**WEEK-5**

**MICRO SERVICES**

**Exercise-1:** **Creating Microservices for account and loan**

**ACCOUNT MICROSERVICES:**

**src/main/java/com/cognizant/account/controller/AccountController.java:**

package com.cognizant.account.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

return Map.of(

"number", number,

"type", "savings",

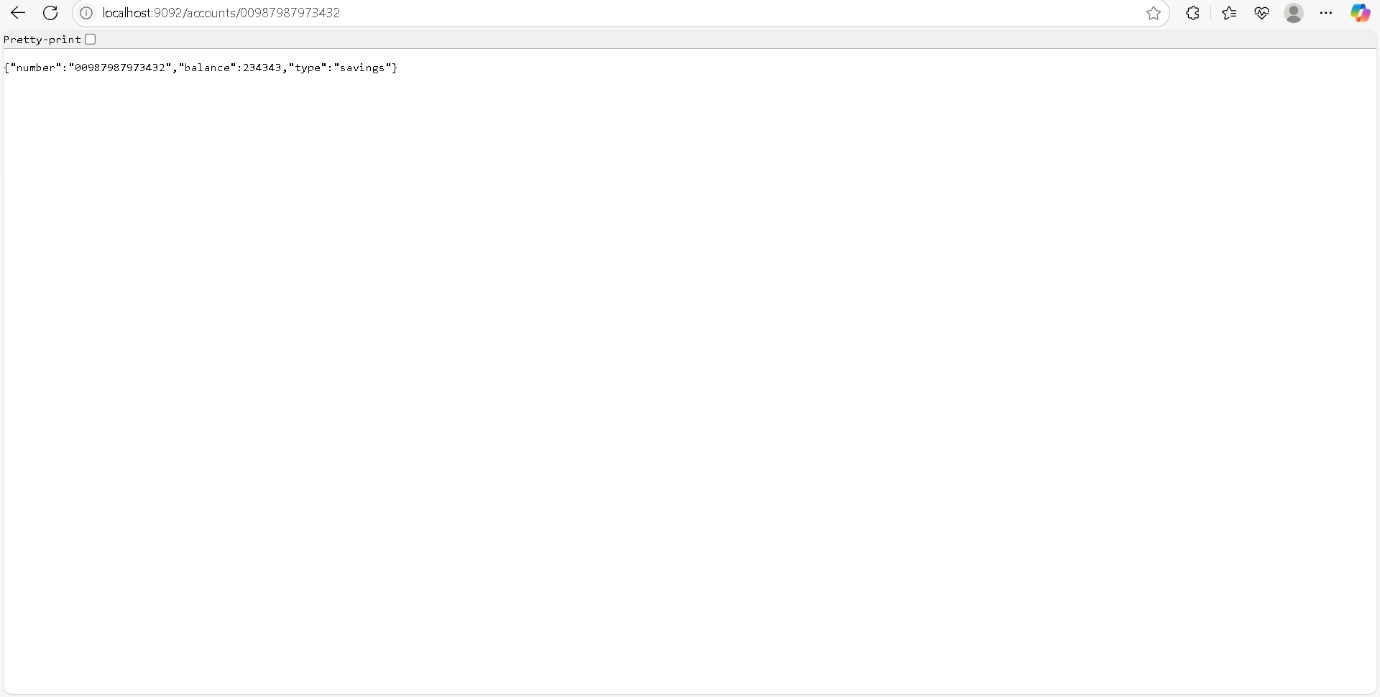
"balance", 234343

);

