**Digital Nurture 4.0**

Week 5-Micro Services with Spring Boot3 and Spring Cloud

# File Name : 2. Microservices with API gateway

# HandsOn: Creating Microservices for account and loan

Program:

* Create Spring Boot Project
* Go to <https://start.spring.io>

**Project Setup:**

* **Project**: Maven
* **Language**: Java
* **Spring Boot**: 3.1.x or 3.5.x
* **Group**: com.cognizant
* **Artifact**: account
* **Name**: account
* **Package name**: com.cognizant.account

**Dependencies:**

* Spring Boot DevTools
* Spring Web
* **Click**: "Generate", then unzip.
* Import the project into IntelliJ IDEA
* Create package and the file AccountApplication.java in src/main/java/com/cognizant/account/AccountApplication.java

**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);  
 }  
  
}

* Add the following code in the application.properties:

Application.properties:

spring.application.name=account  
server.port=8080

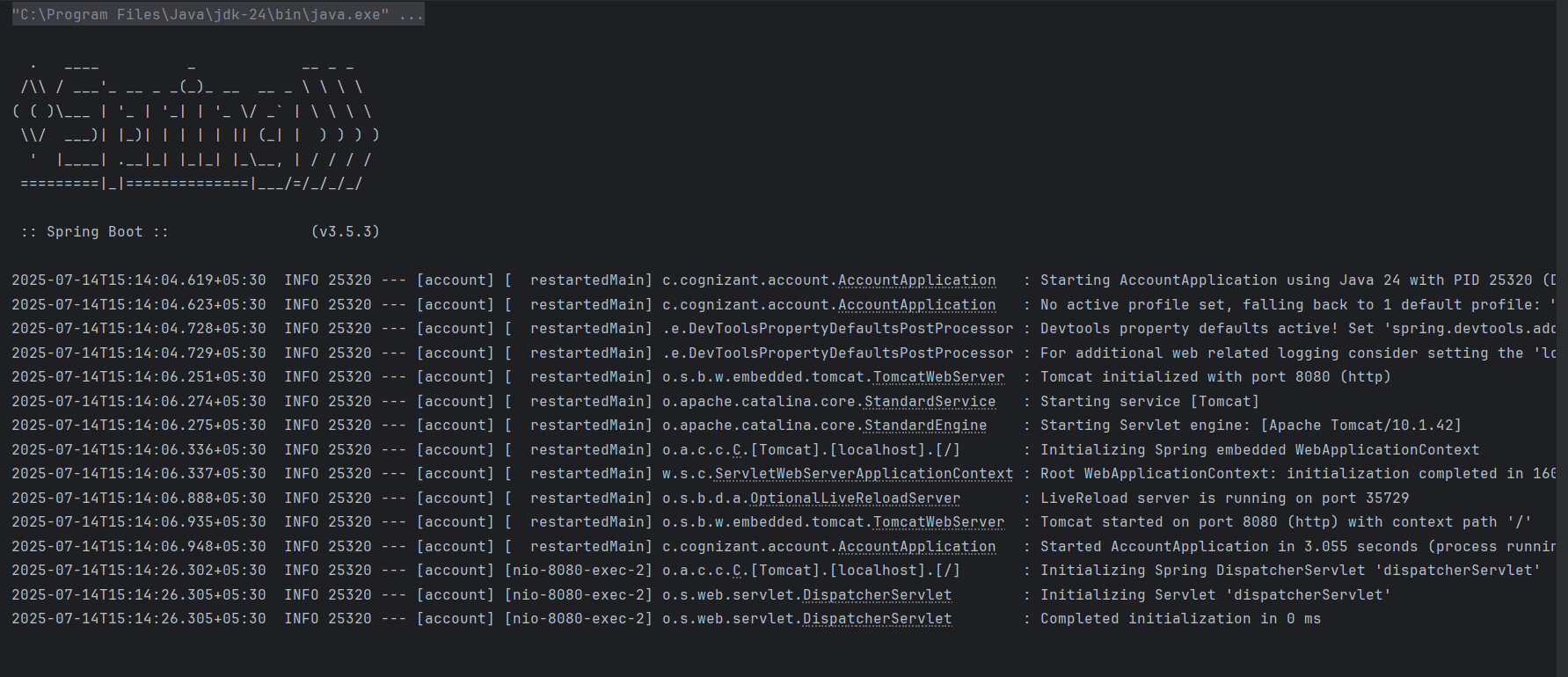
* Create the file AccountController.java in src/main/java/com/cognizant/account/controller/AccountController.java.

AccountController.java:

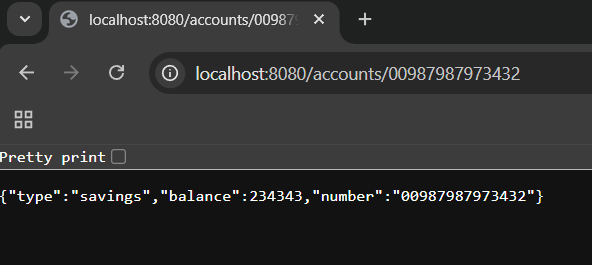
package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 return Map.*of*(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
 }  
}

* Run the AccountApplication.java and run the urlmin the browser <http://localhost:8080/accounts/00987987973432>

Output:



In browser:



# HandsOn: Create Eureka Discovery Server and register

# Microservices

Program:

* **Eureka Discovery Server**

**Spring Initializr Configuration**

* **Group:** com.cognizant
* **Artifact:** eureka-discovery-server
* **Dependencies:**
  + Spring Boot DevTools
  + Spring Web
  + Spring Cloud Discovery > Eureka Server
* Download the file,unzip and open in the intel ij.
* Edit the EurekaDiscoveryServerApplication.java

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);  
 }  
}

* Edit the pom.xml.

