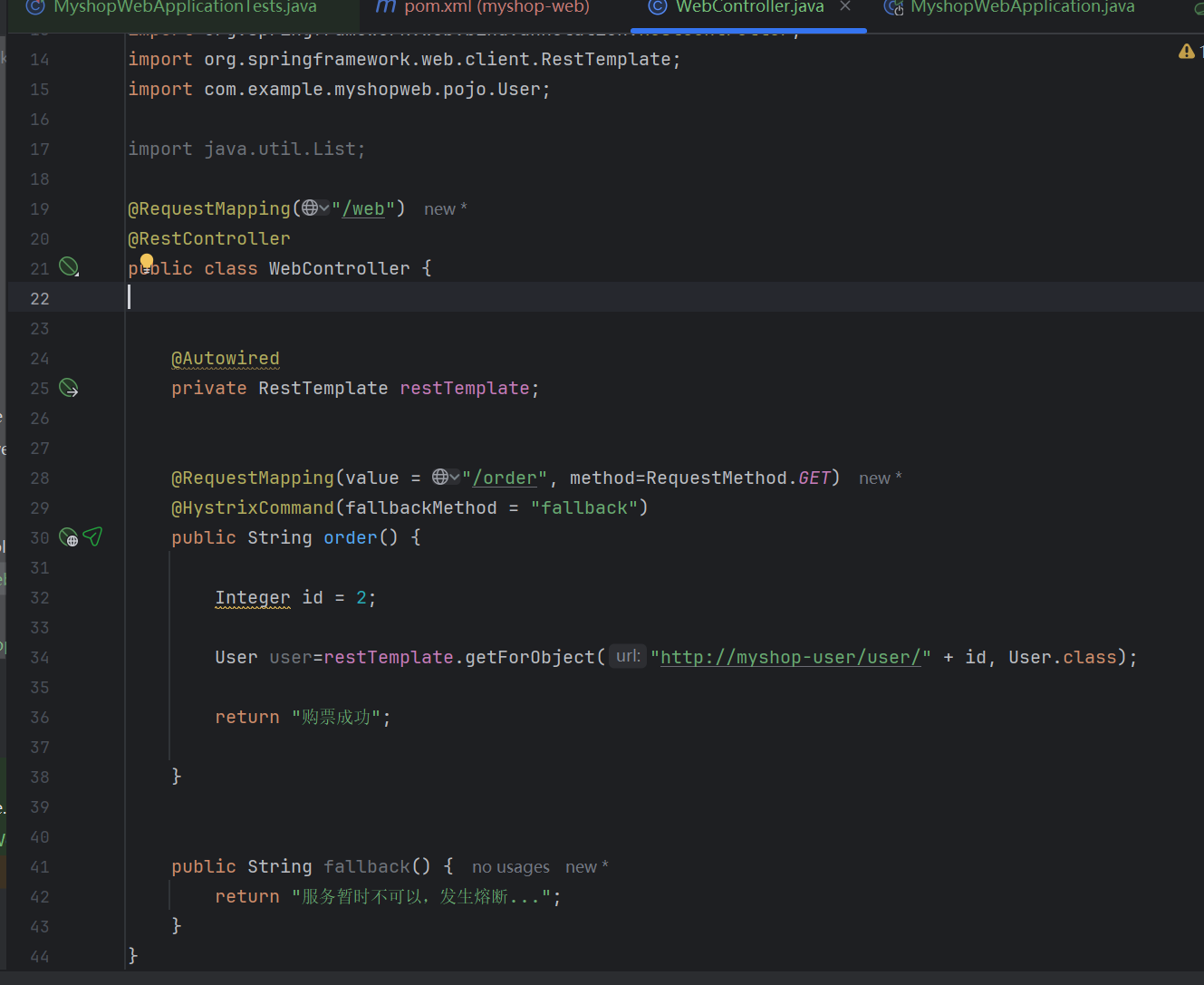
# Spring Cloud熔断器

熔断器是用来保护服务调用方不被出错的被调用方拖垮，因此，熔断器应该添加在服务调用方。本例中购票微服务（myshop-web）调用用户微服务（myshop-web），因此，以下Hystrix设置均在购票微服务中进行。

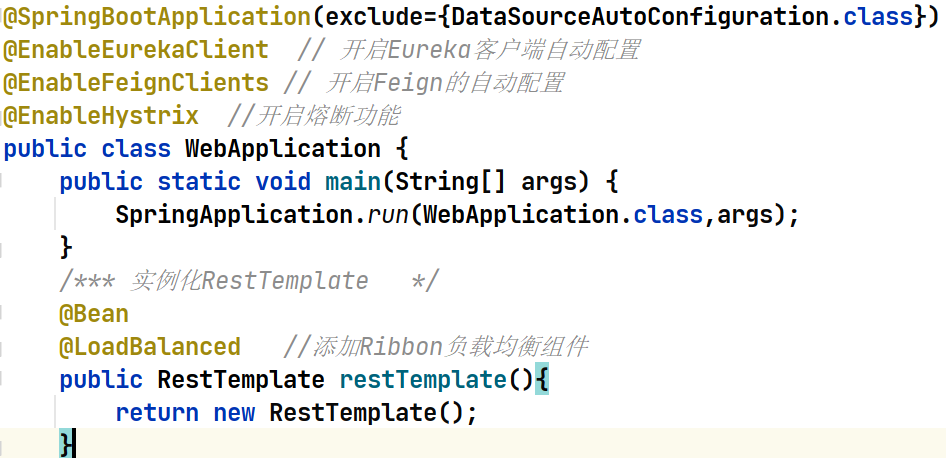
1. 方案1：RestTemplate+ ribbon+Hystrix
2. 导入hystrix的依赖
3. WebController类的order（）方法使用@HystrixCommand注解声明fallback回滚方法，当order方法中远程调用的用户微服务故障时跳转到fallback方法执行