}

}

**OUTPUT:**



**LOAN MICROSERVICES:**

**com.cognizant.loan.controller.LoanController.java:**

package com.cognizant.loan.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

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

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

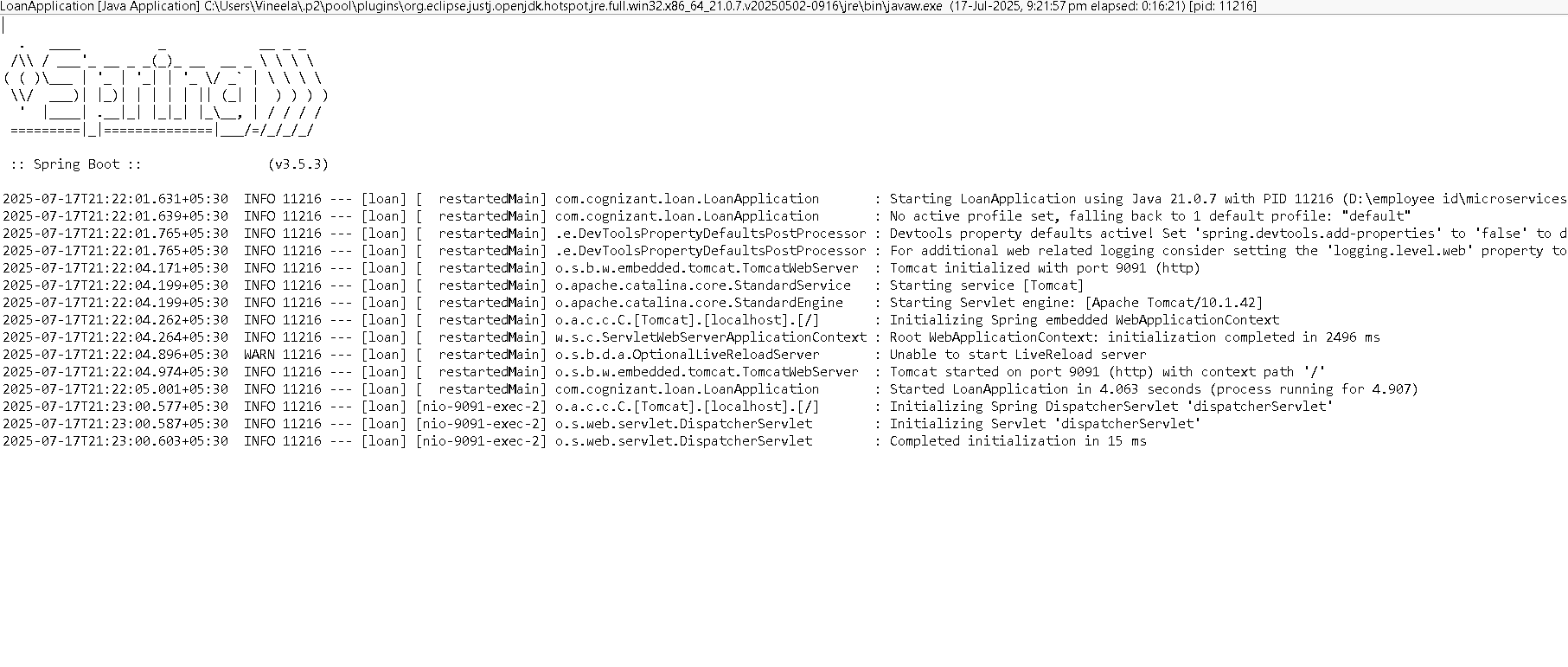
"tenure", 18

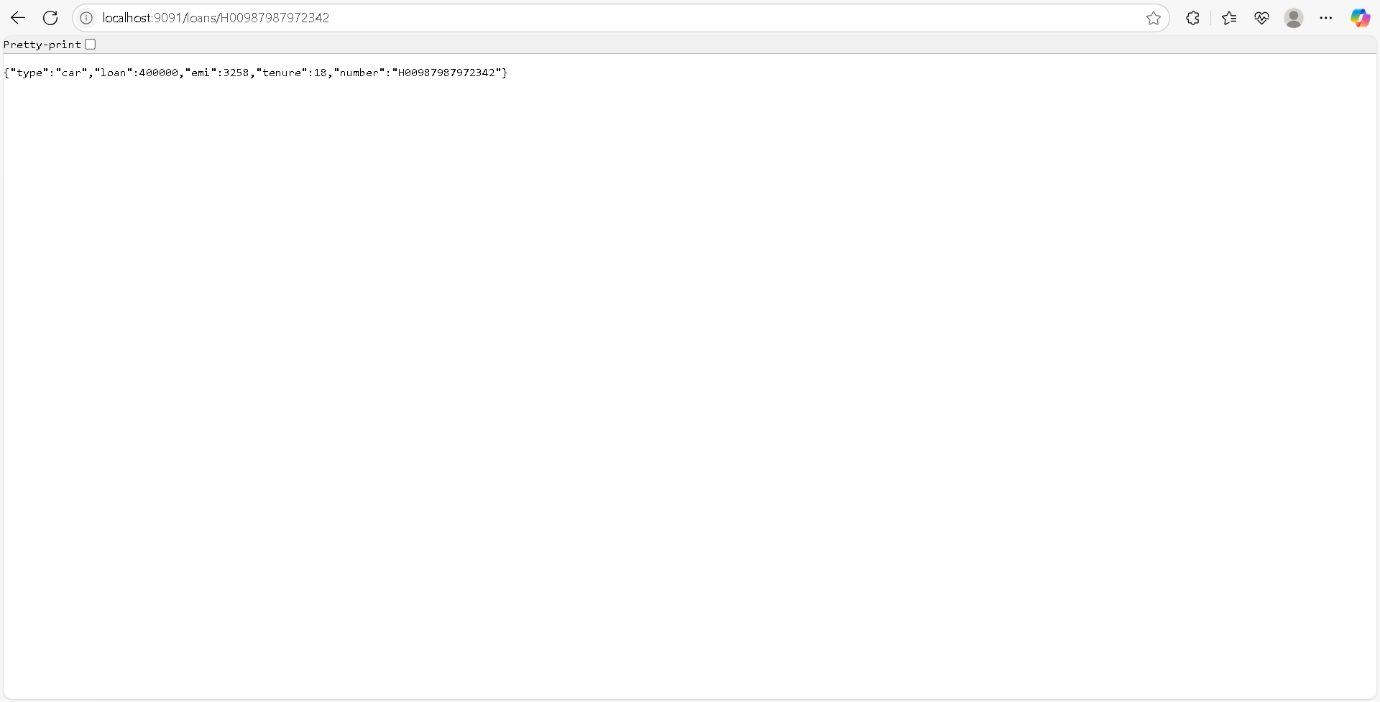
);