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>eureka-discovery-server</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>eureka-discovery-server</name>  
 <description>Eureka Discovery Server</description>  
  
 <properties>  
 <java.version>17</java.version>  
 <spring-cloud.version>2023.0.1</spring-cloud.version>  
 </properties>  
  
 <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>  
  
 <dependencies>  
 <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>  
 </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>

* + Configure the application.properties

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

* GREET Microservice-(greet-service)
* **Spring Initializr Configuration**
* **Group:** com.cognizant
* **Artifact:** greet-service
* **Dependencies:**
* Spring Boot DevTools
* Spring Web
* Eureka Discovery Client
* Download the file,unzip and open in the intel ij.
* Open the GreetServerApplication.java and edit it.

GreetServerApplication.java:

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 GreetServiceApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(GreetServiceApplication.class, args);  
 }  
}

* Create a package named controller in src/main/java/com/cognizant/greet/controller/GreetController.java.

GreetController.java:

package com.cognizant.greet.controller;  
  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
public class GreetController {  
  
 @GetMapping("/greet")  
 public String greet() {  
 return "Hello World";  
 }  
}

* Configure the application.properties.

Application.properties:

spring.application.name=greet-service  
server.port=8082  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

API Gateway

Spring Initializr Configuration

* **Group:** com.cognizant
* **Artifact:** api-gateway
* **Dependencies:**
  + Spring Boot DevTools
  + Spring Cloud Discovery > Eureka Discovery Client
  + Spring Cloud Gateway > Spring Cloud Gateway
* Configure the pom.xml & application.properties.

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>3.3.1</version>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>api-gateway</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>api-gateway</name>  
 <description>Spring Cloud Gateway</description>  
  
 <properties>  
 <java.version>17</java.version>  
 <spring-cloud.version>2023.0.1</spring-cloud.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-gateway</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-starter-webflux</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>

Application.properties:

server.port=9090  
spring.application.name=api-gateway  
  
spring.main.web-application-type=reactive  
  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
spring.cloud.compatibility-verifier.enabled=false  
  
spring.cloud.gateway.discovery.locator.enabled=true  
spring.cloud.gateway.discovery.locator.lower-case-service-id=true

* Edit the ApiGatewayApplication.java.

ApiGatewayApplication.java:

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);  
 }  
}

* Create a package named filter in src/main/java/com/cognizant/api\_gateway/filter/LogFilter.java.

LogFilter.java:

package com.cognizant.apigateway.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.annotation.Order;  
import org.springframework.stereotype.Component;  
import org.springframework.web.server.ServerWebExchange;  
import reactor.core.publisher.Mono;  
  
@Component  
@Order(1)  
public class LogFilter implements GlobalFilter {  
 private 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);  
 }  
}

Account Microservice

* Spring Initializr Setup
* Group: com.cognizant
* Artifact: account-service
* Dependencies: Spring Web, Eureka Discovery Client, Spring Boot DevTools
* Download the file,unzip,open in the intel ij.
* Configure the pom.xml,application.properties

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>24</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>

application.properties:

spring.application.name=account-service  
server.port=8080  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

* The main class AccountServiceApplication.java

AccountServiceApplication.java:

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

* Create a package named filter in src/main/java/com/cognizant/account/controller/AccountController.java.

AccountController.java:

package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 return Map.*of*(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
 }  
}

Loan MicroService

**Spring Initializr Setup**

* Group: com.cognizant
* Artifact: loan-service
* Dependencies: Spring Web, Eureka Discovery Client, Spring Boot DevTools
* Download the file,unzip,open in the intel ij.
* Configure the application.properties.

application.properties:

spring.application.name=loan-service  
server.port=8081  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

* Create the main class LoanServiceApplication.java

LoanServiceApplication.java:

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

* Create a package named filter in src/main/java/com/cognizant/loan/controller/LoanController.java.

