

# RedRock电商

## 关于本接口文档和作业

1. 必须完成全部的基础功能，截至时间：2025年2月16日23:59:59前
2. 将你的项目文件上传至GitHub，提交时只提交你的项目仓库地址，至 `be@redrock.team`
3. 基础功能的接口已在本文档给出，仅作参考请根据自己的实际进行修改。
4. 本接口文档对于用户信息的验证全部为**token**验证。
5. 本文档采用RESTful风格的API虽然写的很拉，如果有路径参数会在访问路径上用 `{}` 标明，并且会在请求参数中解释其含义。

对于返回值，我们规定返回以下格式的 `json`：

```
1  {
2      status: 10000,      // 一个5位数的状态码，正常返回下为10000，其他非正常状态下的
                           状态码需要自行定义
3      "info": "success", // 本次请求的状态文字信息，正常返回下为success，其他非正常状
                           态下的需要自行定义
4      "data": {           // 一个json对象，里面包含其他要返回的数据的键值对
5          "param1": "...",
6          "param2": "..."
7      }
8  }
```

我们解释返回参数时，只解释 `data` 字段里的参数值

1. Web框架可使用**gin**、**Hertz**两个框架，如果使用了其他第三方库，请在你的 `README.md` 文件中展示你对该库的理解与该库是如何，以及你为什么使用这个第三方库，以及这个库在你的作业中的起了什么作用。
2. 对于某个接口的讲解如下：
3. 请注意一个操作所牵扯到的其他数据，并且要注意操作的原子性。
4. 一定要对入参进行**合法性校验**
5. 商品内容的图片展示和存储等不做要求，只需要像商品信息的字段link给出一个链接即可（可以自己编造的，也可以像这个链接一样**IPone16**）
6. 有任何疑问请问学姐

# 基础功能

类似一个商品网站，可参考[淘天集团](#) [淘宝](#)

- 用户的注册与登录
- 修改用户的信息
- 查询商品
- 查看商品详情
- 查看分类下的商品
- 给商品进行评论（评论的增删查改）
- 商品加入购物车
- 获取购物车所有商品
- 下单（只需向前端返回一个订单order，实际业务较为复杂可能会接触一些微服务相关内容）

## 加分项

- 完善订单服务
- 嵌套评论
- 匿名评论
- 浏览的商品的历史记录
- 商家和管理员的出现（可以自由增删商品）
- 与商家客服的聊天
- 单一的登录较为不安全，试着加入验证码操作
- 使用缓存
- 设计一套热度算法，使得能让该用户常看的商品分类下的商品出现在首页
- 用户可以与客服进行聊天
- 部署到自己的服务器上，并且可以访问
- 考虑安全性（xss，sql注入，csrf等）
- 你任何想加的功能

## 基础功能接口

此处的接口文档仅作参考，请根据你的项目实际情况对参数以及返回示例进行增删改查，如有错误请忽略联系学姐

返回示例未作错误示例的处理，请自行结合代码实际情况对返回示例的异常返回进行展示

## 用户相关

### 注册

请求路径：POST /user/register

请求头：

请求参数：application/json

A	B	C	D	E
名称	位置	类型	必选	说明
username	body	string	是	用户名
password	body	string	是	密码

返回参数：

返回示例：

成功返回：

```
1 {
2   "status": 10000,
3   "info": "success"
4 }
```

### 登录（获取token）

请求路径：GET /user/token

请求头：

请求参数：application/json

A	B	C	D
名称	类型	必选	说明
username	string	是	用户名/用户 手机号/邮箱
password	string	是	密码

返回参数：

A	B	C	D
字段名	必选	类型	说明
refresh_token	是	Bearer \$token	refresh_token
token	是	Bearer \$token	token

返回示例：

成功返回

```
1  {
2    "status": 10000,
3    "info": "success",
4    "data": {
5      "refresh_token": "{refresh_token}",
6      "token": "{token}"
7    }
8  }
```

刷新token（维持登录状态）

请求路径： GET /user/token/refresh

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数: application/json

A	B	C	D	E
名称	位置	类型	必选	说明
refresh_token	query	string	是	refresh_token

返回参数：

A	B	C	D
字段名	必选	类型	说明
refresh_token	是	Bearer \$token	新refresh_token
token	是	Bearer \$token	新token

返回示例：

成功返回

```
1 {
2   "status": 10000,
3   "info": "success",
4   "data": {
5     "refresh_token": "{refresh_token}",
6     "token": "{token}"
7   }
8 }
```

修改用户密码

请求路径：PUT /user/password

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：application/json

A	B	C	D
名称	类型	必选	说明
old_password	string	是	旧密码
new_password	string	是	新密码

返回参数：

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000
4 }
```

获取用户信息

请求路径：GET /user/info/{user\_id}

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：query

A	B	C	D	E
名称	位置	类型	必选	说明
user_id	path	string	是	用户id

返回参数：

A	B	C	D
字段名	必选	类型	说明
user	是	复杂数据类型	用户信息

返回示例：

成功返回

