一．在服务消费者上修改

1.pom配置依赖

<dependency>

   <groupId>org.springframework.cloud</groupId>

   <artifactId>spring-cloud-starter-hystrix</artifactId>

</dependency>

2. 修改服务消费者启动入口类

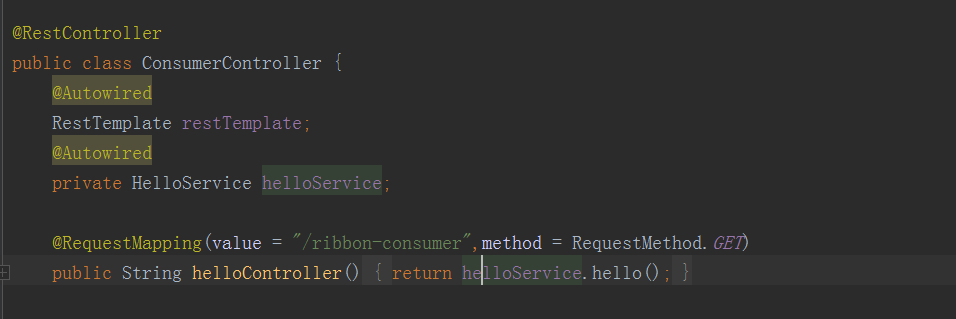


3. 创建一个HelloService类



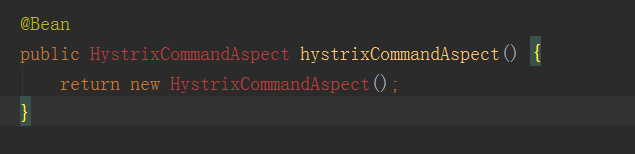
1.RestTemplate执行网络请求的操作我们放在HelloService中来完成。  
2.error方法是一个请求失败时回调的方法。  
3.在hello方法上通过@HystrixCommand注解来指定请求失败时回调的方法。

4. Controller



二．以上是同步调用，异步调用

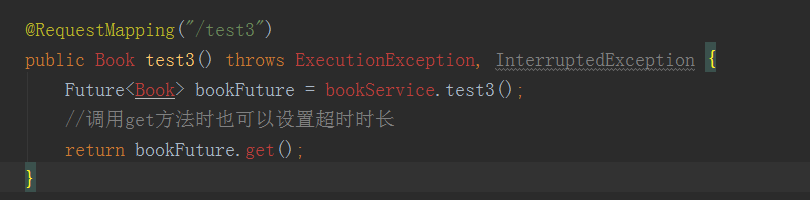
1. 在项目的入口类中配置一个HystrixCommandAspect的Bean



2. 使用@HystrixCommand注解



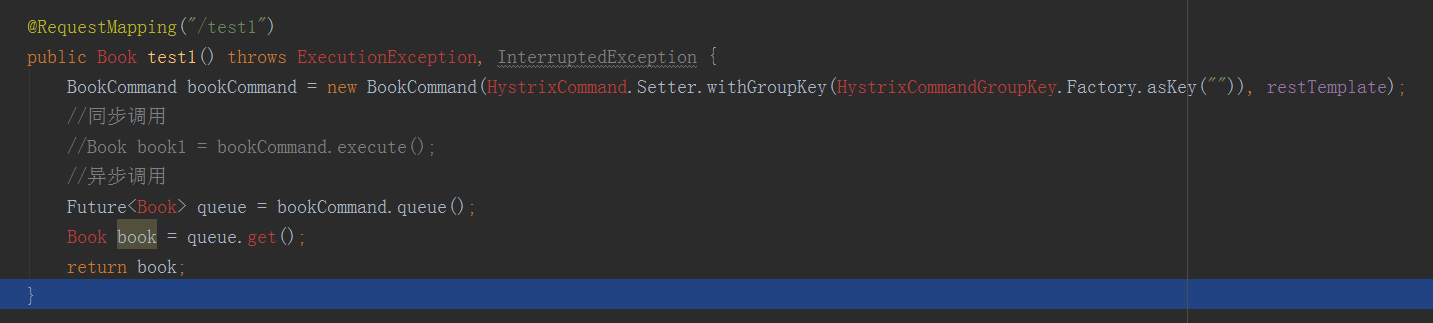
3.异步调用



三．自定义类继承自HystrixCommand

1. 在自定义类中注入RestTemplate，然后重写两个方法：一个是getFallback，这个方法将在服务调用失败时回调；另一个是run方法，执行请求时调用。构造方法的第一个参数主要用来保存一些分组信息。

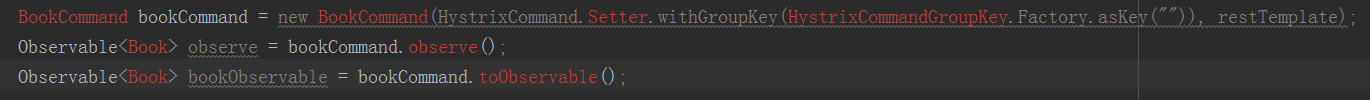


1. Controller中调用

1.获取到BookCommand对象之后，我们有两种方式来执行请求，一种是调用execute方法发起一个同步请求，另一种是调用queue方法发起一个异步请求。  
2.同步请求中可以直接返回请求结果。  
3.异步请求中我们需要通过get方法来获取请求结果，在调用get方法的时候也可以传入超时时长。

四．对响应式函数编的支持

1. 获取到BookCommand对象之后，也可以通过如下两种方式来获取到一个Observable来对数据进行二次处理



1.observe命令在调用的时候会立即返回一个Observable。  
2.toObservable则不会立即返回一个Observable，订阅者调用数据的时候才会执行。

2. 响应式函数编程通过注解来实现



@HystrixCommand(observableExecutionMode = ObservableExecutionMode.EAGER)表示使用observe模式来执行

@HystrixCommand(observableExecutionMode = ObservableExecutionMode.LAZY)表示使用toObservable模式来执行