

[首页](#)[博客](#)[学院](#)[下载](#)[GitChat](#)[TinyMind](#)[论坛](#)[问答](#)[商城](#)

...

[写博客](#)[发Chat](#)

fox9916的博客

[RSS订阅](#)

冰与火

spring cloud : 三、断路器 (hystrix)

2018年03月07日 18:05:48

阅读数 : 149

在分布式系统中，根据业务来拆分成一个个的微服务，服务与服务之间可以相互调用（RPC），在spring cloud 中可以用RestTemplate+ribbon和feign来调用。为了保证其高可用，单个服务有时候会集群部署，由于网络或程序自戕 的原因，服务并不能保证百分百可靠可用，如果单个服务出现问题，调用这个服务就出现线程阻塞，此时若有大量的请求涌入，servlet容器的线程资源就会被消耗完毕导致服务瘫痪。服务与服务之间的依赖性，故障会传播，会对整个微服务系统造成不可估量的严重后果，这就是常说的服务故障的“雪崩效应”。为了解决这个问题，有人就提出了一种解决问题的思路，断路器模型。就是每一个调用服务的接口处加一个断路器，默认是关闭的，当对服务调用时，不可用的次数达到一个阈值时，断路器就会打开，通过回调方法迅速返回一个值结束调用，避免出现连锁故障。

一、改造上一篇文章的zhangsanservice（ribbon+restTemplate方式），在pom.xml中加入hystrix的依赖

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

改造启动类ZhangsanServiceApplication.java:

```
package com.gaox.zhangsanService;

import com.netflix.hystrix.contrib.javanica.annotation.HystrixCommand;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.netflix.hystrix.EnableHystrix;
import org.springframework.context.annotation.Bean;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
@EnableEurekaClient
@EnableHystrix
@RestController
public class ZhangsanServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(ZhangsanServiceApplication.class, args);
    }

    @Bean
    @LoadBalanced
    public RestTemplate restTemplate(){
        return new RestTemplate();
    }

    @Autowired
    private RestTemplate restTemplate;

    @RequestMapping("/hello")
    @HystrixCommand(fallbackMethod = "helloError")
```

```

public String hello(String name){
    String result=restTemplate.getForObject("http://HELLOSERVICE/hello?name="+name,String.class);
    return result;
}
public String helloError(String name){
    return "hello ,"+name+"!  sorry ,error !";
}
}
}

```

依次启动 , serviceCenter,helloService:8762,helloService:8763,zhangsanService,在浏览器中访问http://localhost:8764/hello?name=zhangsan

浏览器依次会显示

你好 , zhangsan。我是helloService,端口是8762

你好 , zhangsan。我是helloService,端口是8763

此时关闭helloService:8763, http://localhost:8764/hello?name=zhangsan

浏览器依次会显示

你好 , zhangsan。我是helloService,端口是8762

hello ,zhangsan! sorry ,error !

这说明当zhangsanService调用的服务helloService:8763不可用时, 会执行快速失败, 直接返回一个字符串, 而不是等待响应超时, 这就很好的控制了当大量请求涌入的时候线程不会阻塞

二、feign是自带断路器的, spring cloud的有些版本中, 默认是关闭的, 需要在配置文件中打开它, 下面上一篇文章中的lisiService为基础来改造一下

Application.properties:

```

server.port=8765
spring.application.name=lisiService
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
feign.hystrix.enabled=true

```

LisiServiceApplication.java:

```

package com.gaox.lisiService;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.netflix.feign.EnableFeignClients;
import org.springframework.cloud.netflix.feign.FeignClient;
import org.springframework.stereotype.Component;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@EnableEurekaClient
@EnableFeignClients
@RestController
public class LisiServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(LisiServiceApplication.class, args);
    }
}

