菜购项目

测试文档

项目名称：菜购

组长：翁宏晖

组员：王少聪

组员：陈志君

组员：傅灿宇

组员：韦皓晨

组员：卓尚宇

组员：廖龙垲

组员：钟焓

组员：艾齐

组员：杨彬

2021 年 5 月 14 日

目录

[1测试团队成员 2](#_Toc17444)

[1.1 前端 2](#_Toc5162)

[1.2 后端 2](#_Toc1547)

[2后端测试 3](#_Toc24508)

[2.1 单元测试 3](#_Toc1307)

[2.2 API测试 13](#_Toc31837)

[3 前端测试 21](#_Toc9319)

[3.1Android测试 21](#_Toc30474)

# 

# 1测试团队成员

## 1.1 前端

测试成员：041802224 翁宏晖、221801422 卓尚宇

使用工具：Junit、LeakCanary

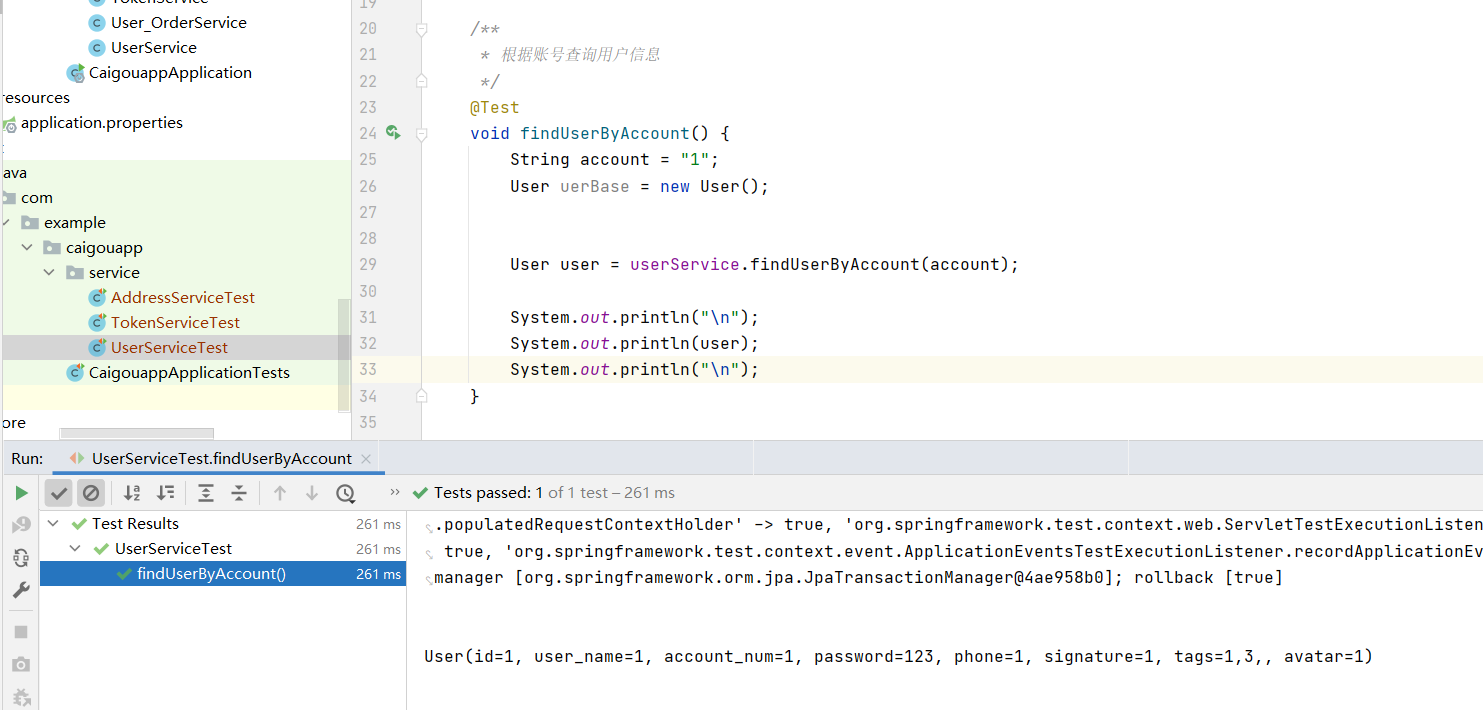
## 1.2 后端

测试成员：221801404 韦皓晨、221801224 钟焓、221801413 艾齐；

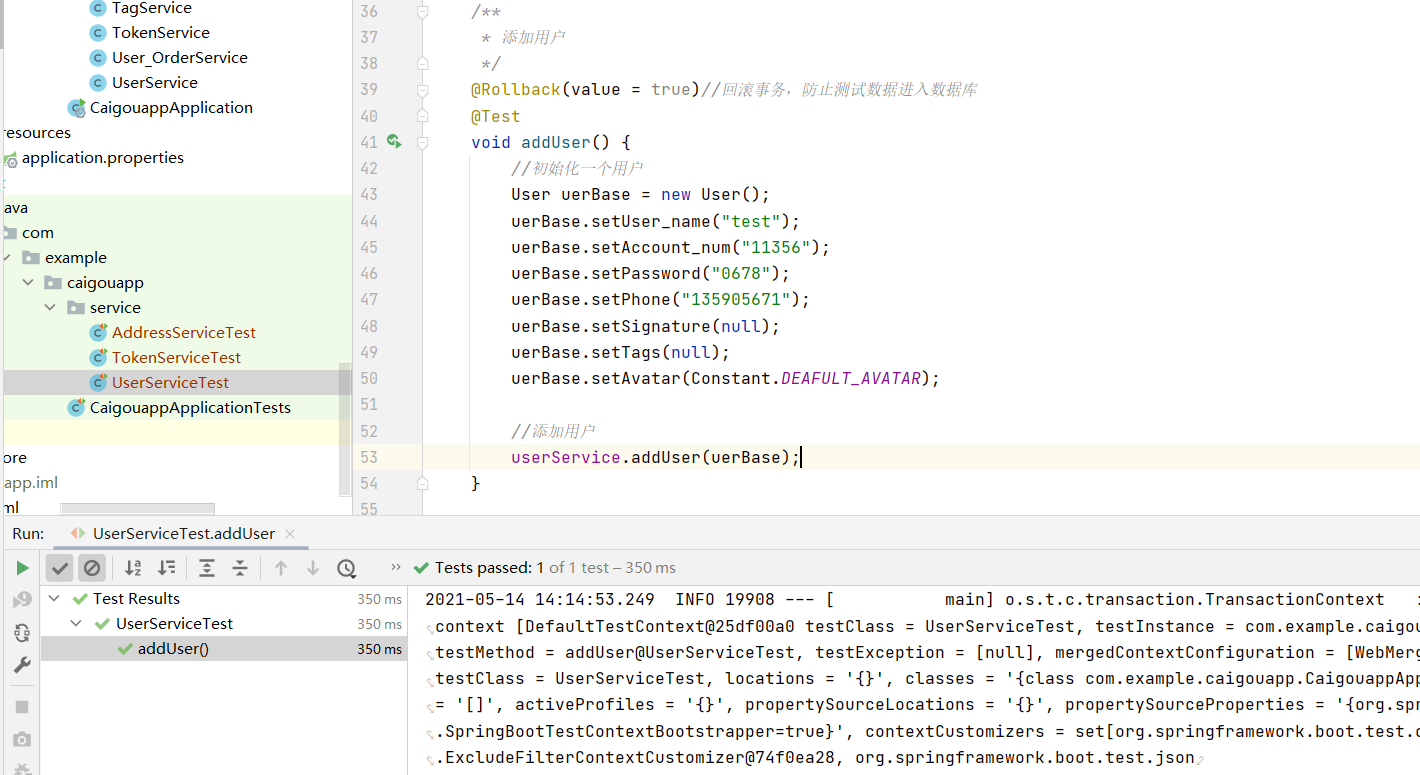
使用工具：Junit、PostMan、ApiRun

# 2后端测试

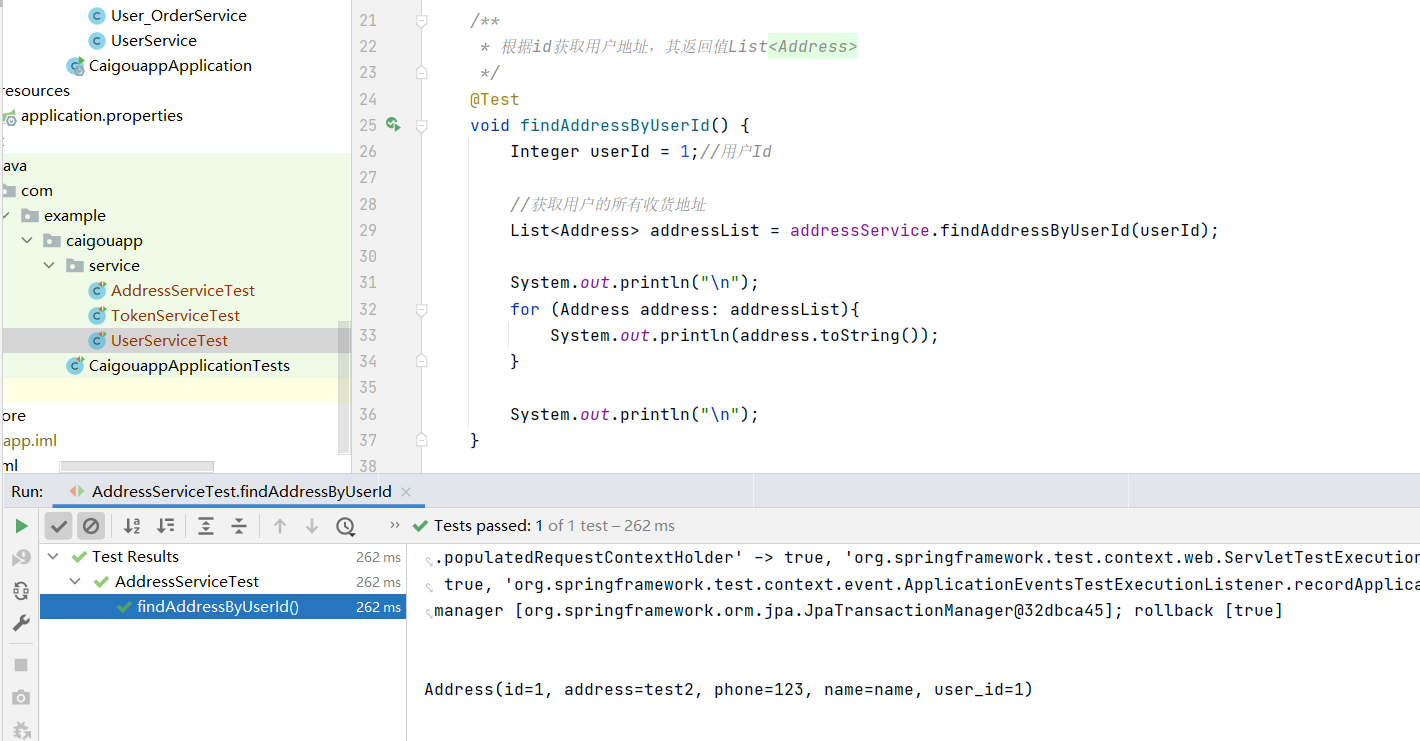
## 2.1 单元测试

1.通过账号account查找用户

2.通过Id查找用户

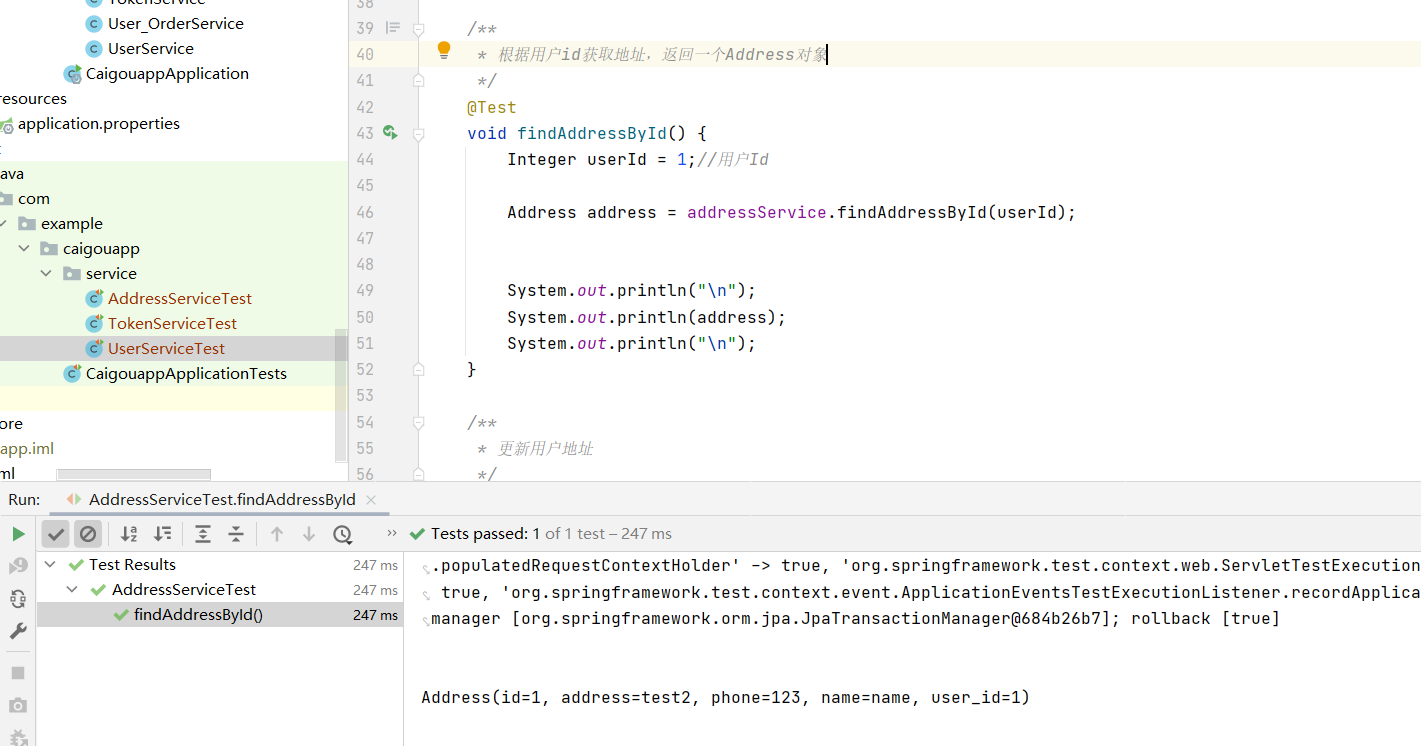
3.添加一个用户

4.根据用户信息生成token

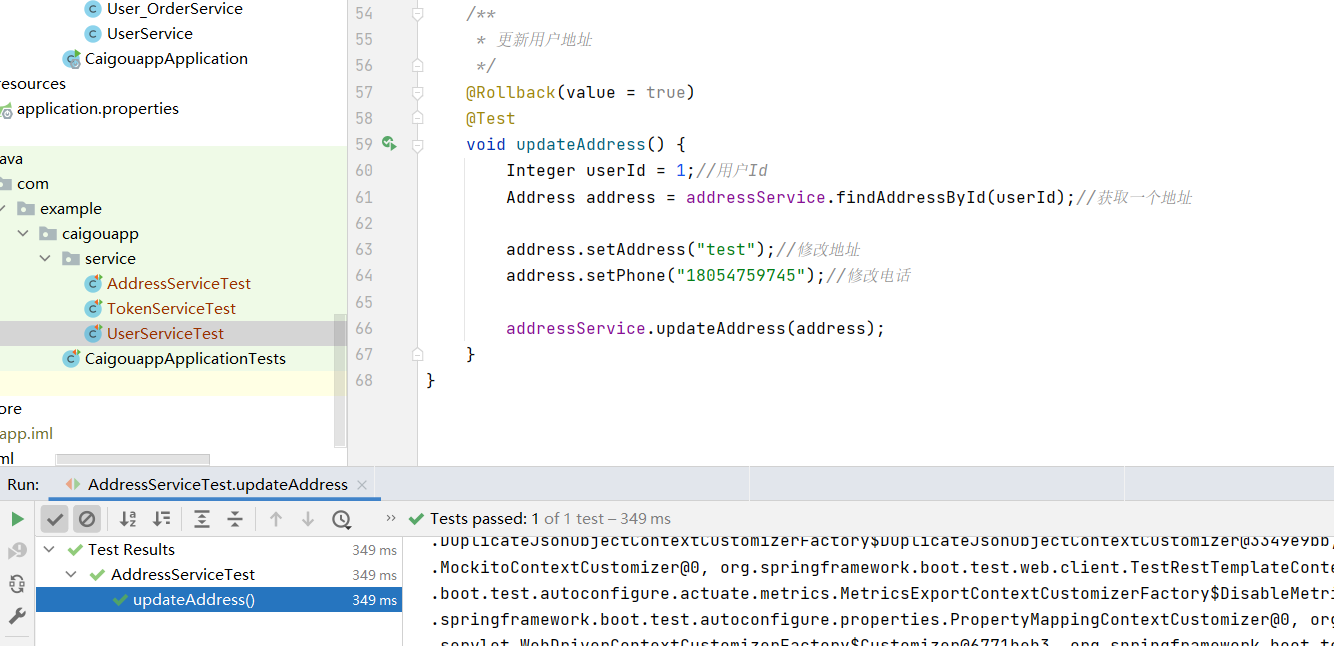


5.通过用户账号account查找其收货地址信息

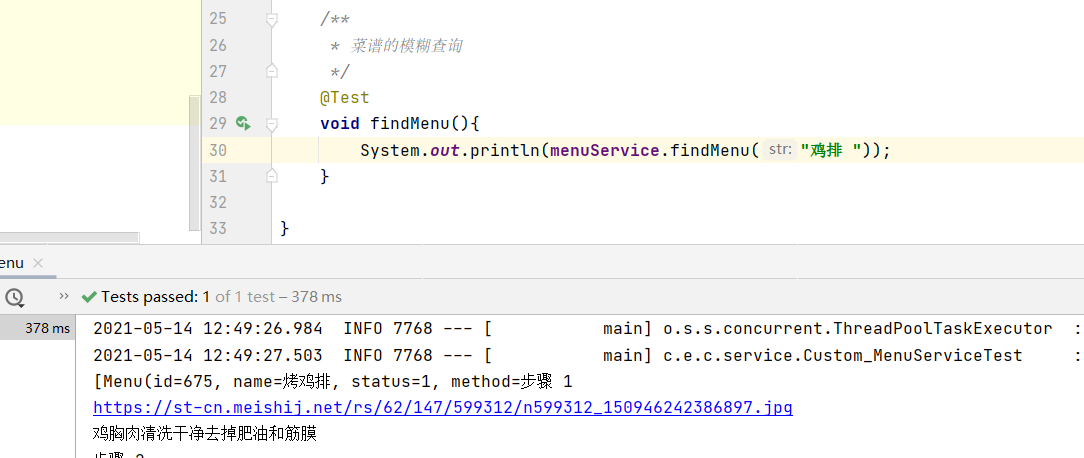
