# Creating Microservices for account and loan

## Account Microservice

## AccountController.java:

## package com.cognizant.account.controller;

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

## import java.util.\*;

## *@RestController*

## *@RequestMapping*("/accounts")

## public class AccountController {

## *@GetMapping*("/{number}")

## public Map<String, Object> getAccountDetails(*@PathVariable* String number) {

## Map<String, Object> response = new HashMap<>();

## response.put("number", number);

## response.put("type", "savings");

## response.put("balance", 234343);

## return response;

## }

## }

## Output:

## 

## Loan Microservice

## LoanController.java:

## package com.cognizant.loan.controller;

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

## import java.util.\*;

## @RestController

## @RequestMapping("/loans")

## public class LoanController {

## @GetMapping("/{number}")

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

## Map<String, Object> response = new HashMap<>();

## response.put("number", number);

## response.put("type", "car");

## response.put("loan", 400000);

## response.put("emi", 3258);

## response.put("tenure", 18);

## return response;

## }

## }

## Output:

## 

# Create Eureka Discovery Server and register microservices

**EurekaDiscoveryServerApplcation.java:**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

*@EnableDiscoveryClient*

*@SpringBootApplication*

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.*run*(LoanApplication.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

**AccountApplication.java:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

*@EnableDiscoveryClient*

*@SpringBootApplication*

public class AccountApplication {

public static void main(String[] args) {

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

}

}

**LoanApplication.java:**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

*@EnableDiscoveryClient*

*@SpringBootApplication*

public class LoanApplication {

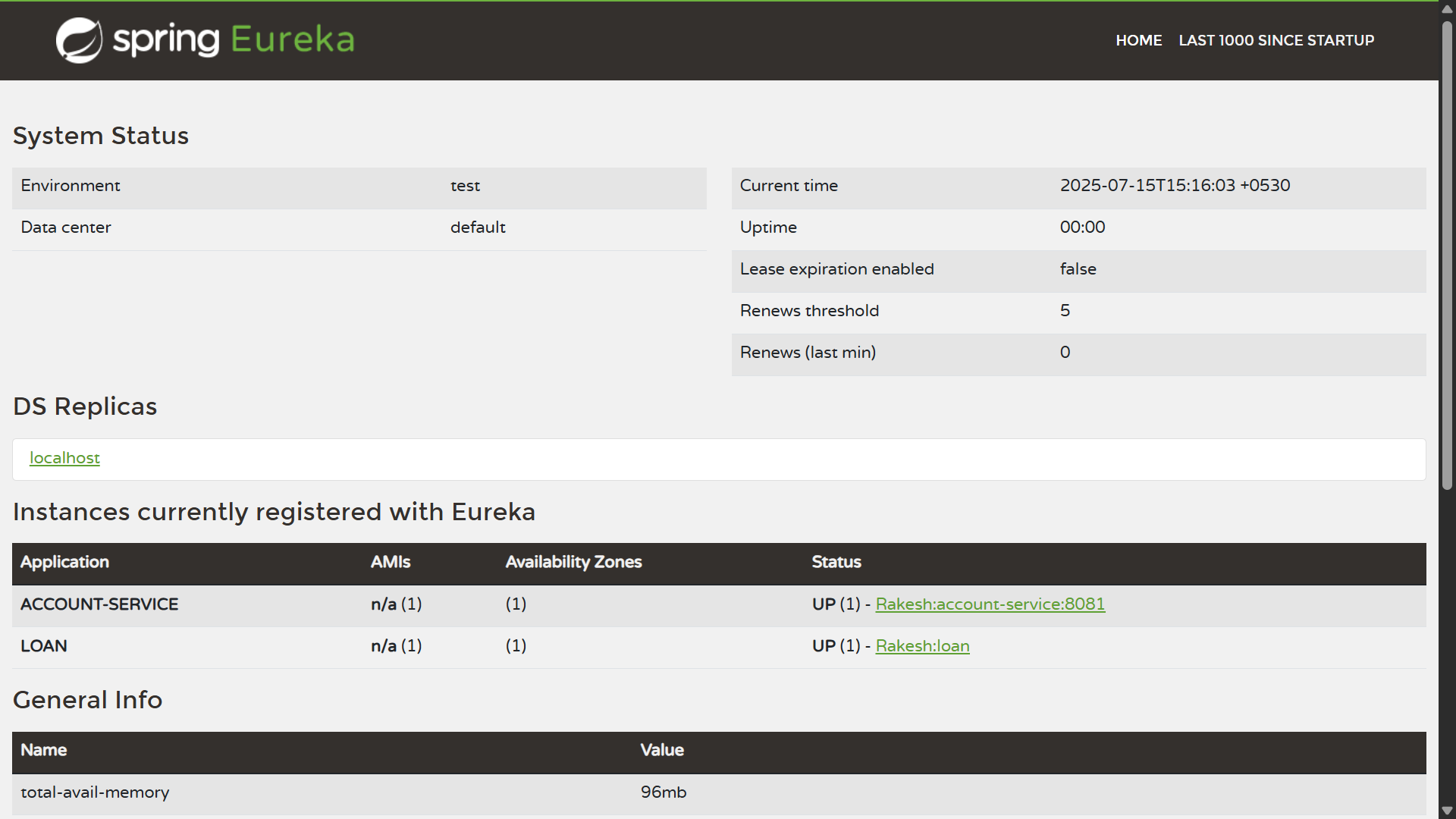
public static void main(String[] args) {