```



```

@FeignClient(value = "helloService" ,fallback = HelloError.class)
public interface HelloService {
    @RequestMapping(value = "/hello", method = RequestMethod.GET)
    String hello(@RequestParam("name") String name);
}
@Component
public class HelloError implements HelloService {
    @Override
    public String hello(String name){
        return "hello ," + name + "! sorry ,error !";
    }
}
@Autowired
private HelloService helloService;

@RequestMapping("/hello")
public String hello(@RequestParam("name")String name){
    return helloService.hello(name);
}
}

```

依次启动serviceCenter,helloService:8762,helloService:8763,lisiService ,然后在浏览器输入http://localhost:8765/hello?name=lisi ,浏览器会交替显示

你好 , lisi。我是helloService,端口是8762

你好 , lisi。我是helloService,端口是8763

.....

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

你好 , lisi。我是helloService,端口是8762

hello , lisi! sorry ,error !

这说明断路器起作用了，当lisiService调用的服务helloService:8763不可用时，会执行快速失败，直接返回一个字符串，而不是等待响应超时，这就很好的控制了当大量请求涌入的时候线程不会阻塞

三、在feign中使用hystrix dashboard(仪表盘)功能

Pom.xml添加以下内容：

```

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-hystrix</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-hystrix-dashboard</artifactId>
</dependency>

```

LisiServiceApplication.java启动类上加两个注解@EnableHystrixDashboard和@EnableCircuitBreaker

```
package com.gaox.lisiService;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.circuitbreaker.EnableCircuitBreaker;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.netflix.feign.EnableFeignClients;

```



```
import org.springframework.cloud.netflix.feign.FeignClient;
import org.springframework.cloud.netflix.hystrix.dashboard.EnableHystrixDashboard;
import org.springframework.stereotype.Component;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@EnableEurekaClient
@EnableFeignClients
@RestController
@EnableHystrixDashboard
@EnableCircuitBreaker
public class LisiServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(LisiServiceApplication.class, args);
    }

    @FeignClient(value = "helloService", fallback = HelloError.class)
    public interface HelloService{
        @RequestMapping(value = "/hello", method = RequestMethod.GET)
        String hello(@RequestParam("name") Stringname);
    }
}
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

```
@Override
public String hello(Stringname){
    return "hello ," +name+"! sorry ,error !";
}

@Autowired
private HelloService helloService;

@RequestMapping("/hello")
public String hello(@RequestParam("name")Stringname){
    return helloService.hello(name);
}
}
```

打开浏览器：访问<http://localhost:8765/hystrix>,界面如下：

第一个输入框：<http://localhost:8765/hystrix.stream>

Delay：默认不变即可

Title：（这个可以随便输入）




0


收藏


评论


微信


微博


QQ

然后点击按钮进入，如图



四、在ribbon中使用hystrix dashboard(仪表盘)功能

Pom.xml添加以下内容：

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

LisiServiceApplication.java启动类上加个注解@EnableHystrixDashboard

```
package com.gaox.zhangsanService;

import com.netflix.hystrix.contrib.javanica.annotation.HystrixCommand;
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.netflix.hystrix.EnableHystrix;
import org.springframework.cloud.netflix.hystrix.dashboard.EnableHystrixDashboard;
import org.springframework.context.annotation.Bean;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

@EnableEurekaClient
@EnableHystrix
@EnableHystrixDashboard
@RestController
public class ZhangsanServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(ZhangsanServiceApplication.class, args);
    }

    @Bean
    @LoadBalanced
    public RestTemplate restTemplate(){
        return new RestTemplate();
    }

    @Autowired
    private RestTemplate restTemplate;

    @RequestMapping("/hello")
    @HystrixCommand(fallbackMethod = "helloError")
    public String hello(String name){
        String result=restTemplate.getForObject("http://HELLOSERVICE/hello?name="+name,String.class);
        return result;
    }

    public String helloError(String name){
        return "hello ,"+name+"! sorry ,error !";
    }
}
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册 X

👍
0

🔖
收藏

💬
评论

👤
微信

🐦
微博

👤
QQ

然后打开浏览器之后的操作和上面feign方法的一样。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/fox9916/article/details/79474868>