1. 编写fallback回滚方法逻辑。如上图，远程方法出错时跳到fallback回滚方法执行



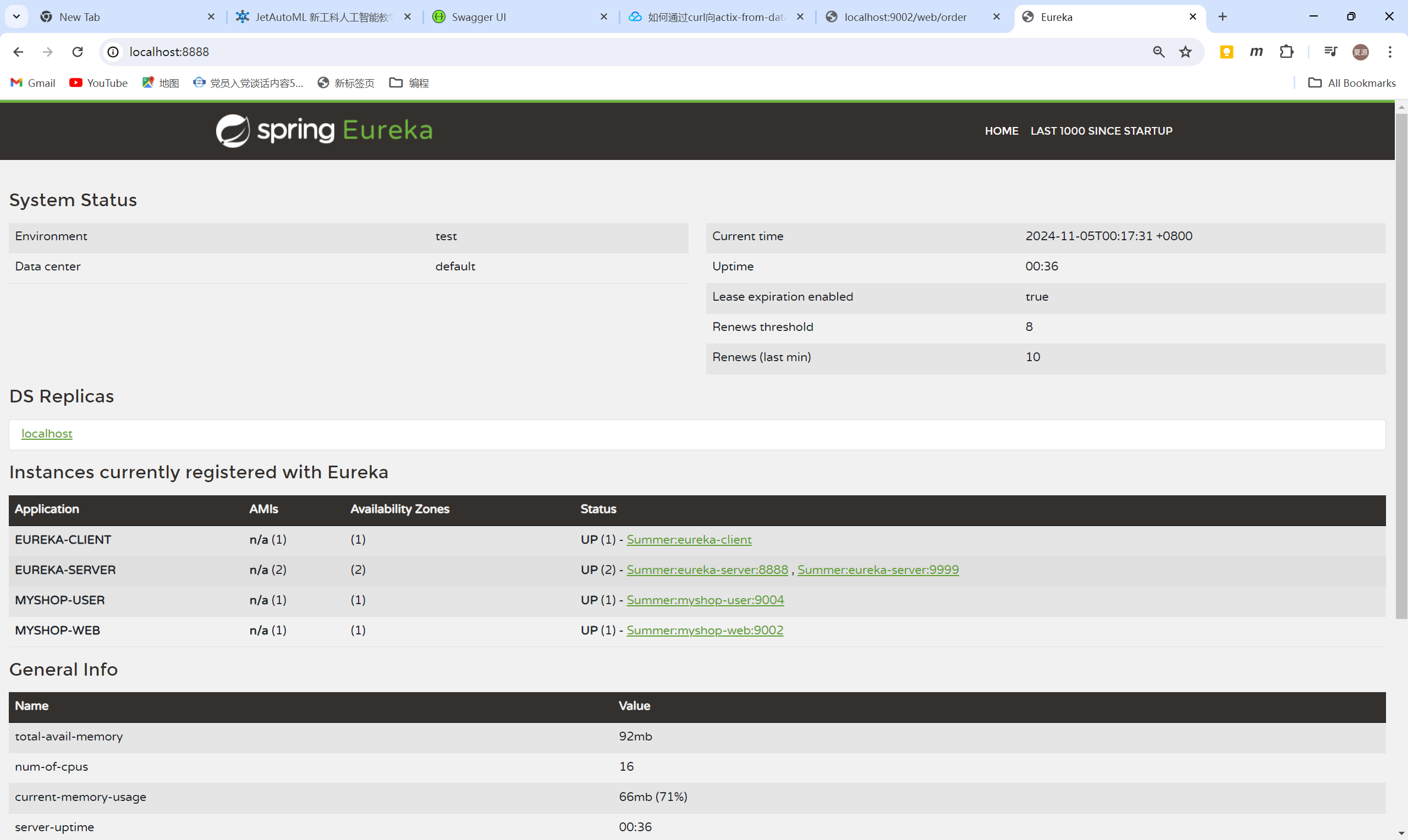
1. 在启动类添加@EnableHystrix注解，注意恢复实例化RestTemplate的方法



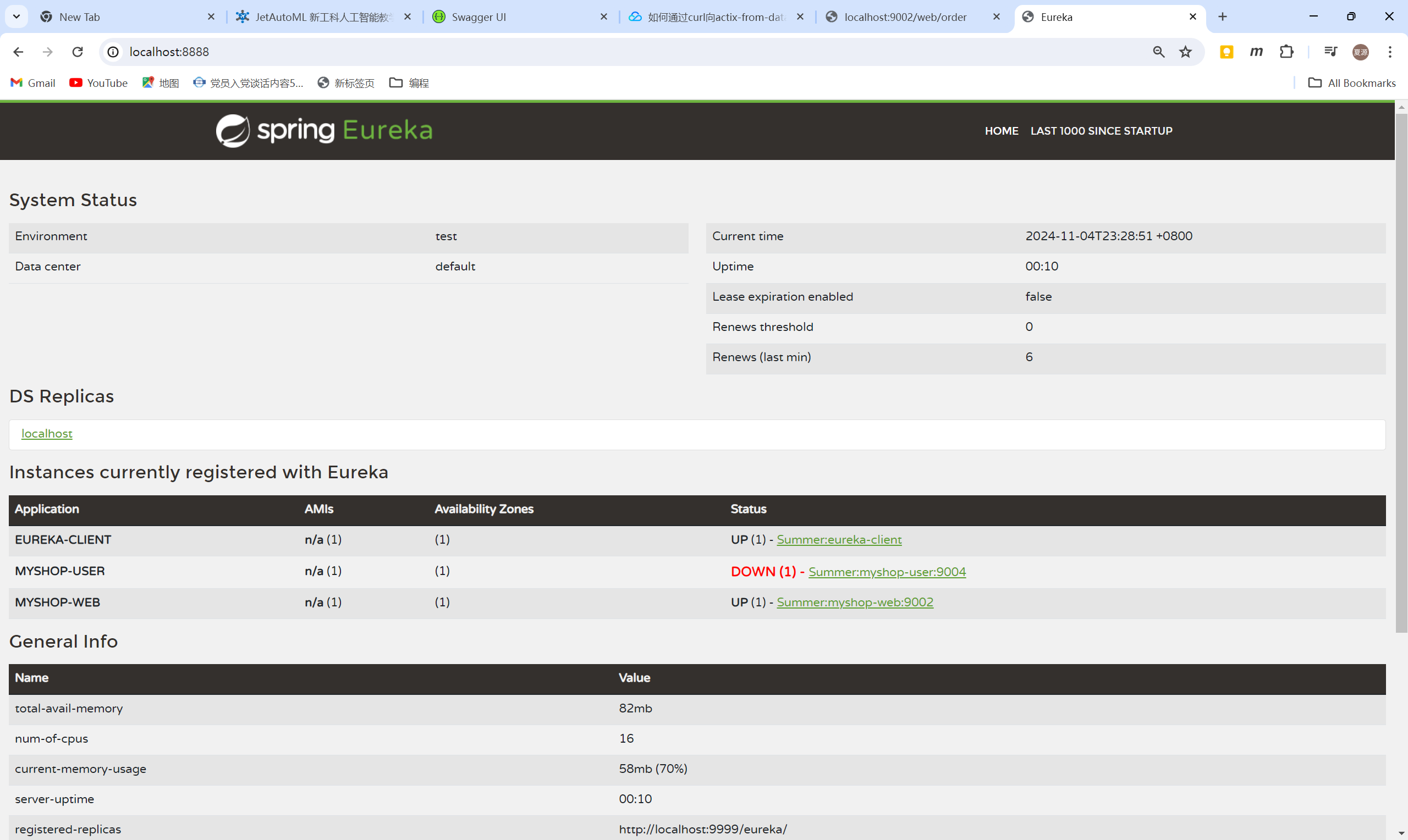
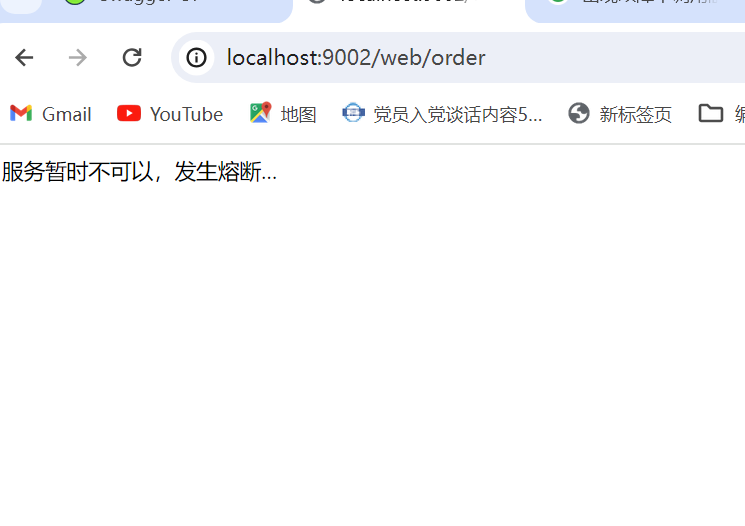