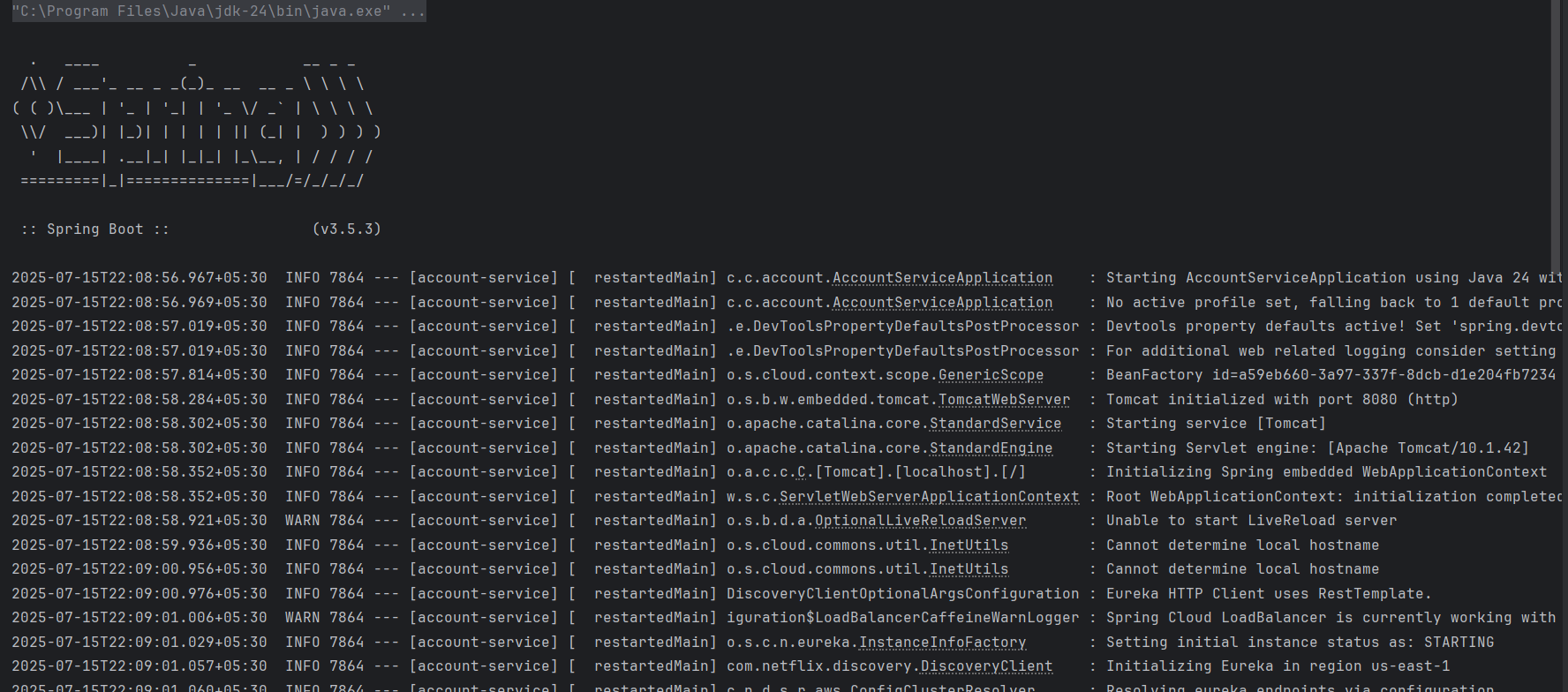
LoanController.java:

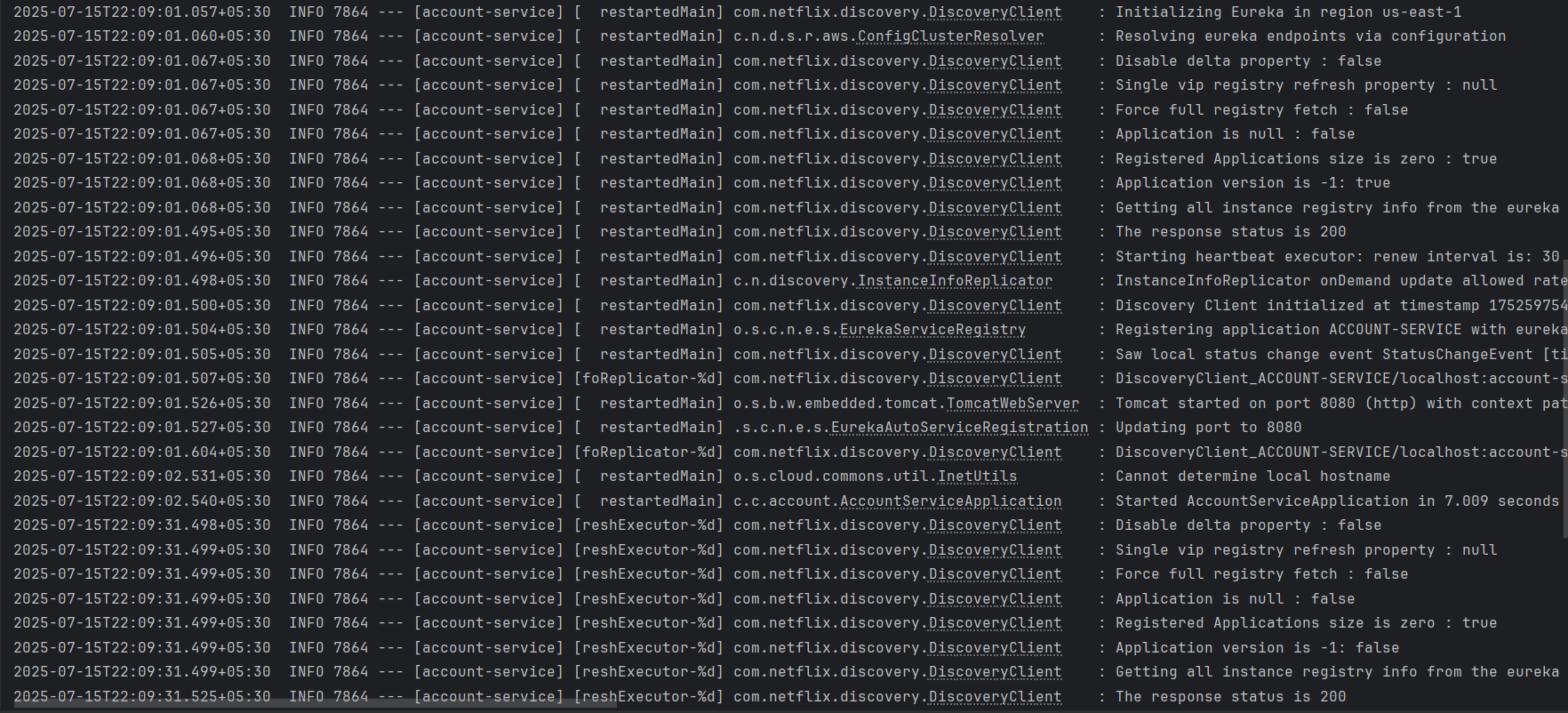
package com.cognizant.loan.controller;  
  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Map;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 return Map.*of*(  
 "number", number,  
 "type", "car",  
 "loan", 400000,  
 "emi", 3258,  
 "tenure", 18  
 );  
 }  
}