6.根据收货地址Id查找收货地址信息



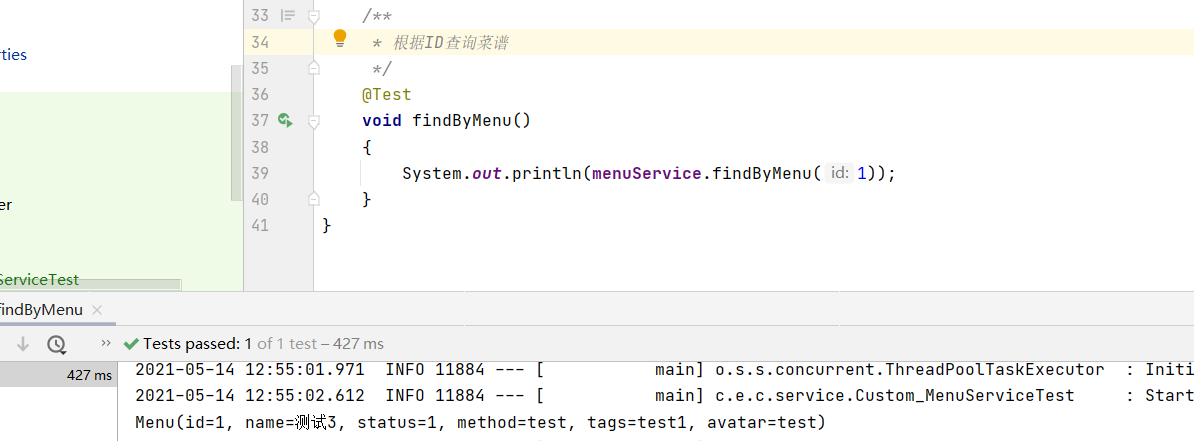
7.更新地址信息



8菜谱的模糊查询



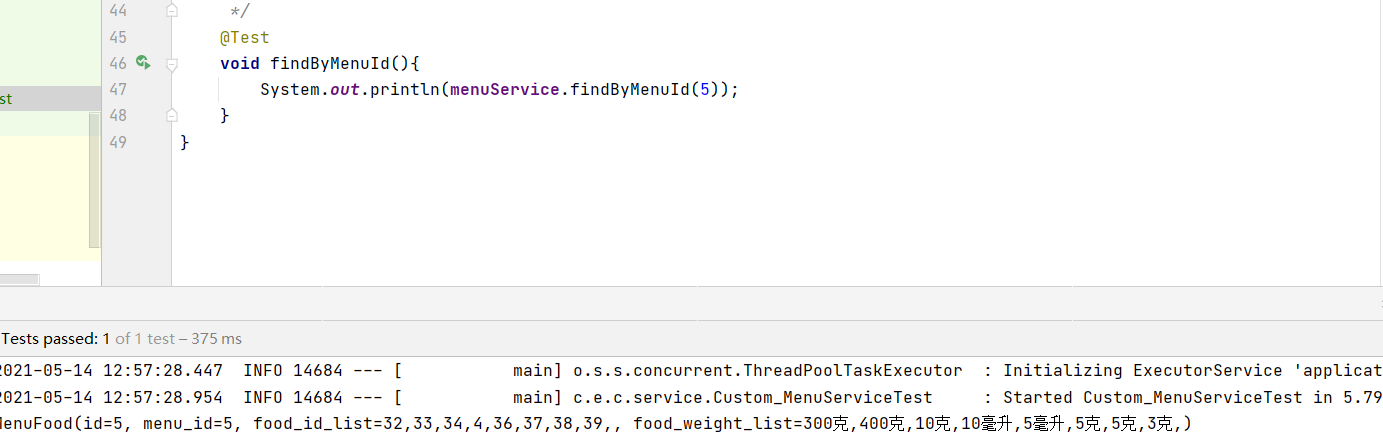
9.根据菜谱ID进行查询



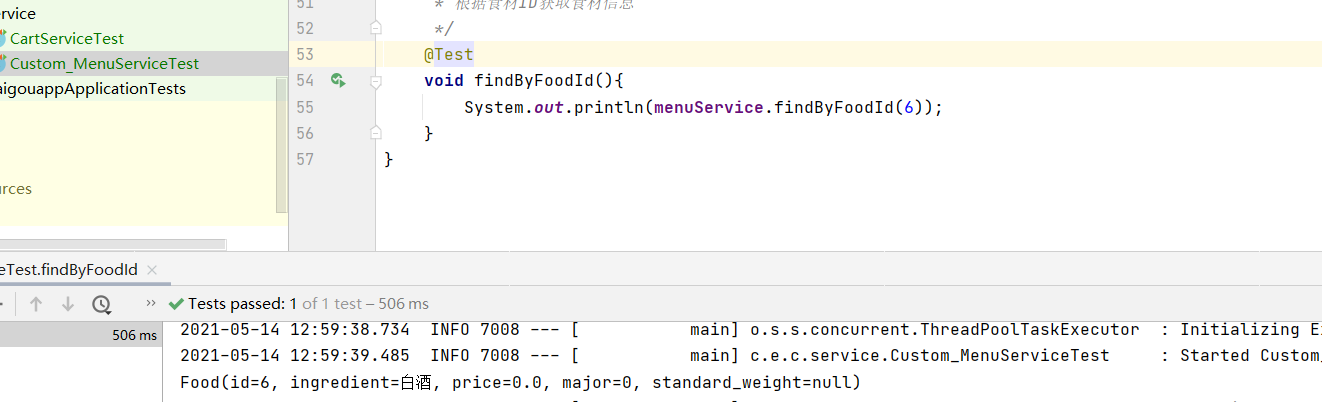
10.根据菜谱的ID查询菜谱的食材列表



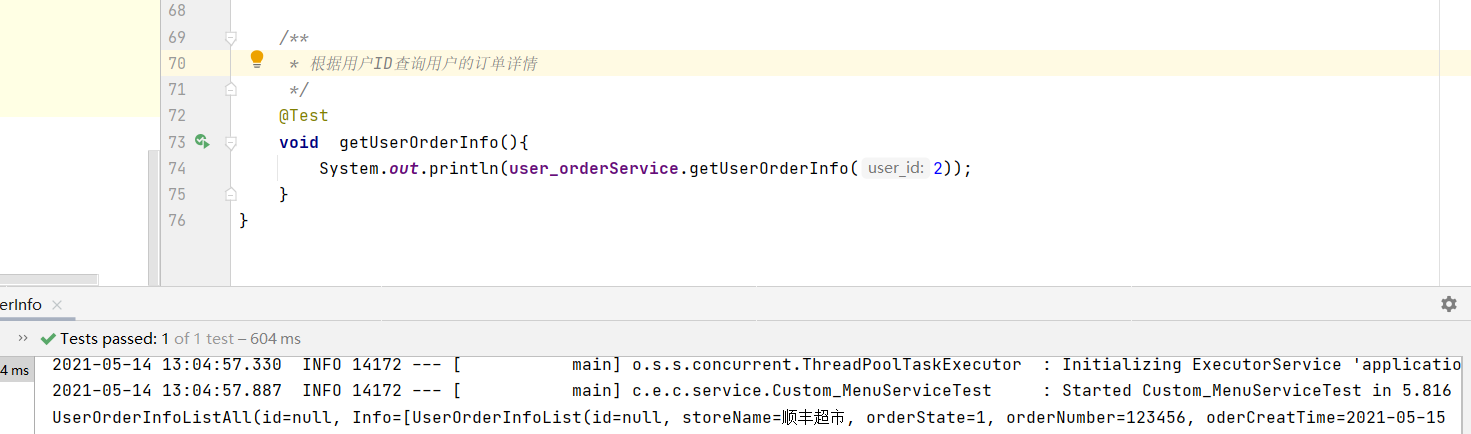
11.根据菜谱的ID查询所需食材的所有信息



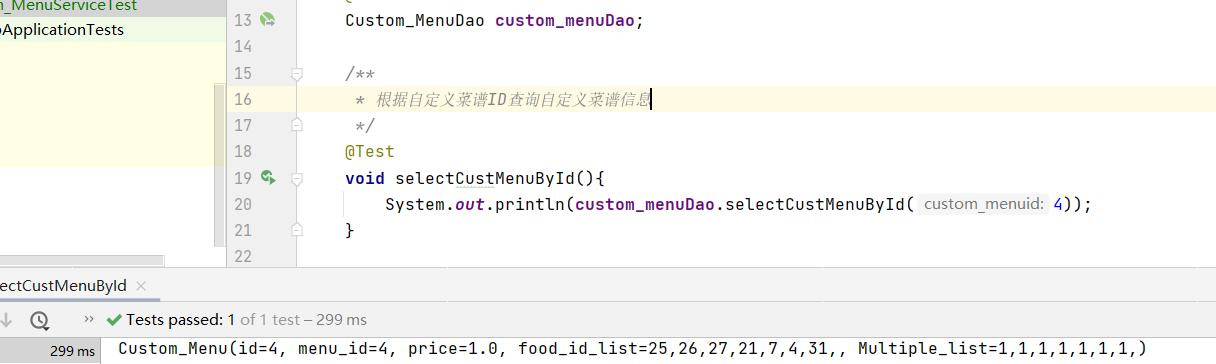
12.根据食材的ID获取食材的信息



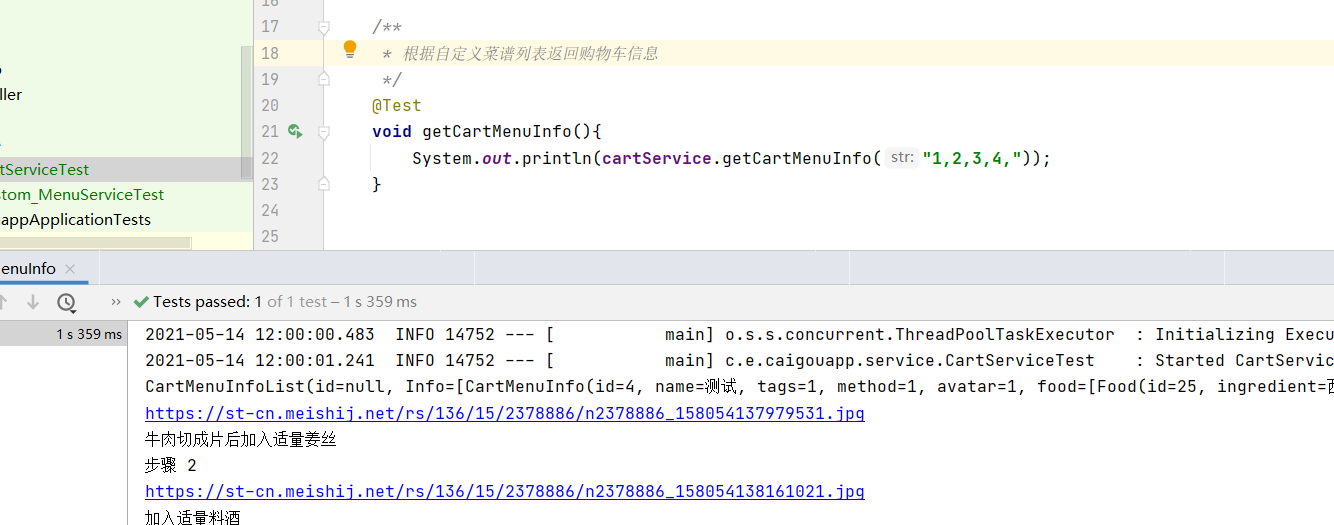
13根据用户ID查询订单详情



14.根据自定义菜谱ID查询该菜谱的信息