1. 测试。

* 开启用户微服务、购票微服务，浏览器请求order方法，记录eureka界面及请求返回界面。（截图）





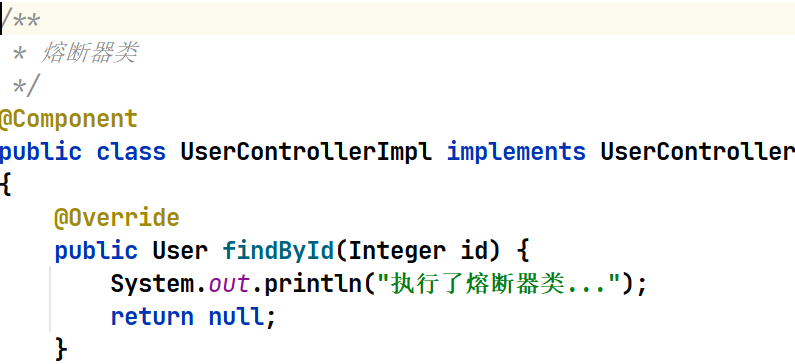
* 关闭用户微服务（模拟用户微服务故障），记录eureka界面及请求返回界面，重启用户微服务后再次记录eureka界面及请求返回界面。（截图）
* 
* 

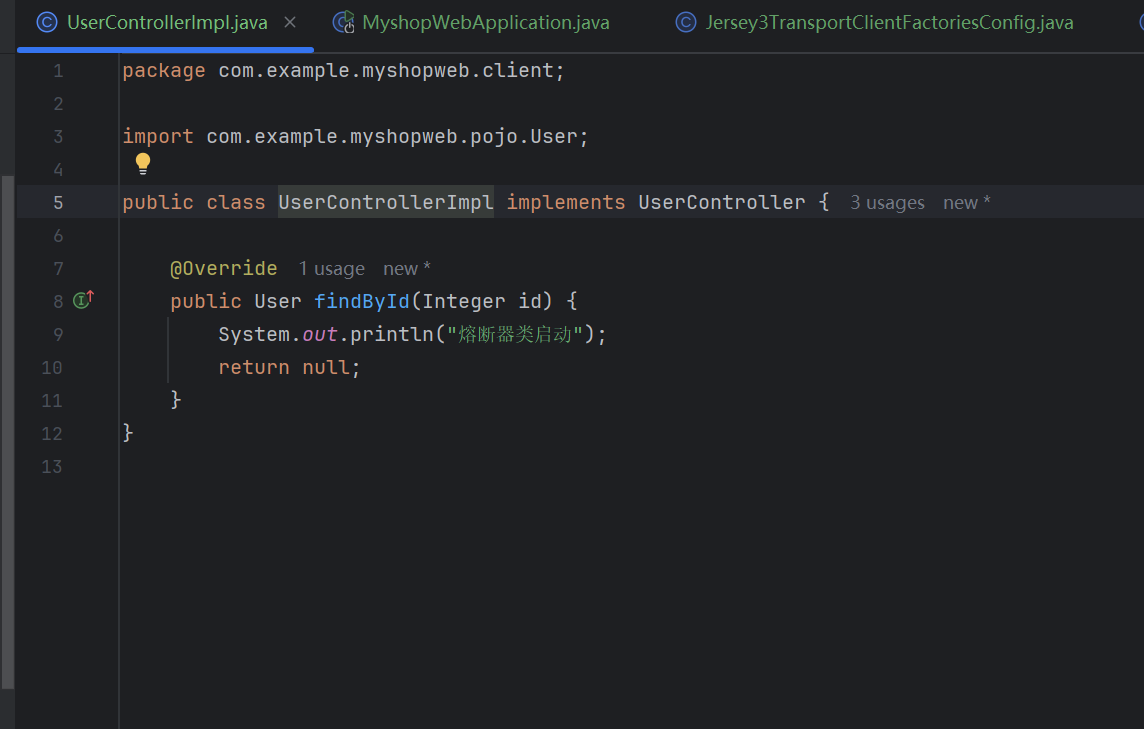
1. 方案2：OpenFeign+Hystrix
2. OpenFeign开启Hystix功能