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

}

}

**Output:**



**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")

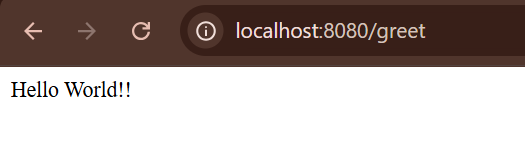
public String sayHello() {

return "Hello World!!";

}

}

**Output:**



**Application.properties:(Eureka Server)**

server.port=8761

spring.application.name=eureka-server

eureka.client.fetch-registery=false

eureka.client.register-with-eureka=false

**EurekaServerApplication.java:**

package com.cognizant.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) {

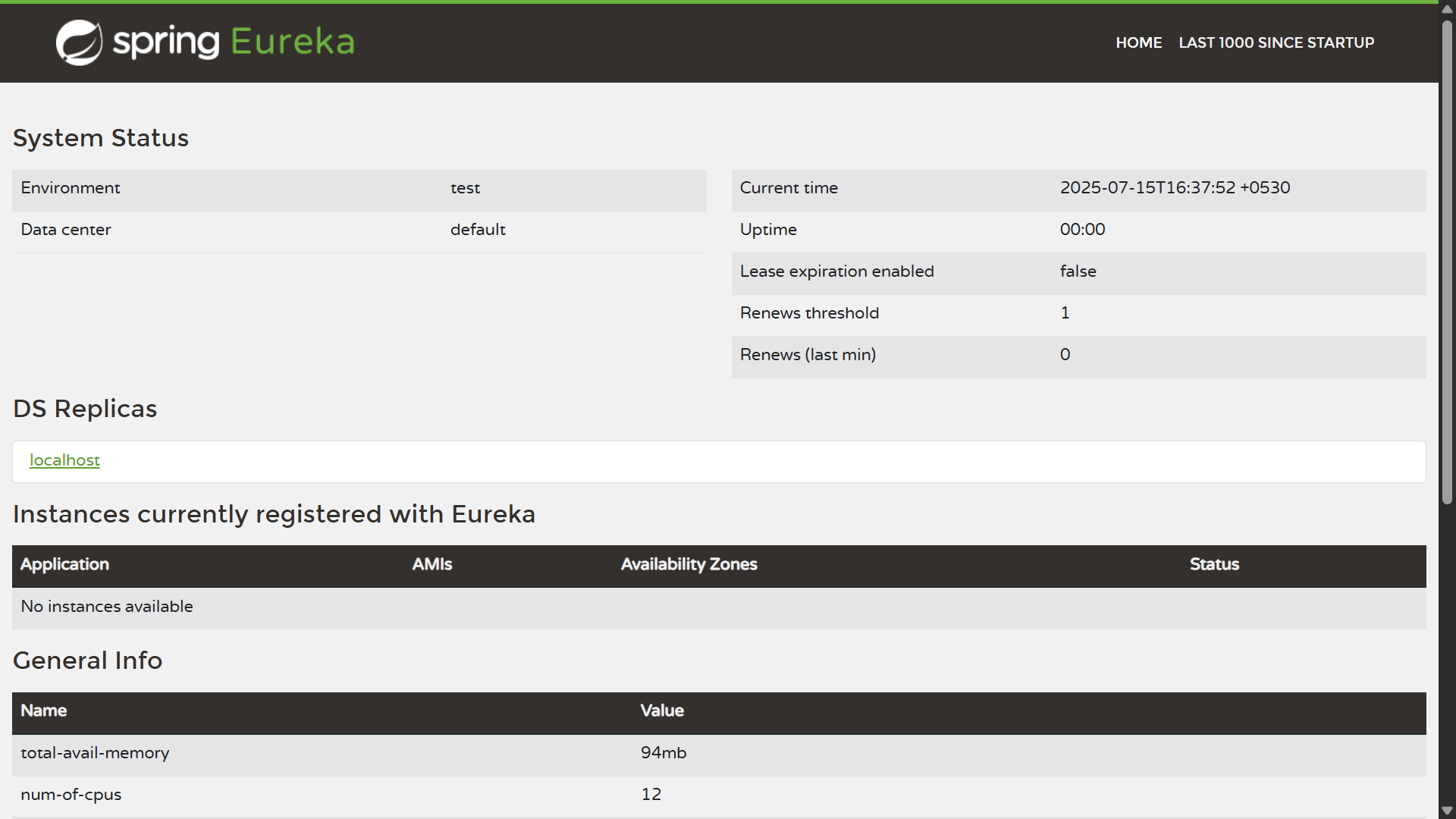
SpringApplication.run(EurekaServerApplication.class, args);

