Circuit Breaker

<https://www.baeldung.com/spring-cloud-circuit-breaker>

1. Add the dependency for resilience4j

Dependency to use for circuit breaker (Spring cloud)

resilience4j

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-circuitbreaker-resilience4j</artifactId>

<version>1.0.2.RELEASE</version>

</dependency>

1. application.properties

resilience4j.timelimiter.configs.default.timeout-duration=3s

1. MyController

package com.ust.jag;

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

import org.springframework.cloud.client.circuitbreaker.CircuitBreaker;

import org.springframework.cloud.client.circuitbreaker.CircuitBreakerFactory;

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

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

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

import org.springframework.web.client.RestTemplate;

@RestController

@RequestMapping("/album")

public class MyController {

@Autowired

private CircuitBreakerFactory circuitBreakerFactory;

@GetMapping

public String getDefaultAlbumList() {

CircuitBreaker circuitBreaker = circuitBreakerFactory.create("circuitbreaker");

String url = "https://jsonplaceholder.typicode.com/albums";

RestTemplate restTemplate=new RestTemplate();

return circuitBreaker.run(() -> restTemplate.getForObject(url, String.class));

}

@GetMapping("/wrong")

public String getAlbumList()

{

CircuitBreaker circuitBreaker = circuitBreakerFactory.create("circuitbreaker");

String url = "http://localhost:1234/not-real";

RestTemplate restTemplate=new RestTemplate();

return circuitBreaker.run(() -> restTemplate.getForObject(url, String.class),

throwable -> getDefaultAlbumList());

}

}

1. Go to <http://localhost:8080/album/wrong>

Java Messaging Service

What is a message?

Suppose, if an object (a) calls a method of object (b). Then we can say, a is sending a message to b.

Usually, when we call methods, they are called synchronously.

How can we make asynchronous messaging?

Point to point communication

Publish Subscribe model

Receivers subscribe

Sender publish

Type of medium for communication can be :

Queue

Topic

JMS

Glassfish server localhost:4848

RabbitMQ

In this demo:

We will create 2 java programs:

1. Sender
2. Receiver

Sender publishes messages

Receiver receives messages

<https://www.erlang.org/patches/otp-26.0.2>

<https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.12.2/rabbitmq-server-3.12.2.exe>

JMS – knowledge base

<https://reintech.io/blog/java-message-queues-implementing-asynchronous-communication-with-jms>