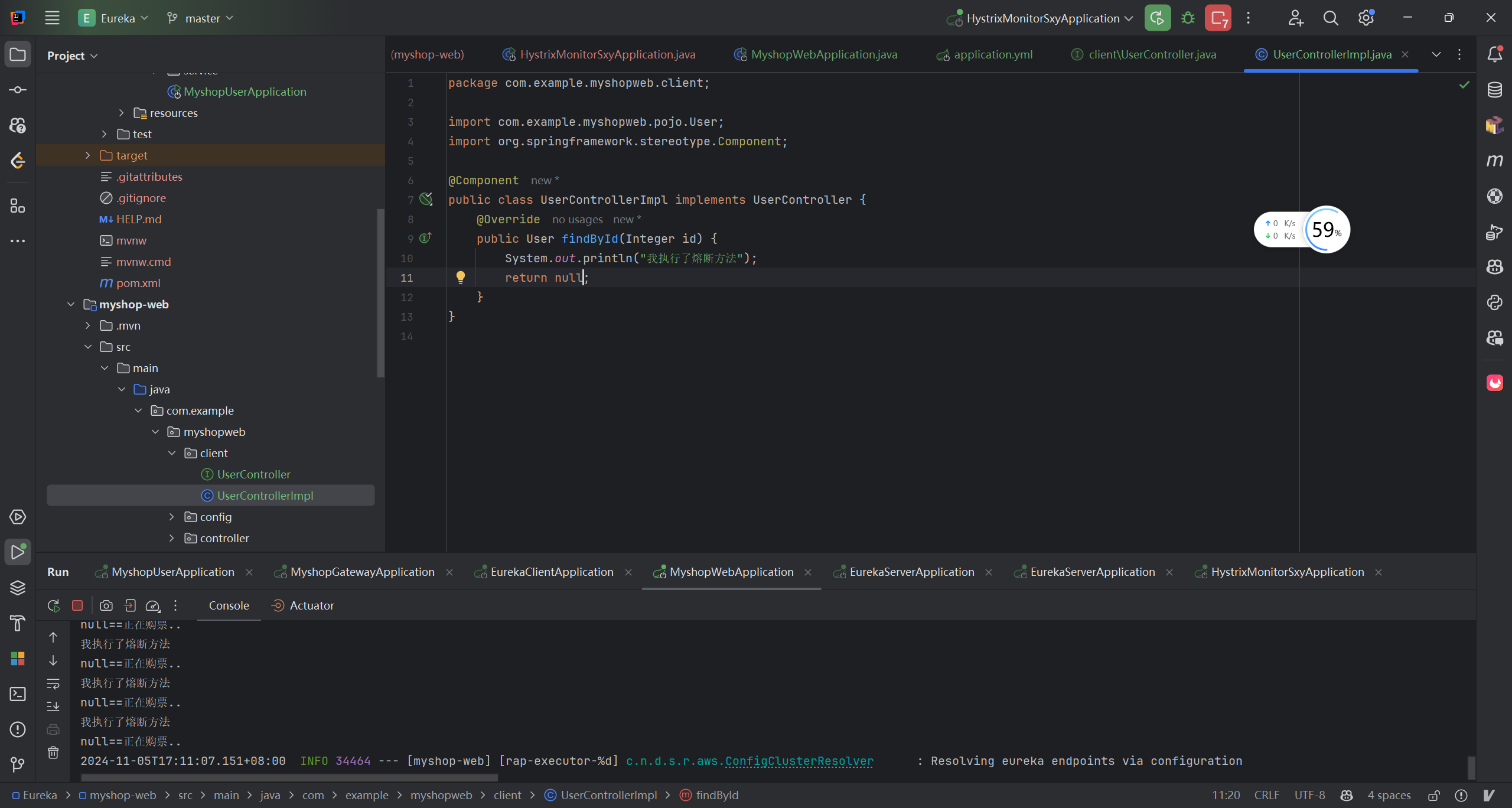
OpenFeign有对Hystix的集成，因此不需要导依赖。但默认情况下Hystix是关闭的。需要在application.yml里通过下面的参数来开启：



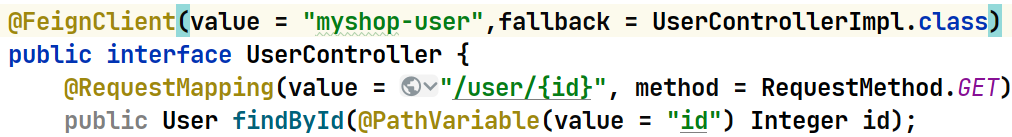
1. 编写熔断器类UserControllerImpl（client包），实现lab4中的UserController 接口



