**COGNIZANT DIGI NURTURE 4.0**

**WEEK 5 MANDATORY HANDS ON BY (6421405)**

**Microservices with Spring Boot 3 and Spring Cloud**

Exersice 1: Creating Microservices for account and loan

**Account**

**Generate Spring Boot Project**

* + **Group:** com.cognizant
  + **Artifact:** account
* **Dependencies:**
  + *Spring Boot DevTools*
  + *Spring Web*
* Click **Generate**.
* Download the ZIP file.

**JAVA CLASS :**

Package : account

Class : AccountController

**AccountController.java**

package com.cognizant.account.account;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Map<String, Object> getAccount(@PathVariable String number) {

return Map.of(

"name ": "Uthaya",

"balance": 12000,

"type": "savings",

"number": "00987987973432"

);

}

}

**AccountApplication.java**

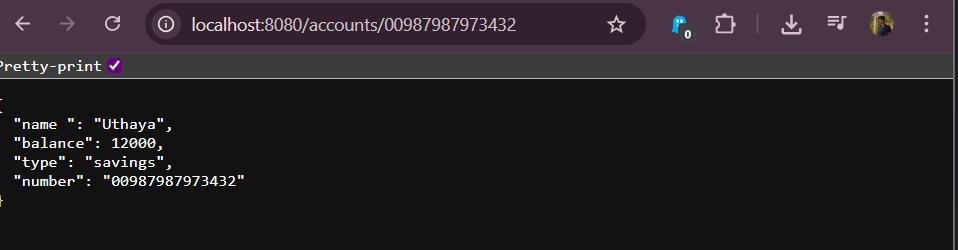
package com.cognizant.account;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AccountApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(AccountApplication.class, args);  
 }  
  
}

**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>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>account</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>account</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>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</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>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

OUTPUT**:**

http://localhost:8080/accounts/00987987973432



**Loan Microservice**

• Follow similar steps specified for Account Microservice and implement a service API to get loan account details o Method: GET o Endpoint: /loans/{number} o Sample Response. Just a dummy response without any backend connectivity.

**Generate Spring Boot Project**

* + **Group:** com.cognizant
  + **Artifact:** account
* **Dependencies:**
  + *Spring Boot DevTools*
  + *Spring Web*
* Click **Generate**.
* Download the ZIP file.

**JAVA CLASS :**

Package : Loan

Class : LoanController

**LoanController.java**

package com.cognizant.loan.loan;  
  
  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Map;  
  
@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "Car",  
 "loan", 500000,  
 "emi", 4597,  
 "tenure", 18  
 );  
 }  
}

**LoanApplication.java**

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);  
 }  
  
}

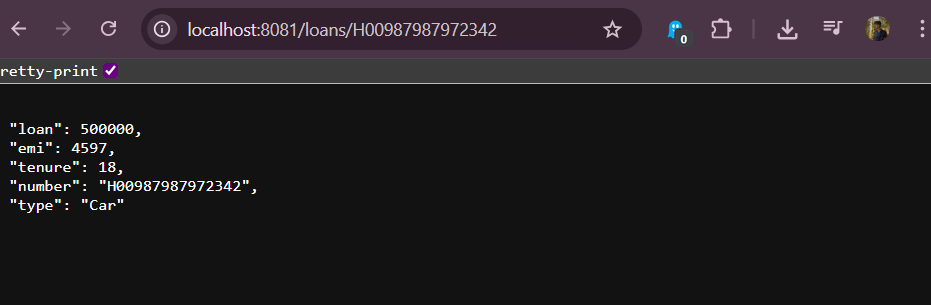
* Change the port because the port number 8080 is already in use make changes in application.propeties
* Server.port = 8081

**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>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>loan</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>loan</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>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</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>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**OUTPUT:**

http://localhost:8081/loans/H00987987972342



**ADDITIONAL HANDSON**

**Create Eureka Discovery Server and register microservices**

Create and Launch Eureka Discovery Server

• Using https://start.spring.io generate a project with following configuration:

* + - Group: com.cognizant
    - Artifact: eureka-discovery-server
    - Module: Spring Cloud Discovery > Eureka Server

**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>3.5.3</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</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-server</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>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-actuator</artifactId>  
 </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>

**EurekaDiscoverServerApplication:**

package com.cognizant.eurekadiscoveryserver;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  
  
@EnableEurekaServer  
@SpringBootApplication  
public class EurekaDiscoveryServerApplication {  
  
 public static void main(String[] args) {  
  
 SpringApplication.*run*(EurekaDiscoveryServerApplication.class, args);  
 }  
  
}

**Application.properties**

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  
# Expose all actuator endpoints  
management.endpoints.web.exposure.include=\*  
  
# Optional: Add info metadata  
info.app.name=account-service  
info.app.version=1.0.0

**JAVA CLASS :**

Package : account

Class : AccountController

**AccountController.java**

package com.cognizant.account.account;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Map<String, Object> getAccount(@PathVariable String number) {

return Map.of(

"name ": "Uthaya",

"balance": 12000,

"type": "savings",

"number": "00987987973432"