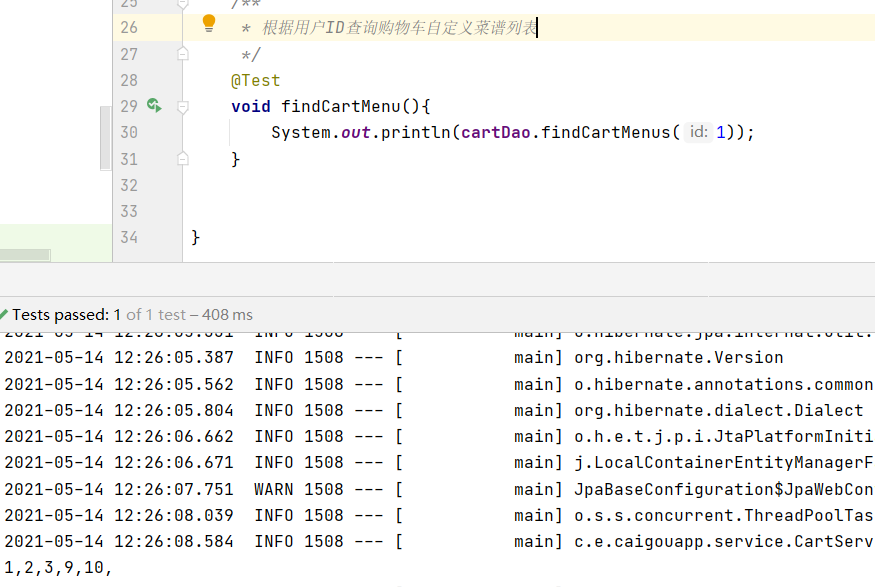
15获取购物车的菜谱信息



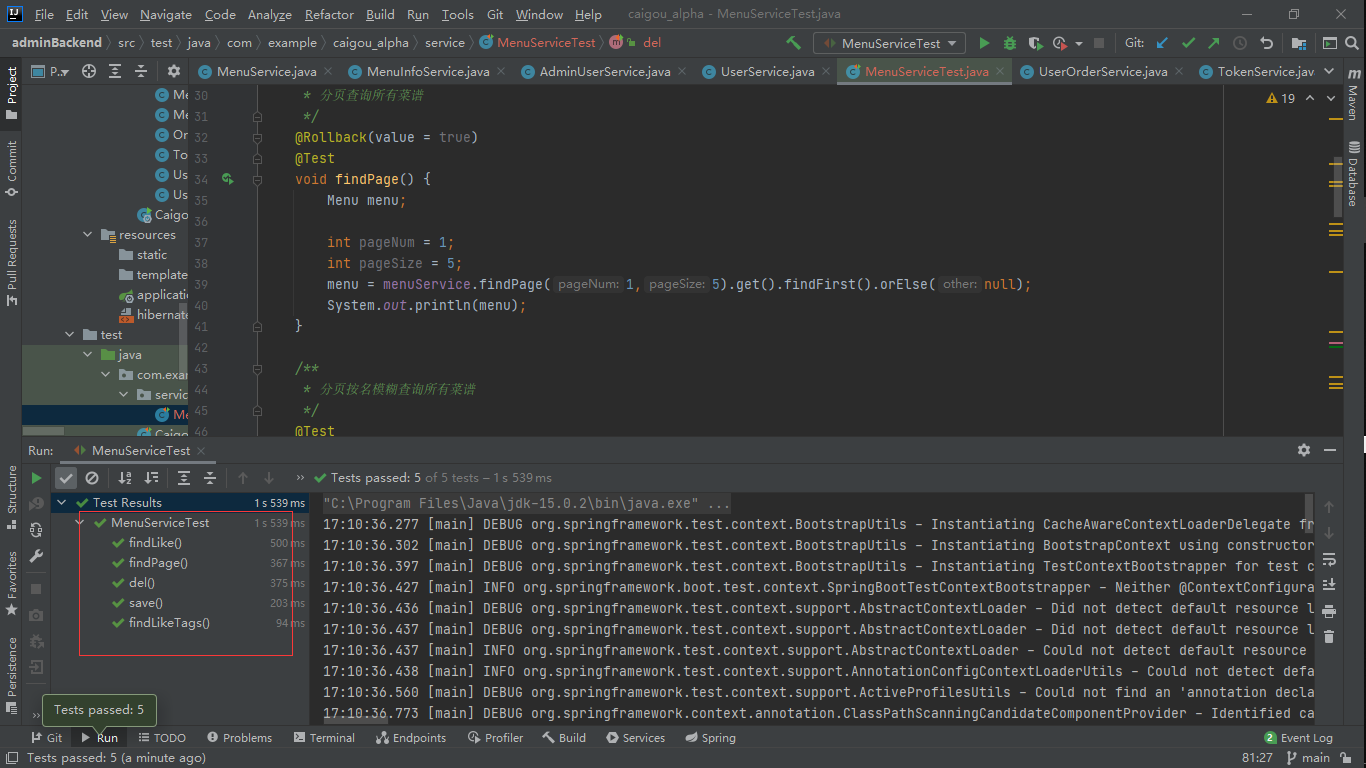
16.获取购物车信息



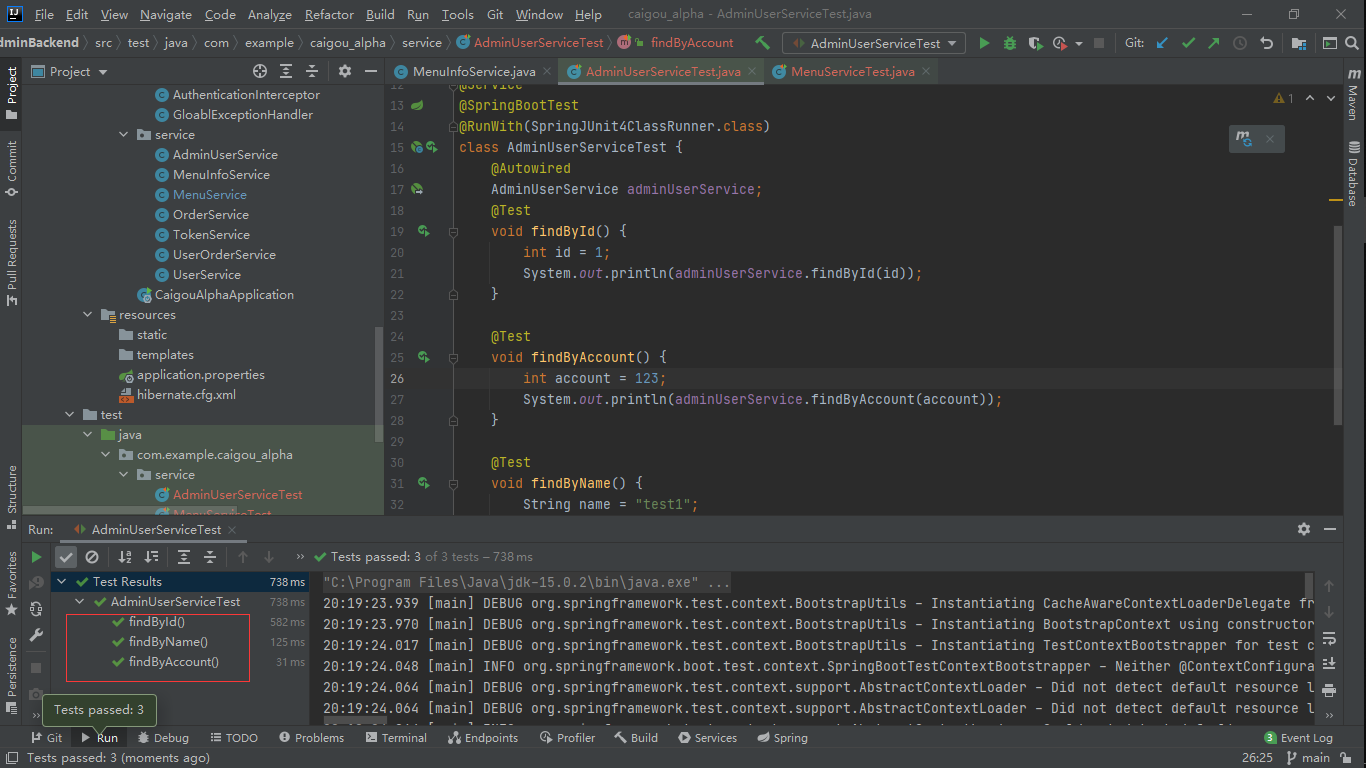
17.获取购物车菜谱ID列表



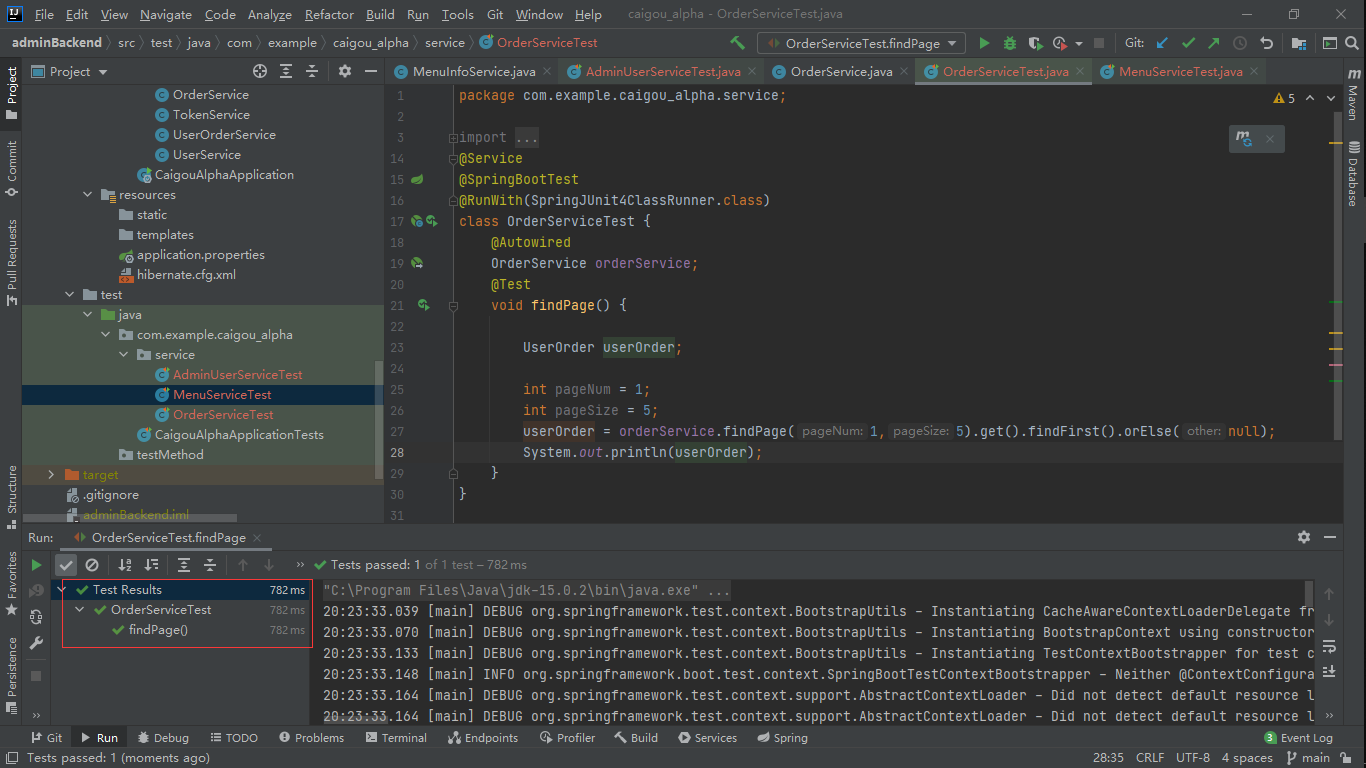
1. 菜谱相关后台操作(分页查询所有、分页模糊查询名字、保存、更新等）



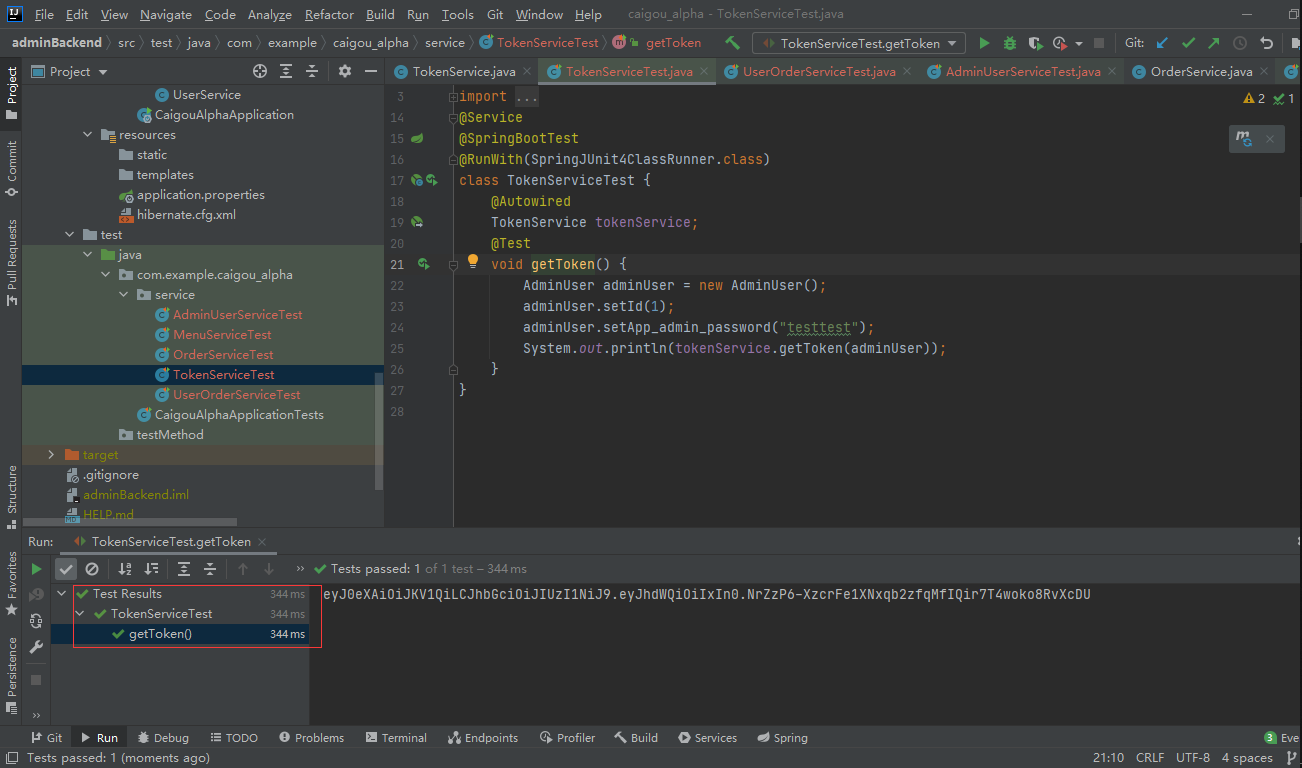
1. 管理员用户操作部分



1. 用户订单分页查询

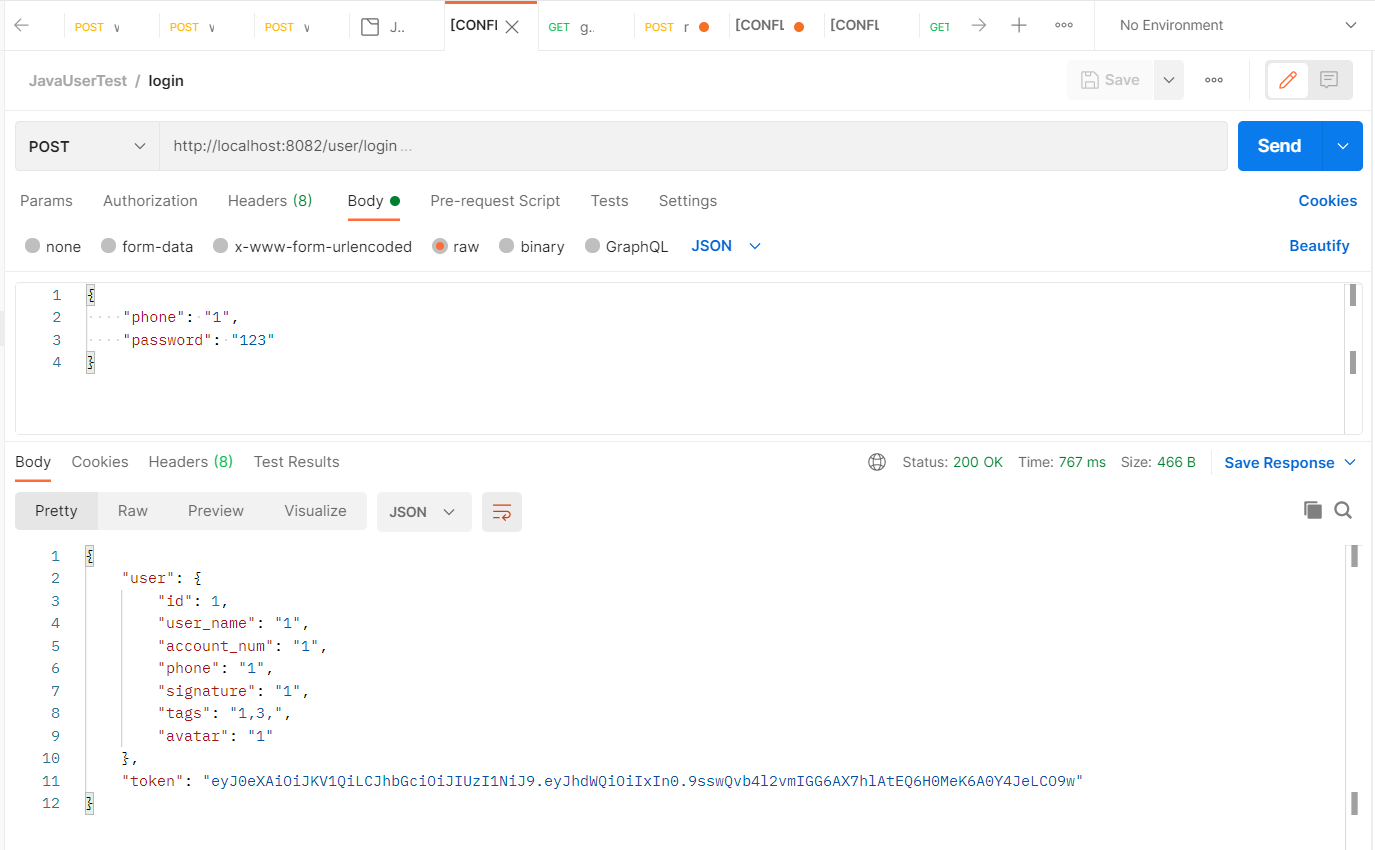
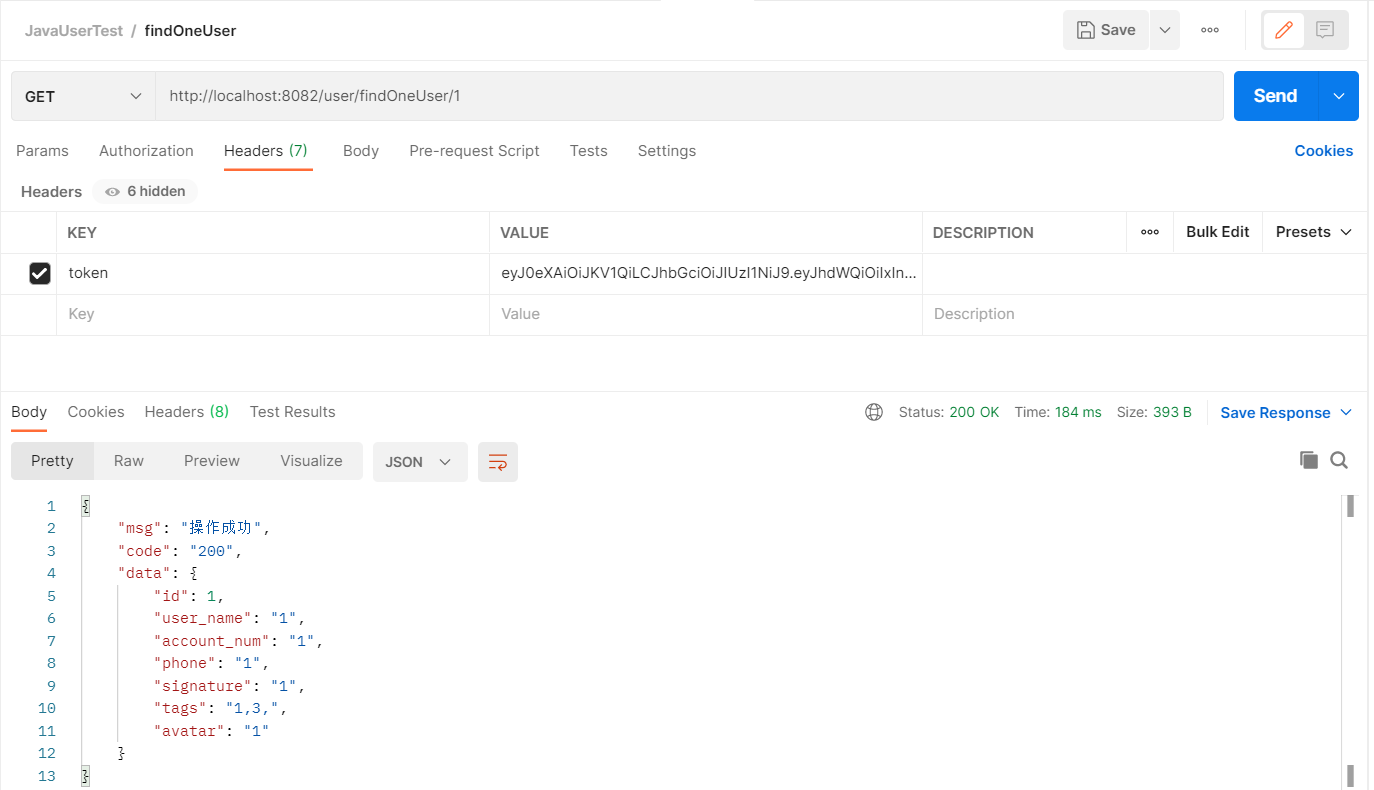


