**Creating Microservices for account and loan**

**STEP 1:** Create the Spring project -eureka-server

**Dependencies** (when creating in Eclipse)

* Eureka Server
* Spring Boot Actuator

**pom.xml**

<project xmlns="https://maven.apache.org/POM/4.0.0" xmlns:xsi="https://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.example</groupId>

<artifactId>eureka-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eureka-server</name>

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

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</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-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>

**EurekaServerApplication.java**

package com.example.eureka\_server;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

**application.yml**

server:

port: 8761

spring:

application:

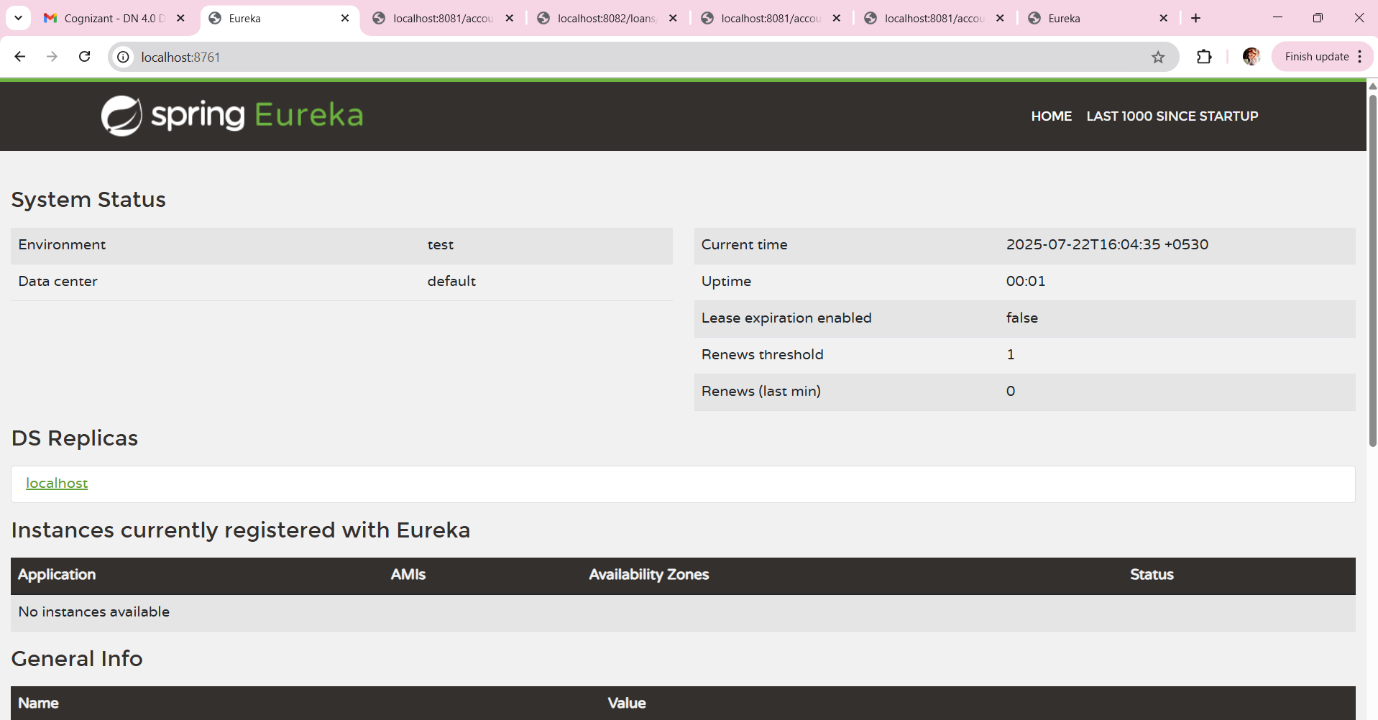
name: eureka-server

eureka:

client:

register-with-eureka: false

fetch-registry: false

****

**STEP 2:** Create next spring project ->account-service

**Dependencies:**

* Spring Web
* Eureka Discovery Client
* Spring Boot Actuator

**AccountServiceApplication.java**

package com.example.account\_service;

