**Sample Hands-on Exercises on Edge Services and API Gateway with Spring Boot 3 and Spring Cloud**

**Exercise 1: Implementing Edge Services for Routing and Filtering**

**Application.properties**

spring.cloud.gateway.routes[0].id=example\_route

spring.cloud.gateway.routes[0].uri=http://example.org

spring.cloud.gateway.routes[0].predicates[0]=Path=/example/\*\*

**LoggingFilter**

package com.example.gateway\_service.filter;

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;

import org.springframework.cloud.gateway.filter.GatewayFilterChain;

@Component

@Order(1)

public class LoggingFilter implements GlobalFilter {

@Override

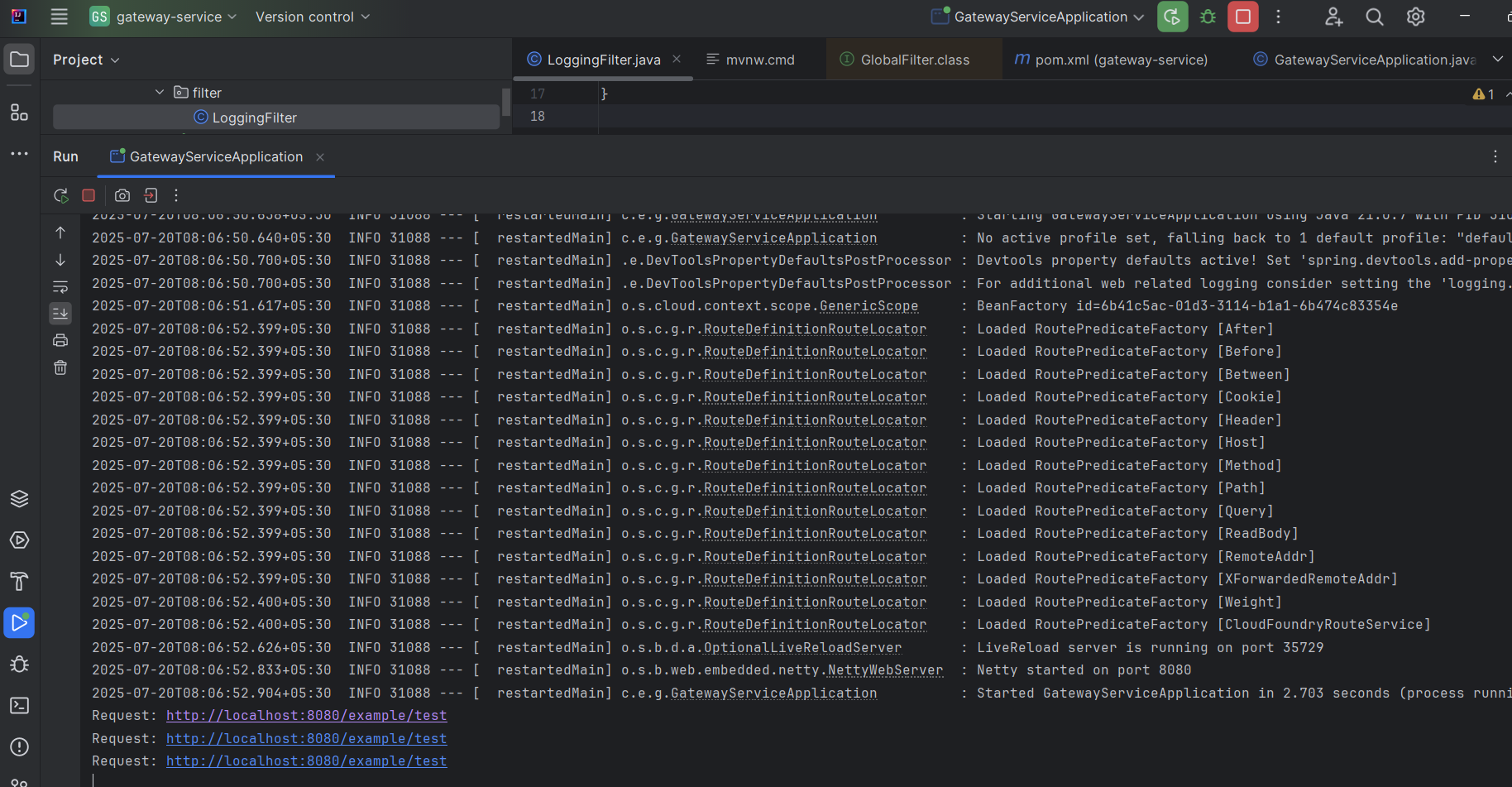
public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