1. 管理员用户登录



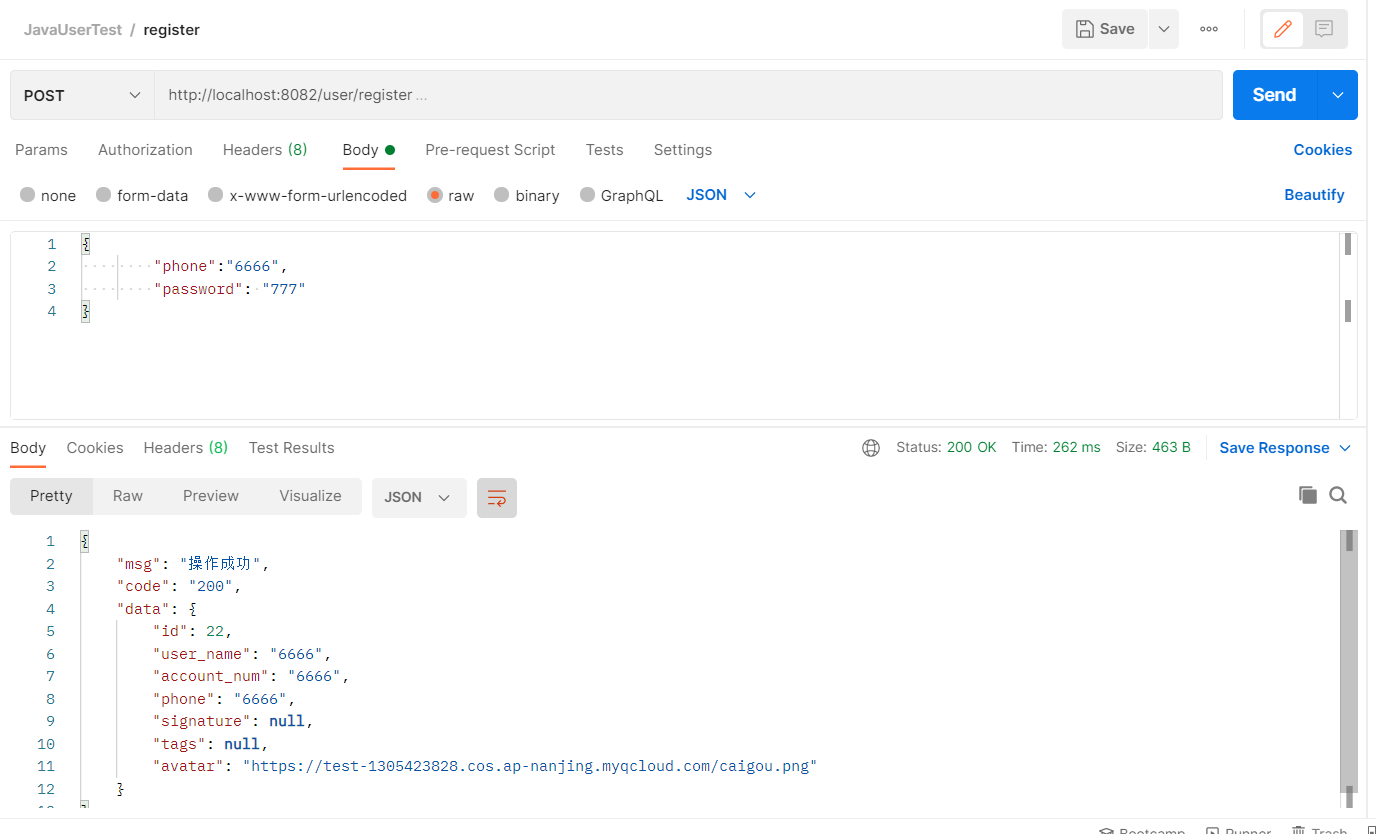
## 2.2 API测试

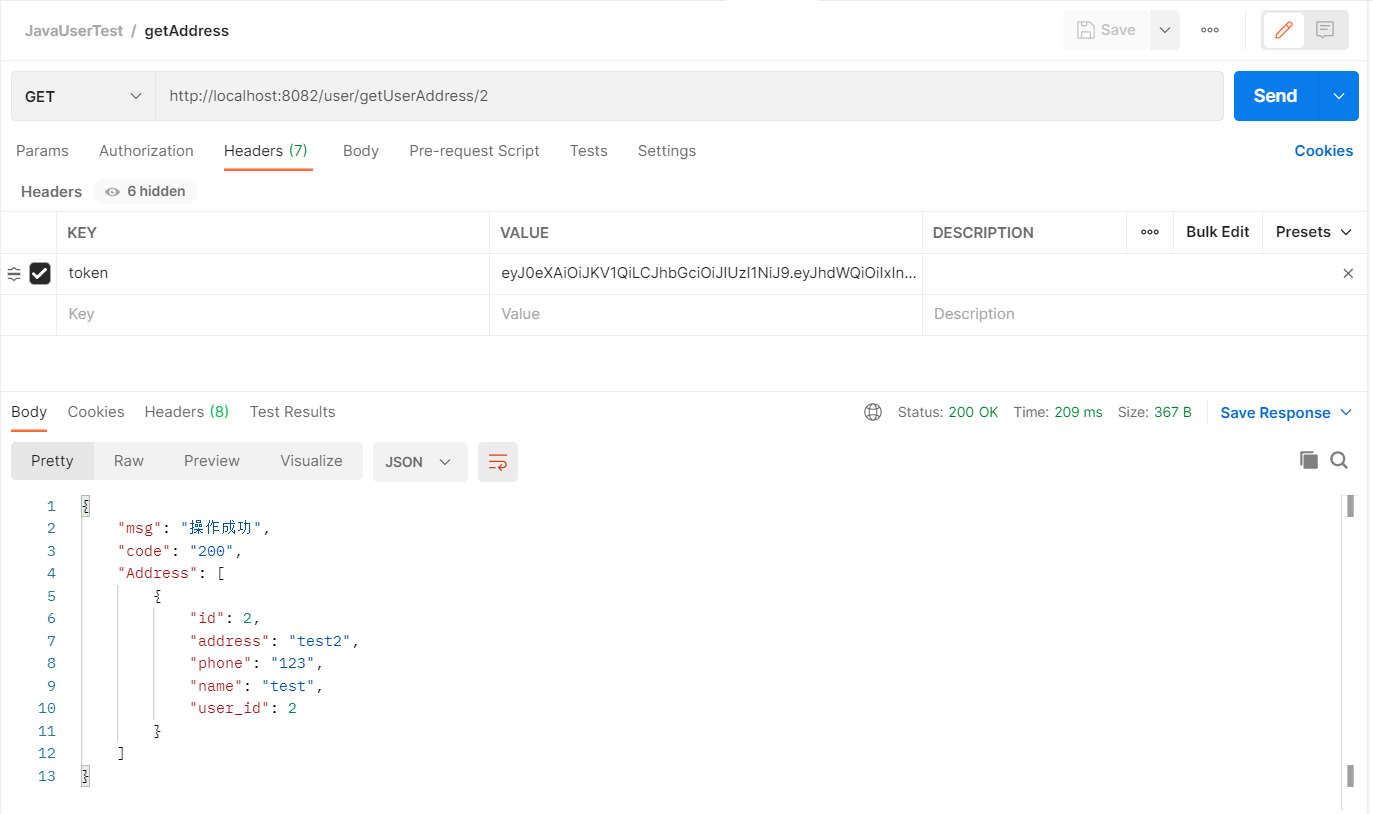
1.用户登录



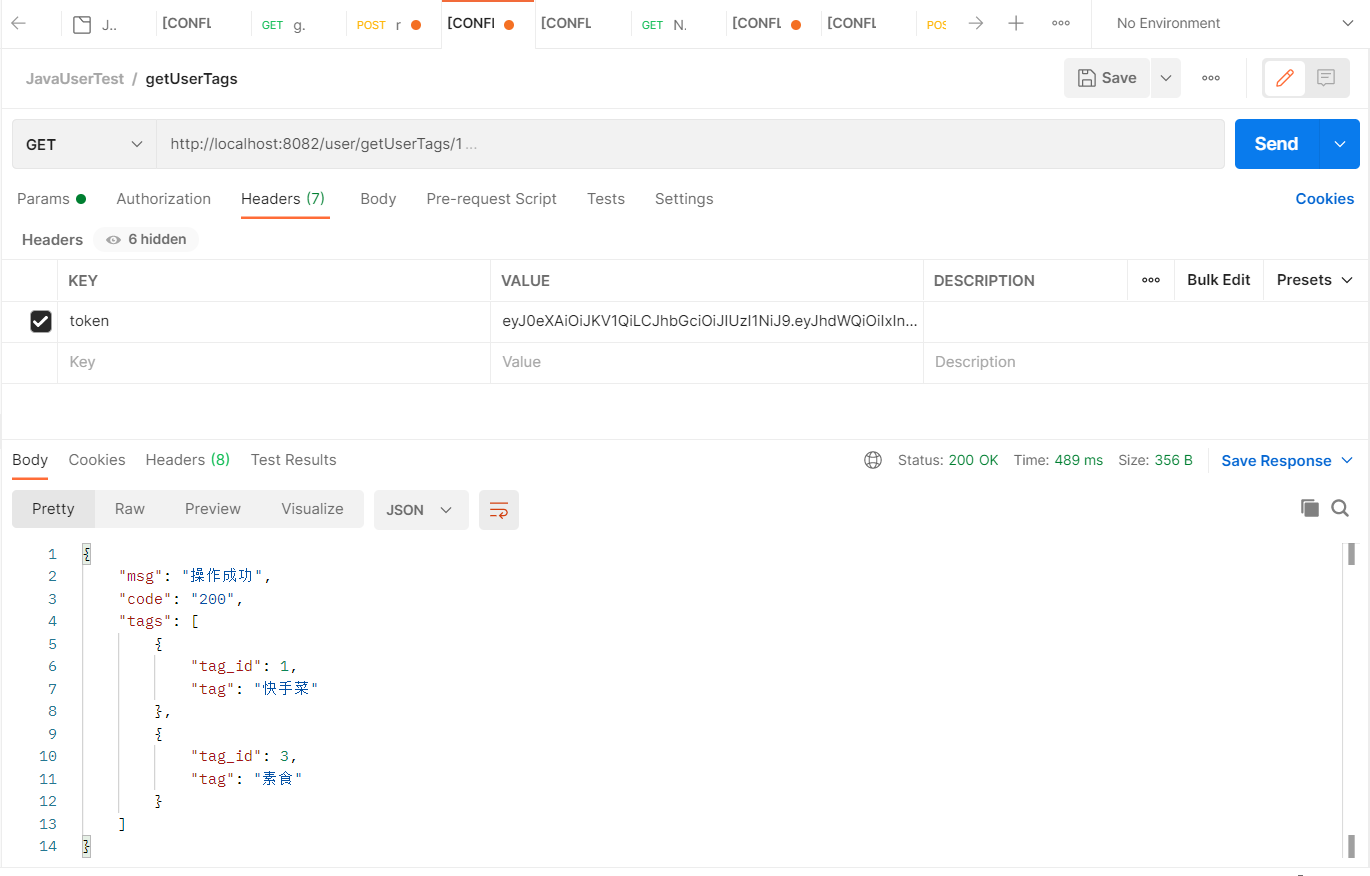
2.用户注册

3.根据account查询用户信息

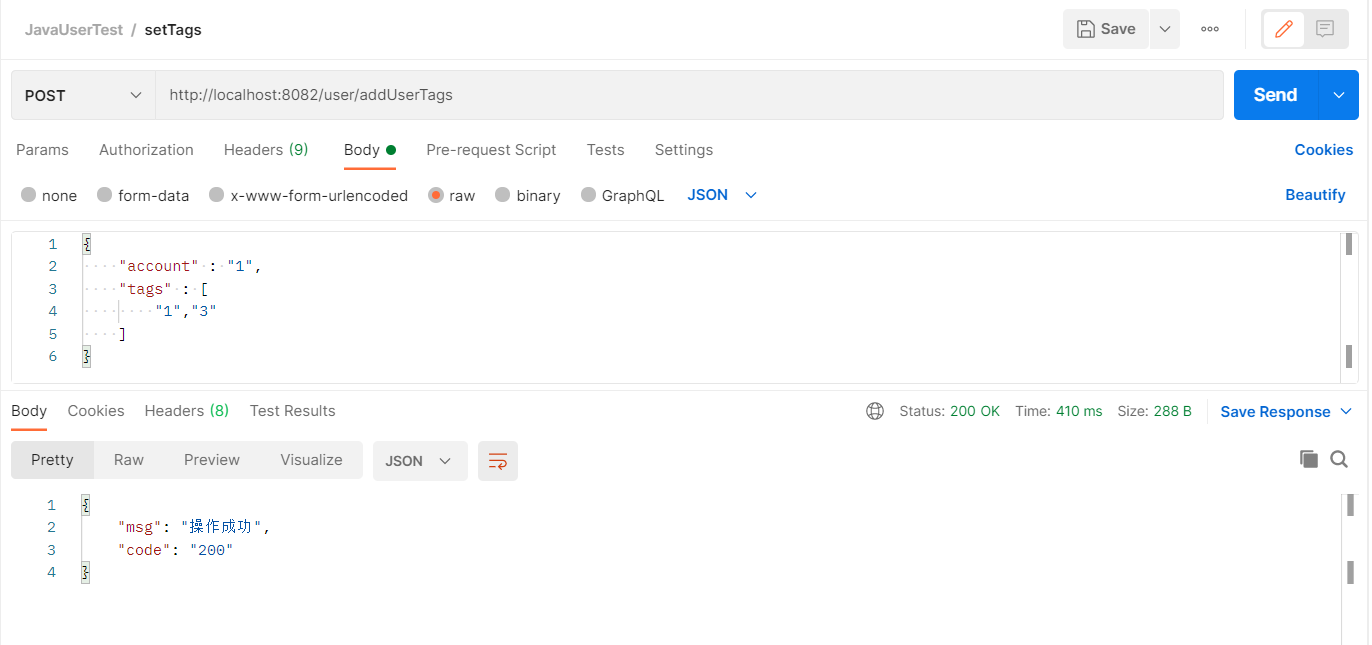
4.根据account查询用户收货地址

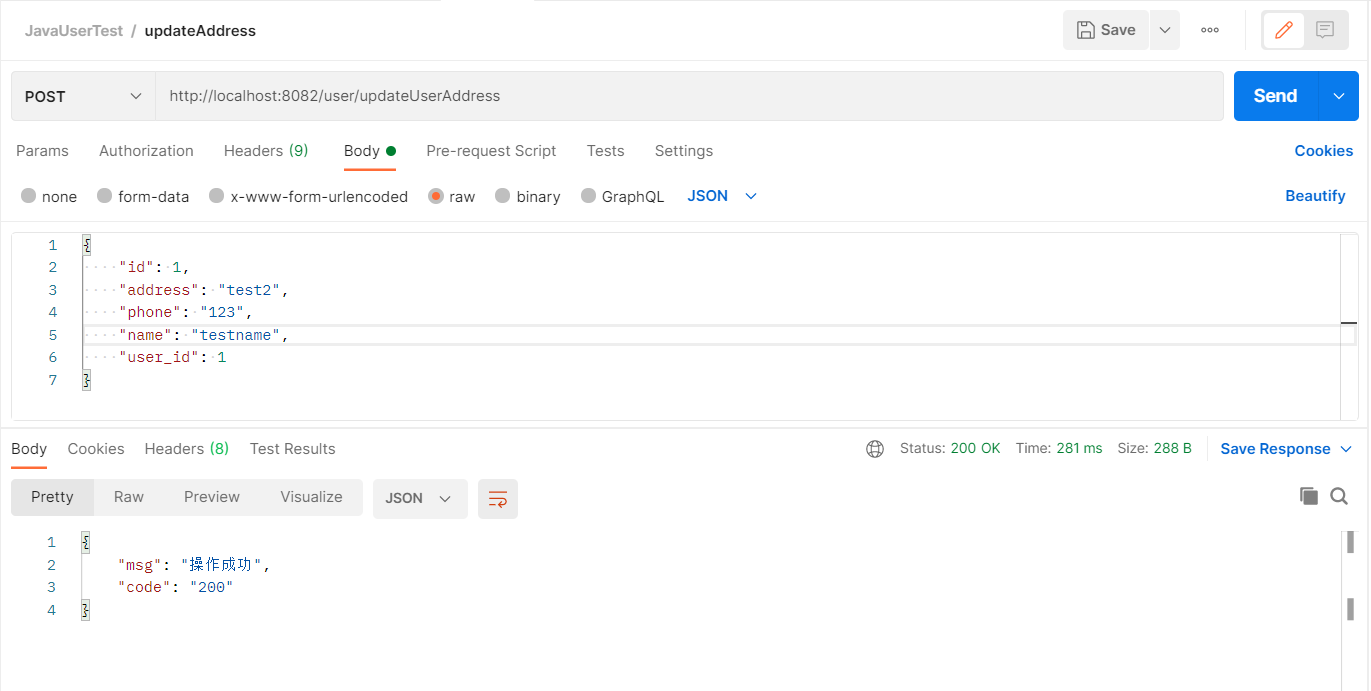


5.根据account查询用户喜爱标签

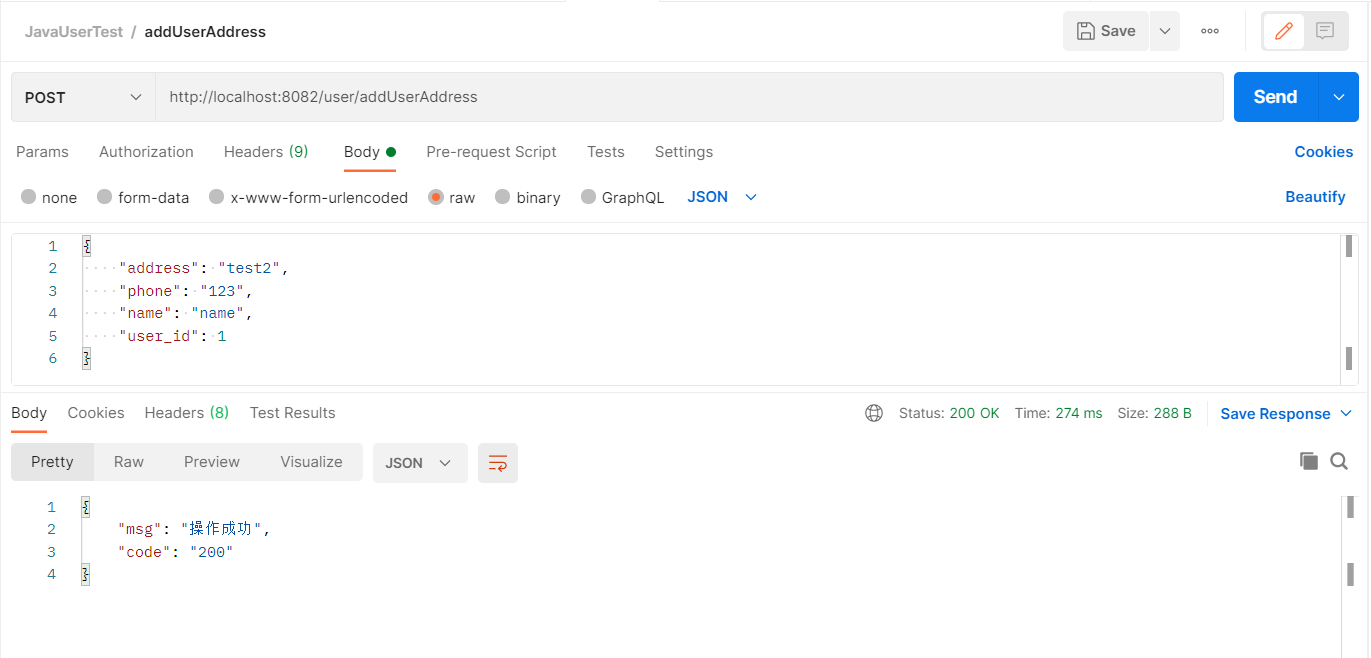


6.用户添加喜爱标签



7.用户更新收货地址

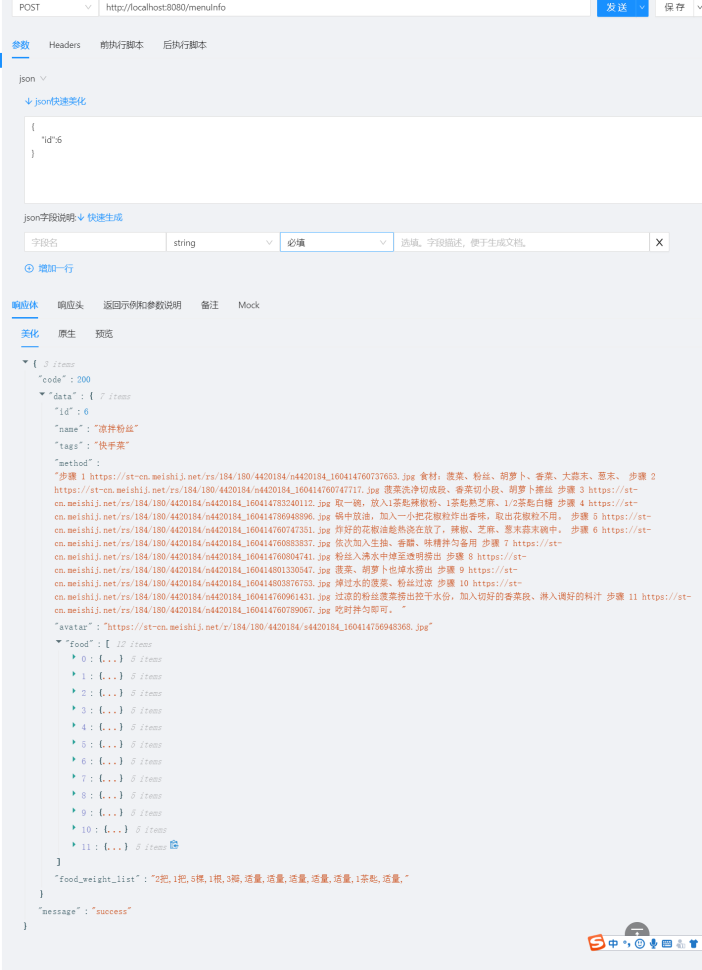
8.用户添加一个地址



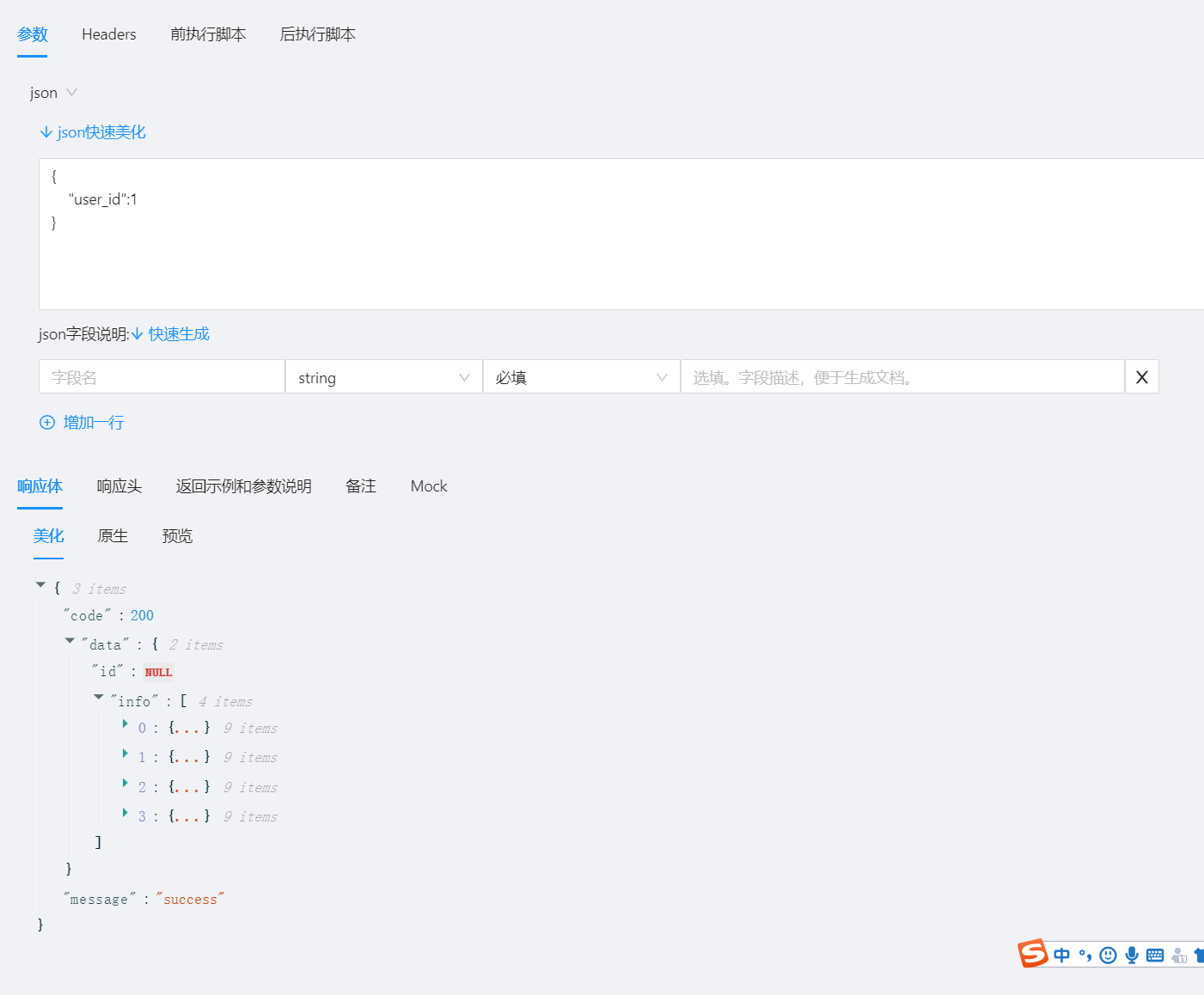
9.根据菜谱的名字和标签进行模糊查询



10.根据菜谱ID查询菜谱的详细信息



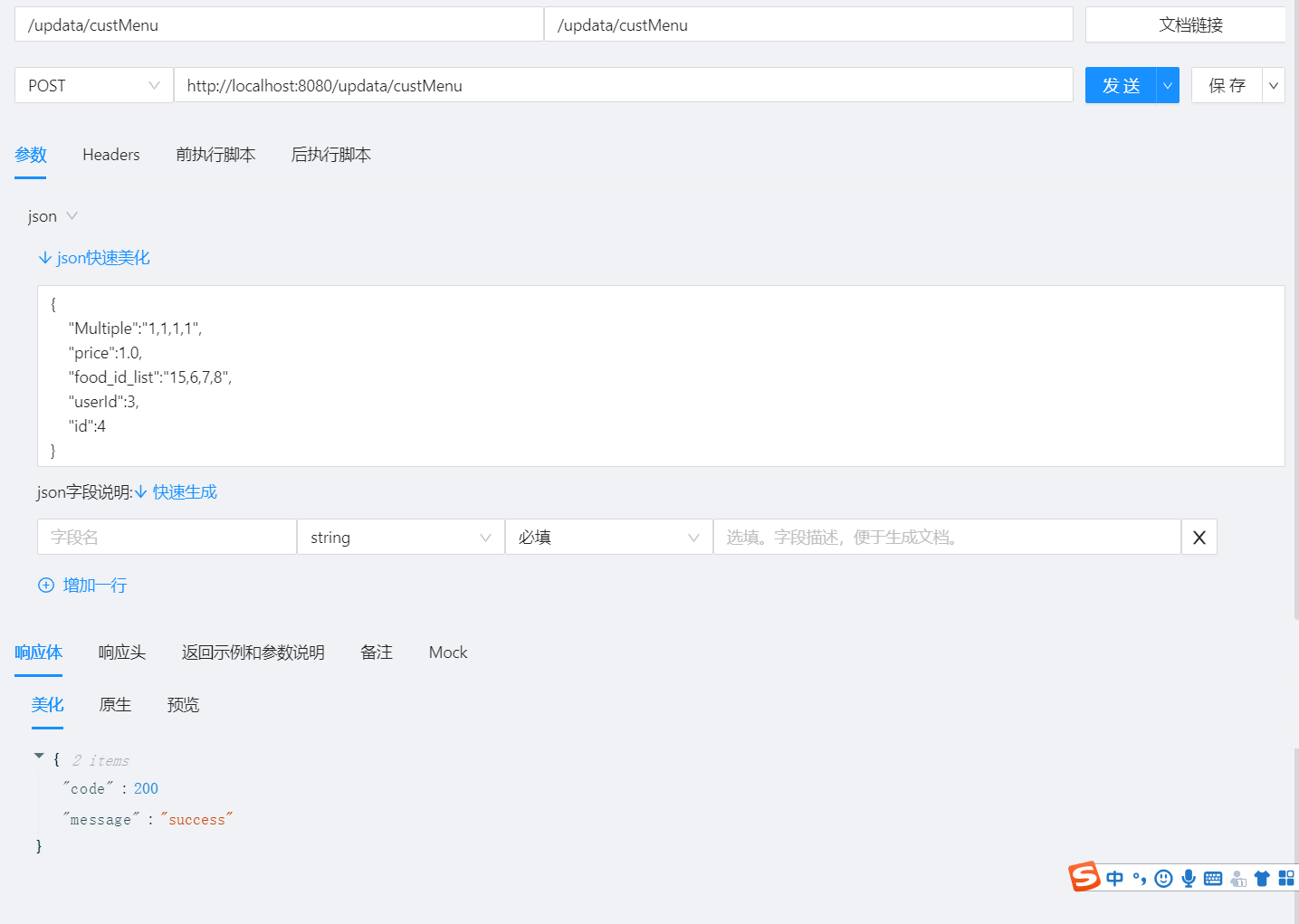
11.查看购物车列表信息



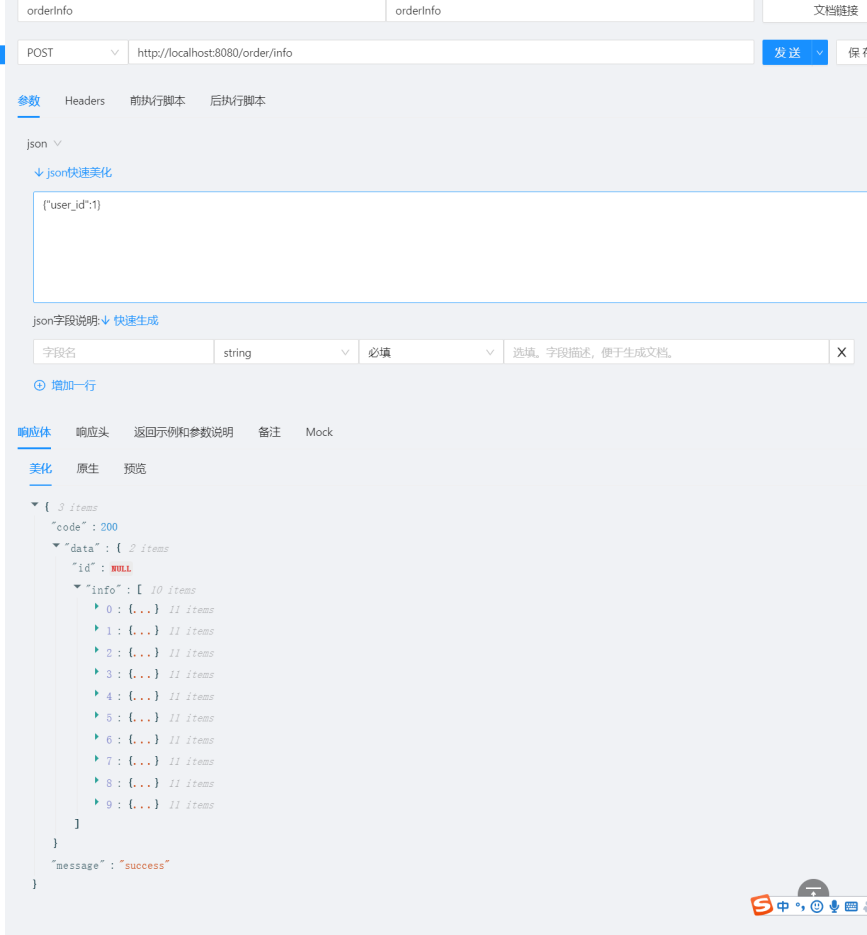
12生成订单



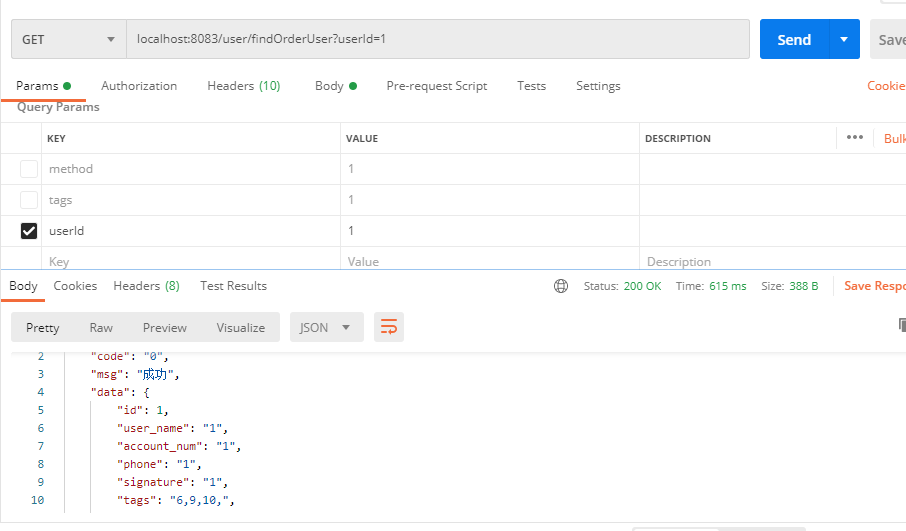
13更新购物车和自定义菜谱



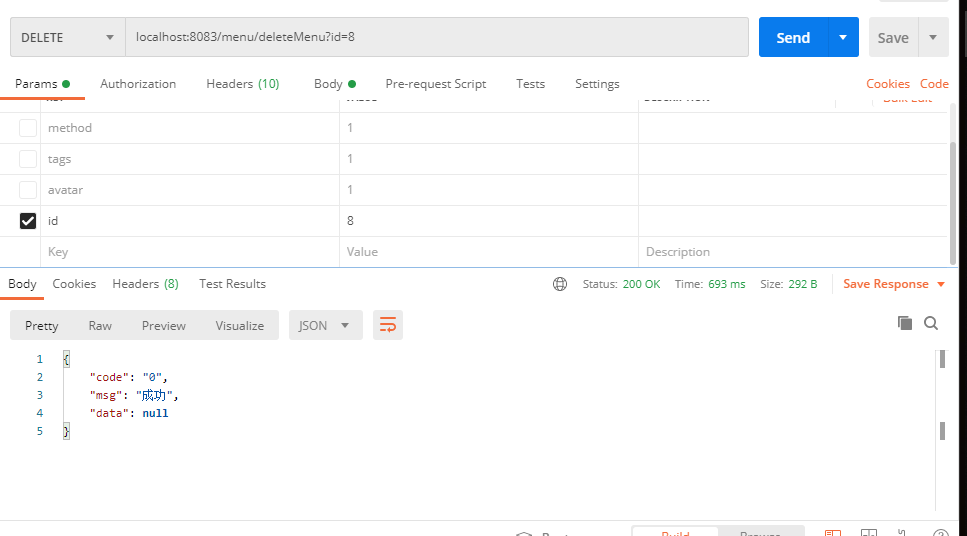
14.查看用户的订单信息

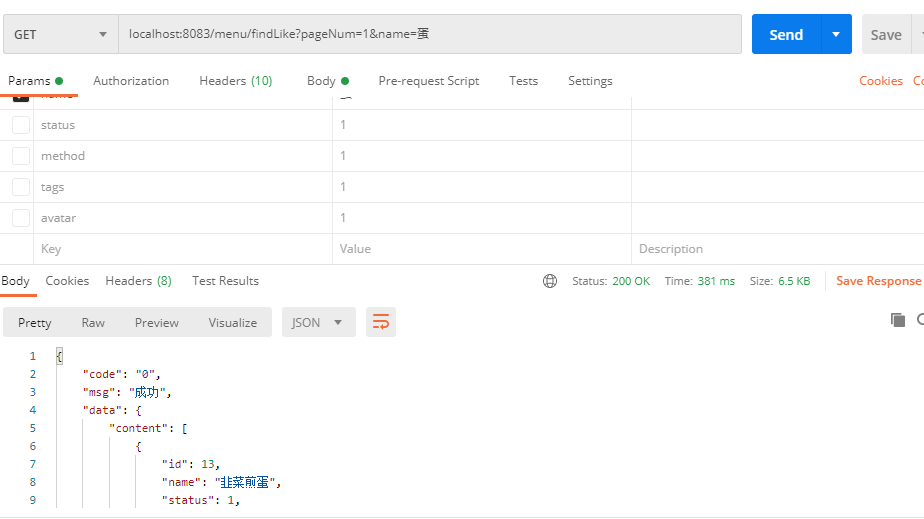


15.根据订单中的ID查到客户信息