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);

}

}

**application.yml**

server:

port: 8081

spring:

application:

name: account-service

eureka:

client:

service-url:

defaultZone: <http://localhost:8761/eureka>

**AccountController.java**

package com.example.account\_service.controller;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

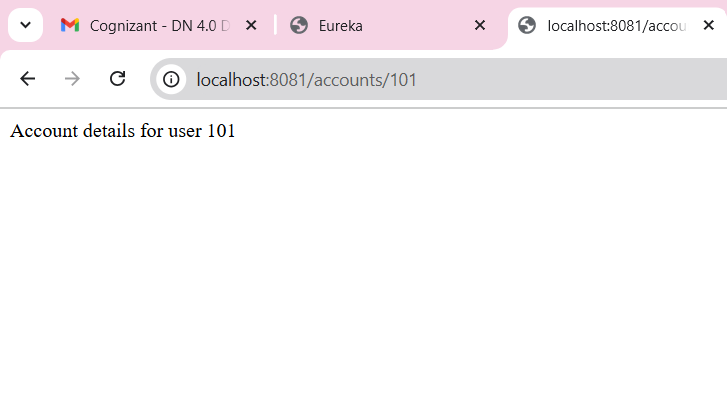
@GetMapping("/{id}")

public String getAccount(@PathVariable String id) {

return "Account details for user " + id;

}

**OUTPUT:**

****

**STEP 3: Create loan-service**

**Dependencies:**

* Spring Web
* Eureka Discovery Client
* Spring Boot Actuator

**LoanServiceApplication.java**

package com.example.loan\_service;

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);

}

}

**application.yml**

server:

port: 8082

spring:

application:

name: loan-service

eureka:

client:

service-url:

defaultZone: <http://localhost:8761/eureka>

**LoanController.java**

package com.example.loan\_service.controller;

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

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{id}")

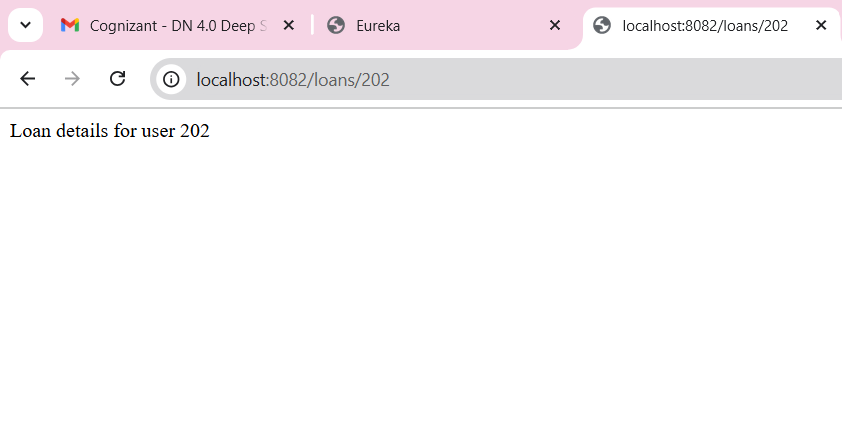
public String getLoan(@PathVariable String id) {

return "Loan details for user " + id;

}

}

**OUTPUT:**

****

**STEP 4: Create api-gateway (Custom, Manual)**

**Dependencies:**

* Spring Web
* Eureka Discovery Client
* Spring Boot Actuator

**ApiGatewayApplication.java**

package com.example.api\_gateway;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

import org.springframework.context.annotation.Bean;

import org.springframework.web.client.RestTemplate;

@SpringBootApplication

@EnableDiscoveryClient

public class ApiGatewayApplication {

public static void main(String[] args) {

SpringApplication.run(ApiGatewayApplication.class, args);