个人分类： [spring cloud](#)

所属专栏： [spring cloud](#)

[查看更多>>](#)

想对作者说点什么？ [我来说一句](#)

spring cloud: Hystrix断路器(熔断器)

1.Hystrix客户端Netflix已经创建了一个名为Hystrix的库,实现了断路器的模式。在microservice架构通常有多个层的服务调用。 低水平的服务的服务失败会导致级联故障一直给到用...

 zhuchuangang 2016-05-01 10:06:31 阅读数：27758

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！ [登录](#) [注册](#) [×](#)

电子断路器：Hystrix ClientsNetflix创建了一个库叫Hystrix实现了电子断路器模块。在一个微服务架构中它一般有多个服务调用层。 一个底层的服务错误会引起级联错误一直反馈到用户。...

 chenqipc 2016-11-24 17:23:34 阅读数：1835

新出的微赚钱方式被曝光，网友：白玩这么多年！

汇奇林科技·顶新

5. 断路器(Hystrix)

1.Hystrix介绍 (1)在微服务中,单个服务为了高可用性防止单点故障通常会集群部署。由于网路或其他原因服务变得不可用,服务调用者会出现长等待的线程阻塞,此时会有大量的其他请求涌入,servlet...

 qq_34448345 2017-12-05 18:02:00 阅读数：96

断路器hystrix原理及使用

原理在微服务架构中，服务之间形成调用链路，链路中的任何一个服务提供者都可能面临着相应超时、宕机等不可用的情况，在高并发的情况下，这种情况会随着并发量的上升恶化，形成“雪崩效应”，而断路器hystrix...

 jrn1012 2017-09-04 15:08:56 阅读数：652


Spring Cloud构建微服务架构（四）断路器（Hystrix）

在微服务架构中，根据业务来拆分成一个个的服务，服务与服务之间可以相互调用（RPC），在Spring Cloud可以用RestTemplate+Ribbon和Feign来调用。为了保证其高可用，单个服务...

 cheidou123 2017-12-17 17:31:10 阅读数：122

spring cloud之断路器hystrix（五）

1.Hystrix介绍 断路器：Hystrix客户端Netflix的创造了一个调用的库Hystrix实现了断路器图案。在微服务架构中，通常有多层服务调用。图1.微服务图较低级别的服务中的服务故障可能导致...

 u014401141 2017-11-28 16:22:49 阅读数：233

 0

 收藏

 评论

 微信

 微博

 QQ

男人如何补肾，分享一个有效的补肾妙方

乾云 · 顶新

spring cloud学习(五)断路器 Hystrix

断路器 Hystrix 在分布式环境中，其中的应用程序执行访问远程资源和服务的操作，有可能对这些操作的失败是由于瞬时故障，如慢的网络连接，超时，或者被过度使用的资源或暂时不可用。这些故障一般之后的短时...

u011244202 2017-02-19 21:17:00 阅读数：640

springcloud实战之7 断路器（Hystrix）

在微服务广泛使用的架构中，成群的服务通过服务注册与订阅来建立关联，但每个微服务都是独立的部分（跨进程，跨机器，跨机房），服务之间通过远程调用的方式相互访问，这种架构就增加了因为网络原因造成阻塞的风险，...

u012806787 2017-12-14 11:52:53 阅读数：173

Spring Cloud中的断路器Hystrix

什么是微服务?举个简单的例子，我想做一个用户管理项目，里边就三个功能：用户注册、用户登录、用户详情浏览。按照传统的软件开发方式直接创建一个Web项目，分分钟就把这三个功能开发出来了，但是我现在想使用微...

u012702547 2017-09-16 17:04:38 阅读数：976

断路器（Hystrix）

在微服务架构中，根据业务来拆分成一个个的服务，服务与服务之间可以相互调用（RPC），在Spring Cloud可以用RestTemplate+Ribbon和Feign来调用。为了保证其高可用，单个服务...