}

}

**Output:**

<http://localhost:8761>



**Pom.xml(greet-service):**

<?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>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-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>

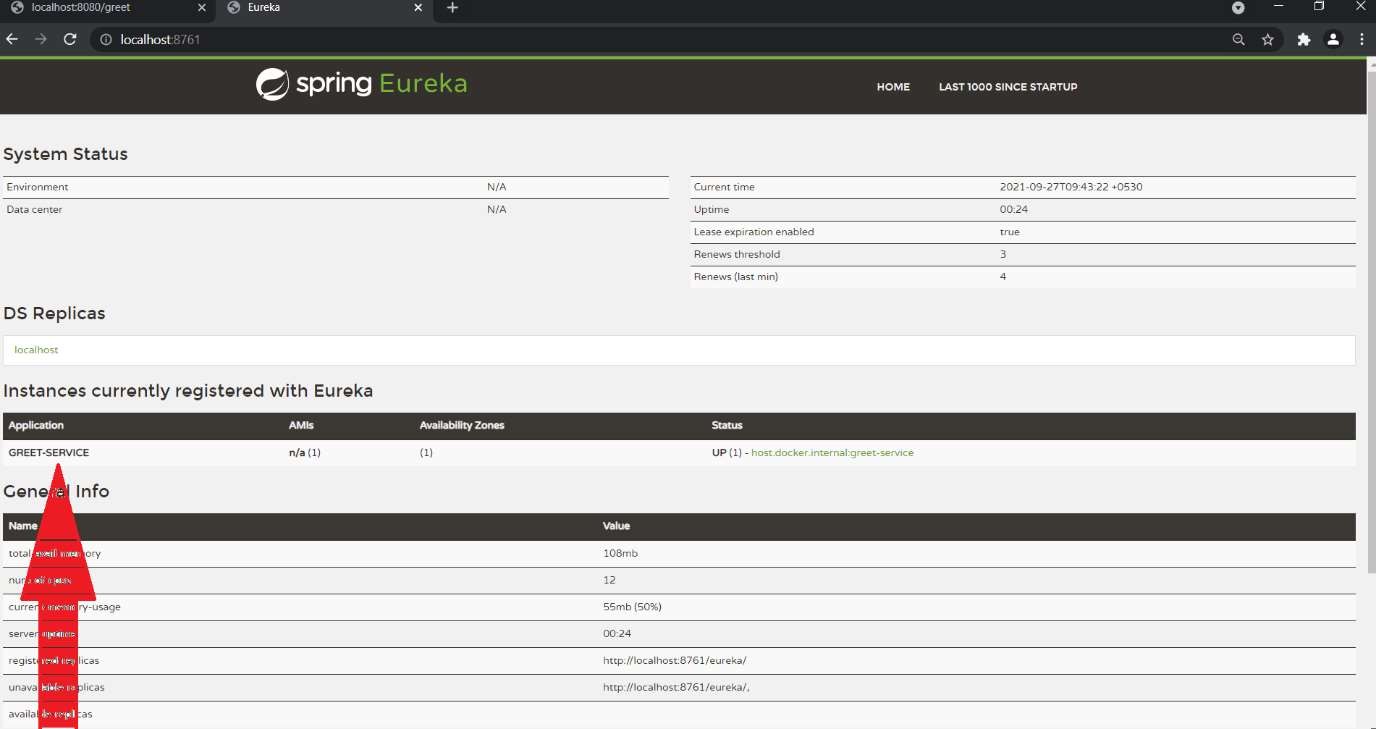
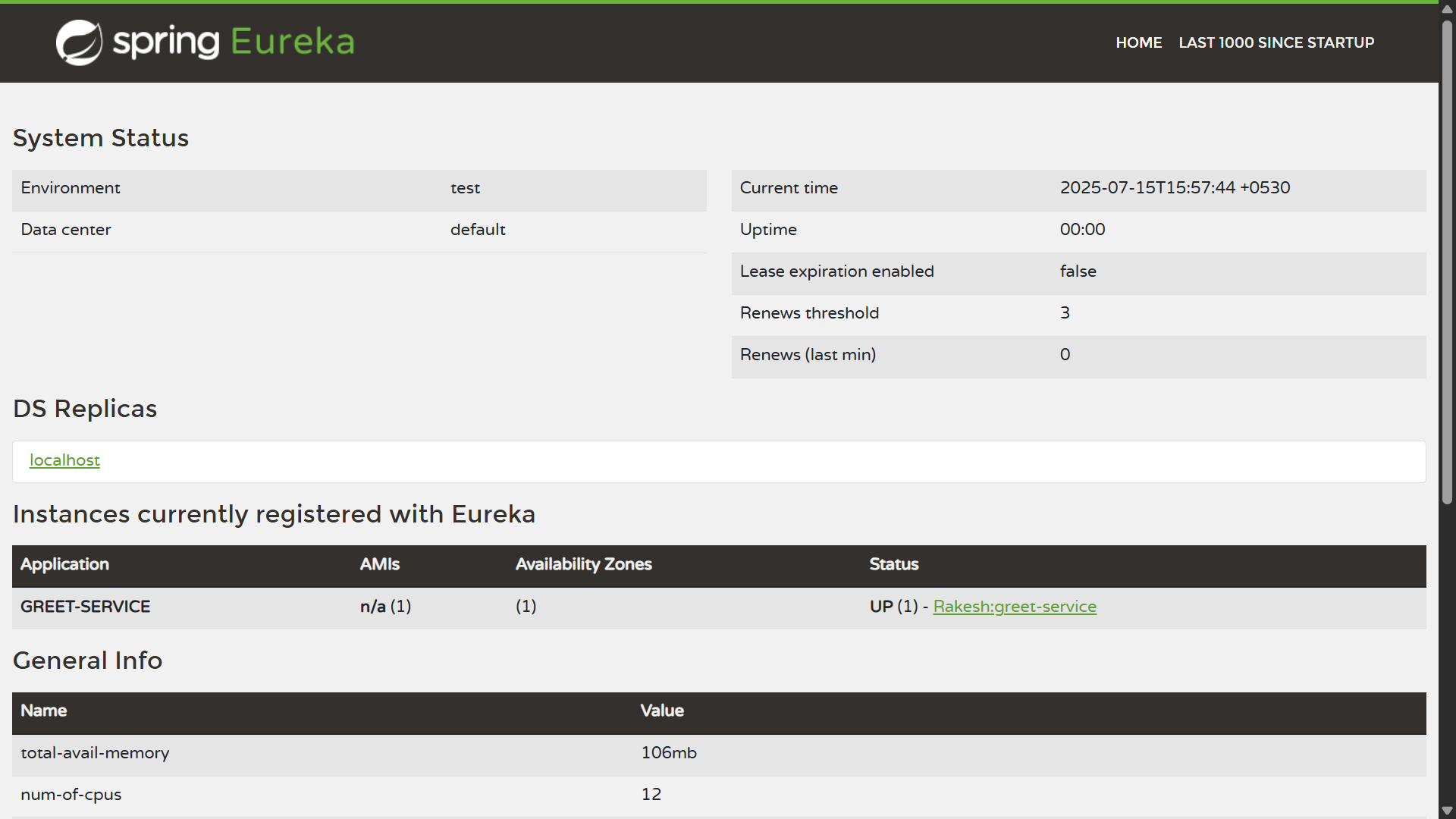
</plugins>