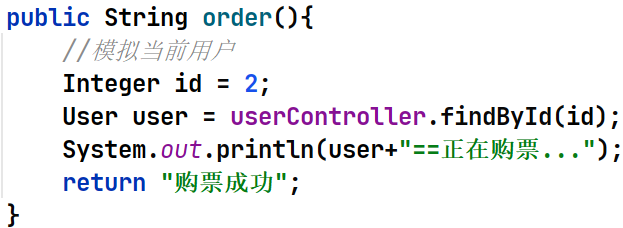


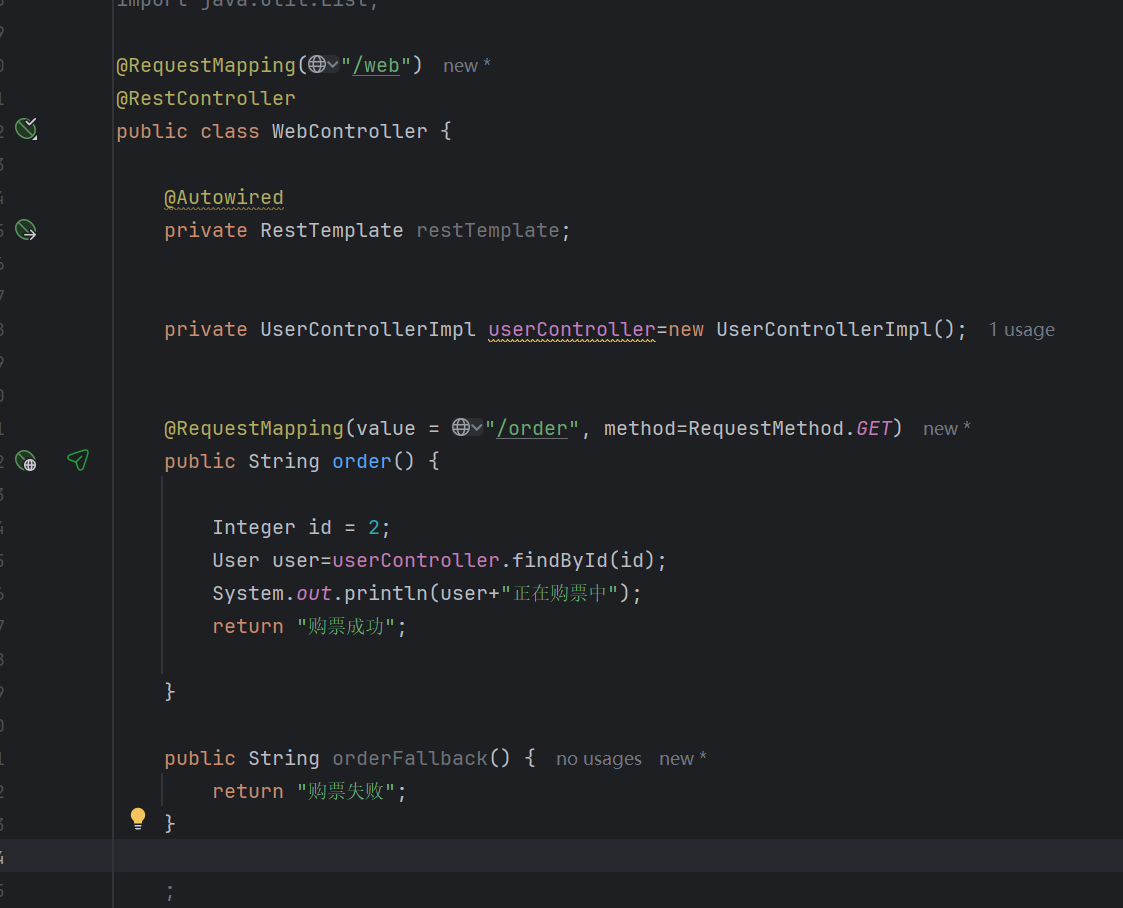


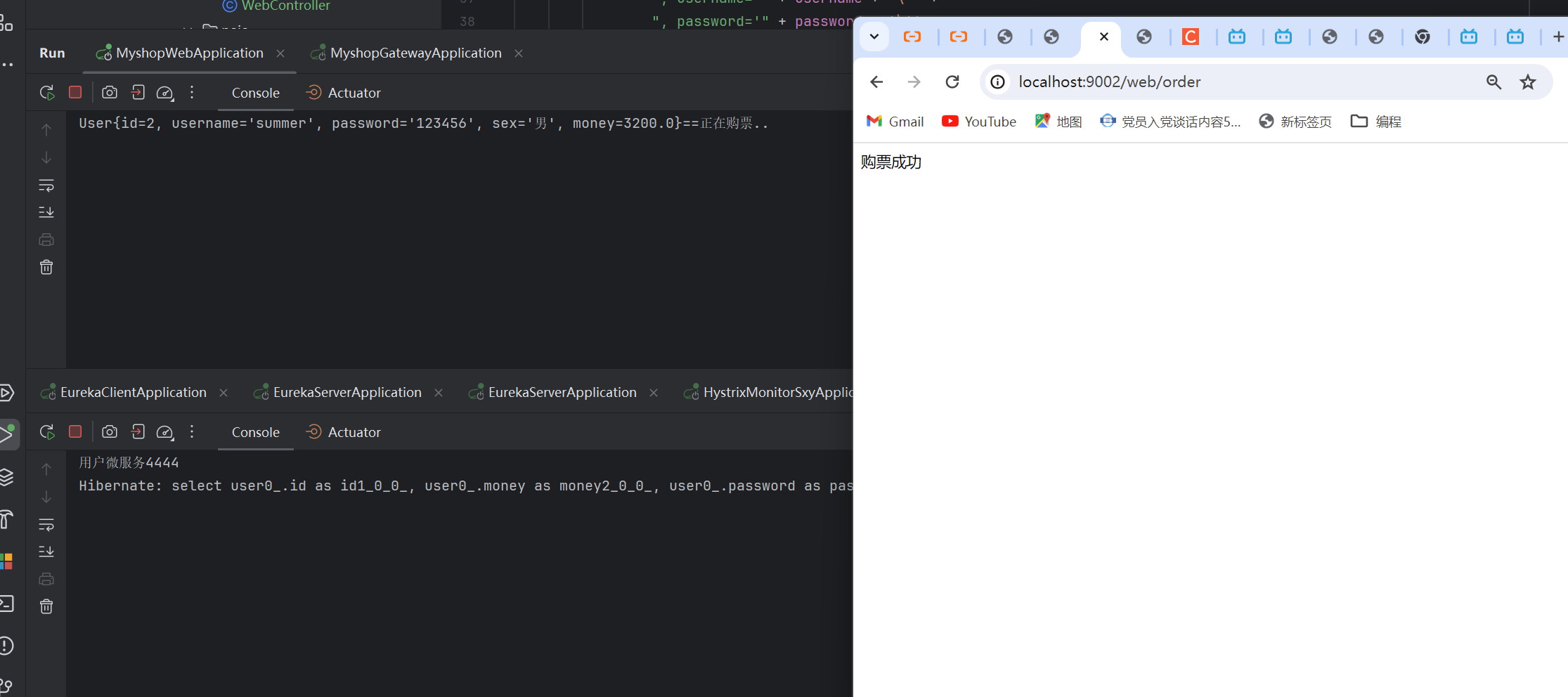
1. UserController接口上增加fallback，指向UserControllerImpl熔断器类