```
1 {
2   "status": 10000,
3   "info": "success",
4   "data": {
5     "user": {
```

```
6      "id": "123456",
7      "avatar": "{avatar_url}",
8      "nickname": "你好世界",
9      "introduction": "永远少年",
10     "phone": 12345678900,
11     "qq":123456789,
12     "gender": "未知",
13     "email": "test@gmail.com",
14     "birthday": "2022-01-02"
15 }
16 }
17 }
```

## 修改用户信息

### 请求路径：PUT /user/info

该接口用于更改用户信息，除token外，全部请求参数都是非必选的，要求该接口有某个参数才修改对应的信息。比如我请求时只选了nickname和introduction字段，那么我们只改用户昵称和简介的信息，其他信息保持原状。

### 请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

### 请求参数：application/json

A	B	C	D	E
名称	位置	类型	必选	说明
nickname	body	string	否	昵称
avatar	body	string	否	头像
introduction	body	string	否	简介
telephone	body	string	否	手机号
qq	body	string	否	qq号
gender	body	string	否	性别
email	body	string	否	邮箱
birthday	body	string	否	生日，格式为YYYY-MM-DD

返回参数：

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000
4 }
```

## 商品相关

获取商品列表

请求路径：GET /product/list

请求头：

请求参数：

字段名	类型	说明
products	复杂数据类型数组	商品的信息

返回参数：

返回示例：

成功返回

```
1 {
2   "status": 10000,
3   "info": "success",
4   "data": {
5     "products": [
6       {
7         "product_id": "1",
8         "name": "傲慢与偏见",
9         "description": "一本书",
10        "type": "book",
11        "comment_num": 35,
12        "price": 9.8,
```



```
13     "is_addedCart":true
14     "cover": "http://127.0.0.1/picture_url1",
15     "publish_time": "1980-11-07",
16     "link": "http://127.0.0.1/test1"
17 },
18 {
19     "product_id": "2",
20     "name": "T-shirt",
21     "description": "一件短袖",
22     "type":"clothes",
23     "comment_num": 100,
24     "price": 88.88,
25     "is_addedCart":false
26     "cover": "http://127.0.0.1/picture_url2",
27     "publish_time": "1980-11-07",
28     "link": "http://127.0.0.1/test2"
29 }
30 ]
31 }
32 }
```

## 搜索商品

请求路径：GET /book/search

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	否	Bearer \$token	验证token，没token则is_addedCart字段为false

请求参数：query

A	B	C	D	E
名称	位置	类型	必选	说明
product_name	query	string	是	商品名称/代号（如果有相关的字段）

返回参数：

A	B	C	D
字段名	必选	类型	说明
products	是	复杂数据类型	商品信息

返回示例：

成功返回

```
1 {
2   "status": 10000,
3   "info": "success",
4   "data": {
5     "products": [
6       {
7         "product_id": "1",
8         "name": "傲慢与偏见",
9         "description": "一本书",
10        "type": "book",
11        "comment_num": 35,
12        "price": 9.8,
13        "is_addedCart": true
14        "cover": "http://127.0.0.1/picture_url1",
15        "publish_time": "1980-11-07",
16        "link": "http://127.0.0.1/test1"
17      }
18    ]
19  }
```

加入购物车

请求路径：PUT /product/addCart

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：x-www-form-urlencoded/form-data

A	B	C	D
名称	类型	必选	说明
product_id	string	是	商品id

返回参数：

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000
4 }
```

获取购物车商品列表

请求路径： GET /product/cart

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数： query

A	B	C	D	E
名称	位置	类型	必选	说明
user_id	path	string	是	用户id

返回参数

字段名	类型	说明
products	复杂数据类型数组	商品的信息

返回示例：

```
1 {
```

```
2  "status": 10000,
3  "info": "success",
4  "data": {
5    "products": [
6      {
7        "product_id": "1",
8        "name": "傲慢与偏见",
9        "type": "book",
10       "price": 9.8,
11       "cover": "http://127.0.0.1/picture_url1",
12       "link": "http://127.0.0.1/test1",
13       "num": 114
14     },
15     {
16       "product_id": "2",
17       "name": "T-shirt",
18       "type": "clothes",
19       "price": 88.88,
20       "cover": "http://127.0.0.1/picture_url2",
21       "link": "http://127.0.0.1/test2",
22       "num": 514
23     }
24   ],
25   "accout": 114514
26 }
27 }
```

## 获取商品详情

请求路径：GET /product/info/{product\_id}

请求头：

请求参数：query

A	B	C	D	E
名称	位置	类型	必选	说明
product_id	path	string	是	商品id

返回参数：

A	B	C
字段名	类型	说明
name	string	商品名称
description	string	商品概述
type	string	类型
comment_num	int	评论数量
price	float	商品价格
is_addedCart	bool	是否加入购物车
cover	string	封面链接
publish_time	string	发布时间
link	string	链接

返回示例：

