

# 二手交易平台

# ( 线上市场 ) 测试用例

文件状态: 正式发布

文件标识: 测试用例

当前版本: v2.1.0

作者:

姓名: 张子文、崔辰垚...

学号: 2018011358、201...

团队: 啊对对队

完成日期: 2021/11/11

版本更新信息:

v2.1.0 记录了对所有测试项目和测试用例的详细信息

## 1 引言

### 1.1 使用人员:

二手商品买家、二手商品卖家、系统管理员

### 1.2 编写目的:

记录所有测试用例测试的过程、目的和结果

### 1.3 背景:

a. 名称: 二手交易平台(线上市场)

b. 本项目由啊对对队提出并由俞昊洋担任项目经理, 由俞昊洋、宋雪霞担任配置组, 由吴昊、徐伟、崔辰垚、张弈凡、刘科宏、张子文、刘晨阳担任开发组, 由俞昊洋、宋雪霞、吴昊、崔辰垚担任管理组, 周俊毅、吴浩涵、汪筱青担任质保组和用户组, 并在个人笔记本上进行运维。旨在为有二手商品买卖需求的用户提供一个可供交易的平台。

### 1.4 定义与缩写

术语: 运维

解释:

指互联网运维, 通常属于技术部门, 与研发、测试、系统管理同为互联网产品技术支撑的4大部门, 这个划分在国内和国外以及大小公司间都会多少有一些不同。

一个互联网产品的生成一般经历的过程是: 产品经理、需求分析、研发部门开发、测试部门测试、运维部门部署发布以及长期的运行维护。

运维, 本质上是对网络、服务器、服务的生命周期各个阶段的运营与维护, 在成本、稳定性、效率上达成一致可接受的状态。

### 1.5 参考资料

•

## 2 测试项目

**测试项目名称:**

后端代码接口测试

测试人员: 张子文

测试时间: 2021/11/10

**测试内容:**

Controller层的接口正确性测试

用户功能测试功能如下:

1. 用户登录
2. 用户注册
3. 获取待验证的注册用户列表
4. 删除待验证表的允许登录的注册用户
5. 允许登录的注册用户信息插入用户表
6. 用户更改个人信息
7. 获取注册商家的用户信息
8. 获取店铺等级
9. 充值

购物车功能测试功能如下:

1. 添加购物车
2. 从购物车删除商品
3. 更改加入购物车的商品数量
4. 用户已经下单的商品
5. 商家确认发货
6. 商家退款订单
7. 用户确认收货表
8. 用户历史订单

评论功能测试功能如下:

1. 添加评价
2. 获取商品所有评论

商品功能测试功能如下:

1. 管理员获取待审核的商品
2. 允许商品上架
3. 获取所有允许上架的商品展示在主页面
4. 获取所有商品
5. 商品详情页
6. 商家下架商品
7. 允许商品上架
8. 修改商品数量
9. 修改商品价格

**测试环境与系统配置:**

postman、JDK1.8、eclipse、MySQL

**测试输入数据:**

根据对应功能类所规定需要输入的参数进行传参, 如用户登录的方法要求的传参为用户id和密码, 测试输入的参数即为id和word, 分别测试了普通用户登录、管理员登陆以及登陆失败三种场景, 以此类推。

测试次数: 30

**预期结果:**

postman可以输出请求的对应数据, 如果测试的功能不返回参数, eclipse可以返回对应功能类中的打印信息。

**测试过程：**

在postman中创建对应的测试功能的文件，在本地环境中运行后端代码、MySQL数据库，在postman中输入本地localhost地址和测试功能的方法名，根据对应功能设计传参，运行查看结果。

**测试结果：**

所有输出结果符合预期，测试通过。

**测试结论：**

接口测试通过，后端代码接口调用相关功能通过。

**实现限制：**

只能在本地环境中进行单个接口的测试。

**备注：****测试项目名称：**

评论功能测试

**测试人员：**张子文

**测试时间：**2021/11/4

**测试内容：**

评论功能测试功能如下：

1. 添加评价
2. 获取商品所有评论

**测试环境与系统配置：**

JDK1.8、eclipse、MySQL

**测试输入数据：**

1. 添加评论功能  
用户id、用户密码、商品id、评论内容、评论状态
2. 获取商品评论  
商品id

**测试次数：**4

**预期结果：**

1. 添加评论功能  
第一次向没有对应用户评论的商品添加评论，结果添加成功；  
第二次向对应用户已经评论过的商品添加评论，结果添加失败。
2. 获取商品评论  
第一次获取数据库评论表中第一个商品评论，返回评论结果成功；

第二次获取一个数据库中不存在评论的商品，返回结果为null。
<b>测试过程：</b> 连接花生壳服务器，并将所有请求参数打包上传，通过服务器连接后台服务和数据库，并将结果通过服务器返回至本地。
<b>测试结果：</b> 与预期结果相符。
<b>测试结论：</b> 测试结果通过。
<b>实现限制：</b> 数据库建表时，对应的列元素属性定义为Varchar（255），无法进行数组越界的情况测试。
<b>备注：</b>

<b>测试项目名称：</b> 商品测试
<b>测试人员：</b> 周俊毅
<b>测试时间：</b> 2021/11/7
<b>测试内容：</b>
<b>测试环境与系统配置：</b>
<b>测试输入数据：</b>

函数名称	应用场景	测试用例
allowgood(String id,String word,String goodid)	允许商品上架	id=1 word=123 goodid=1
		id=1 word=999 (255位) goodid=1
		id=999 (255位) word=1 goodid=1
		id= word=123 goodid=1
		id=1 word= goodid=1
		id=1 word=1 goodid=

函数名称	应用场景	测试用例
goodinfo(String goodid)	插入商品id到列表	goodid=1
		goodid=
		goodid=99999 (255位)
		goodid=99999 (256位)

测试次数: 2

#### 预期结果:

测试用例	预期结果	实际结果	过关失败	
id=1 word=123	返回商品列表	返回商品列表	通过	
id= word=123	无返回商品列表	无返回商品列表	通过	
id=1 word=	无返回商品列表	无返回商品列表	通过	
id=-1 word=123	无返回商品列表	无返回商品列表	通过	
id=1 word=999999 (256位)	无返回商品列表	无返回商品列表	通过	
id=-1 word=9999 (256位)	无返回商品列表	无返回商品列表	通过	
id=1 word=999999 (255位)	返回商品列表	返回商品列表	通过	
应用场景	测试用例	预期结果	实际结果	过关失败
允许商品上架	id=1 word=123 goodid=1	商品被允许上架	商品被允许上架	通过
	id=1 word=999 (255位) goodid=1	商品被允许上架	商品被允许上架	通过
	id=999 (255位) word=1 goodid=1	商品被允许上架	商品被允许上架	通过
	id= word=123 goodid=1	商品不允许上架	商品不允许上架	通过
	id=1 word= goodid=1	商品不允许上架	商品不允许上架	通过
	id=1 word=1 goodid=	商品不允许上架	商品允许上架	失败

#### 测试过程:

使用junit编程, 进行函数测试

#### 测试结果:

如上图

#### 测试结论:

大部分函数功能正常, 存在部分函数逻辑不够完善。

实现限制：

--

备注：

--

测试项目名称：

用户登录
------

测试人员：

崔辰垚

测试时间：

2021/11/12

测试内容：

输入用户名和密码，若输入正确，则操作成功，允许登录。

测试环境与系统配置：

VSCode、IDEA、JDK8、MySQL

测试输入数据：

1. 用户名及密码均正确
2. 用户名和密码分别为空
3. 用户名和密码中分别有非法字符
4. 用户名正确，密码错误
5. 用户名错误，密码正确

测试次数：

5

预期结果：

测试1操作成功，登录进入系统  
除测试1之外，其余项目均为操作失败，返回“用户名或密码错误”信息。

测试过程：

在数据库中书写测试用例，通过使用IDEA向服务器中传送数据库中的数据，并在代码中将返回语句输出至“console”中，通过观察console中的返回信息，确认测试结果。

测试结果：

除第一次操作成功外，其余4次实验均返回“用户名或密码错误”信息。

测试结论：

此次测验结果良好，验证通过。

实现限制：

--

备注：

--

测试项目名称：

用户注册
------

测试人员：

崔辰垚

测试时间：

2021/11/12

测试内容：

用户通过提交用户名、密码、性别等基本信息后，点击提交按钮，将信息提交至数据库中，并返回成功消息。

测试环境与系统配置：

VSCode、IDEA、JDK8、MySQL
------------------------

测试输入数据：

1. 用户正确提交所有基本信息。
2. 所有的信息分别为空或其中的几项为空。
3. 信息中出现非法字符。
4. 提供的信息超出范围。

测试次数：

15

预期结果：

所有基本信息提交正确时，返回注册成功的信息。  
在提交信息时出现为空项时，判断该项是否可以为空，若不可为空，则返回注册失败，若可以为空，则注册成功。  
在提交信息出现非法字符时，返回注册失败  
提交信息超出范围时，返回注册失败。

测试过程：

在数据库中书写测试用例，通过使用IDEA向服务器中传送数据库中的数据，并在代码中将返回语句输出至“console”中，通过观察console中的返回信息，确认测试结果。

测试结果：

第1次测试为正确信息，注册成功。  
第2-5次测试，空项均为必填项，均注册失败。  
第6-10次测试，空项均为可填项，注册成功。  
第11-15次测试，不同的项中超出范围时，均注册失败。

测试结论：

此次测验结果良好，验证通过。

实现限制：

--

备注：

--

测试项目名称：

购物车功能测试
---------

测试人员：刘晨阳

测试时间：2021/11/11

测试内容：

主要测试了Controller层购物车的相关功能，包括：

1. 向购物车添加商品
2. 用户购物车的所有商品
3. 从购物车删除商品
4. 用户付款的商品
5. 用户已经下单的商品
6. 退款
7. 商家待发货
8. 商家确认发货
9. 商家退款订单
10. 商家拒绝退款
11. 同意退款
12. 用户确认收货表
13. 用户确认收货改变商品状态为5
14. 用户历史订单

测试环境与系统配置：

Junit、JDK1.8、eclipse、MySql
----------------------------

测试输入数据：

--



函数名称	应用场景
addgood ()	在购物车里加商品
List<car> car ()	查看购物车中的商品
pay ()	付款商品
测试用例	
id="" word="" goodid=""	
id="admin" word="123" goodid=""	
id="admin" word="" goodid="shopper"	
id="admin" word="123" goodid="shopper"	
测试用例	
id="" word=""	
id="admin" word=""	
id="admin" word="123"	
测试用例	
id="" word="" goodid="" address="" phone="" uname="" money=""	
全部不为空时	
有一个为空时	
测试次数: <input type="text" value="14"/>	
预期结果:	
<div></div>	
测试过程:	
修改参数, 用Junit进行测试	
测试结果:	
<div></div>	

Finished after 6.681 seconds

Runs: 15/15   Errors: 0   Failures: 0

▼ CarControllerTest [Runner: JUnit 5] (3.423 s)

- horder (1.394 s)
- addGood (0.368 s)
- tuikuan (0.137 s)
- deleteGoodFromCar (0.122 s)
- Car (0.090 s)
- pay (0.109 s)
- tui (0.138 s)
- fahuo (0.085 s)
- jutui (0.091 s)
- payed (0.085 s)
- ushou (0.110 s)
- horder1 (0.406 s)
- sendgood (0.083 s)
- allowtui (0.082 s)
- shouhuo (0.123 s)

≡ Failure Trace

```
1077
1100         BufferedReader responseReader = r
1101         while ((readLine = responseReader
1102             formSB.append(readLine).appen
1103         }
1104         responseReader.close();
1105         result = formSB.toString();
1106     } else {
1107         result = "{\"code\":\"\" + resultC
1108     }
1109     out.close();
1110 } catch (Exception e) {
1111     logger.error(e.toString());
1112 }
1113 } finally {
1114     connection.disconnect();
1115 }
1116 System.out.println(result);
1117
1118 }
1119 }
1120
```

Problems   Javadoc   Declaration   Console   Progress

<terminated> CarControllerTest [JUnit] C:\Program Files\Java\jdk1.8.0

```
2021-11-18 21:05:15.139 INFO 2488 --- [          main]
2021-11-18 21:05:15.143 INFO 2488 --- [          main]

2021-11-18 21:05:15.223 INFO 2488 --- [          main]
2021-11-18 21:05:15.227 INFO 2488 --- [          main]

2021-11-18 21:05:15.304 INFO 2488 --- [          main]
2021-11-18 21:05:15.308 INFO 2488 --- [          main]
[{"id":"admin","shopid":"shopper","goodid":"cf89d005-3c8

2021-11-18 21:05:15.426 INFO 2488 --- [          main]
2021-11-18 21:05:15.432 INFO 2488 --- [extShutdownHook]
2021-11-18 21:05:15.433 INFO 2488 --- [extShutdownHook]
2021-11-18 21:05:15.443 INFO 2488 --- [extShutdownHook]
```

**测试结论:**

数据库定义的数据类型都是varchar255, 无法测试边界值等

**实现限制:**

需要一个一个修改测试用例才能够进行测试

备注:

测试项目名称:

vue前端单元测试

测试人员: 吴浩涵

测试时间: 2021/11/15

测试内容:

```
import login from '@/components/login';
import {expect} from 'chai'
import {mount} from '@vue/test-utils'

describe('登陆', () => {
  it('测试用例一', () => {
    expect(login('userid=admin')).to.be.deep.equal({userid:'admin'})
    expect(login('userword=123')).to.be.deep.equal({userid:'123'})
  });
  it('测试用例二', () => {
    expect(login('userid=buyer')).to.be.deep.equal({userid:'buyer'})
    expect(login('userword=123')).to.be.deep.equal({userid:'123'})
  });
  it('测试用例三', () => {
    expect(login('userid=buyer')).to.be.deep.equal({userid:'buyer'})
    expect(login('userword=123')).to.be.deep.equal({userid:'456'})
  });
  it('测试用例四', () => {
    expect(login('userid=buyer')).to.be.deep.equal({userid:'zxc'})
    expect(login('userword=123')).to.be.deep.equal({userid:'456'})
  });
  it('测试用例五', () => {
    expect(login('userid')).to.be.lengthOf(0)
    expect(login('userid')).to.be.lengthOf(0)
  });
});

describe('inputvalue', () => {
  expect(login(value)).ot.be.deep.equal. keysend. space
  expect(login(value)).ot.be.deep.equal. keysend. space
});
```

```

describe('messageid', ()=>{
  it(' iderrmsg', ()=>{
    let wrapper = mount(login);
    wrapper.setProps({msg:'请输入账号'})
    expect(wrapper.find('data()').text()).to.be.contain('请输入账号')
  })
  it(' worderrmsg', ()=>{
    let wrapper = mount(login);
    wrapper.setProps({msg:'请输入密码'})
    expect(wrapper.find('data()').text()).to.be.contain('请输入密码')
  })
})

```

```

import zhuce from '@components/zhuce';
import {expect} from 'chai'

```

```

describe('注册', ()=>{
  it(' 测试用例一', ()=>{
    expect(zhuce('value=test')).to.be.deep.equal({value:'test'})
  });
  it(' 测试用例二', ()=>{
    expect(zhuce(value)).ot.be.deep.equal.keysend. space
    expect(zhuce(value)).ot.be.deep.equal.keysend. space
  });
  it(' 测试用例三', ()=>{
    expect(zhuce('sex=男')).to.be.deep.equal({sex:'男'})
  });
  it(' 测试用例四', ()=>{
    expect(zhuce('sex=女')).to.be.deep.equal({sex:'女'})
  });
  it(' 测试用例五', ()=>{
    expect(zhuce('sex=1')).to.be.deep.equal({sex:'女'})
  });
  it(' 测试用例六', ()=>{
    expect(zhuce('phone=1111111111')).to.be.lengthOf(11)
  });
  it(' 测试用例七', ()=>{
    expect(zhuce('phone=11111111')).to.be.lengthOf(11)
  });
  it(' 测试用例八', ()=>{
    expect(zhuce('account=0000000000000000')).to.be.lengthOf(16)
  });
}

```

```
});
it('测试用例九', ()=>{
    expect(zhuce('acount=00000000')).to.be.lengthOf(16)
});
it('测试用例十', ()=>{
    expect(zhuce('city=beijing')).to.be.deep.equal({city:'beijing'})
});
it('测试用例十一', ()=>{
    expect(zhuce('city=beijing')).to.be.deep.equal({city:'test'})
});
});
```

#### 测试环境与系统配置:

前端测试环境: pc/Windows/vs/vue

#### 测试输入数据:

admin 123 zxc 456 空 1111111111 1111111 0000000000000000 00000000 beijing test

测试次数:

#### 预期结果:

admin 123 正确  
zxc 456 错误  
空 正确  
1111111111 正确  
1111111 错误  
0000000000000000 正确  
00000000 错误  
beijing 正确  
test 错误

#### 测试过程:

创建vue test, 编写测试用例, 编写完成cd进入myvue, npm run test运行测试

#### 测试结果:

登录测试  
测试用例一、测试用例二成功。测试用例三测试用例四用例五失败  
注册测试  
测试用例一到三成功、测试用例四五失败、测试用例六八成功、测试用例七九失败、测试用例十成功、测试用例十一失败

#### 测试结论:

前端单元测试

#### 实现限制:

前端输入框按照数据字典进行限制

**备注:**

A complete log of this run can be found in:  
D:\nodejs\node\_cache\\_logs\2021-11-15T06\_10\_54\_854Z-debug.log  
Access has been denied to resolve the property "statements" because it is not an "own  
property" of its parent.