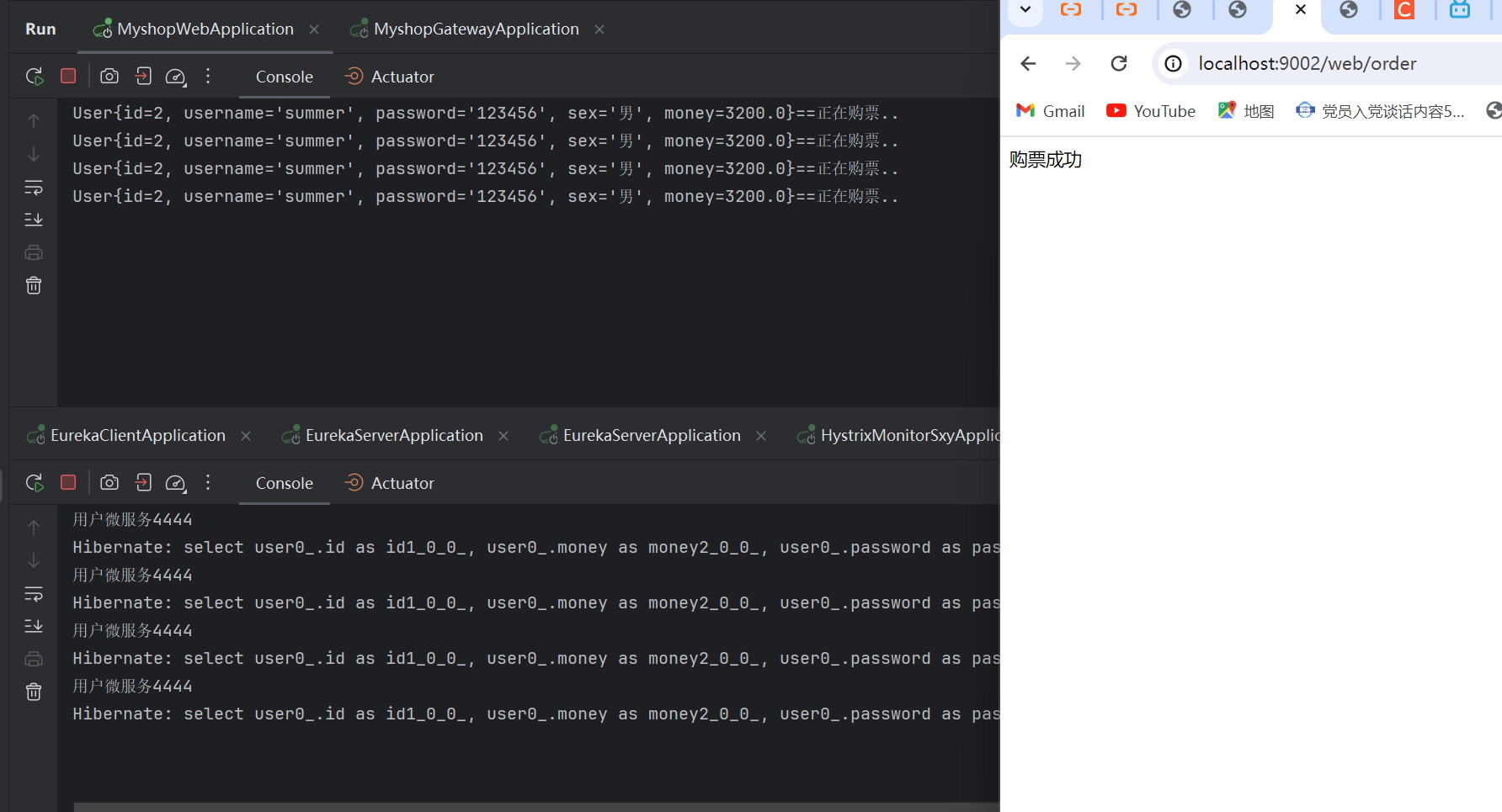


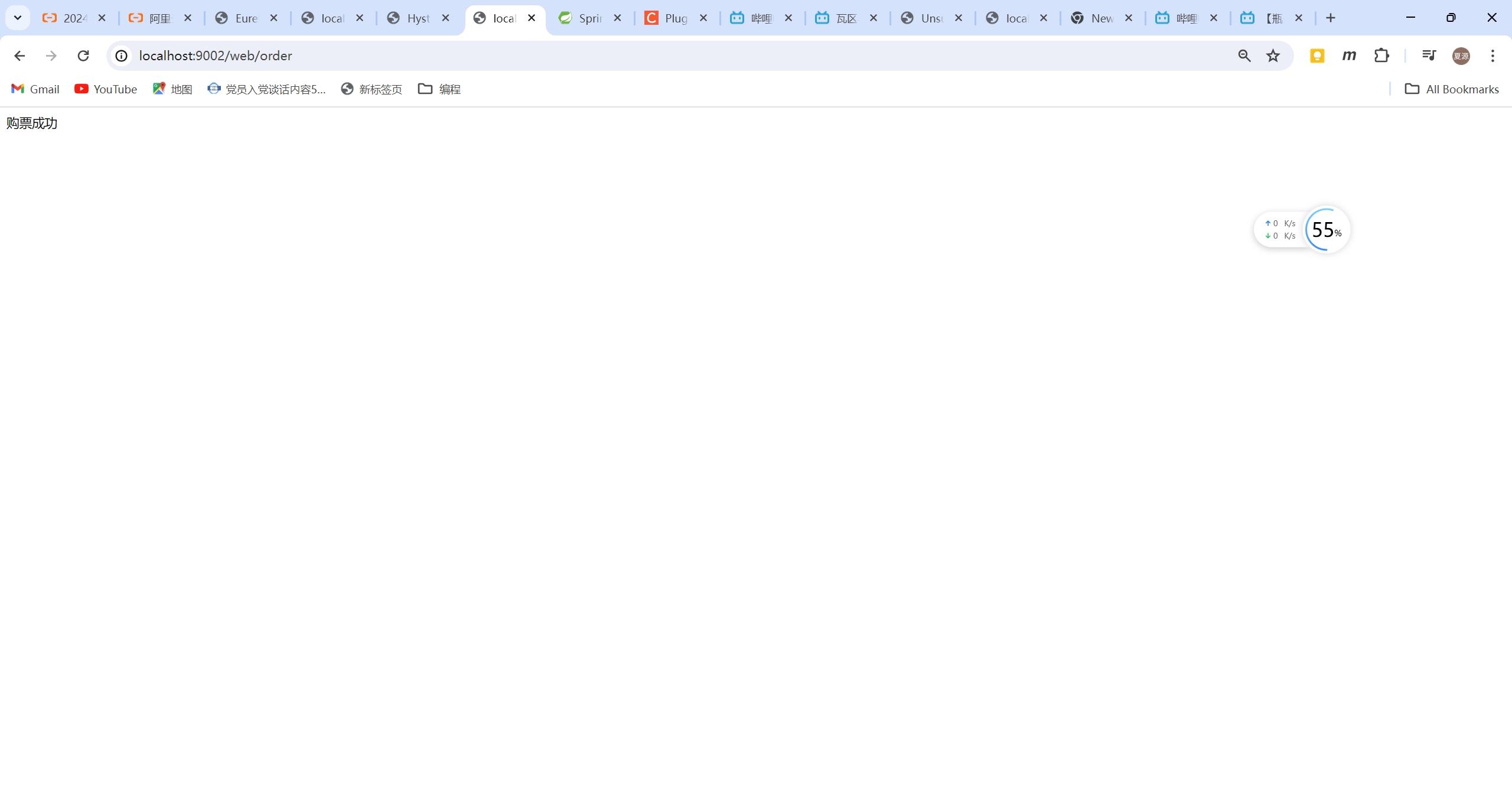
1. 将WebController改