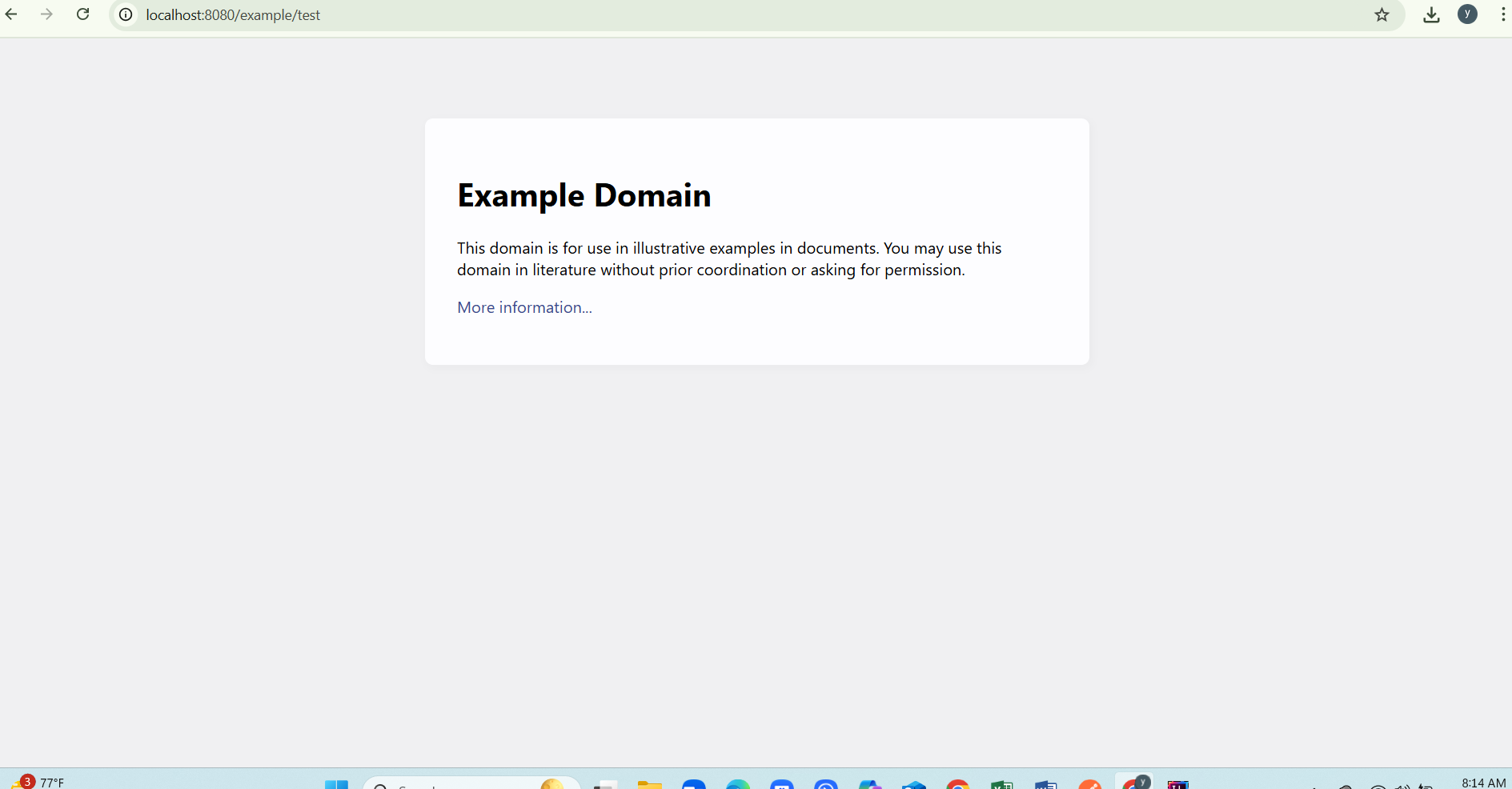
System.out.println("Request: " + exchange.getRequest().getURI());

return chain.filter(exchange);

}

}





**Exercise 2: Load Balancing in an API Gateway**

**LoadBalancer**

package com.example.gateway.config;

import org.springframework.cloud.client.ServiceInstance;

import org.springframework.cloud.client.loadbalancer.LoadBalancerClientFactory;

import org.springframework.cloud.loadbalancer.core.RandomLoadBalancer;

import org.springframework.cloud.loadbalancer.core.ReactorLoadBalancer;

import org.springframework.cloud.loadbalancer.support.ServiceInstanceListSupplier;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.core.env.Environment;

@Configuration

public class LoadBalancerConfiguration {

@Bean

public ReactorLoadBalancer<ServiceInstance> randomLoadBalancer(

Environment environment,

LoadBalancerClientFactory loadBalancerClientFactory) {

String name = environment.getProperty(LoadBalancerClientFactory.PROPERTY\_NAME);

return new RandomLoadBalancer(

loadBalancerClientFactory.getLazyProvider(name, ServiceInstanceListSupplier.class),

name);

}

}

**Controller**

package com.example.gateway.controller;

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

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class TestController {

@Value("${server.port}")

private String port;

@GetMapping("/loadbalanced/hello")

public String hello() {

return "Hello from instance on port " + port;

}

}

**App.yml**

server:

port: 8081

spring:

application:

name: gateway-service