}

}

**OUTPUT:**





**Exercise-2: Create Eureka Discovery Server and register microservices**

**src/main/java/EurekaDiscoveryServerApplication:**

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);

}

}

**src/main/resources/application.properties:**

spring.application.name=eureka-discovery-server

server.port=8761

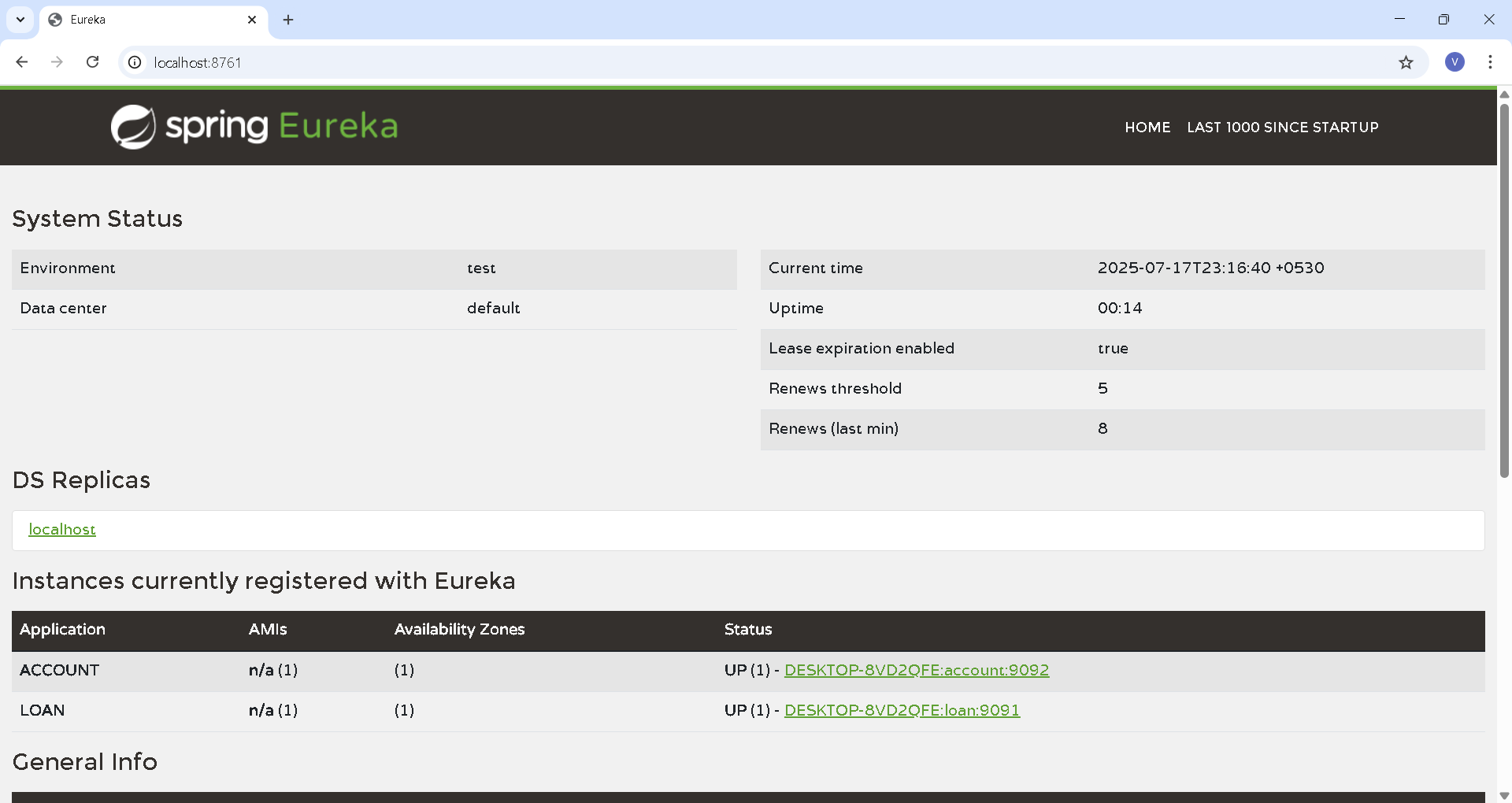
eureka.client.register-with-eureka=false