</build>

</project>

**Output:**

http://localhost:8761



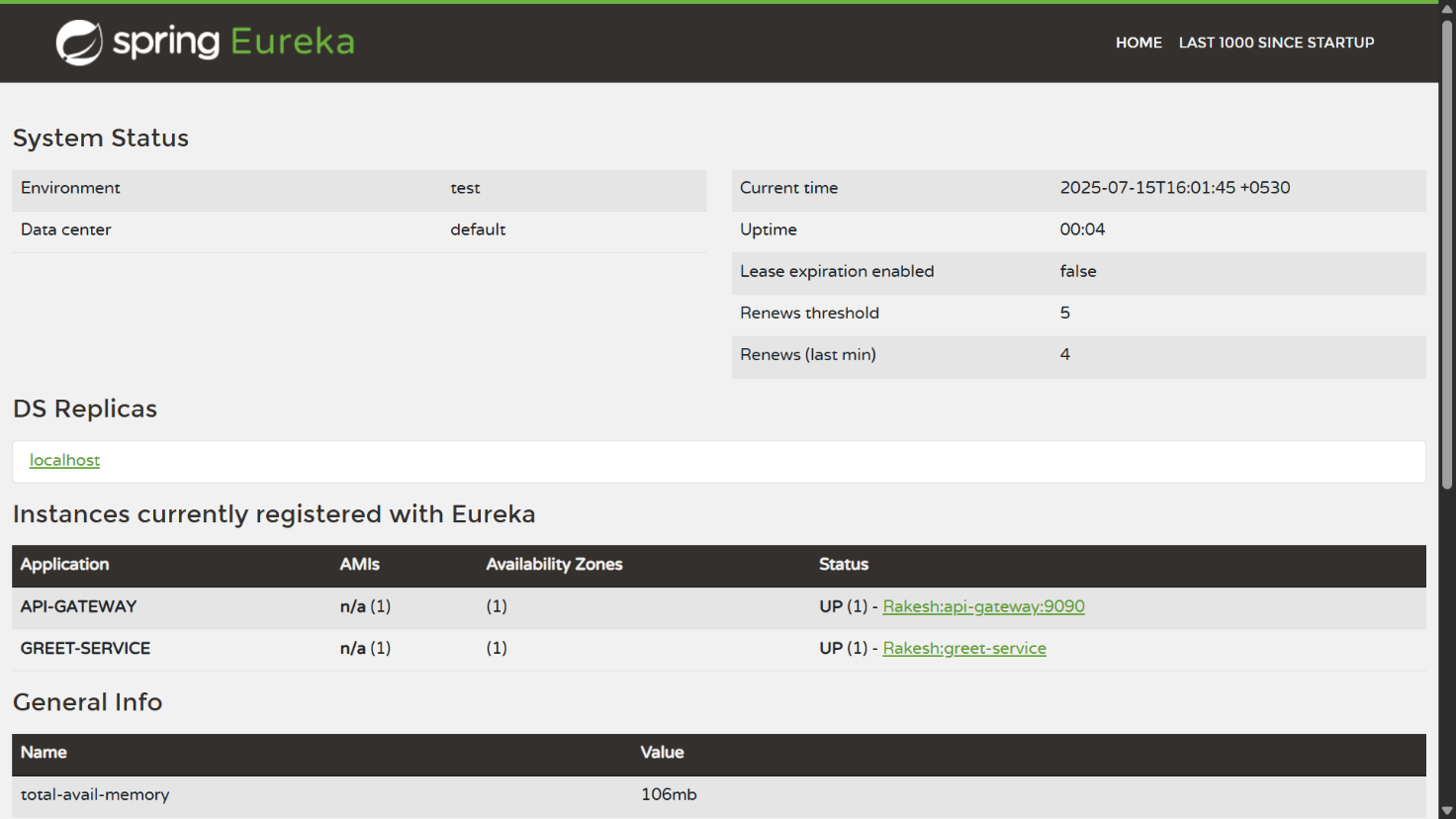
**Application.properties (api-gateway):**