Run the all the files and related urls in the browser:

**Output:**

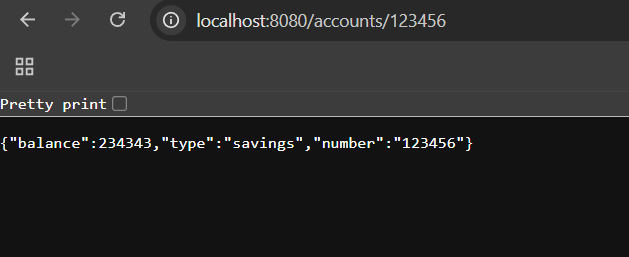
* AccountServiceApplication.java:





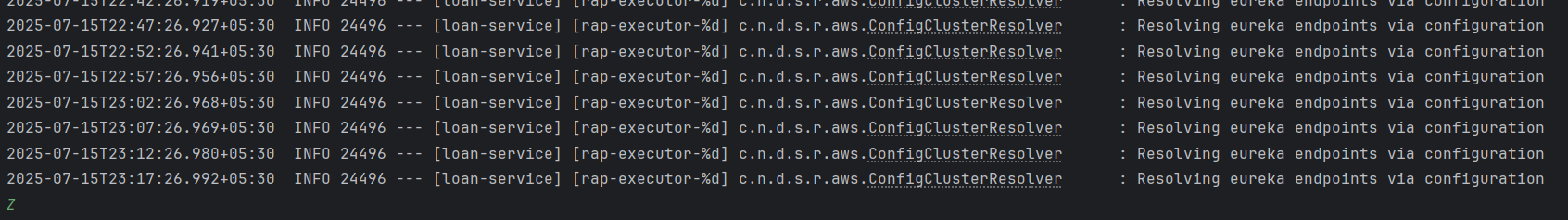
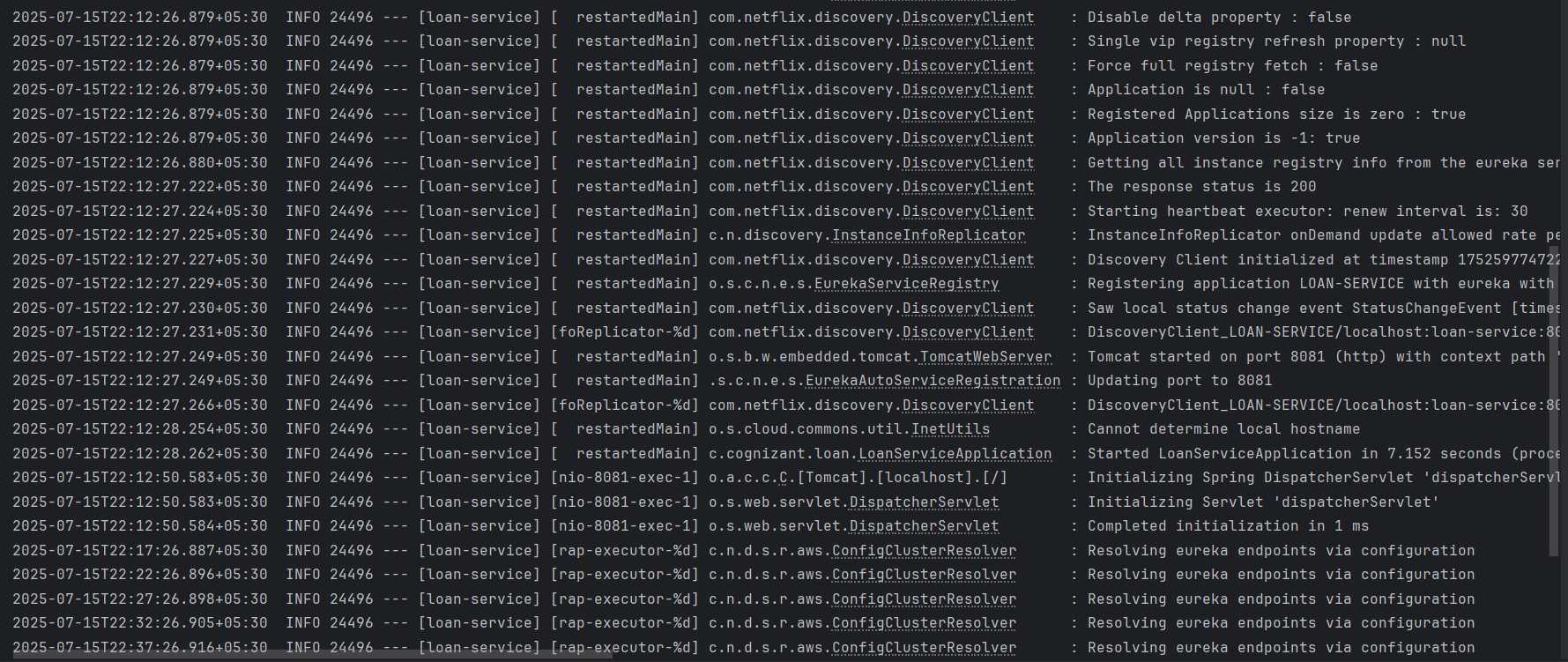


In Browser run the url <http://localhost:8080/accounts/123456>:

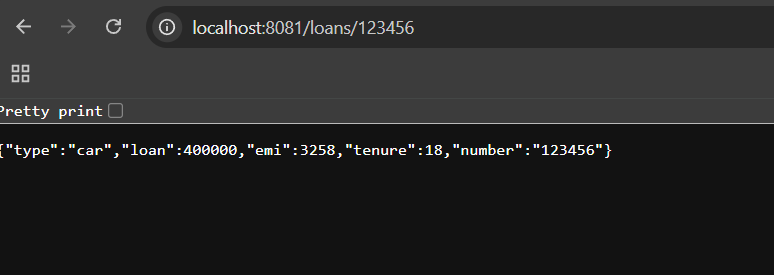


LoanServiceApplication.java:

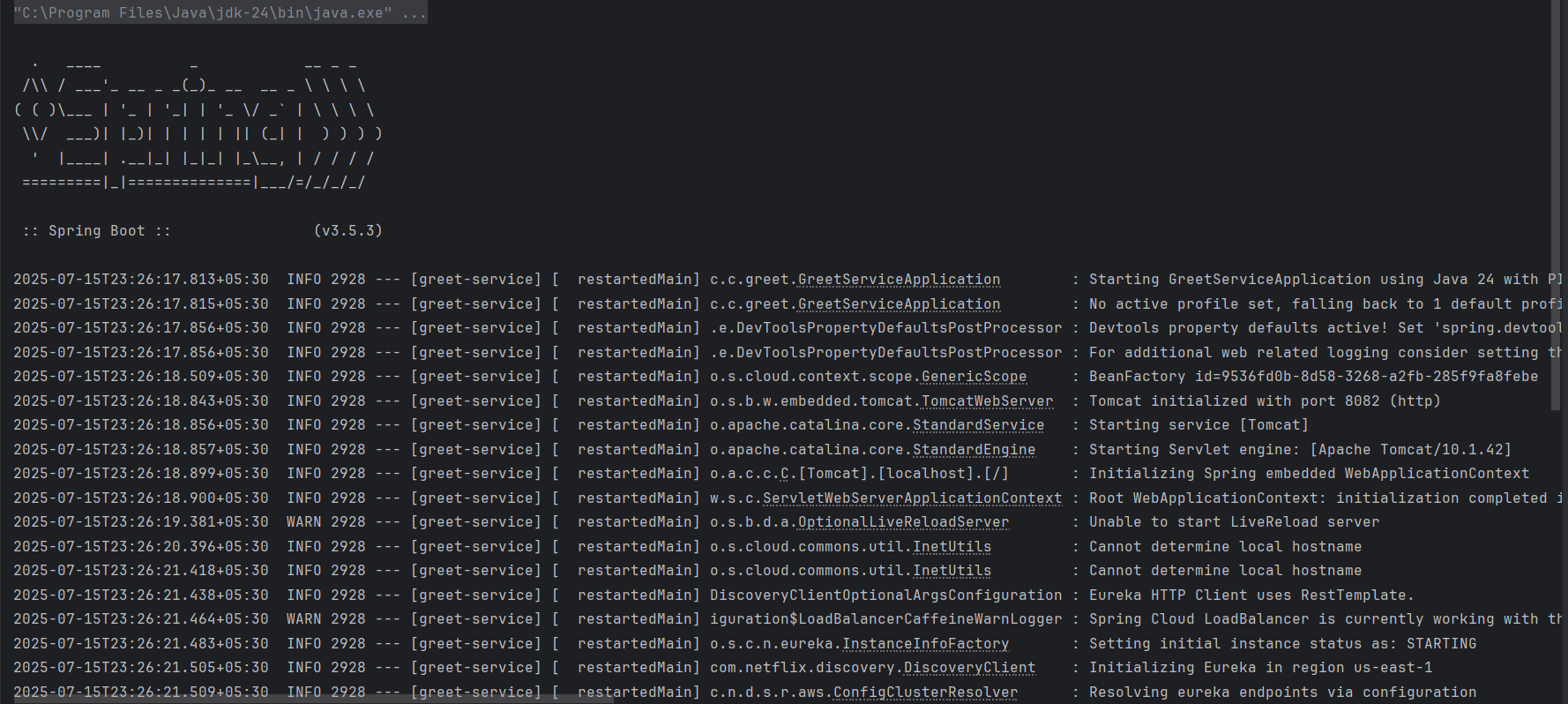


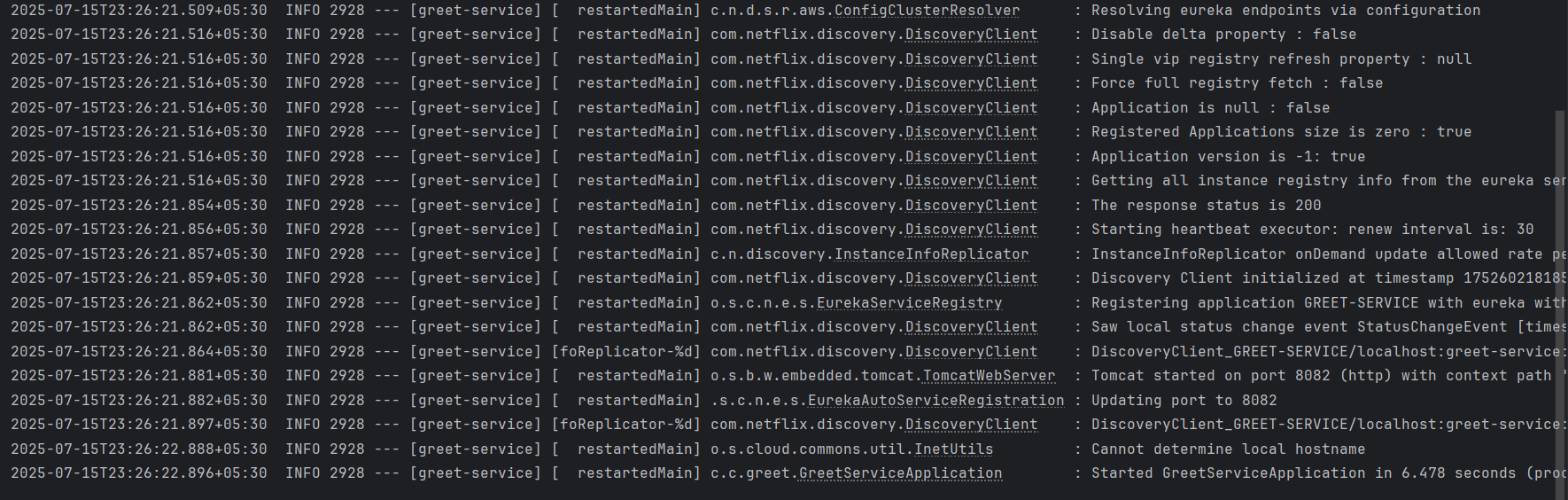


