

主要的需求：

- 1、系统可以根据进入者的身份展现不同的功能
- 2、通用功能有看个人信息、修改个人信息（包括密码）查看和自己相关的订单信息
- 3、商家可以上架和下架商品，但不能使用下单功能
- 4、用户对订单的操作主要有，下单、退订和确认收货，但使用不了和商家有关的操作
- 5、外卖员可以抢单，还有可以选择可以抢的订单，查看自己抢到的单的信息
- 6、商家下架商品需要实现检查该订单被哪些顾客下单过，而且还没出餐，然后根据商品的价格逐一将钱退回给顾客，然后将订单删除。对于出餐的，或者被骑手抢单的就无需操作

Category (id, 类)

Users (账号, 密码, 用户名, 头像地址, 类别)

Menu (商家账号 (FK), 菜名, 价格)

Order (订单编号 (递增), 商家账号 (FK), 菜名 (FK), 顾客 (FK), 骑手 (FK))

接口：

请求方式：

参数：

响应：

Category: 1——外卖员；2——商家；3——顾客

下单的 pay 需要验证是否存在这个商品

接口需求

User

1. 注册接口：user/register

请求方式：get

参数：password, username, category

响应：一个账号

```
{  
  "code": 0,
```

```
"message": "操作成功",  
"data": 366659464  
}
```

2. 登录接口: user/login

请求方式: post, 请求体里使用表单提交。

参数: account, password

响应: 一个 token

3. 更改密码接口: user/alterPwd

请求方式: post

参数: newPwd (新密码)

响应:

4. 更改其他信息的接口: user/alterOther

请求方式: post

参数: Nun (新的用户名, 可以为 null), Npic (新头像, 可以为 null)。前端记得当两者为空时提示, 提交的参数最好至少一个不为空。

响应:

5. 个人信息接口: user/mymessage

请求方式: post, 请求体里使用表单提交。

参数: 无

响应: 一个 json: [

```
{  
  "account": 37797662,  
  "password": "3d2172418ce305c7d16d4b05597c6a59",  
  "picture": "null",  
  "balance": 30,  
  "category": 2,  
  "username": "test2"  
}  
]
```

Customer

1. 下单接口：

请求方式：post

参数：JSON (seller, 卖方账号, name, 菜名, price, 价格) 前端根据点击哪个菜将菜的信息通过 dom 元素 (就是 json 的参数) 选出来然后发给后端。

响应：Result 类

2. 退订接口：customer/withdraw

请求方式：post

参数：和上面的一样通过 dom 获取 json 对于菜单 order

```
{
    "num": 23,
    "seller": 641832841,
    "name": "kfc5",
    "customer": 54206030,
    "rider": 0
}
```

记得还要在这里判断是否满足退订的条件，即 rider 是 0 而不是一个具体的数据串。

3. 确认收货接口：customer/confirm

请求方式：post

参数：和上面的一样，一个 order 的 json

响应：Result 对象

4. 查看菜单接口：customer/getallmenu

请求方式：get

参数：无

响应：

```
[
    {
        "seller": 641832841,
        "name": "kfc2",
        "price": 27
    },
    {
        "seller": 641832841,
        "name": "kfc3",
        "price": 28
    }
]
```

```
    },  
    {  
      "seller": 641832841,  
      "name": "kfc4",  
      "price": 29  
    },  
    {  
      "seller": 641832841,  
      "name": "kfc5",  
      "price": 30  
    }  
  ]  
}
```

5. 请求自己的订单信息: customer/myorder

请求方式: get

参数: 无

响应: 和上面的一样

Ridder

//这个接口意义不大, 所以没做

1. 请求所有的订单信息: rider/allorder

请求方式: get

参数: 无

响应: {
 "num": 23,
 "seller": 641832841,
 "name": "kfc5",
 "customer": 54206030,
 "rider": 0
}

2. 请求没被强过的订单: rider/orderable

请求方式: get

参数: 无

响应: 就是一个 order{

```
"num": 23,  
"seller": 641832841,  
"name": "kfc5",  
"customer": 54206030,  
"rider": 0  
}
```

3. 查看自己抢到的订单: rider/myorder

请求方式: get

参数: 无

响应: {

```
"num": 23,  
"seller": 641832841,  
"name": "kfc5",  
"customer": 54206030,  
"rider": 0  
}
```

4. 抢单: rider/rob

请求方式: post

参数: {

```
"num": 23,  
}
```

响应: Result 对象

Seller

1. 上菜: seller/provide

请求方式: put

参数: 写在 url 的 name 和 price

响应：Result 对象

2. 下架菜品：seller/delete

请求方式：del

参数：写在 url 的 name

响应：Result 对象

3. 查看自己上架的所有菜品：seller/mymenu

请求方式：get

参数：无

响应：[
 {
 "seller": 641832841,
 "name": "kfc2",
 "price": 27
 },
 {
 "seller": 641832841,
 "name": "kfc3",
 "price": 28
 }
]

4. 查看自己的所有订单：seller/myorder

请求方式：get

参数：无

响应：{
 "num": 23,
 "seller": 641832841,
 "name": "kfc5",
 "customer": 54206030,
 "rider": 0
}