**Payment-service**

**week5/Microservices-SpringBoot/payment-system/payment-service/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.example</groupId>

<artifactId>payment-service</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>payment-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>21</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.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<dependency>

<groupId>io.github.resilience4j</groupId>

<artifactId>resilience4j-spring-boot3</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-logging</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>

**week5/Microservices-SpringBoot/payment-system/payment-service/src/test/java/com/example/payment/PaymentServiceApplicationTests.java**

package com.example.payment;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class PaymentServiceApplicationTests {

@Test

void contextLoads() {

}

}

**week5/Microservices-SpringBoot/payment-system/payment-service/src/main/resources/application.yml**

server:

port: 8083

spring:

application:

name: payment-service

management:

endpoints:

web:

exposure:

include: "\*"

resilience4j:

circuitbreaker:

instances:

paymentCB:

failureRateThreshold: 50

slowCallRateThreshold: 50

slowCallDurationThreshold: 2s

slidingWindowSize: 5

minimumNumberOfCalls: 5

waitDurationInOpenState: 10s

permittedNumberOfCallsInHalfOpenState: 2

automaticTransitionFromOpenToHalfOpenEnabled: true

**week5/Microservices-SpringBoot/payment-system/payment-service/src/main/java/com/example/payment/PaymentClient.java**

package com.example.payment;

import io.github.resilience4j.circuitbreaker.annotation.CircuitBreaker;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Service;

import org.springframework.web.client.RestTemplate;

@Service

public class PaymentClient {

private final Logger logger = LoggerFactory.getLogger(PaymentClient.class);

private final RestTemplate restTemplate = new RestTemplate();

private static final String EXTERNAL\_URL = "https://deelay.me/5000/https://jsonplaceholder.typicode.com/todos/1";

@CircuitBreaker(name = "paymentCB", fallbackMethod = "fallbackPayment")

public String makePayment() {

logger.info("Simulating delay...");

try {

Thread.sleep(5000); // 5-second delay to trigger timeout

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

String response = restTemplate.getForObject("https://jsonplaceholder.typicode.com/todos/1", String.class);

logger.info("Received response: {}", response);

return response;

}

public String fallbackPayment(Throwable t) {

logger.warn("Fallback triggered due to: {}", t.getMessage());

return "Fallback response: Payment system is temporarily unavailable.";

}

}

**week5/Microservices-SpringBoot/payment-system/payment-service/src/main/java/com/example/payment/PaymentController.java**

package com.example.payment;

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

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

@RestController

public class PaymentController {

private final PaymentClient paymentClient;

public PaymentController(PaymentClient paymentClient) {

this.paymentClient = paymentClient;

}

@GetMapping("/api/v1/pay")

public String pay() {

return paymentClient.makePayment();

}

}

**week5/Microservices-SpringBoot/payment-system/payment-service/src/main/java/com/example/payment/PaymentServiceApplication.java**

package com.example.payment;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class PaymentServiceApplication {

public static void main(String[] args) {

SpringApplication.run(PaymentServiceApplication.class, args);

}

}