spring.application.name=api-gateway

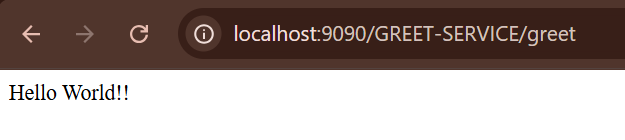
server.port=9090

spring.cloud.gateway.discovery.locator.enabled=true

**Output:**



http://localhost:9090/GREET-SERVICE/greet



**Application.properties (api-gateway):**

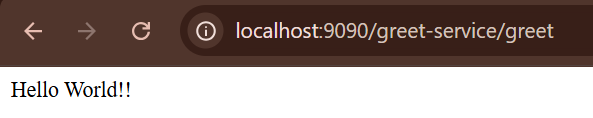
spring.application.name=api-gateway

server.port=9090

spring.cloud.gateway.discovery.locator.enabled=true

spring.cloud.gateway.discovery.locator.lower-case-service-id=true

**Output:**



**LogFilter.java (api-gateway):**

package com.cognizant.api\_gateway.filters;

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.stereotype.Component;

import org.springframework.web.server.ServerWebExchange;

import reactor.core.publisher.Mono;

*@Component*

public class LogFilter implements GlobalFilter {

Logger logger =LoggerFactory.*getLogger*(LogFilter.class);

*@Override*

public Mono<Void> filter (ServerWebExchange exchange, GatewayFilterChain chain) {

logger.info("====>Request URL {}", exchange.getRequest().getURI());

return chain.filter(exchange);

}

}

**Application.properties (api-gateway):**

spring.application.name=api-gateway

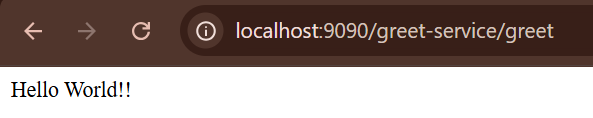
server.port=9090

spring.cloud.gateway.discovery.locator.enabled=true

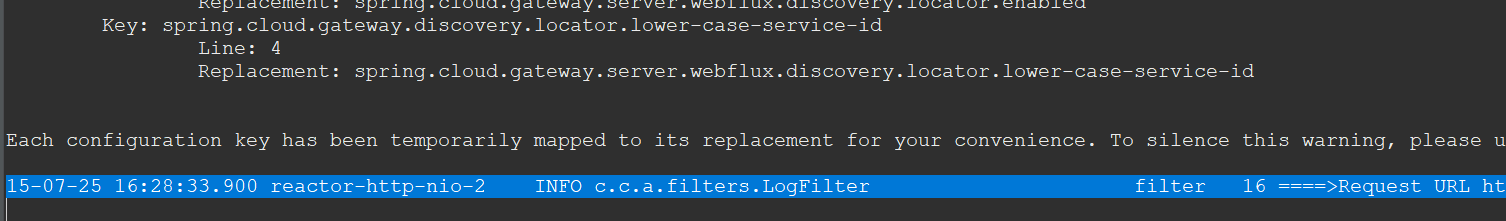
spring.cloud.gateway.discovery.locator.lower-case-service-id=true

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger**{25}** %25M %4L %m%n

**Output:**



**Logs (api-gateway):**



15-07-25 16:28:33.900 reactor-http-nio-2 INFO c.c.a.filters.LogFilter filter 16 ====>Request URL http://localhost:9090/greet