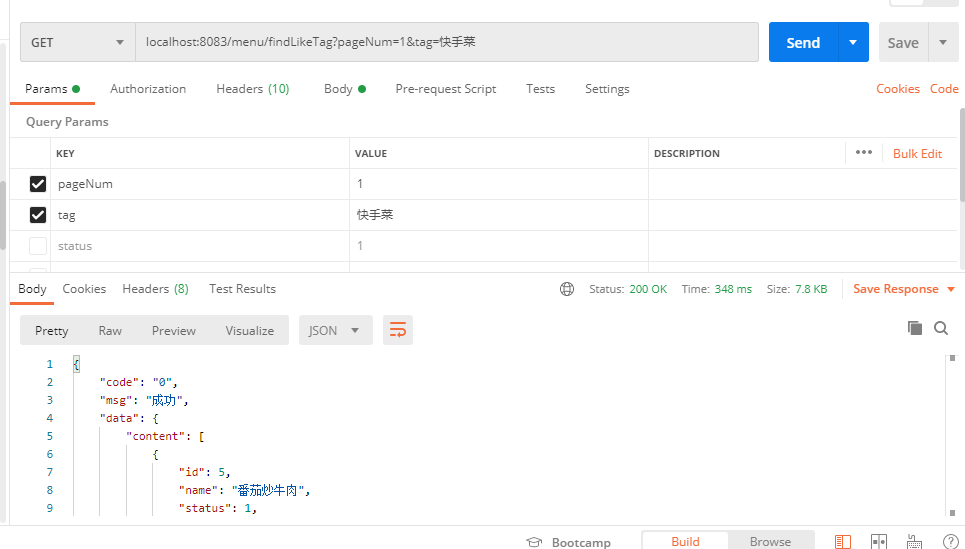


16、根据菜谱ID删除菜谱

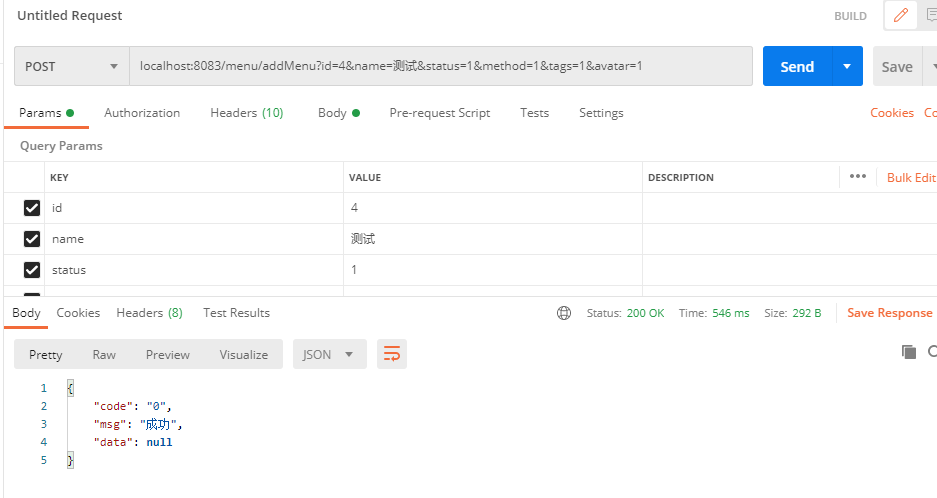


17、根据名字模糊分页查询菜谱

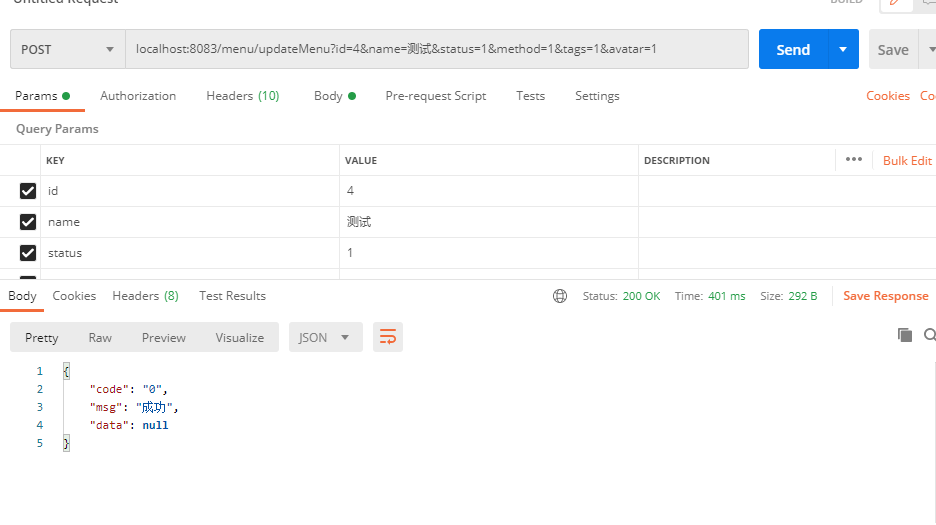
18、根据标签模糊分页查询菜谱



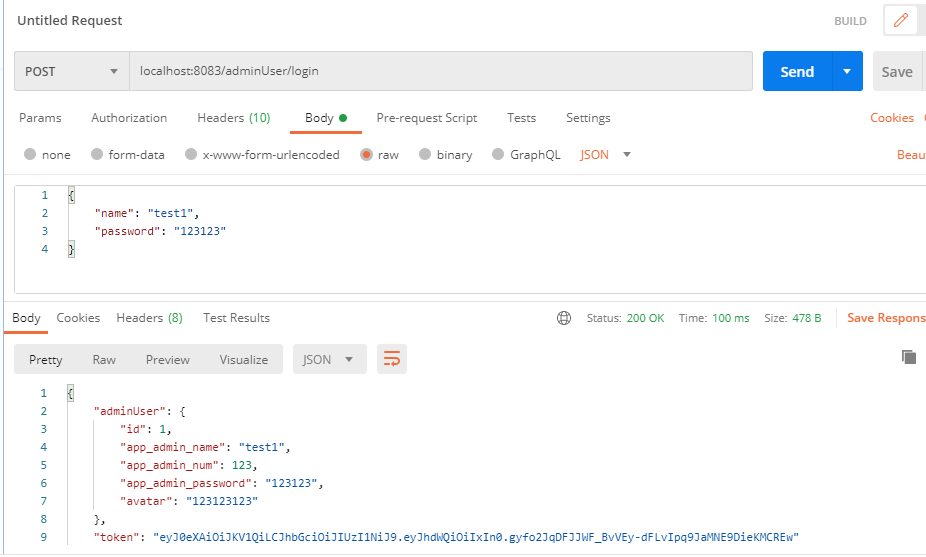
19、增加菜谱



20、修改菜谱



21.后台管理员登陆

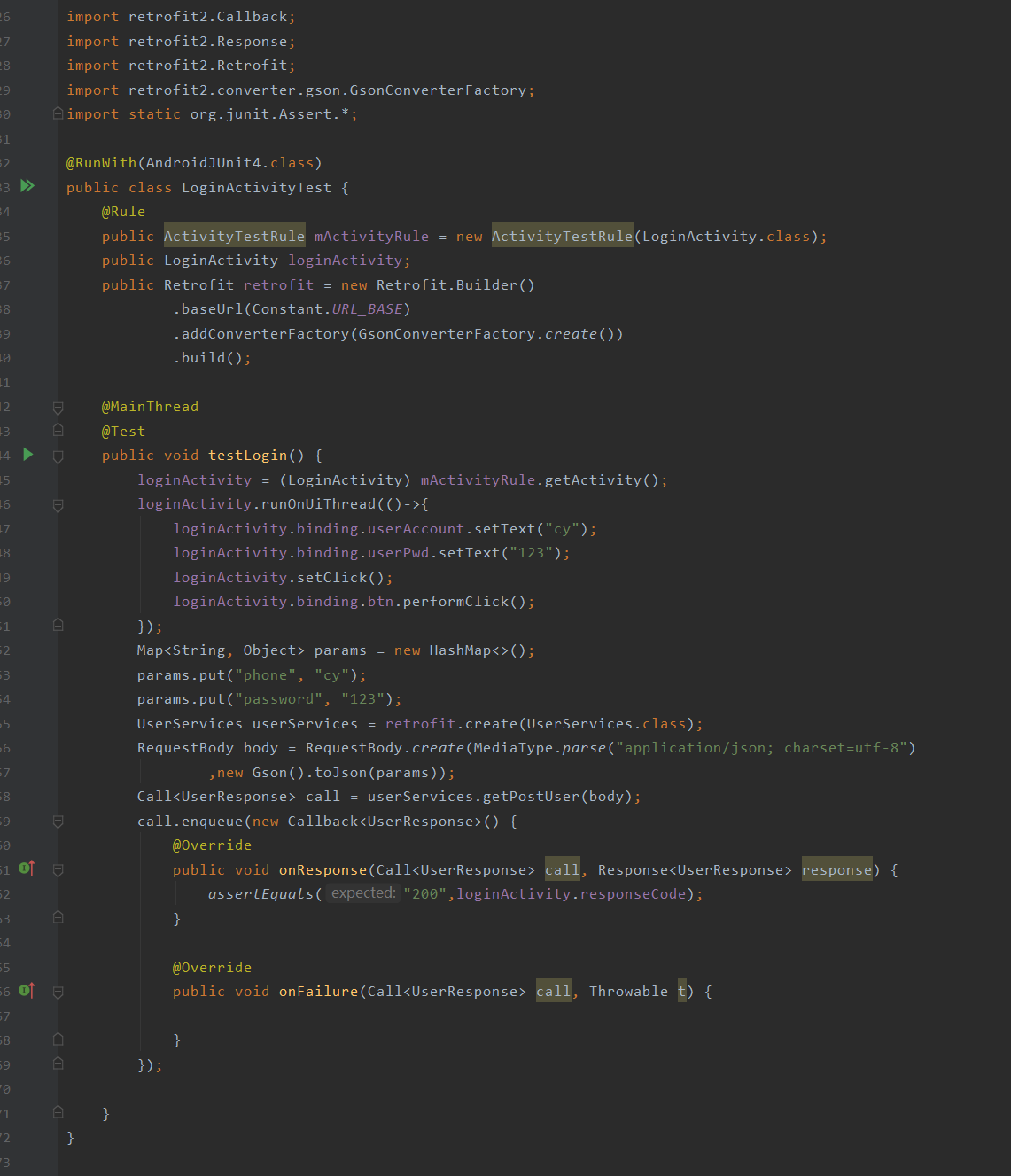


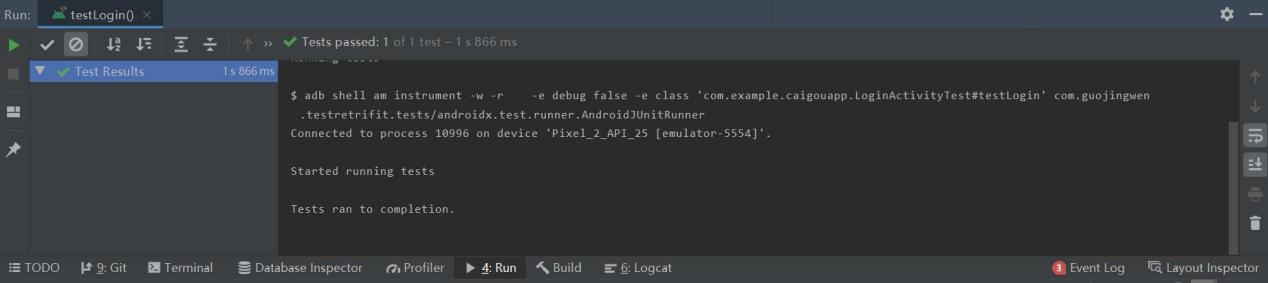
# 3 前端测试

# 3.1Android测试

#### 3.1.1 Activity单元测试

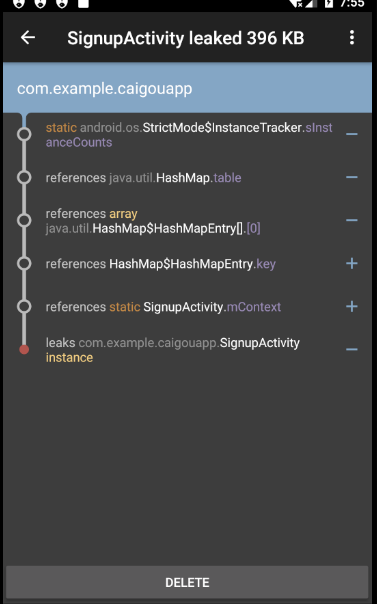
安卓单元测试一般不适合测试与系统有复杂交互的UI。这里使用JUNIT4框架测试，测试用例使用登录的网络请求为测试过程：



测试结果：

#### **3.1.2 性能测试**

Android Studio 自带的 Profile 工具、MAT(Memory Analyzer Tool)、以及LeakCanary都可以对安卓的性能泄露进行检测，这里使用LeakCanary工具对app进行检测。



这里发现主要是SignupActivity的静态对象mContext持有Activity的引用，导致Activity资源无法被回收。在onDestroy()方法中将mContext置空，解除mContext持有的activity引用。另一个是登录和注册界面的来回调用导致泄露，考虑使用SingleTask的启动模式来保持对activity的单例使用以保证资源不被重复占用。