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>

MSK Managed Streaming for Apache Kafka

In our system, docker is installed. So we can run docker image of kafka

Kafka services and MSK (Managed Services for Kafka)

-----------------------

what is kafka service?

Publish - Subscribe model

Publisher = Producer

Subscriber = Consumer

See practical demo:

-------------------

The producer and consumer are not online.

They are related via "Topic"

JMS i) queue ii) topic

docker

use docker images

kafka

zookeeper

how to run two docker images together?

docker-compose.yml

0) create a "docker-compose.yml" file:

-----------------------------------

version: '2'

services:

zookeeper:

image: wurstmeister/zookeeper

ports:

- "2181:2181"

kafka:

image: wurstmeister/kafka

ports:

- "9092:9092"

environment:

KAFKA\_ADVERTISED\_HOST\_NAME: 127.0.0.1

KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181

KAFKA\_AUTO\_CREATE\_TOPICS\_ENABLE: 'false'

1) docker compose up

in a location in cmd prompt where "docker-compose.yml" is present

this is going to run the docker images

2) lets create a producer

open a new cmd prompt

C:\Users\rjaga>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

98a2ea29060f wurstmeister/kafka "start-kafka.sh" About an hour ago Up About an hour 0.0.0.0:9092->9092/tcp kafka\_kafka\_1

2c9ea685d4f3 wurstmeister/zookeeper "/bin/sh -c '/usr/sb…" About an hour ago Up About an hour 22/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp kafka\_zookeeper\_1

C:\Users\rjaga>docker exec -it 98a2ea29060f bash

bash-5.1# kafka-topics.sh --create --zookeeper zookeeper:2181 --replication-factor 1 --partitions 1 --topic test

Created topic test.

bash-5.1#

bash-5.1# kafka-console-producer.sh --broker-list localhost:9092 --topic test

>this is the message

abcd

>efgh

>ijkl

>emop

>qrst

3) lets create a receiver

docker ps -a

note down the container id of kafka

C:\Users\rjaga>docker exec -it 98a2ea29060f bash

bash-5.1# kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test

this is also a message

abcd

1234

efgh

ijkl

mnop

qrst

Spring cloud API gateway

<https://spring.io/guides/gs/gateway/#initial>

steps:

1. Create a spring boot app

Dependencies: gateway

1. In the Application or configuration class:

Define a bean:

@Bean

public RouteLocator myRoutes(RouteLocatorBuilder builder) {

return builder.routes()

.route(p -> p

.path("/get")

.filters(f -> f.addRequestHeader("Hello", "World"))

.uri("http://httpbin.org:80"))

.build();

}

1. Run the application and check  
   localhost:8090/get