* Run this url in browser http://localhost:8081/loans/123456

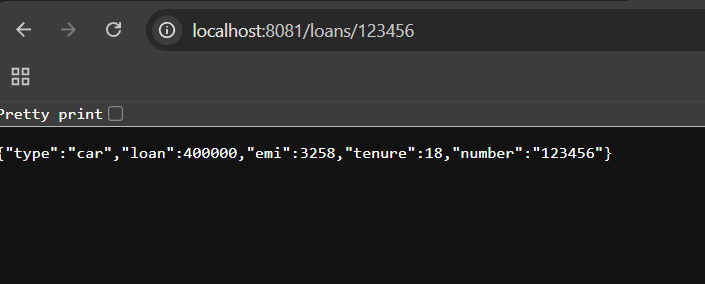


* Run GreetServiceApplication.java

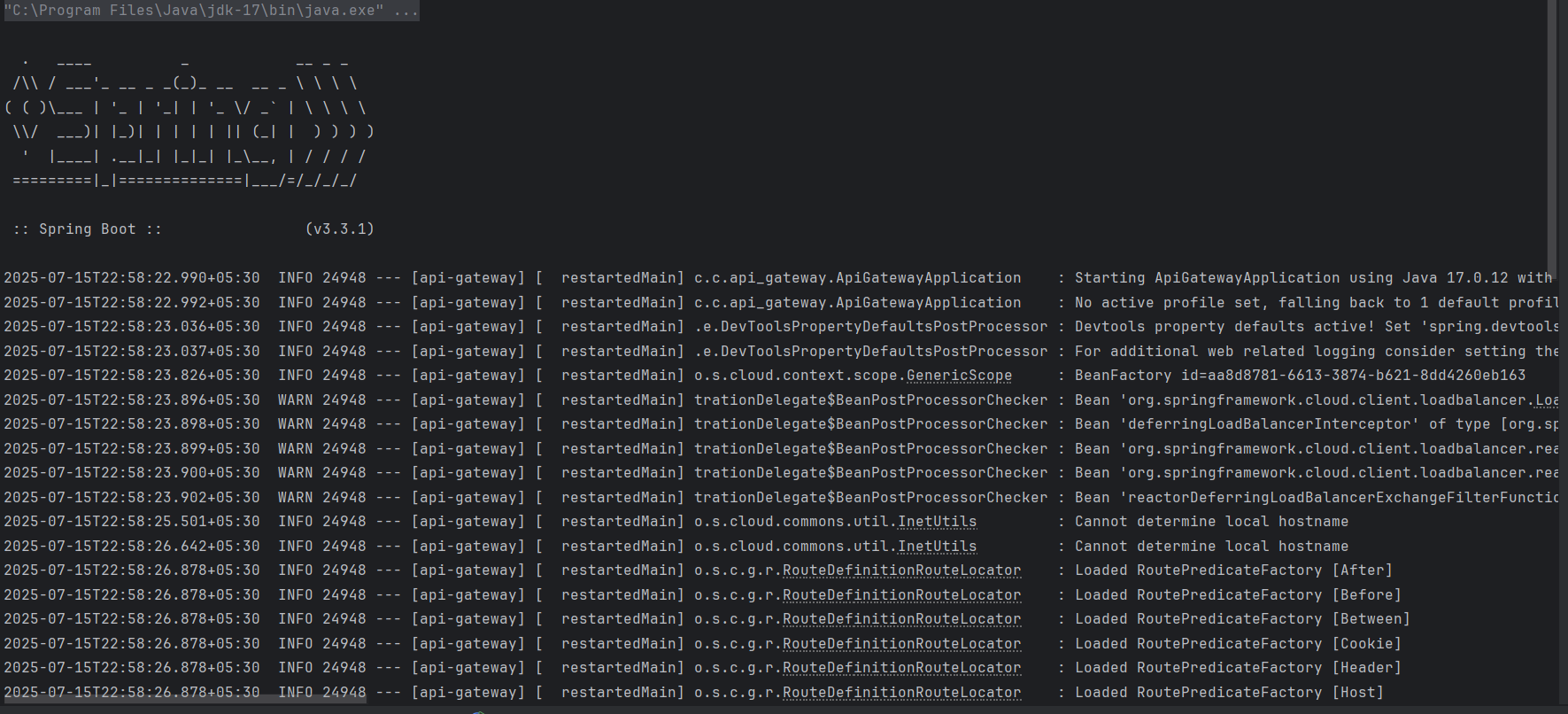


