

## 1. 菜品查询

发起请求的网址:

<http://localhost:3000/api/v1/searchFood>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: get

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

发起请求的例子:

<http://localhost:3000/api/v1/searchFood?username=YoshiPark&key=s2354dasd#123>

发起请求返回的 json 格式:

```
[
{
  "foods": {
    "0": {
      "name": "宫保鸡丁",
      "price": 15,
      "oldprice": 20,
      "sellcount": "0",
      "count": "0",
      "rating": "100",
      "info": "好吃",
      "icon": null,
      "image": null
    },
    "1": {
      "name": "扁豆",
      "price": 1,
      "oldprice": 2,
      "sellcount": "0",
      "count": "0",
```

```
        "rating": "100",
        "info": "难吃",
        "icon": null,
        "image": null
    }
},
"name": "热销榜",
"type": 0
},
{
    "foods": {
        "0": {
            "name": "咸骨粥",
            "price": 5,
            "oldprice": 10,
            "sellcount": "0",
            "count": "0",
            "rating": "100",
            "info": "好吃",
            "icon": null,
            "image": null
        }
    },
    "name": "热菜",
    "type": 1
}
]
```

```
{
  "data": [
```

```
{
  "Username": "YoshiPark",
  "Dishname": "宫保鸡丁",
  "DishOldprice": 20,
  "Dishprice": 15,
  "Dishtype": 0,
  "Dishimage": null,
  "Dishicon": null,
  "Dishinfo": "好吃",
  "Dishdescription": "好吃",
  "Dishtypename": "热销榜"
}
]
```

#### 4.更新菜品

发起请求的网址:

<http://localhost:3000/api/v1/searchFood/updatefood>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

dishname(菜品的名字)

disholdprice(菜品原先的价格, 如果商家没有填写, 就不需要传上来, 或者以 null 值传上来也可)

dishprice(菜品现在的价格)

dishtype(菜品代表所在的类型的数字, 比如是热销榜的话, 为 0, 甜品的话, 为 1, 到时候前端给一个表对应即可)

dishtypename(菜品所在的类型, 比如热销榜, 甜品等)

dishinfo(前端给的信息)

dishdescription(前端给的信息)

另: 图片 image 和 icon 的上传, 由另外的 url 进行处理, 到时候再给出 url 地址  
请求成功返回的例子:

Result = { code:200, msg:"更新成功"}

## 5.删除菜品

发起请求的网址:

<http://localhost:3000/api/v1/searchFood/delfood>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

dishname(菜品的名字)

发起请求的例子:

<http://localhost:3000/api/v1/searchFood/delfood?username=YoshiPark&dishname=宫保鸡丁&key=s2354dasd#123>

请求成功返回的例子:

Result = { code:200, msg:"删除成功"}

6.商家全部订单查询(按照时间排序,在这里由于是全部订单查询, 为了优化不给出订单详情, 如果需要商家想要查看订单详情, 需要点击某个订单查看)

发起请求的网址:

<http://localhost:3000/api/v1/searchOrder>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: get

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

发起请求的例子:

<http://localhost:3000/api/v1/searchOrder?username=YoshiPark&key=s2354dasd#123>

请求成功返回的例子:

```
{
  "data": [
    {
      "Username": "YoshiPark",
      "Ordernumber": "011",
      "Ordertime": "2017-07-12 06:30:21",
      "Tablenumber": 3,
      "Price": 20
    },
    {
      "Username": "YoshiPark",
      "Ordernumber": "010",
```

```
    "Ordertime": "2017-01-22 04:30:15",
    "Tablenumber": 2,
    "Price": 15
  },
  {
    "Username": "YoshiPark",
    "Ordernumber": "012",
    "Ordertime": "2016-06-23 04:30:21",
    "Tablenumber": 4,
    "Price": 50
  }
]
```

## 7. 插入一个订单

发起请求的网址:

<http://localhost:3000/api/v1/searchOrder/addorder>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

ordernumber(订单号, 如果商家没有指定, 那么就直接用商家注册的账号+日期+桌号代替)

ordertime(订单下单的时间, 格式为 yy-mm-dd hh:mm:ss)

tablenumber(订单所在的桌号)

price(订单的总金额)

orderdetail(订单详情, 这是一个数组, 数组里面有四个元素)

orderdetail[i].dishname(菜名)

orderdetail[i].dishcount(点这个菜的数量)

orderdetail[i].tasternote(客户点这个菜的备注)

orderdetail[i].price(当时这个菜的单价)

请求成功返回的例子:

Result = { code:200, msg:"增加成功"}

## 8. 查询一个订单

发起请求的网址:

<http://localhost:3000/api/v1/searchOrder/queryorder>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

username(商家注册的账号)

key(安全起见, 需要加一个 key, 这个 key 由商家注册时候服务器给出, 每次商家登陆时候服务器都会给前端返回一个 key, 需要前端保管)

ordernumber(订单号, 如果商家没有指定, 那么就直接用商家注册的账号+日期+桌号代替)

## 9.商家注册:

发起请求的网址:

<http://localhost:3000/users/addUser>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

username(商家注册的账号)

password(商家注册的密码, 希望前端能够进行 MD5 加密)

shopname(商家注册的商家名)

tablename(商家拥有的台数)

phonenum(商家的电话号码)

请求成功返回的例子:

Result = { code:200, msg:"增加成功"}

## 10.商家提供银行卡

发起请求的网址:

<http://localhost:3000/bankcard/addCard>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

account: 银行卡的账号

accountname:银行卡的持有人

username:商家注册的账号

请求成功返回的例子:

Result = { code:200, msg:"增加成功"}

## 11.商家删除银行卡

发起请求的网址:

<http://localhost:3000/bankcard/delCard>

注:将来 localhost 改为服务器的 IP 地址

发起请求的方式: post

发起请求的参数:

account: 银行卡的账号

**accountname:**银行卡的持有人

**username:**商家注册的账号

请求成功返回的例子:

**Result** = { code:200, msg:"删除成功"}