Date : 04-Jun-21
Spring Boot 9AM
Mr. RAGHU

Spring Cloud Circuit Breaker

=> Fault Tolerance API : Reduce error rate in application for user request.
=> To avoid Cascade Exceptions,
MS#1--->MS#2--->MS#3
If MS#2 is not working problem , then do not go to MS#3, return from
MS#2 with one Dummy Response.

=> Fallback method : In case of Actual logic got exception, then we are
executing one dummy method which returns dummy response.

Msg: Unable to Process Request! try after sometime, Server Down.

=> Circuit Breaker is implemented by Netflix Hystix.

*** Current Version of Hystrix is having bugs.

=> we can even modify default values and enable dashboard.
=> @HystrixProperty
circuitBreaker.requestVolumeThreshold (default 20)
: max no.of failed request to open a circuit
circuitBreaker.sleepWindowInMilliseconds (default 5)
: max time for next re-try to close circuit if it is open.

======Hystrix Dashboard(current view)==========

S#1 Add Addtionally Hystrix-Dashboard and Actuator

S#2 Enabe Endpoint : hystrix.stream (or) use *
in proeprties file:
management.endpoints.web.exposure.include=*

S#3 At starter class
@EnableHystrixDashboard

S#4 Run Eureka and MS# apps check URL of MS#
http://localhost:9690/actuator/hystrix.stream

S#5 Enter this URL
http://192.168.0.8:9690/payment/pay

S#6 (Dashboard View)
Taken another tab and start dashboard
http://192.168.0.8:9690/hystrix

In Dashboard Enter URL like
http://localhost:9690/actuator/hystrix.stream

```
Success : Actual Method is executed without any exception
Short-Circuited : (Open Cirduit) Request went to fallback method only
Bad Request : Request processing returned with status 400
Timeout     : Unable to process request due to resource issues
(n/w,...)
Rejected    : Unable to find fallback method, problem in execution of
                        actual method.
Failure  : (Closed Circuit) Request went to actual method, exception
is thrown
                fallback method is found and executed.
================================================================
====
Q) CB is used to handle Exceptions?
A) No. It is used to avoid casecase exceptions.
  To handle exceptions write try-catch block
  and re-throw them (or global Exception handledr)

Q) Can we call CB used for Fault-Tolerance Service?
A) It reduces failure rate of System for inter connected system
   if continioues exceptions are comming.

Q) What is Circuit status : OPEN ?
A) If the current request is directly sent to fallback method
   insted of actual method then that is called Tripped Circit
   or OPEN Circuit.

Q) When fallback method is executed?
A)
   Case#1  If OPEN Circuit
   Case#2  Closed Circuit but exception is thrown

*)Hint : please use <spring-cloud.version>Hoxton.SR4</spring-
cloud.version>
         inplace of SR11 use SR4

=======================(Eureka Application)=========================
1. Eureka (same as before)
2. PaymentMS
a)
Name: SpringCloudPaymentHystrix
Dep : Web, eureka discovery client, Hystrix, Hystrix Dashboard,
Actuator

b) At starter class: @EnableHystrix, @EnableHystrixDashboard

c) Rest Controller:
package in.nareshit.raghu.rest;

import java.util.Random;

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

import com.netflix.hystrix.contrib.javanica.annotation.HystrixCommand;
```

```
import
com.netflix.hystrix.contrib.javanica.annotation.HystrixProperty;

@RestController
@RequestMapping("/payment")
public class PaymentRestController {

        @GetMapping("/pay")
        @HystrixCommand(
                        fallbackMethod = "doPaymentFallback",
                        commandKey = "paymentFallback",
                        commandProperties = {

@HystrixProperty(name="circuitBreaker.requestVolumeThreshold",value =
"6"),

@HystrixProperty(name="circuitBreaker.sleepWindowInMilliseconds",value
= "10000")
                        })
        public String doPayment() {
                System.out.println("FROM ACTUAL METHOD");
                if(new Random().nextInt(80)<50) {
                        throw new RuntimeException("FROM DUMMY
EXCEPTION!");
                }
                return "SUCCESS!!";
        }

        public String doPaymentFallback() {
                System.out.println("FROM FALLBACK METHOD");
                return "FAILED! TRY AFTER SOME TIME!!";
        }
}

d) application.properties
server.port=9690
spring.application.name=PAYMENT-SERVICE

eureka.client.service-url.defaultZone=http://localhost:8761/eureka
management.endpoints.web.exposure.include=*

e)Execution order

*) Run Eureka and MS# apps check URL of MS#
  http://localhost:9690/actuator/hystrix.stream

*) Enter this URL(MS# URL)
http://192.168.0.8:9690/payment/pay

*) GOTO Dashboard View
Taken another tab and start dashboard
http://192.168.0.8:9690/hystrix

*) In Dashboard Enter URL like
http://localhost:9690/actuator/hystrix.stream
================================================================
*) We need only ready made service 'hystrix.stream' that gets
```

```
  Ciruit details, meta data(Succ,Fails, Error..)
  into Dashboard View and display all current progress..

Activate using:
  management.endpoints.web.exposure.include=hystrix.stream
(or for all)
  management.endpoints.web.exposure.include=*
-----------------------------------------------------------------
Spring Boot Security:
https://www.youtube.com/watch?v=XTyQIrlyWfQ
https://www.youtube.com/watch?v=7wA46kRh2u8
https://www.youtube.com/watch?v=oLlSs-p6OEs
https://www.youtube.com/watch?v=cnrJ-Nnvoik
https://www.youtube.com/watch?v=rgG2_T-OB8g

https://www.youtube.com/watch?v=feETfZbvu-k
https://www.youtube.com/watch?v=Hzkw846jIOU
https://www.youtube.com/watch?v=bJAsHOH4lMk


Task:-
#1. Spring Boot + ELK
https://www.youtube.com/watch?v=uSYExRWbC9Y
#2.
Spring Boot + Angular
https://www.youtube.com/c/NareshIT/search?
query=Spring%20boot%20angular
#3.
Spring Boot + ReactJS
https://www.youtube.com/c/NareshIT/search?query=Spring%20boot%20react

Cloud Deployment: (PCF)
https://www.youtube.com/watch?v=QOwgiJWmZ9k
=================================================================
=
Redis As DB
https://www.youtube.com/watch?v=HBmlNMGh9O0
Redis As Cache
https://www.youtube.com/watch?v=IwYEdZOmY6g
```