}

@Bean

public RestTemplate restTemplate() {

return new RestTemplate();

}

}

**application.yml**

server:

port: 8080

spring:

application:

name: api-gateway

eureka:

client:

service-url:

defaultZone: <http://localhost:8761/eureka>

**GatewayController.java**

package com.example.api\_gateway.controller;

import org.springframework.beans.factory.annotation.Autowired;

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

import org.springframework.cloud.client.ServiceInstance;

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

import org.springframework.web.client.RestTemplate;

import java.util.List;

@RestController

public class GatewayController {

@Autowired

private RestTemplate restTemplate;

@Autowired

private DiscoveryClient discoveryClient;

private String getServiceUrl(String serviceName) {

List<ServiceInstance> instances = discoveryClient.getInstances(serviceName);

if (instances.isEmpty()) {

throw new RuntimeException("No instance available for " + serviceName);

}

return instances.get(0).getUri().toString();

}

@GetMapping("/accounts/{id}")

public String forwardToAccount(@PathVariable String id) {

String url = getServiceUrl("account-service") + "/accounts/" + id;

return restTemplate.getForObject(url, String.class);

}

@GetMapping("/loans/{id}")

public String forwardToLoan(@PathVariable String id) {

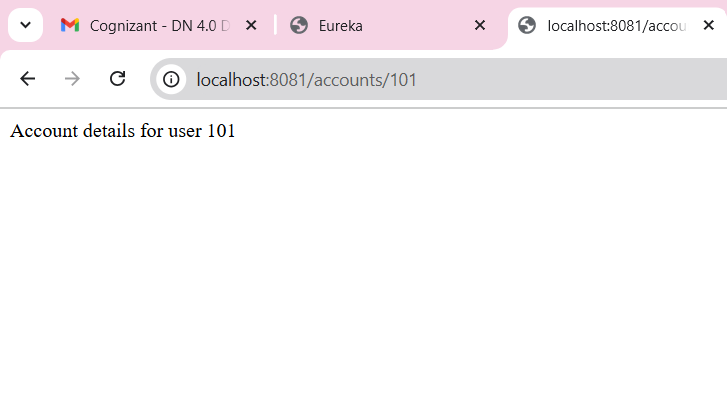
String url = getServiceUrl("loan-service") + "/loans/" + id;

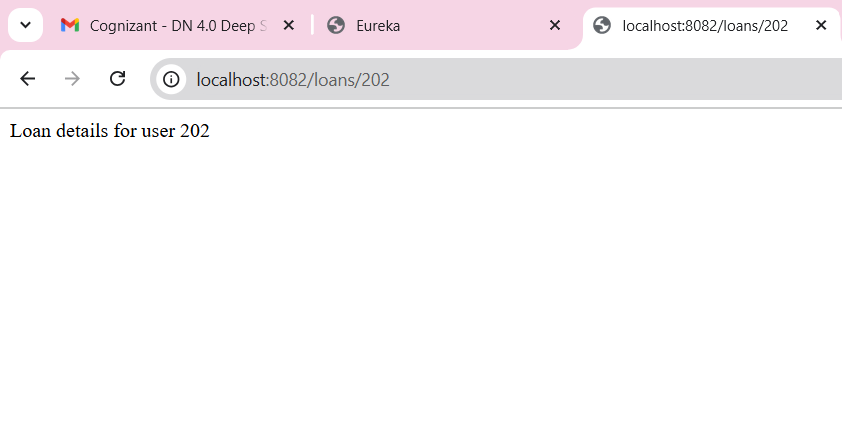
return restTemplate.getForObject(url, String.class);

}

}

**OUTPUT:**

****

****

**STEP 5:** Run All Projects

Run in this order:

1. eureka-server
2. account-service
3. loan-service
4. api-gateway

**STEP 6**: Test in Browser