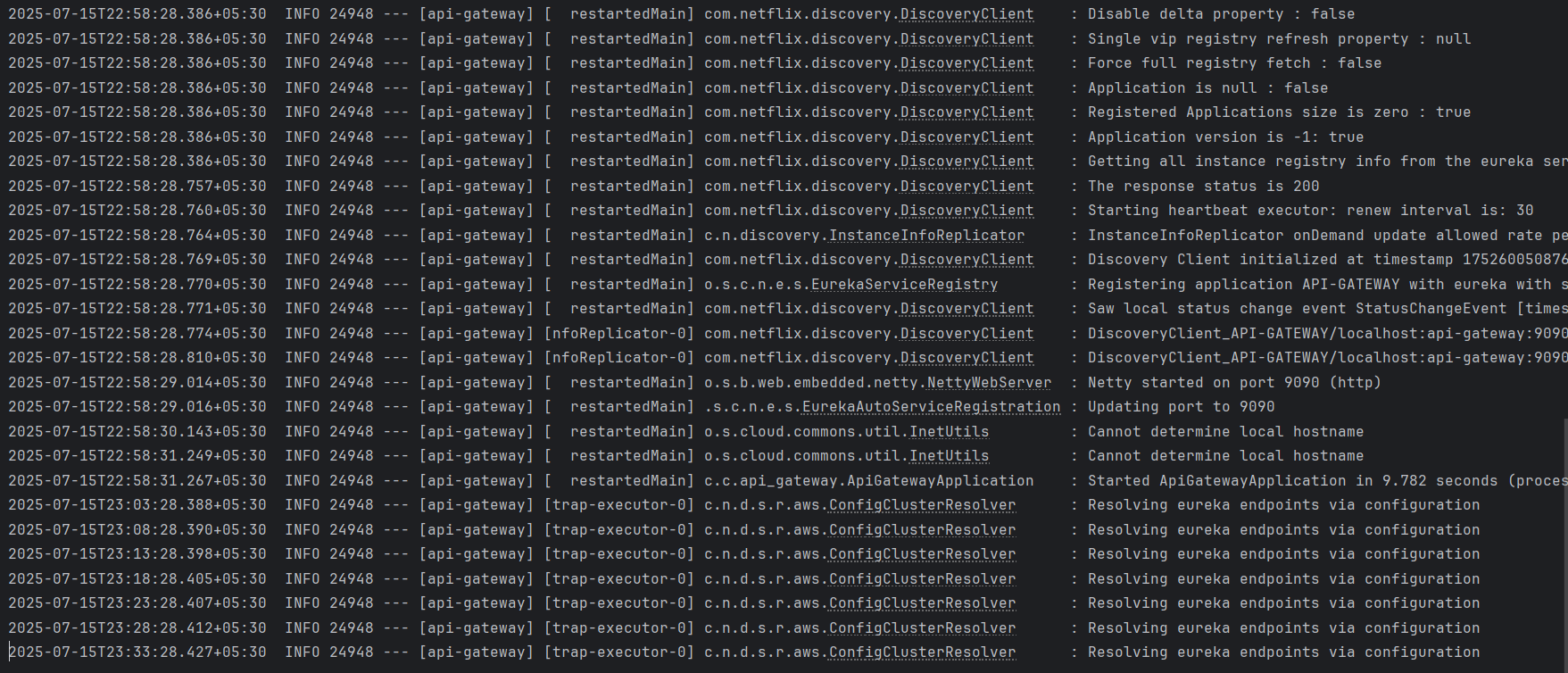


In browser run this url http://localhost:8082/greet.

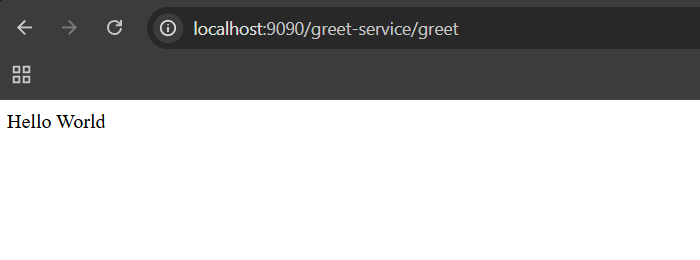


* Run the ApiGatewaySollution.