belalds 2018-05-29 17:14:37 阅读数：13

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册 X

服务就会出现线程阻...

java_ying 2018-04-10 14:26:53 阅读数：46

SpringCloud Hystrix 断路器

雪崩效应在微服务架构中通常会有多个服务层调用，大量的微服务通过网络进行通信，从而支撑起整个系统。各个微服务之间也难免存在大量的依赖关系。然而任何服务都不是100%可用的，网络往往也是脆弱的，所以难免有...

w_x_z_ 2016-12-03 14:09:30 阅读数：4268

Spring基础：快速入门spring cloud（5）：断路器之Hystrix

Spring Cloud是Spring总多的Project中的一个,它提供了一整套的工具帮助系统架构师们在进行分布式设计的时候可以更加方便,在创建和发布微服务时极为便捷和有效。本系列文章将会使用最简...

liumiaocn 2017-01-04 08:00:56 阅读数：7776

50万码农评论：英语对于程序员有多重要！

不背单词和语法，老司机教你一个数学公式秒懂天下英语

一起来学SpringCloud之 - 断路器Hystrix（Feign）

上一篇中我们讲了 断路器Hystrix（Ribbon）本章讲解Feign+Hystrix已经Request请求传递，各种奇淫技巧.....

memmsc 2017-09-28 13:38:11 阅读数：2179

史上最简单的SpringCloud教程 | 第四篇:断路器（Hystrix）

在微服务架构中，我们将业务拆分成一个个的服务，服务与服务之间可以相互调用（RPC）。为了保证其高可用，单个服务又必须集群部署。由于网络原因或者自身的原因，服务并不能保证服务的100%可用，如果单个服务...

forezp 2017-04-09 21:14:05 阅读数：254476

Spring Cloud 断路器Hystrix实战

微服务架构中,因分布式环境引起的请求状态通常包含如下三种 成功失败超时 对于单机系统来说可以很容易的做到快速失败,这是没有问题的,对于分布式环境而言由于网络的不可靠性而导致的请求状态不确定的情...

u010820702

2017-03-18 15:44:56

阅读数：2545

【SpringCloud】（十一）：超时机制和断路器及 Hystrix简单实践

上篇文章我们配置了Eureka集群，实现了高可用。在微服务框架中，一个服务消费者可能是其他服务消费者的提供者，而当低层次的服务提供者出现问题时，会导致系统资源被耗尽。出现雪崩效应。 Hystrix...

u013045552

2017-08-14 07:49:18

阅读数：3213

Spring cloud系列八 Hystrix的用处、解决的问题、工作流程图、断路器流程

1. 概述本文对Hystrix进行介绍，本文内容如下： 什么是Hystrix Hystrix要解决的问题和产生的原因分析 Hystrix设计原则 Hystrix的处理流程图和详细流程说明 断路器工作流程...

hry2015

2017-11-14 18:50:12

阅读数：1125

免费云主机试用一年

免费云主机试用一年厂家批发

百度广告



springcloud微服务四：服务容错保护Hystrix断路器

通过前边的学习，服务注册中心、服务提供者和服务消费者都成功建立并运行起来，而且通过默认的配置restTemplate及@loadbalanced注解开启了负载均衡。在默认的情况下，负载均衡策略是线...

tuzongxun

2017-05-26 10:56:56

阅读数：4403

引起其他的正常的应用也挂了，最终导致雪崩，断路器会在每个请求的...

shunzi1046

2017-04-19 17:31:39

阅读数：2038

SpringCloud 学习记录(二)：断路器(hystrix)

在上一篇中我们学习了如何使用feign调用远程接口，但当服务提供方宕机，或网络问题造成连接不可用，或服务提供者异常时，又该如何处理？这篇我们来看如何使用断路器-hystrix。首先我们启动e...

MyNoteBlog

2017-08-10 13:57:18

阅读数：250

(笔记) SpringCloud之Hystrix断路器 属性详解

Command属性：主要用来控制 HystrixCommand 命令的行为，主要有下面5种类型的属性配置： execution配置 控制HystrixCommand.run() 的如何执行 该配...

Weixiaohuai

2018-01-09 10:51:12

阅读数：206

Spring Cloud——断路器监控Hystrix Dashboard&Turbine

在前文谈论过，服务客户端的负载均衡中，为了保证高可用的特性，我们使用了断路器Hystrix。回顾一下，Hystrix做了什么事情呢？在标注了HystrixCommand注解的方法中，如果该方法执行...

a60782885

2017-04-15 19:05:09

阅读数：3825

全局规划导航式扫地机器人

618狂欢科沃斯直降600



SpringCloud之断路器聚合监控Spring Cloud Hystrix Turbine实例

一、简介 监控端点/turbine.stream是对集群使用的。从端点的命名中，可猜测到这里我们将引入Turbine, 通过它来汇集监控信息，并将聚合后的信息提供给Hystrix Dashboard...

smartdt

2018-01-11 13:24:29

阅读数：165

史上最简单的SpringCloud教程 | 第十二篇: 断路器监控(Hystrix Dashboard)

在我的第四篇文章[断路器](#)讲述了如何使用[断路器](#)，并简单的介绍了下[Hystrix Dashboard](#)组件，这篇文章更加详细的介绍[Hystrix Dashboard](#)。一、[Hystrix Dashboard](#)简介...

 forezp

2017-04-17 23:01:06

阅读数：59391

springCloud学习02之断路器Hystrix-turbine监控-ribbon/feign对Hystrix的支持

随着Docker的推进，微服务越来越热了.在微服务架构中，我们将业务拆分成一个个的服务，服务与服务之间可以相互调用。为了保证其高可用，单个服务又必须集群部署。由于网络原因或者自身的原因，服务并不能保证...

 dream_broken

2017-08-03 17:39:14

阅读数：1467

springcloud实战之8 断路器-仪表盘-单例监控（Hystrix）

[Spring Cloud](#)不但跟[Hystrix](#)能够做到优雅的综合，还能够整合他的仪表盘组件[Hystrix Dashboard](#)，它主要用来实时监控[Hystrix](#)的各项指标信息。通过[Hystrix Dashboard](#)...

 u012806787

2017-12-19 15:51:09

阅读数：422

六、SpringCloud断路监控面板Hystrix Dashboard

一、简介 [Hystrix Dashboard](#)是[Hystrix](#)的一个组件，[Hystrix Dashboard](#)提供一个[断路器](#)的监控面板，可以使我们更好的监控服务和集群的状态，仅仅使用[Hystrix Dashboard](#)...

 qq_36027670

2018-04-09 21:21:52

阅读数：233

免费云主机试用一年

有哪些可以免费试用一年左右的云服务器

百度广告



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

数据了。这就...

 forezp

2017-04-18 21:54:21

阅读数：52261

Spring Cloud（五）断路器监控(Hystrix Dashboard)

在上两篇文章中讲了，服务提供者 Eureka + 服务消费者 Feign，服务提供者 Eureka + 服务消费者（rest + Ribbon）修改加入 [断路器](#)监控...

 u012889902

2017-12-08 01:18:14

阅读数：460

spring cloud使用hystrix实现断路保护机制

断路保护机制：即容错性，在微服务架构中，系统之间都是相互依赖的，如果一个系统出现了异常，就会出现如下情况。 [hystrix](#)的大致原理如下： [spring ...](#)

 mazhen1991

2017-11-09 16:52:55

阅读数：137

Spring Cloud微服务断路由Hystrix与服务网关学习笔记

上一篇文章介绍了如何使用[Spring Cloud](#)与[Spring Boot](#)以及Docker容器搭建一个微服务模式的示例(<https://github.com/JackmanGo/SpringCloudL...>)

 canot

2017-03-20 23:51:18

阅读数：5859

spring cloud学习笔记-断路器

[断路器](#)

 zpz448561636

2017-04-23 12:23:24

阅读数：1590

好用的语音识别api，让您的设备听得懂话

适用语音导航、室内设备控制、语音搜索、实时语音输入等场景。



Spring Cloud构建微服务架构-Hystrix断路器

[断路器](#) [断路器](#)模式源于Martin Fowler的Circuit Breaker一文。“[断路器](#)”本身是一种开关装置，用于在电路上保护线路过载，当线路中有电器发生短路时，“[断路器](#)”能够及时的切断故...

0

收藏

评论

微信

微博

QQ

文章结合，上两篇文章中代码进行

路保护机制就是为了解决这种情

qq_40354025 2017-11-27 09:17:49 阅读数：296

一起来学SpringCloud之 - 断路器Hystrix (Ribbon)

微服务架构中，根据业务划分成若干个服务，各单元应用间通过服务注册与订阅的方式互相依赖，依赖通过远程调用的方式执行，该方式难以避免因网络或自身原因而出现故障或者延迟，从而并不能保证服务的100%可用，此...

memmsc 2017-09-28 13:30:38 阅读数：510

SpringCloud学习记录——Feign客户端负载均衡及Hystrix断路器

点我点我:项目github地址。1、搭建service-feign服务调用multipl（两个节点提供），实现客户端负载均衡效果。新建maven module，命名为micro-service-fei...

u012459871 2018-04-04 22:07:11 阅读数：99

springCloud Finchley 微服务架构从入门到精通【八】断路器 Hystrix(feign)

一、前言 上节写了ribbon 整合hystrix的实例，本节准备实现feign，由于比较简单，只写关键代码，具体看github源码 二、代码实现 创建fallback包存放feign的fa...

wcblog 2018-05-09 12:28:40 阅读数：43

Spring Cloud 的 Hystrix 更多参数配置信息

1、背景 在复杂的分布式 架构 的应用程序有很多的依赖，都会不可避免地在某些时候失败。高并发的依赖失败时如果没有隔离措施，当前应用服务就有被拖垮的风险。例如:一个依赖30个SOA服务的系统,每个...

djCode 2017-09-09 16:00:15 阅读数：1236

免费云主机试用一年



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

spring cloud hystrix 配置属性大全

以下的属性都是spring cloud 1.5.9版本的。commandProperties execution.isolation.strategy 使用命令调用的隔离模式 采用线程隔离 ex...

wjg19890301 2017-12-27 12:34:14 阅读数：354

十八、Spring cloud：Hystrix原理

一、运行原理 Hystrix是国外知名的视频网站Netflix所开源的非常流行的高可用架构框架。Hystrix能够完美的解决分布式架构中打造高可用服务面临的一系列技术难题。Hystrix是豪猪...

qq_34288630 2018-01-18 16:33:33 阅读数：130

个人资料



冰与火IAF

关注

原创	粉丝	喜欢	评论
21	9	7	9

等级： 博客 2 访问：2万+

积分：414 排名：12万+

勋章：





墙体彩绘机



最新文章

- 关于springboot+shiro+thymeleaf页面级元素的权限控制的问题
- Spring +mybatisplus+shiro权限管理集成整合
- spring cloud ：五、分布式配置中心（spring cloud config）
- spring cloud ：四、路由网关（zuul）
- spring cloud ：二、服务的消费（ribbon和feign两种方式）

博主专栏



spring cloud

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

spring	3篇
Java基础	5篇
database	1篇
git	2篇
springboot	4篇
展开	

归档	
2018年5月	2篇
2018年3月	6篇
2018年2月	2篇
2017年10月	5篇
2017年7月	1篇
展开	

热门文章

- spring cloud ：五、分布式配置中心（spring cloud config）
阅读量：11397
- git本地项目代码上传至码云远程仓库总结
阅读量：3891
- spring webflow 学习小结
阅读量：3630
- 关于spring boot属性文件中中文参数输出到页面乱码
阅读量：573
- spring cloud ：一、服务的注册与发现
阅读量：268

0

收藏

评论

微信

微博

QQ