为使用OpenFeign的代码



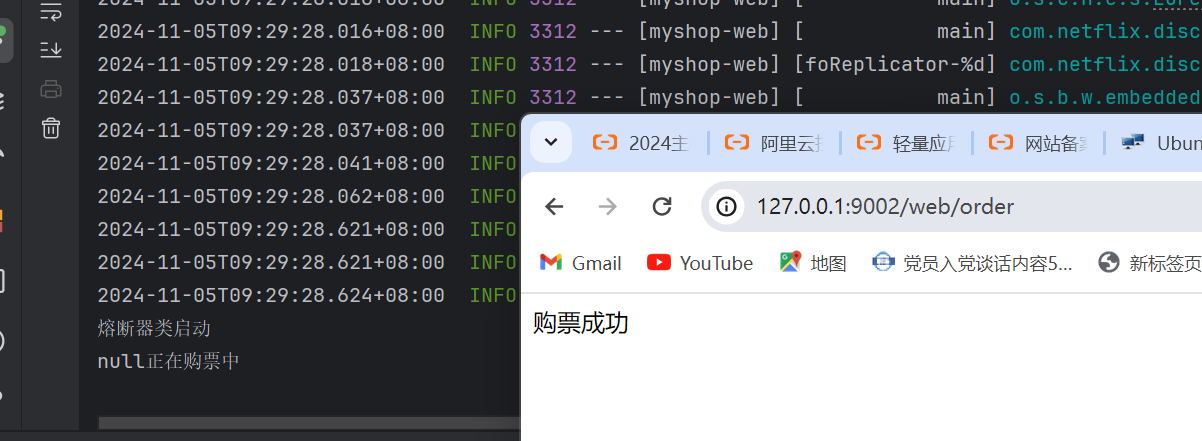
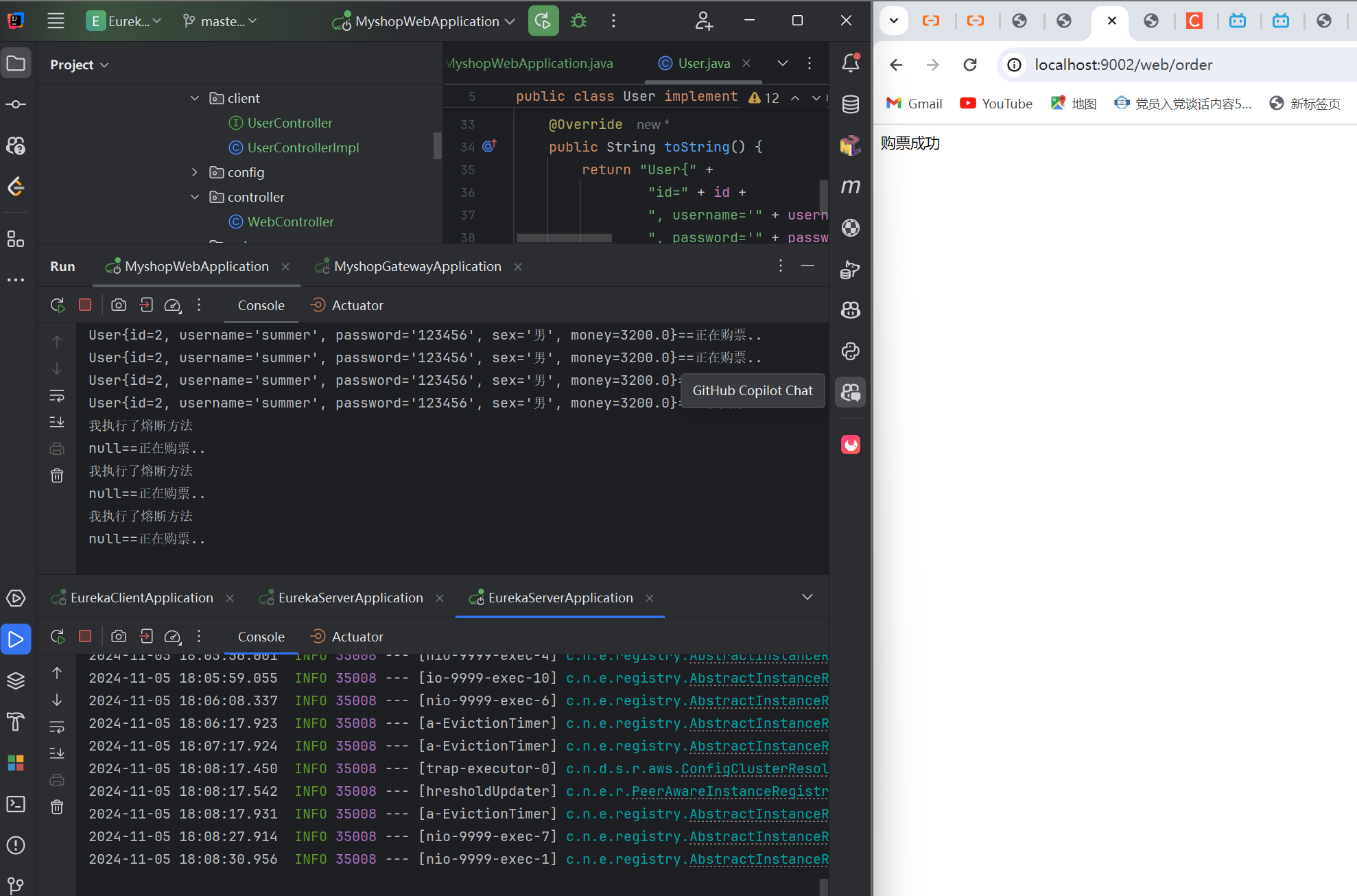