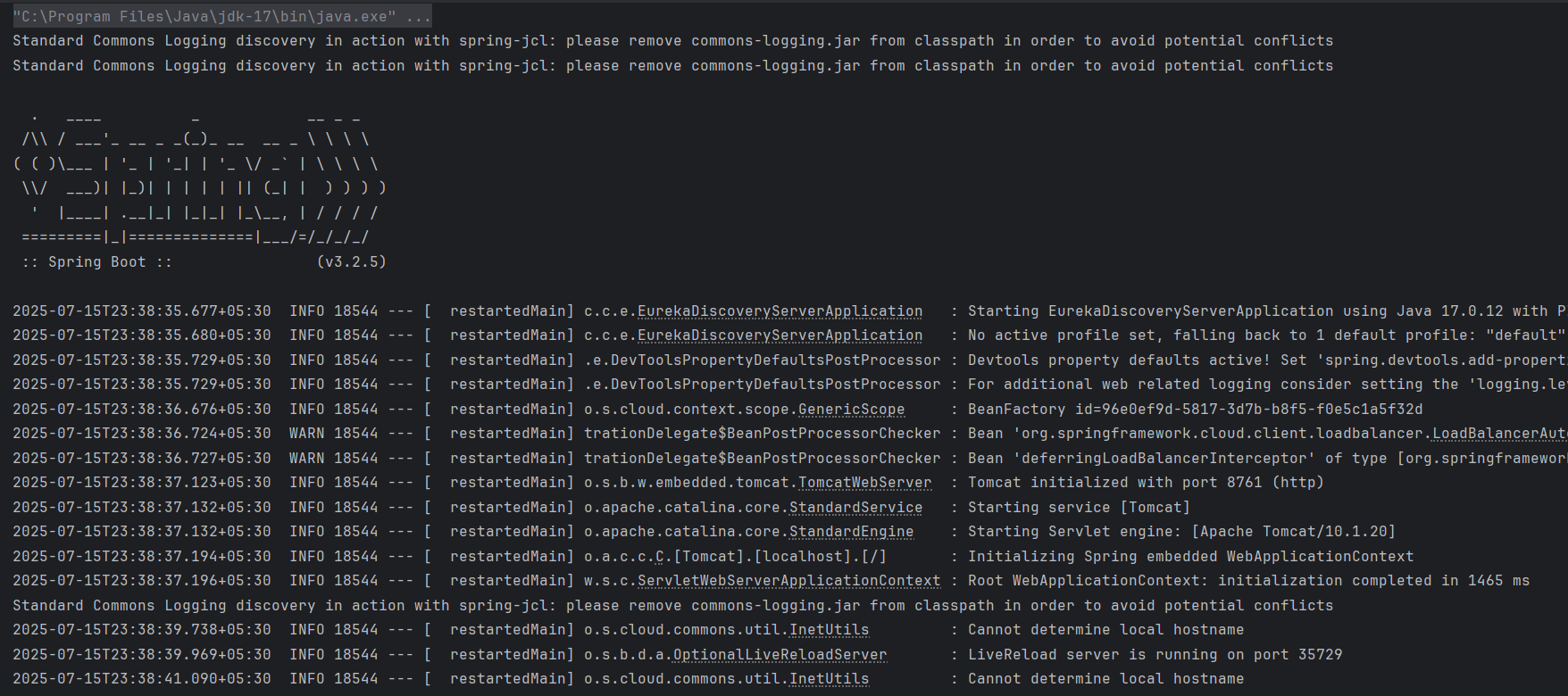


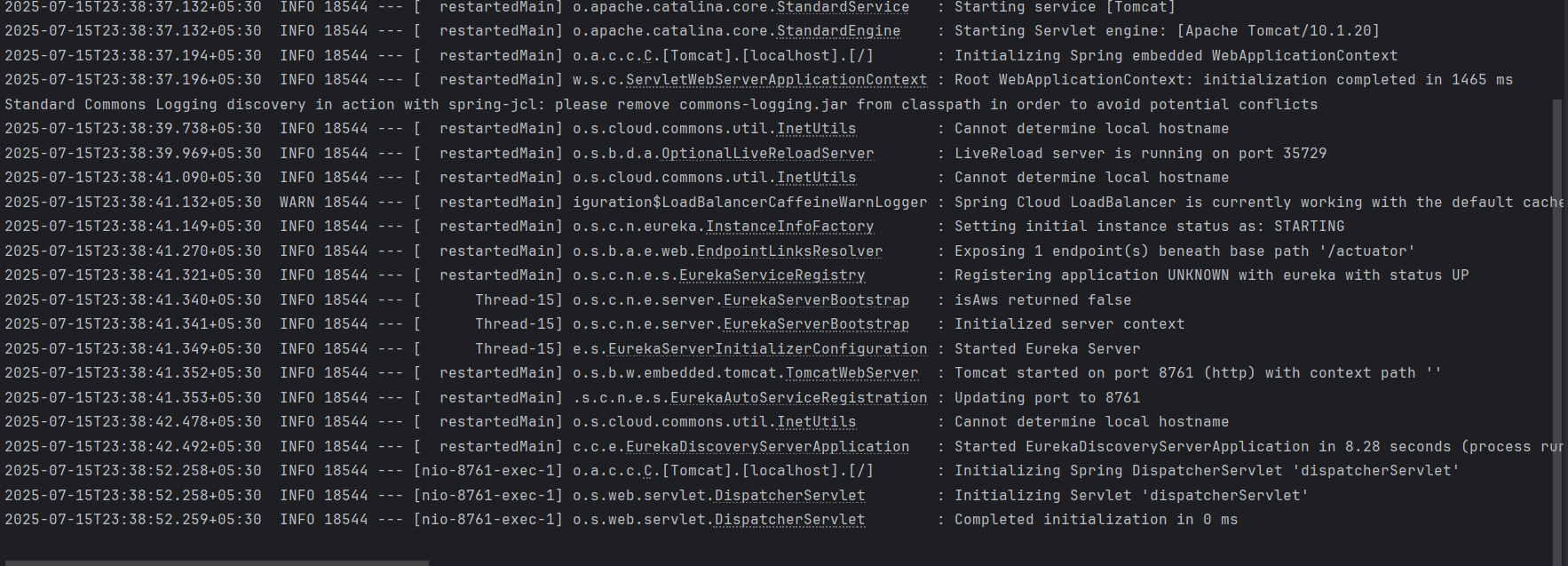


Run the url <http://localhost:9090/greet-service/greet> in browser.



* Run the EurekaDiscoveryServerApplicstion.java.





* Run the url <http://localhost:8761> in the browser.