```
1 {
2   "status": 10000,
3   "info": "success",
4   "product_id": "1",
5   "name": "傲慢与偏见",
6   "description": "一本书",
7   "type":"book",
8   "comment_num": 35,
9   "price": 9.8,
10  "is_addedCart":true
11  "cover": "http://127.0.0.1/picture_url1",
12  "publish_time": "1980-11-07",
13  "link": "http://127.0.0.1/test1"
14 }
```

获取相应标签的商品列表

请求路径 GET /product/{type}

请求头：

请求参数：query

A	B	C	D	E
名称	位置	类型	必选	说明
t'y'petype	query	string	否	商品标签

返回参数：

A	B	C	D
字段名	必选	类型	说明
data	是	复杂数据类型	商品信息

返回示例：

成功返回

```
1 {
2   "status": 10000,
3   "info": "success",
4   "data": {
5     "products": [
6       "product_id": "114",
7       "name": "一个商品",
8       "description": "仙贝",
9       "type": "other",
10      "comment_num": 35,
11      "price": 9.8,
12      "is_addedCart": true
13      "cover": "xxxx",
14      "publish_time": "2001-12-05",
15      "link": "xxxxxx"
16    ],
17  }
18 }
```

评论相关

获取商品的评论

请求路径： GET /comment/{product\_id}

请求头：

请求参数： query

A	B	C	D	E
名称	位置	类型	必选	说明
product_id	path	string	是	商品id

返回参数：

A	B	C	D
字段名	必选	类型	说明
comments	是	复杂数据类型数组	信息不完全的书评信息的数组

返回示例:

成功返回

```
1 {
2   "info": "success",
3   "status": 10000,
4   "comments": [
5     {
6       "post_id": "79",
7       "publish_time": 503270377478,
8       "content": "occaecat est voluptate qui officia",
9       "user_id": "55",
10      "avatar": "http://dummyimage.com/100x100",
11      "nickname": "康静",
12      "praise_count": 65,
13      "is_praised": 1, (0为未处理, 1为点赞, 2为点踩)
14      "product_id": "99"
15    },
16    {
17      "post_id": "80",
18      "publish_time": 503270377478,
19      "content": "occaecat est voluptate qui officia",
20      "user_id": "55",
21      "avatar": "http://dummyimage.com/100x100",
22      "nickname": "康静",
23      "praise_count": 65,
24      "is_praised": 1,
25      "product_id": "99"
26    }
27  ]
28 }
```

给商品评论

请求路径: POST /comment/{product\_id}

请求头:

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：application/json

A	B	C	D
名称	类型	必选	说明
book_id	string	是	商品id
content	string	是	评论内容

返回参数：

A	B	C	D
字段名	必选	类型	说明
data	是	int	此评论的id

返回示例：

```
1 {
2   "info": "success",
3   "status": 10000,
4   "data": "{comment_id}"
5 }
```

## 删除书评

删除评论的时候需要删除相关点赞信息

请求路径：DELETE /comment/{comment\_id}

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：query



A	B	C	D	E
名称	位置	类型	必选	说明
comment_id	path	string	是	评论id

返回参数：

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000
4 }
```

更新评论

请求路径： PUT /comment/{comment\_id}

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

返回参数：

请求参数： application/json

A	B	C	D
名称	类型	必选	说明
comment_id	string	是	评论id
content	string	是	评论内容

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000
4 }
```

## 操作相关

### 点赞点踩

请求路径：PUT /comment/praise

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：form-data/x-www-form-urlencoded

A	B	C	D
字段名	必选	类型	说明
model	是	int	模型标识。当model为1时，为点赞；model为2时，为点踩
comment_id	是	string	要点赞的id。

返回参数：

返回示例：

成功返回

```
1 {
2     "info": "success",
3     "status": 10000
4 }
```

### 下单

请求路径：POST /operate/order

请求头：

A	B	C	D
字段名	必选	数值	说明
Authorization	是	Bearer \$token	验证token

请求参数：application/json

A	B	C	D
字段名	必选	类型	说明
user_id	是	int	用户id
orders	是	复杂类型数组	订单内容
address	是	复杂类型	地址信息
total	是	float	总价

返回参数：

A	B	C	D
名称	类型	必选	说明
order_id	string	是	订单id

返回示例：

成功返回

```
1 {
2   "info": "success",
3   "status": 10000,
4   "order_id":233
5 }
```

## 结语

亲爱的同学们，我们本学期的课程伴随着本次的寒假作业的开始而结束了，很感谢同学们的坚持，即使我们的课讲得并不好，也依旧坚持学习（虽然可能有的同学中途因为种种事而脱节了）。无论如何，你们在过去的一学期里，都付出了努力。可能部分同学的基础较为薄弱，深受打击也不要因此而放弃，可以借此时机慢慢努力进行补充。每个人都有自己的节奏与学习方式，不要受到其他人的影响，只要在进步那么一切都是有意义的。

新的一年即将到来，在这里提前先祝福大家新年快乐，蛇年阖家幸福（希望不要大年初一还在敲代码，同家人一起开心过年才是重要的）。

--by 红岩网校后端研发部

隔壁家叔叔 过年 了还在



你可千万  
不要变成  
那样