1. 重启购票微服务测试。记录开启用户微服务及关闭微服务状态下浏览器请求返回界面及IDEA控制台信息。（截图）
2. 
3. 







1. 
2. 
3. Hystrix Dashboard监控面板
4. 搭建Hystrix Dashboard工程

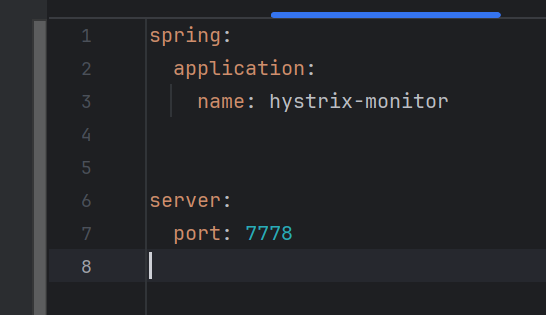
新建名为hystrix-monitor的微服务

1. hystrix-monitor微服务导入依赖

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

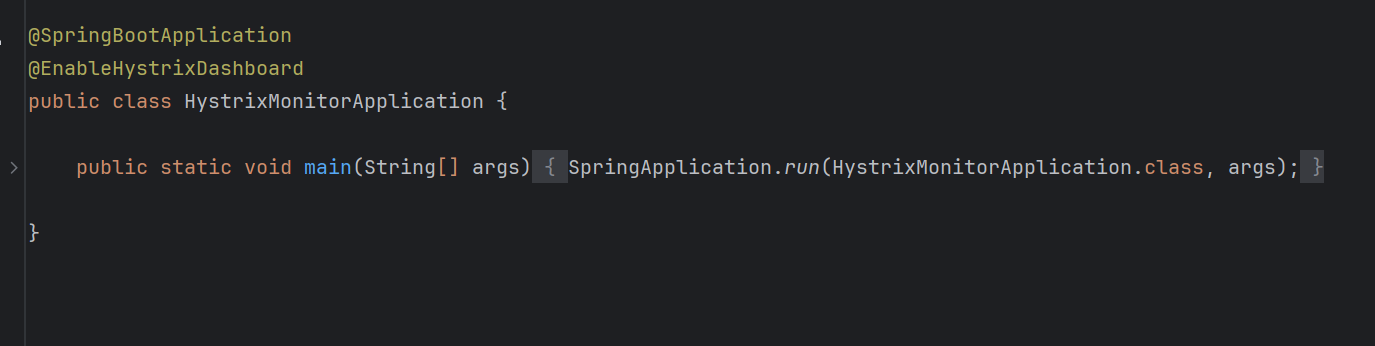
1. 编写application.yml配置文件

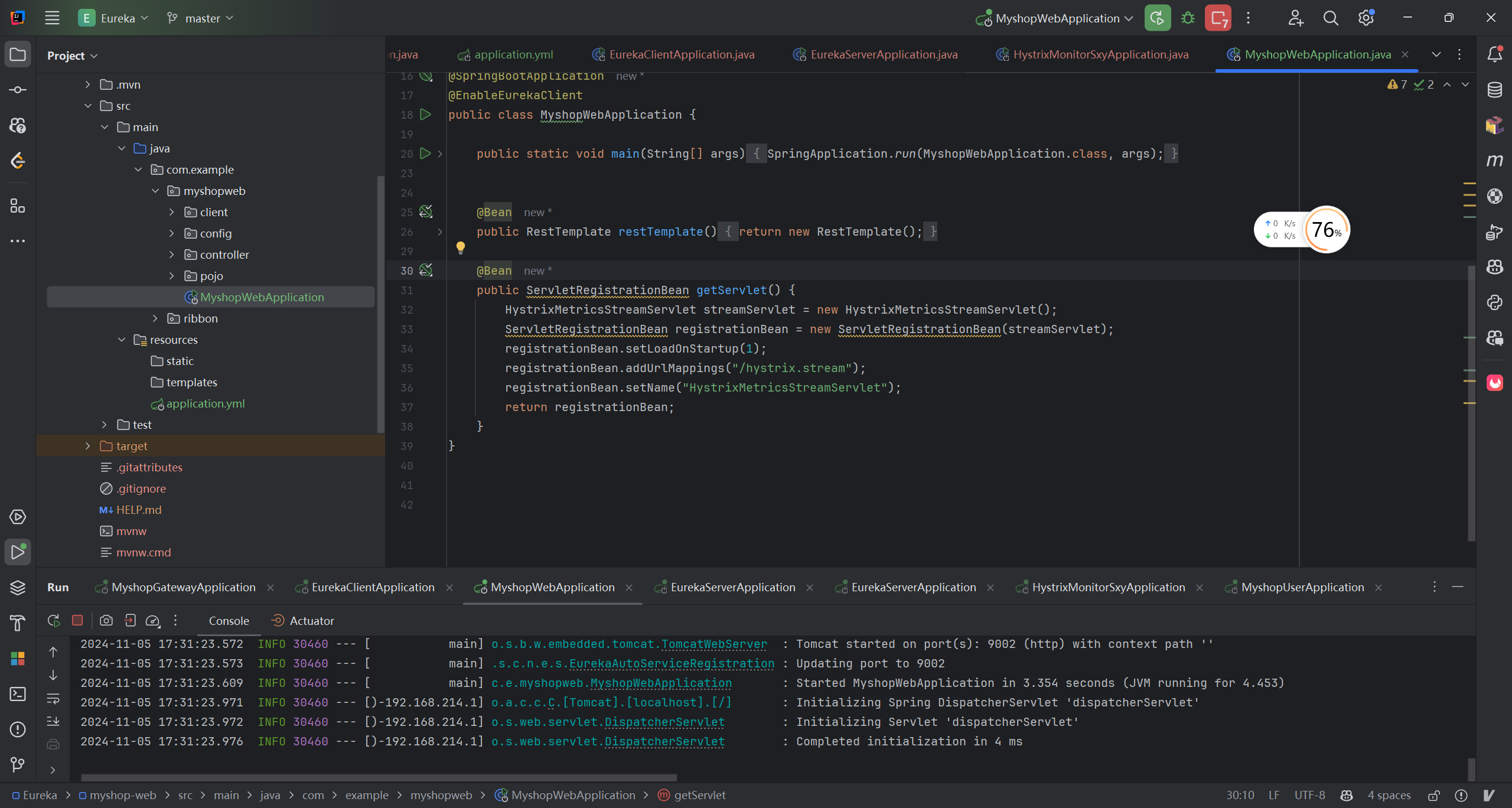




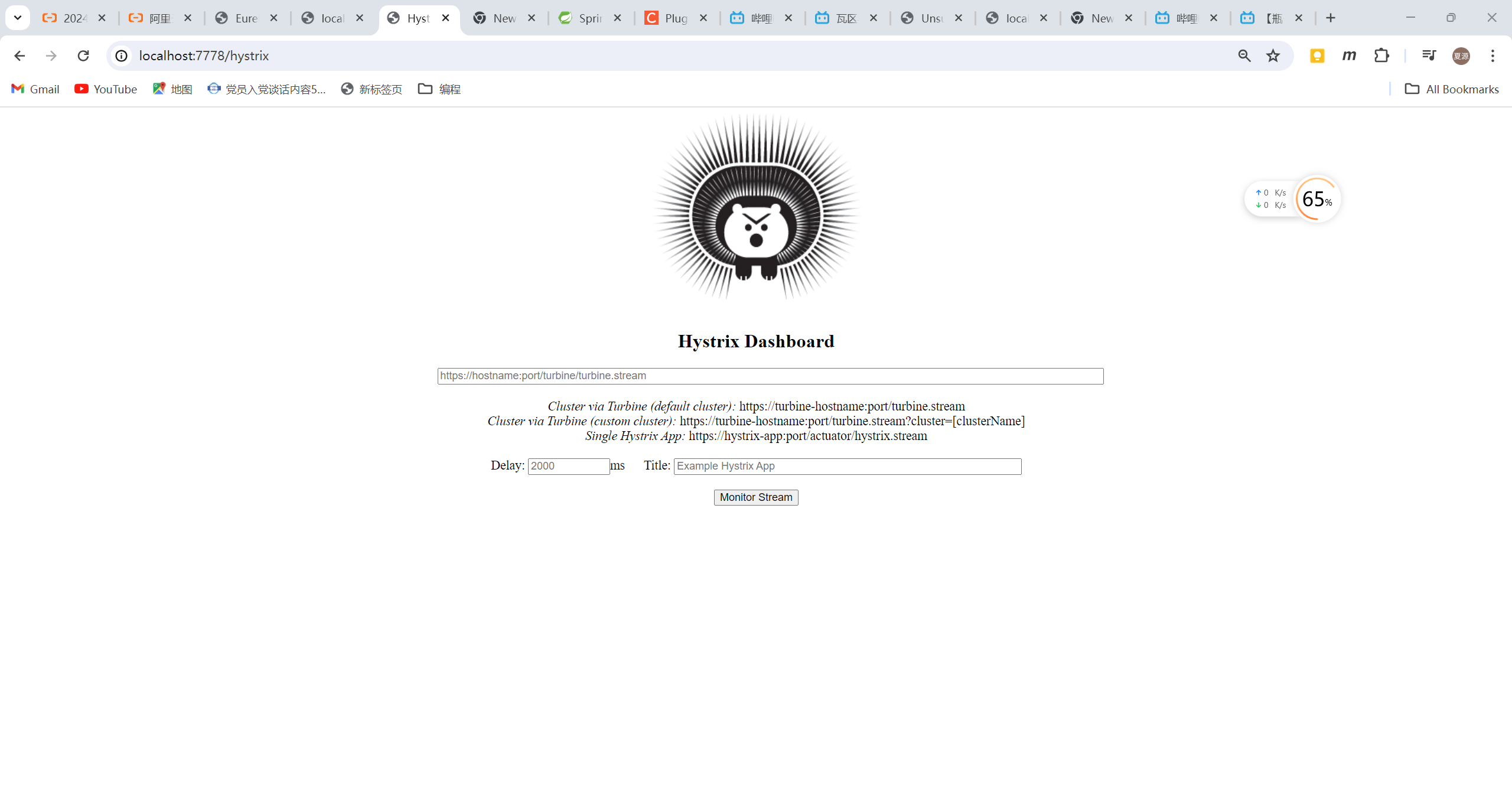
1. 启动类添加@EnableHystrixDashboard注解







启动Hystrix Dashboard监控面板微服务，并访问其用户界面（截图）



1. 在服务调用方（购票微服务）的启动类，添加监听方法的Servlet

启动类添加如下ServletRegistrationBean：

@Bean  
 public ServletRegistrationBean getServlet() {  
 HystrixMetricsStreamServlet streamServlet = new HystrixMetricsStreamServlet();  
 ServletRegistrationBean registrationBean = new ServletRegistrationBean(streamServlet);  
 registrationBean.setLoadOnStartup(1);  
 registrationBean.addUrlMappings("/hystrix.stream");  
 registrationBean.setName("HystrixMetricsStreamServlet");  
 return registrationBean;  
 }

1. 启动各微服务，访问hystrix dashboard。浏览器多次请求order方法，记录监控界面。（截图）

