
1 目录

Day02. Spring Cloud 微服务.....	1
1 回顾.....	1
2 ribbon 重试.....	2
3 hystrix.....	3
4 feign.....	3
4.1 声明式客户端.....	3
4.2 feign 整合 ribbon.....	4
4.3 feign 整合 hystrix.....	4
5 作业.....	4

Day02. Spring Cloud 微服务

2 回顾

- eureka
 - 服务注册,服务治理
 - 维护一个主机列表
 - 微服务向注册中心注册,或从注册中心拉取注册表
 - 每 30 秒发送一次心跳数据
 - 90 秒没有收到心跳数据,认为服务不可用

- 默认不会剔除服务
- 客户端每 30 秒拉取一次新的注册数据
- eureka 配置
 - ◆ pom.xml
 - eureka-server
 - ◆ 主程序
 - @EnableEurekaServer
 - ◆ application.yml
 - application.name=eureka1
 - server.port=2001
 - defaultZone=http://localhost:2001/eureka
- ribbon
 - 作用：负载均衡和重试
 - 封装 RestTemplate,生成动态代理,来添加负载均衡和重试的代码
 - ribbon
 - ◆ pom.xml
 - starter-ribbon
 - 如果有 eureka-client 依赖,ribbon 依赖不需要单独添加
 - ◆ 创建 RestTemplate 的方法上,添加 @LoadBalanced
 - ◆ restTemplate 向远程主机发送请求,使用 http:// 服务 id/....

3 ribbon 重试

单独使用 ribbon,开启重试需要自己写代码,添加重试参数配置

后面 ribbon 被 feign 和 zuul 封装,不做任何配置,不写任何代码,可以直接使用负载均衡和重试

- 添加 spring-retry 依赖
- yml 中配置重试参数
- RequestFactory 中配置超时参数

4 hystrix

断路器

降级和熔断

降级:

当服务不可用,或者访问超时,可以执行降级代码,可以返回错误信息,或者返回缓存数据

熔断:

10秒内请求数量达到20个,并且,有50%出现失败降级的情况,会打开断路器,所有的请求,都直接降级,不再向后台微服务转发

断路器打开5秒后,会处于半开状态,当请求到达时,会尝试向后台转发,如果成功,会关闭断路器,如果仍然失败,保持打开状态

5 feign

- 声明式客户端
- 整合 ribbon 和 hystrix

5.1 声明式客户端

```
@FeignClient(name="item-service",
fallback=ItemServiceFB.class)
interface ItemService {
    @GetMapping("/{orderId}")
    JsonResult getItems(@PathVariable String orderId);
}

result = service.getItems("123");
```

- 请求 <http://localhost:8001/123> 并得到返回的响应结果

```
class ItemServiceFB implements ItemService {  
    public JsonResult getItems(String orderId) {  
        return JsonResult.err("....");  
    }  
}
```

5.2 feign 整合 ribbon

不需要任何配置,已经启用 ribbon 的负载均衡和重试

5.3 feign 整合 hystrix

启用 hystrix

```
feign.hystrix.enabled=true
```

添加 hystrix 依赖

暴露 hystrix.stream 监控端点

6 作业

<http://github.com/benwang6>

下载 springcloud10

重做 11 到 17

