**Create Eureka Discovery Server and register microservices**

**CODE:**

**Greet-Service: -**

**GreetController Class: -**

package com.cognizant.greet\_service.controller;  
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 World!!";  
 }  
}

**application.properties file: -**

spring.application.name=greet-service  
server.port=8083  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

**GreetServiceApplication Class: -**

package com.cognizant.greet\_service;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
  
@EnableDiscoveryClient  
@SpringBootApplication  
public class GreetServiceApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(GreetServiceApplication.class, args);  
 }  
  
}

**pom.xml file: -**

<?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>greet-service</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>greet-service</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-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>

**Api-Gateway : -**

**ApiGatewayApplication Class: -**

package com.cognizant.api\_gateway;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
  
@EnableDiscoveryClient  
@SpringBootApplication  
public class ApiGatewayApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(ApiGatewayApplication.class, args);  
 }  
  
}

**LogFilter Class: -**

package filters;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.cloud.gateway.filter.GlobalFilter;  
import org.springframework.cloud.gateway.filter.GatewayFilterChain;  
import org.springframework.stereotype.Component;  
import org.springframework.web.server.ServerWebExchange;  
import reactor.core.publisher.Mono;  
  
@Component  
public class LogFilter implements GlobalFilter {  
  
 private static final Logger *logger* = LoggerFactory.*getLogger*(LogFilter.class);  
  
 @Override  
 public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {  
 *logger*.info("Request path: {}", exchange.getRequest().getPath());  
 return chain.filter(exchange);  
 }  
}

**application.properties file: -**

spring.application.name=api-gateway  
server.port=9090  
  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
  
spring.cloud.gateway.discovery.locator.enabled=false  
  
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-service/\*\*  
spring.cloud.gateway.routes[0].filters[0]=StripPrefix=1

**pom.xml file: -**

<?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> <!-- You can also use 3.5.3 if compatible -->  
 <relativePath/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>api-gateway</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>api-gateway</name>  
 <description>Spring Cloud Gateway for Microservices</description>  
  
 <properties>  
 <java.version>21</java.version>  
 <spring-cloud.version>2025.0.0</spring-cloud.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Actuator for health checks -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-actuator</artifactId>  
 </dependency>  
  
 <!-- Spring Cloud Gateway (WebFlux-based) -->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-gateway</artifactId>  
 </dependency>  
  
 <!-- Eureka Client -->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>  
 </dependency>  
  
 <!-- Devtools for live reload -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Testing -->  
 <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>

**Eureka-Discovery: -**

**EurekaDiscoveryServerApplication Class: -**

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 file: -**

spring.application.name=eureka-discovery-server  
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

**pom.xml file: -**

<?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 Spring Cloud Microservices</description>  
  
 <properties>  
 <java.version>21</java.version>  
 <spring-cloud.version>2025.0.0</spring-cloud.version>  
 </properties>  
  
 <dependencies>  
 <!-- Eureka Server dependency -->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  
 </dependency>  
  
 <!-- Spring Boot DevTools (optional) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Spring Boot Test -->  
 <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>  
 <!-- Spring Boot Maven Plugin -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <configuration>  
 <mainClass>com.cognizant.greet\_service.GreetServiceApplication</mainClass>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

OUTPUT:



