

# 电子商城文档-前后端接口

## 功能概述

### 登录界面

不管是用户还是平台管理端，都通过一个界面登录，这个界面登录上去之后，response里面有role的信息，role为admin则进入平台管理端，role为用户则进入用户端。

session里面要存：ID，role，token

这也就意味着，如果直接访问admin 或者 user 相关的界面，session里面 没有相关role等信息的话，要被拦截

加一个注册功能，让用户可以直接注册

### 用户 (User) :

6个界面

#### 界面一：查看商品概览（默认界面）

搜索+筛选（有一个Type信息）查看商品（分页），可以对搜索结果进行按照价格排序（升序、降序）或者商品编号排序（排序在前端做还是后端做？），右边有一个商品详情可以点击，进入一个新的界面：商品详情。

#### 界面二：查看购物车

可以看到余额以及购物车信息，购物车中每个商品的右边最好有一个+和-的按钮可以按，调整购买商品数量，每次调整完记得刷新页面重新请求购物车信息,下方还要有汇总信息，一共多少钱。（不设计优惠券和打折）

#### 界面三：查看订单

对于订单（有三种，未发货、待收货、已完成，可以是三个按钮or一个选项之类的，也可以是三个类似界面，点一个看一类）：可以分页查看，待收货可以确认收货

#### 界面四：查看个人信息

里面可以看到账户相关的信息，以及自己的余额（不可修改，不提供充值按钮）和收货地址（有一个添加收货地址的功能）。

地址维护（一个收货地址包括收货人姓名、电话、住址、住址类型（家、学校））

修改密码

维护个人信息（真实姓名，邮箱）

#### 界面五：商品详情

还是最基本的商品的信息，多了一些评论，然后点击购买会弹出一个框框输入数量，可以选择立即下单or加入购物车。立即购买还需要

填地址信息和备注等到对订单付款的时候干。

(ps.如何处理立即下单和加入购物车这两个小功能的同与不同交给前端同学)

## 界面六：用户自己的评价界面

可以查看自己的可评价和已评价

每一条评价会显示是哪个订单的哪个物品及数量（maybe有个时间）

**五步走：先下单，此时不用太多信息，然后付款，此时顺便填地址和备注，然后管理员发货，然后用户确认收货，之后可以去评价**

## 平台管理端 (Admin)

### 界面一：管理商品

搜索+筛选（有一个Type信息和出版社信息）查看商品（分页），右边可以选择删除or修改商品信息。

右上角有个按钮可以增加商品

### 界面二：管理用户

可以分页查看用户，看不到用户密码，右边有个按钮：充值

添加用户

### 界面三：订单管理

对于订单（有四种，未支付、未发货、待收货、已完成，可以是四个按钮or一个选项之类的，也可以是四个类似界面，点一个看一类）：可以分页查看，**未发货可以确认发货**，要求填写一个物流信息（一个简单的string）

### 界面四：管理出版社

比较容易

### 界面五：管理商品类型Tag

比较容易

### 界面六：管理商品媒介形式Medium

比较容易

---

## 设计风格（前后端交互）

接口调用使用Restful的风格。

所有的**response**都是**data+code+msg**。

code为200表示请求成功，使用data中的数据，如果code不是200，则弹出一个表示失败的弹窗，并显示msg的内容

除了登录以外，**所有的request**都需要在Header中携带auth=token，ID=ID。role不用携带，但是建议保存在session中，用于前端界面路由筛选。

request的Body如果携带多个参数，就把多个参数以结构体的方式在JSON中打包传送，如果只有一个参数，则直接以键值对的方式传输

**所有的请求都使用POST的方式**，不刻意区分delete、post、get、update

命名风格：ID全大写，基本使用驼峰命名法

---

## 接口文档（前后端）

### 1.登录

POST请求：/auth/login

header里面可以不携带东西

在RequestBody里面携带JSON类型的数据

```
{
  "ID": "admin",
  "password": "admin"
}
```

成功样例的Response的Body

```
{
  "code": 200,
  "msg": "登录成功",
  "data": {
    "ID": "admin",
    "role": "admin",
    "token": "aa",
    "refreshToken": "aa"
  }
}
```

失败样例的Response的Body

```
{
  "code": 1000,
  "msg": "登录失败",
  "data": null
}
```

---

## 用户功能/user

## 1.注册

/auth/register

唯一不需要携带header中信息的接口之一

```
{
  "ID": "1234zjx",
  "password": "1234",
  "name": "zjx",
  "email": "20302010032@fudan.edu.cn",
  "phone": "13906638321"
}
```

```
{
  "code": 200,
  "msg": "注册成功",
  "data": "1234zjx"
}
```

## 1.2修改密码

/user/changePassword

```
{
  "oldPassword": "1234",
  "newPassword": "12345"
}
```

## 2.1获取个人信息

/user/info

header中携带好ID和token，Body中不需要参数，

注意，response里的参数叫userID，而不是ID

```
{
  "code": 200,
  "msg": "获取个人信息成功",
  "data": {
    "phone": "13906638321",
    "name": "张佳洵",
    "userID": "1234zjx",
    "email": "20302010032@fudan.edu.cn"
  }
}
```

## 2.2修改个人信息

/user/updateInfo

header中携带好ID和token, Body中需要参数

```
{
  "phone": "13906638321",
  "name": "张佳洵",
  "email": "20302010032@fudan.edu.cn"
}
```

获得响应之后如果是code==200表示修改成功, 记得刷新页面重新获得新的数据

## 3.1用户添加收货地址

/user/addAddress

请求体里面携带要新添加的location,发完请求如果是成功 (code==200) 记得**刷新一下界面**, 因为会有新的地址, 重新请求所有地址

```
{
  "name": "张佳洵",
  "phone": "13906638321",
  "details": "浙江省温州市瓯海区娄桥街道",
  "description": "家"
}
```

```
{
  "code": 200,
  "msg": "地址添加成功",
  "data": "浙江省温州市瓯海区娄桥街道"
}
```

## 3.2用户删除收货地址

user/deleteAddress

RequestBody里面直接传一个string表示是要删除掉的**收货地址的id**就行

```
2
```

code==200的话记得刷新

## 4.获取自己的所有收货地址

/user/allAddress

不需要参数, 携带好ID和token,得到的response里面的data, 是之前地址的**数组**, 当然还加上了ID, **这个ID可以不显示给用户看**

```
{
  "code": 200,
  "msg": "succ",
  "data": [
```

```

    {
      "addressID": 2,
      "name": "张佳洵",
      "phone": "13906638321",
      "details": "浙江省温州市瓯海区娄桥街道",
      "description": "家"
    },
    {
      "addressID": 3,
      "name": "张佳洵",
      "phone": "13906638321",
      "details": "上海市复旦大学",
      "description": "学校"
    }
  ]
}

```

## 4.2获取自己的所有收货地址(String形式)

/user/allAddressString

不需要参数, 携带好ID和token

```

{
  "code": 200,
  "msg": "succ",
  "data": [
    "上海市复旦大学 张佳洵 13906638321 (学校)"
  ]
}

```

####

## 5.获得购物车中的数据

/user/getShoplist

不需要参数, 携带好ID和token,得到的response里面的data, 是一个结构体, 它有totalPrice和一个数组

```

{
  "code": 200,
  "msg": "获得购物车成功",
  "data": {
    "totalPrice": 44.00,
    "records": [
      {
        "goodsId": "BNO1234",
        "name": "金瓶梅",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "黄色",
        "imageUrl": "nb",
        "details": "机械工业",

```

```

        "price": 12.00,
        "status": "ForSale",
        "num": 2
    },
    {
        "goodsId": "BN06789",
        "name": "计算机网络",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "学习",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 10.00,
        "status": "ForSale",
        "num": 2
    }
]
}

```

## 6.1 添加进购物车（在商品详情界面）

`/user/addToShoplist`

Body中需要的参数：goodID和数量，添加的如果是购物车中已有的，会有类似于商品数量+k的效果

```

{
  "goodID": "aa",
  "num": 2
}

```

如果是在商品详情页面点击的添加进购物车

```

{
  "code": 200,
  "msg": "添加成功",
  "data": null
}

```

## 6.2 调整购物车中商品数量(在购物车界面)

`/user/updateShoplist`

Body中需要的参数：goodID和数量，这里的数量表示**调整后的数量**，**要求非负**，0表示从购物车中删除

```

{
  "goodID": "aa",
  "num": 2
}

```

```
{
  "code": 200,
  "msg": "添加成功",
  "data": null
}
```

### 6.3从购物车中删除一样商品(在购物车界面)

/user/removeFromShoplist

Body中需要的参数: goodID

"aa"

```
{
  "code": 200,
  "msg": "删除成功",
  "data": null
}
```

### 7.立即下单

/user/buy

Body里面需要商品的ID和对应的数量,数量是用户点击立即购买以后填写的,我们要求必须是**正整数**。

```
{
  "goodID": "BN01234",
  "num": 2
}
```

成功或者失败都记得出现弹窗提示一下信息msg

地址信息和备注等付钱的时候一起填

### 8.结算购物车

/user/buyShoplist

Body不需要参数,把Header带好就行

成功或者失败都记得出现弹窗提示一下信息msg,成功还得刷新页面

**注意,这个函数不管结果是不是200都把msg拿出来提示一下,200和310只显示msg,201显示msg+data**

```
{
  "code": 310,
  "msg": "购物车中全为下架商品!\n已自动清空购物车",
  "data": null
}
```



## 9.获取自己的余额

/user/getMoney

Body里面不需要有参数,, 拿到的data直接是一个money类似的数据 (固定两位小数点)

```
{
  "code": 200,
  "msg": "获得余额成功",
  "data": 0.00
}
```

## 10.查看商品 (搜索+分页)

/user/userSearchForSaleGood

```
{
  "medium": [],
  "tag": [],
  "pageNum": 1,
  "pageSize": 10,
  "search": ""
}
```

```
{
  "code": 200,
  "msg": "搜索商品成功",
  "data": {
    "total": 2,
    "records": [
      {
        "goodsId": "BNO1234",
        "name": "金瓶梅",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "黄色",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 12.00,
        "status": "ForSale"
      },
      {
        "goodsId": "BNO6789",
        "name": "计算机网络",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "学习",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 10.00,

```

```
        "status": "ForSale"
      }
    ]
  }
}
```

如果search为null，则表示搜索全部,type为null，则表示搜索全部种类,Tag类似

## 11.清空购物车

/user/clearShoplist

header中携带好auth和ID即可

## 12.1查看待支付订单

user/getToPaidOrder

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "获得待支付订单成功",
  "data": {
    "total": 3,
    "records": [
      {
        "id": 2,
        "buyerId": "1234zjx",
        "totalPrice": 42.0,
        "status": "NotPaid",
        "address": null,
        "notes": null,
        "waybillNum": null
      },
      {
        "id": 4,
        "buyerId": "1234zjx",
        "totalPrice": 36.0,
        "status": "NotPaid",
        "address": null,
        "notes": null,
        "waybillNum": null
      },
      {
        "id": 6,
        "buyerId": "1234zjx",
        "totalPrice": 24.0,
        "status": "NotPaid",

```

```
        "address": null,
        "notes": null,
        "waybillNum": null
    }
}
]
```

## 12.2查看待发货订单

user/getUndeliverOrder

## 12.3查看待收货订单

user/getNotReceivedOrder

## 12.4查看已完成订单

user/getFinishedOrder

## 13.支付订单

支付订单的时候要一起填写备注和收获地址

user/pay

```
{
  "orderId": 4,
  "notes": "不要加辣",
  "address": "xxxx xxx xxx xx"
}
```

```
{
  "code": 200,
  "msg": "支付成功",
  "data": null
}
```

## 14.获得订单详情

user/orderDetail

2

```
{
  "code": 200,
  "msg": "获得订单详情成功",
  "data": [
    {
      "goodsId": "BN01234",
      "name": "金瓶梅",
      "author": "xx",
      "publisher": "机械工业",
    }
  ]
}
```

```

        "medium": "书籍",
        "tag": "黄色",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 12.0,
        "status": "ForSale",
        "num": 1
    },
    {
        "goodsId": "BN06789",
        "name": "计算机网络",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "学习",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 30.0,
        "status": "ForSale",
        "num": 3
    }
]
}

```

这里类似购物车的结构，但是！**price的含义是总价！就是这个商品购买时的价格\*数量**

## 15.确认收货

user/confirmReceive

只需要给一个orderId即可

4

```

{
  "code": 330,
  "msg": "确认收货失败\nnull",
  "data": null
}

```

## 16.获得用户的未评价（分页）

/shop/getNeedComment

注意，这个页面里面可以先把content和stars忽略掉，不显示出来

```

{
  "pageNum": 1,
  "pageSize": 10
}

```

```

{

```

```
"code": 200,
"msg": "成功获得评价信息",
"data": {
  "total": 1,
  "records": [
    {
      "orderId": 4,
      "purchaseID": 5,
      "goodID": "BNO1234",
      "goodName": "金瓶梅",
      "url": "nb",
      "content": null,
      "stars": null
    }
  ]
}
```

## 17. 获得用户的已评价（分页）

/shop/getHaveCommented

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "成功获得评价信息",
  "data": {
    "total": 1,
    "records": [
      {
        "orderId": 4,
        "purchaseID": 5,
        "goodID": "BNO1234",
        "goodName": "金瓶梅",
        "url": "nb",
        "content": "挺好看的",
        "stars": 5
      }
    ]
  }
}
```

## 18.进行评价

/shop/toComment

注意，评价的stars要求是整数

```
{
  "purchaseID": 5,
  "content": "挺好看的",
  "stars": 5
}
```

```
{
  "code": 200,
  "msg": "评价成功",
  "data": null
}
```

## 19.(分页)获得一个商品的评价信息（在商品详情页使用）

shop/getGoodComment

```
{
  "goodID": "BN01234",
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "获得商品评价",
  "data": {
    "total": 1,
    "records": [
      {
        "userID": "1234zjx",
        "content": "挺好看的",
        "stars": 5
      }
    ]
  }
}
```

## 管理员功能

### 1.1获取所有用户信息（分页）（还没写）

/admin/getUserPage

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 1,
    "records": [
      {
        "userID": "1234zjx",
        "name": "张佳洵",
        "email": "20302010032@fudan.edu.cn",
        "phone": "13906638321"
      }
    ]
  }
}
```

## 1.2给用户充值

/money/add

注意，是/money

```
{
  "ID": "1234zjx",
  "money": 20.00//20.00或者"20.00"都可以
}
```

```
{
  "code": 200,
  "msg": "充值成功",
  "data": "1234zjx"
}
```

## 2.添加出版社

/shop/addPublisher

请求

```
{
  "name": "机械工业",
  "details": "nb"
}
```

响应

```
{
  "code": 200,
  "msg": "添加成功",
  "data": "机械工业"
}
```

## 2.2修改出版社信息

/shop/updatePublisher

请求

```
{
  "name": "机械工业",
  "details": "update:nb"
}
```

响应

```
{
  "code": 200,
  "msg": "信息修改成功",
  "data": "机械工业"
}
```

## 2.3删除出版社

/shop/deletePublisher

请求

机械工业

响应

```
{
  "code": 200,
  "msg": "信息修改成功",
  "data": "机械工业"
}
```

## 2.4获得所有出版社的名字

用于下拉框选择

/shop/getAllPublisherName

不需要请求参数

响应



```
{
  "code": 200,
  "msg": "succ",
  "data": [
    "机械工业",
    "机械工业2",
    "机械工业3"
  ]
}
```

## 2.5分页获得出版社

/shop/searchPublisherPage

这里的search给什么都没关系，不会去搜索，就是单纯留着了，前端发消息的时候有or没有这个search不影响后端接收，结果都一样

```
{
  "pageNum": 1,
  "pageSize": 10,
  "search": ""
}
```

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 3,
    "records": [
      {
        "name": "机械工业",
        "details": "nb"
      },
      {
        "name": "机械工业2",
        "details": "nb"
      },
      {
        "name": "机械工业3",
        "details": "update:nb"
      }
    ]
  }
}
```

## 3.添加商品形式/媒介

/shop/addMedium

请求（注意，直接传string的话不需要额外套括号）

书籍

响应

```
{
  "code": 200,
  "msg": "添加成功",
  "data": "书籍"
}
```

### 3.2删除

/shop/deleteMedium

请求

书籍

响应与删除出版社类似

### 3.3获得所有商品媒介

/shop/getAllMedium

请求不需要参数，响应与获取所有出版社名字类似

### 3.4获得所有商品媒介名字

/shop/getAllMediumName

### 4.1添加商品标签Tag

/shop/addGoodsTag

### 4.2删除Tag

/shop/deleteGoodsTag

### 4.3获得所有Tag

/shop/getAllGoodsTag

### 4.3获得所有Tag名字

/shop/getAllGoodsTagName

## 6.添加商品

/shop/addGood

刚添加进来默认是在售的

注意一个商品的tag和medium是**单选框**!!!

```
{
  "goodsId": "BNO1234",
  "name": "金瓶梅",
  "author": "xx",
  "publisher": "机械工业",
  "medium": "书籍",
  "tag": "黄色",
  "imageUrl": "nb", //以后会是传一幅图片，而不是string的url
  "details": "机械工业",
  "price": "12.00" //注意！是string！不用特意搞成数字
}
```

## 7. 查看在售商品（搜索+分页）

/shop/adminSearchForSaleGood

如果search为null或者""，则表示搜索全部，tag为null或者空数组，则表示搜索全部种类，Tag类似

如果选中多个Tag，则表示**满足任何一个Tag**即可，type类似

页面里需要两个多选框和一个文字搜索框

```
{
  "medium": ["书籍", "CS"],
  "tag": ["玄幻", "言情"],
  "pageNum": 1,
  "pageSize": 10,
  "search": "阿衰"
}
```

注意，下面这里比添加商品多了一个**是否在售**的信息，这个信息类似性别的“男”“女”处理，可以做成两个小圆点，也可以做成一个下拉选择框，里面的选项是固定死的**ForSale**和**SoldOut**

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 2,
    "records": [
      {
        "goodsId": "BNO1234",
        "name": "金瓶梅",
        "author": "xx",
        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "黄色",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 12.00,
        "status": "ForSale"
      },
      {
        "goodsId": "BNO6789",
        "name": "计算机网络",
        "author": "xx",

```

```

        "publisher": "机械工业",
        "medium": "书籍",
        "tag": "学习",
        "imageUrl": "nb",
        "details": "机械工业",
        "price": 10.00,
        "status": "ForSale"//只有ForSale和SaleOut两种
    }
}
}
}

```

## 8.查看下架商品

/shop/adminSearchSoldOutGood

参数和查看在售商品完全一样，得到的结果也一样，**建议做成两个页面**

## 9.修改商品信息

/shop/updateGood

上架和下架单独管理，一方面是修改商品的**基本信息**，另一方面是可以修改上架和下架

**goodsID是不让改的**

商品信息下架了的话，用户那里就搜索不到了

```

{
  "goodsId": "BN06789",
  "name": "计算机网络",
  "author": "xx",
  "publisher": "机械工业",
  "medium": "书籍",
  "tag": "学习",
  "imageUrl": "nb",
  "details": "机械工业",
  "price": "10.00",
  "status": "SoldOut"
}

```

```

{
  "code": 200,
  "msg": "修改成功",
  "data": "计算机网络"
}

```

## 10.管理员查看订单详情

/shop/adminOrderDetail

2

```

{
  "code": 200,

```

```
"msg": "获得订单详情成功",
"data": [
  {
    "goodsId": "BN01234",
    "name": "金瓶梅",
    "author": "xx",
    "publisher": "机械工业",
    "medium": "书籍",
    "tag": "黄色",
    "imageUrl": "nb",
    "details": "机械工业",
    "price": 12.0,
    "status": "ForSale",
    "num": 1
  },
  {
    "goodsId": "BN06789",
    "name": "计算机网络",
    "author": "xx",
    "publisher": "机械工业",
    "medium": "书籍",
    "tag": "学习",
    "imageUrl": "nb",
    "details": "机械工业",
    "price": 30.0,
    "status": "ForSale",
    "num": 3
  }
]
}
```

## 11.0查看未支付订单（不搜索+分页）

shop/adminSearchToPaidOrder

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 2,
    "records": [
      {
        "id": 2,
        "buyerId": "1234zjx",
        "totalPrice": 42.0,
        "status": "NotPaid",
        "address": null,
        "notes": null,

```

```
        "waybillNum": null
      },
      {
        "id": 6,
        "buyerId": "1234zjx",
        "totalPrice": 24.0,
        "status": "NotPaid",
        "address": null,
        "notes": null,
        "waybillNum": null
      }
    ]
  }
}
```

## 11.1查看未发货订单（不搜索+分页）

shop/adminSearchUndeliverOrder

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 1,
    "records": [
      {
        "id": 4,
        "buyerId": "1234zjx",
        "totalPrice": 36,
        "status": "NotYetShipped",
        "address": "xxxx xxx xxx xx",
        "notes": "不要加辣",
        "waybillNum": null
      }
    ]
  }
}
```

## 11.2发货

shop/deliver

```
{
  "orderId": 4,
  "deliverNo": "YT1234"
}
```

```
{
  "code": 200,
  "msg": "发货成功",
  "data": null
}
```

## 12.查看待收货订单（不搜索+分页）

shop/adminSearchUnreceiveOrder

```
{
  "pageNum": 1,
  "pageSize": 10
}
```

```
{
  "code": 200,
  "msg": "succ",
  "data": {
    "total": 1,
    "records": [
      {
        "id": 4,
        "buyerId": "1234zjx",
        "totalPrice": 36,
        "status": "Shipped",
        "address": "xxxx xxx xxx xx",
        "notes": "不要加辣",
        "waybillNum": "YT1234"
      }
    ]
  }
}
```

### 13.查看已完成订单（不搜索+分页）

shop/adminSearchFinishOrder

```
{  
  "pageNum": 1,  
  "pageSize": 10  
}
```