api\_gateway;

**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);

}

}

RouteConfig.java Code:

**package** com.microservices.api\_gateway;

**import** org.springframework.cloud.gateway.server.mvc.config.GatewayServerMvcAutoConfiguration;

**import** org.springframework.cloud.gateway.server.mvc.filter.BeforeFilterFunctions;

**import** org.springframework.cloud.gateway.server.mvc.handler.HandlerFunctions;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.servlet.function.RouterFunction;

**import** **static** org.springframework.cloud.gateway.server.mvc.filter.BeforeFilterFunctions.*setPath*;

**import** **static** org.springframework.cloud.gateway.server.mvc.handler.HandlerFunctions.~~http~~;

**import** **static** org.springframework.cloud.gateway.server.mvc.handler.HandlerFunctions.route;

@Configuration

**public** **class** RouteConfig {

@Bean

**public** RouterFunction<?> gatewayRoutes() {

**return** route("account\_route")

.path("/account/\*\*")

.filters(f -> f.prefixPath("/api").before(setPath("/account")))

.uri("http://localhost:8081")

.andRoute("loan\_route")

.path("/loan/\*\*")

.filters(f -> f.prefixPath("/api").before(setPath("/loan")))

.uri("http://localhost:8082")

.andRoute("greet\_route")

.path("/greet/\*\*")

.filters(f -> f.prefixPath("/api").before(setPath("/greet")))

.uri("http://localhost:8083");

}

}

application.properties Code:

# Gateway runs on port 8080

server.port=8080

# Register this gateway with Eureka

spring.application.name=api-gateway

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

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

# Route configuration (send /greet to greet-service)

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

spring.cloud.gateway.routes[0].uri=lb://greet-service

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

# Optional: Logging

logging.level.org.springframework.web=DEBUG

pom.xml Code:

<?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>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.microservices</groupId>

<artifactId>api-gateway</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>api-gateway</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

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

<spring-cloud.version>2025.0.0</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-starter-gateway-server-webmvc</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<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>

Excepted Output:

Hello from Account Service!

Hello from Loan Service!

Welcome to Greek Service!

Output:

