

[博客专区](#) > [稻草鸟人的博客](#) > [博客详情](#)

## 原 荐 微服务常用技能之优雅的重试

稻草鸟人 发表于 2周前 阅读 2512 收藏 168 点赞 4 评论 19

收藏

HOT

一般从事软件开发工作的同时通常需要调用远程服务，但是即使是亚马逊、阿里巴巴、腾讯这些巨头提供的服务也可能因为网络或者服务高可用方面的一些问题，或许会遇到超时等异常情况，这时候我们就必需要考虑到异常情况下我们的应用如何能处变不惊处理这些异常，是直接抛出异常还是说可以自动进行一些处理呢？通常情况下我们首先想到的解决方案一定是重试对吧。

我看过很多代码，他们都怎么写的呢，如下

```
public String invoke(int times) {
    for(int i = 0; i < times; i++) {
        try{
            // do some thing
            // return
        }catch(Exception e) {
            return invoke(i);
        }
    }
}
```

上面的代码意思是首先传入一个次数参数，假设我们传递的是3，那么当遇到异常的时候，重新调用当前方法，次数变为2...

你有没有发现如果调用外部服务很多的时候，这样的for循环写的让人不舒服，因为都是重复的代码，而且显的很啰嗦呢，spring boot给我们带来了一些福音，我们看看spring boot框架下如何进行重试

### 1、首先引入JAR包

```
<dependency>
    <groupId>org.springframework.retry</groupId>
    <artifactId>spring-retry</artifactId>
</dependency>
```

### 2、使用注解 @EnableRetry @Retryable 下面先给出代码，然后我们再讲注解的含义

```
@SpringBootApplication
@EnableRetry
public class LeaningNotesApplication {

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

@Service
public class BookService {
```

```
private Random random = new Random();

@Retryable(include = RuntimeException.class, maxAttempts = 3)
public String readingList() {
    int randomInt= random.nextInt(10) ;
    if(randomInt < 8){ //模拟调用失败情况
        System.out.println("调用失败发生异常");
        throw new RuntimeException("call dependency service fail.");
    }else{
        System.out.println("调用成功");
        return "调用成功 ;number:"+randomInt;
    }
}

@Recover
public String reliable() {
    return "Cloud Native Java (O'Reilly)";
}

}
```

3、我们再写一个接口，通过浏览器来访问这个接口，看看上面的代码帮我们进行了几次重试，另外如果重试了几次都失败，他又执行了什么方法

```
@RestController
public class BookController {

    @Autowired
    private BookService bookService;

    @GetMapping("/book")
    public String book() {
        return bookService.readingList();
    }
}
```

上面的这个代码就比较简单了，我们可以通过<http://localhost:8080/book> 来访问这个接口，然后在控制台看结果，我都迫不及待想要去试试了，你呢？

4、我打开了浏览器输入了地址，然后我看到控制台打印了下面这样的信息

调用失败发生异常

调用成功

是的，你没看错，先输出了一个异常日志，然后又输出了一个成功日志，说明了它在异常的情况下帮我重试了一次，然后才成功的，当然你也可以多尝试几次看看不一样的结果。

5、@Retryable 几个比较重要的参数

value : 抛出某些异常的时候才进行重试

include: 和value的含义类似

exclude: 排除某些异常不进行重试，这里要注意的是，如果value、include、exclude都为空的时候则任何异常都进行重试

maxAttempts: 最大重试次数

backoff: 重试策略，默认使用注解@Backoff 而@BackOff中value的默认为1000L，multiplier的默认值为0，表示默认情况下固定暂停1秒进行重试，如果我们把multiplier调整为2，那么表面第一次进行重试时间为1秒，第二次为2秒，第三次为4秒。

是不是很牛逼，有木有GET到新技能？

注意：注解了@Recover方法的参数可以没有，如果有一一定是和注解@Retryable处理的异常是一样的或者是它的子集，否则recover方法将不能被执行，并不是网上一些人瞎逼逼说一定要和@Retryable处理的异常保持一致

请大家多多关注，让我能有勇气坚持下去



© 著作权归作者所有

分类： 工作日志    字数： 976

标签：    spring    Spring Boot    Spring Cloud    Spring Retry

打赏

点赞

收藏

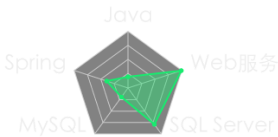
分享



稻草鸟人

后端工程师    上海

粉丝 47 | 博文 11 | 码字总数 9129



相关博客

<div>基于Spring Boot构建Spring Cloud微服务架构</div> <div>中国_码农</div> <div>398    3</div>	<div>基于Spring Boot和Spring Cloud实现微服务架构学习(二)-Spring Boot总结</div> <div>T_Tang</div> <div>525    0</div>	<div>基于Spring Boot和Spring Cloud实现微服务架构学习(三)-Spring Boot应用</div> <div>T_Tang</div> <div>293    0</div>
--	---	---

评论 (19)

Ctrl+Enter 发表评论



itwjc

1楼 2017/04/08 15:24

不错



itwjc

2楼 2017/04/08 21:08

good



ihuotui

3楼 2017/04/08 22:31

如果都是spring环境,就是优雅,但是为了少依赖.



12叔

4楼 2017/04/10 09:50

但是 不用spring的话。。



稻草鸟人

5楼 2017/04/10 11:21

引用来自“12叔”的评论  
但是 不用spring的话。。

现在确实还存在一些没有用spring的项目，大多是一些很老的项目，比如银行的一些内部系统或者是传统企业，这些公司项目要么面临改造，要么就死撑，要么咱就换个地吧



竹隐江南

6楼 2017/04/13 08:50

不错啊~~



大后锋

7楼 2017/04/13 09:17

引用来自“12叔”的评论  
但是 不用spring的话。。

引用来自“稻草鸟人”的评论

现在确实还存在一些没有用spring的项目，大多是一些很老的项目，比如银行的一些内部系统或者是传统企业，这些公司项目要么面临改造，要么就死撑，要么咱就换个地吧

也不一定是老项目啊，你不如用jfinal的，nutz的，play的==



漆黑的烈焰使

8楼 2017/04/13 09:24

这个jar不一定spring boot才能用吧？



稻草鸟人

9楼 2017/04/13 09:29

引用来自“漆黑的烈焰使”的评论  
这个jar不一定spring boot才能用吧?

当然



稻草鸟人

10楼 2017/04/13 09:32

引用来自“漆黑的烈焰使”的评论  
这个jar不一定spring boot才能用吧?

回复@漆黑的烈焰使:任何项目只要需要类似功能的都可以用的,很方便



notreami

11楼 2017/04/13 10:46

无非改成切面实现,自己实现,一样没有问题。



notreami

12楼 2017/04/13 10:46

无非改成切面实现,自己实现,一样没有问题。



HideHai

13楼 2017/04/13 10:54

又涨姿势了



HeartArea

14楼 2017/04/13 15:34

原理就是aop,但是low逼的是为什么引入retry时不能把aspectj-weaver依赖也搞进入啊,还得自己手加。



楼雨

15楼 2017/04/13 17:29

一点都不优雅,当项目越久越大,人员N次更替之后,你这简直是恶梦。  
当然,也不赞同第一种写法。



稻草鸟人

16楼 2017/04/13 17:42

引用来自“楼雨”的评论  
一点都不优雅,当项目越久越大,人员N次更替之后,你这简直是恶梦。  
当然,也不赞同第一种写法。

回复@楼雨:如何写更好?请指教



终曲

17楼 2017/04/14 00:02

生产环境重试有时候很危险,根据不同的异常可能要fail fast,那就把错误处理的逻辑全部隐式转嫁到抛什么异常上了,代码更难读



Zollty  
18楼 2017/04/16 11:55  
没看出有多优雅。。。



稻草鸟人  
19楼 2017/04/16 18:39

引用来自“终曲”的评论  
生产环境重试有时候很危险，根据不同的异常可能要fail fast，那就把错误处理的逻辑全部隐式转嫁到抛什么异常上了，代码更难读

回复@终曲：重试只是其中最常见的方式，不错，有些可能要FAILFAST有些需要FAIL OVER当然还有其他降级等策略

社区

开源项目  
技术问答  
动弹  
博客

众包

开源资讯  
技术翻译  
专题  
招聘

码云

项目大厅  
软件与服务  
接活赚钱  
在线工具

活动

Git代码托管  
Team  
PaaS  
在线工具

关注微信公众号

下载手机客户端

©开源中国(OSChina.NET)

关于我们 广告联系 @新浪微博 合作单位

开源中国社区是工信部 开源软件推进联盟 指定的官方社区 粤ICP备12009483号-3