

微服务架构之Spring Cloud Hystrix

一、简介

1.1 Hystrix是什么

Spring Cloud Hystrix（豪猪）是熔断器，主要用于服务的降级和熔断处理

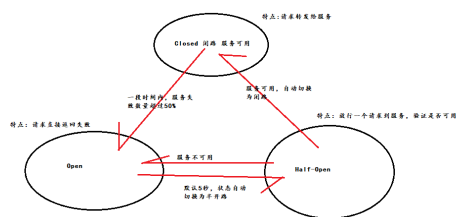
防止单个服务故障，从而耗尽服务器的资源，线程资源，最终导致微服务中服务的级联崩溃

一旦出现故障，就会通过第三方请求方式，实现服务的回退操作。

1.2 Hystrix特点

1、请求熔断

在一定时间内，请求失败数量超过50%，熔断器自动切换成开路状态（Open），这时所有请求都会直接返回失败，不再请求服务器。保持开路状态一段时间后，默认(5秒)，将开路状态改为半开路（Half-Open），放行下一个请求到服务器，如果成功，服务可用，熔断器自动切换为闭路状态(Closed),后续的请求都到服务器。如果失败，重新切换为开路状态(Open),后续持续进行



2、服务降级

Fallback降级处理

如果服务异常（超时、出现bug等）、定义一个fallback方法，当服务异常 的时候，可以去执行fallback方法（返回的结果可以来自缓存）

3、隔离

使用线程池实现资源隔离

参考Docker 容器之间都是隔离

4、请求合并

自动进行操作合并，减小网络间的请求次数

5、缓存

会默认对请求做缓存处理

<http://localhost:8080/a/getByid/1>

二、初体验

2.1 搭建熔断器的监控项目

1、依赖jar

、、

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-hystrix</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-hystrix-dashboard</artifactId>
</dependency>
```

2、开关类上配置

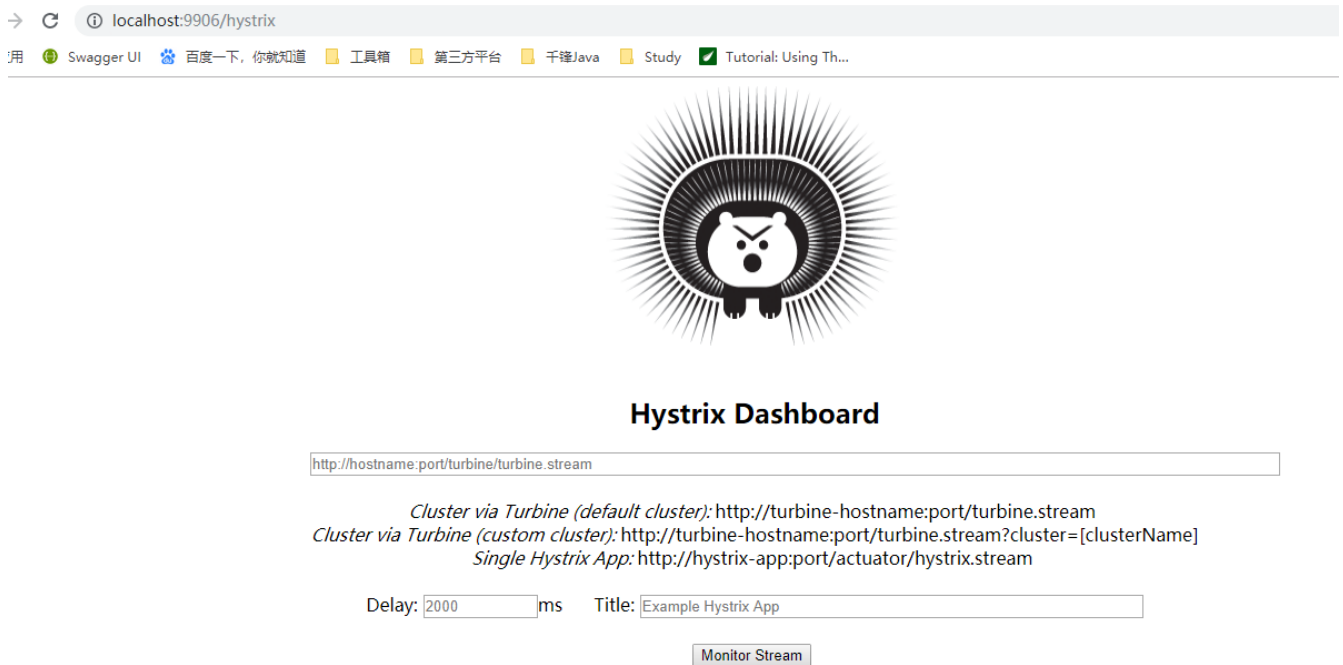
使用注解@EnableHystrixDashboard

3、更改端口号

server.port=9906

4、运行并测试

浏览器输入地址：<http://localhost:9906/hystrix>



2.2 服务提供者中使用熔断器

1、依赖jar

``

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-hystrix</artifactId>
</dependency>
```

2、配置开关类

使用注解，启用Hystrix

@EnableCircuitBreaker 启用熔断器监控

或者是使用：@EnableHystrix

都是启用Hystrix的意思

3、在提供者的控制器中

定义方法实现降级处理

并且通过注解@HystrixCommand实现服务的方法和降级方法的绑定

4、配置Servlet

```
HystrixMetricsStreamServlet
```

Hystrix的数据采集 通过这个Servlet获取服务的状态

5、运行并测试

2.3 服务链路监控

借助Hystrix仪表盘 进行熔断服务的监控


打开仪表盘项目

浏览器输入地址: <http://localhost:9906/hystrix>

在指定位置输入需要监控的服务 数据采集

[localhost:9906/hystrix](#)

Swagger UI 百度一下, 你就知道 工具箱 第三方平台 千锋Java Study Tutorial: Using Th...



服务提供者: 开启了熔断并且设置熔断监控数据的Servlet

Hystrix Dashboard

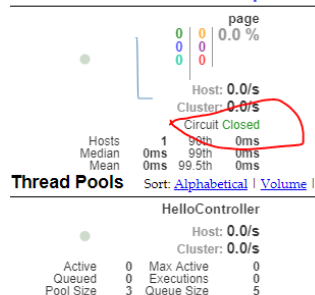
<http://localhost:9902/hystrix.stream>

Cluster via Turbine (default cluster): <https://turbine-hostname:port/turbine.stream>
Cluster via Turbine (custom cluster): [https://turbine-hostname:port/turbine.stream?cluster=\[clusterName\]](https://turbine-hostname:port/turbine.stream?cluster=[clusterName])
Single Hystrix App: <https://hystrix-app:port/actuator/hystrix.stream>

Delay: ms Title: **监控名称**

刷新时间

开启监控



看到服务的健康状态

三、核心技术

1、服务雪崩

多个服务之间相互调用

A服务调用B服务，B服务调用C服务，C服务调用服务 扇出，依赖

如果某个服务崩溃，导致扇出的服务链响应超时，最终会导致A服务占用资源越来越多，最后服务器崩溃，服务雪崩

2、请求熔断

熔断器是针对是服务提供者

3、核心注解

@EnableCircuitBreaker :启用熔断器

@EnableHystrix :启用熔断器

@HystrixCommand :设置服务降级的方法

@EnableHystrixDashboard :启用Hystrix的仪表盘监控 进行服务链路跟踪