);

}

}

**AccountApplication.java**

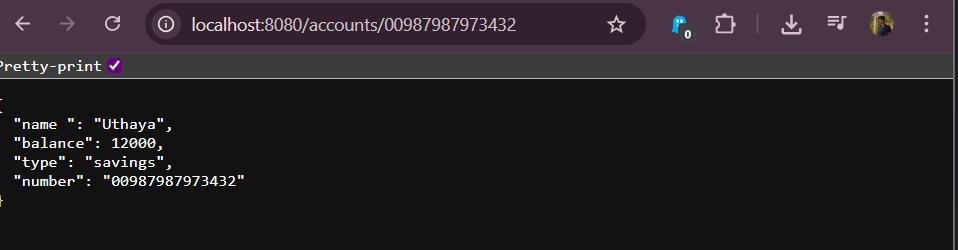
package com.cognizant.account;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AccountApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(AccountApplication.class, args);  
 }  
  
}

**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>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>account</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>account</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>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</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>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

OUTPUT**:**

http://localhost:8080/accounts/00987987973432



**JAVA CLASS :**

Package : Loan

Class : LoanController

**LoanController.java**

package com.cognizant.loan.loan;  
  
  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Map;  
  
@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "Car",  
 "loan", 500000,  
 "emi", 4597,  
 "tenure", 18  
 );  
 }  
}

**LoanApplication.java**

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);  
 }  
  
}

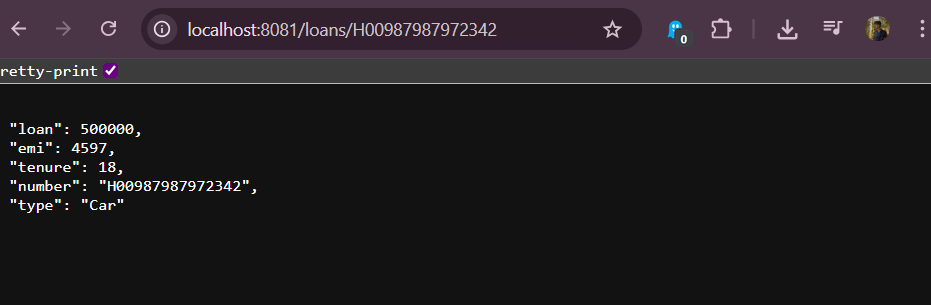
* Change the port because the port number 8080 is already in use make changes in application.propeties
* Server.port = 8081

**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>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>loan</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>loan</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>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</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>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**OUTPUT:**

http://localhost:8081/loans/H00987987972342



**GreetController.java**

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 greet() {  
 return "Hello World";  
 }  
}

**GreetServiceApplication.java**

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);  
 }  
  
}

**Application.properties**  
spring.application.name=greet-service  
server.port=8081  
  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/  
eureka.client.register-with-eureka=true  
eureka.client.fetch-registry=true

**OUTPUT:**



**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>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>api-gateway</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
 <name>api-gateway</name>  
 <description>Spring Cloud Gateway with Eureka</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.5</version> <!-- Latest compatible version -->  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>17</java.version> <!-- Java 17 or higher -->  
 <spring-cloud.version>2023.0.1</spring-cloud.version>  
 </properties>  
  
 <dependencies>  
  
 <!-- Required for Gateway -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-webflux</artifactId>  
 </dependency>  
  
 <!-- Spring Cloud Gateway -->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-gateway</artifactId>  
 </dependency>  
  
 <!-- Eureka Discovery Client -->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>  
 </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>  
 <!-- Spring Boot Plugin -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <configuration>  
 <mainClass>com.cognizant.api\_gateway.ApiGatewayApplication</mainClass>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**LogFilter.java**

// src/main/java/com/example/apigateway/filter/LogFilter.java  
package com.cognizant.api\_gateway.filter;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.cloud.gateway.filter.GatewayFilterChain;  
import org.springframework.cloud.gateway.filter.GlobalFilter;  
import org.springframework.core.Ordered;  
import org.springframework.stereotype.Component;  
import org.springframework.web.server.ServerWebExchange;  
import reactor.core.publisher.Mono;  
  
@Component  
public class LogFilter implements GlobalFilter, Ordered {  
  
 private static final Logger *logger* = LoggerFactory.*getLogger*(LogFilter.class);  
  
 @Override  
 public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {  
 *logger*.info("Incoming Request Path: {}", exchange.getRequest().getPath());  
 return chain.filter(exchange);  
 }  
  
 @Override  
 public int getOrder() {  
 return -1; // High precedence  
 }  
}

**ApiGatewayApplication**

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

**Application.properties**

server.port=9090  
spring.application.name=api-gateway  
  
# Register with Eureka  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
eureka.client.register-with-eureka=true  
eureka.client.fetch-registry=true  
  
# Force lowercase route matching  
spring.cloud.gateway.discovery.locator.enabled=true  
spring.cloud.gateway.discovery.locator.lower-case-service-id=true

**OUTPUT**