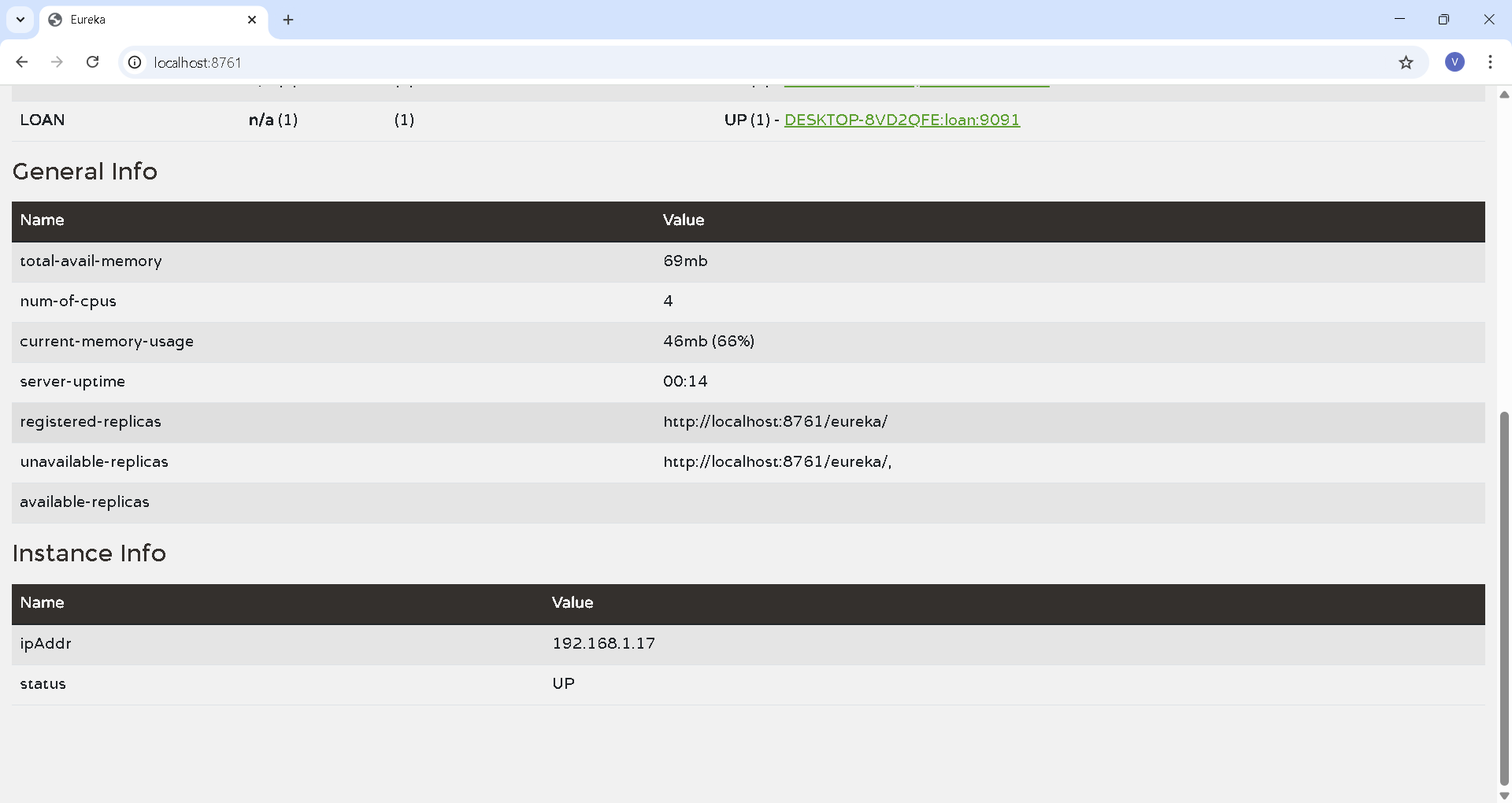
eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF

logging.level.com.netflix.discovery=OFF

**OUTPUT:**



****

****