API 接口文档

基于 axios 封装一个 ajax 请求函数模块

```
/**
 * ajax 请求函数模块
 * 返回值：Promise
*/
import axios from "axios"
export default function ajax(url,data={},type="GET"){
    return new Promise((resolve,reject)=>{
        let promise
        // 处理异步请求
        if(type === "GET"){
            // 拼装请求数据
            let dataStr = "";
            Object.keys(data).forEach((key)=>{
                dataStr += key + "=" + data[key] + "&"
            })

            if(dataStr !== ""){
                dataStr = dataStr.substring(0,dataStr.lastIndexOf("&"))
                url = url + "?" + dataStr
            }
            console.log(url)
            // 发送 ajax 请求
            promise = axios.get(url) // 返回值一个 Promise 的实例对象
        }else if(type === "POST"){
            //POST 请求
            console.log(url)
            console.log(data)
            promise = axios.post(url,data) // 返回值是一个 Promise 的实例对象
        }else{
            promise = axios.delete(url,data)
        }
        promise.then((response)=>{
            resolve(response.data) // 把请求服务器的数据从外层的promise的resolve方法返
回出去
        }).catch((error)=>{
```

```
            reject(error)// 把请求的错误信息给出
        })
```

```
    })
}
```

本次是服务器设置跨域

```
const cors = require('cors')
app.use(cors())
```

具体的代码实现请参考：https://github.com/huweiu/monter_taste_server

前台应用相关接口

1. 获取短信验证码 get 请求

请求 URL：http://localhost:6000/api/users/sendcode

参数列表：

参数	是否必选	类型	说明
phoneNum	Y	string	手机号码

2. 短信验证码注册或者登录 post 请求

请求 URL：http://localhost:6000/api/users/loginSms

参数	是否必选	类型	说明
phoneNum	Y	string	手机号
Code	Y	string	验证码

3. 密码登录 post 请求

请求 URL：http://localhost:6000/api/users/login_pwd

参数	是否必选	类型	说明
userName	Y	string	手机号码
passWord	Y	string	密码
captcha	Y	string	验证码
captcha_id	Y	string	验证码数据库 ID

4. 用户注册 post 请求

请求 URL：http://localhost:6000/api/users/register

参数	是否必选	类型	说明
userName	Y	string	用户名
passWord	Y	string	密码
phoneNum	Y	phoneNum	手机号

5. 查询用户是否已经被注册 get 请求

请求 URL：http://localhost:6000/api/users/check_name

参数	是否必选	类型	说明
userName	Y	string	用户名

6. 获取图形验证码 get 请求
请求 URL：http://localhost:6000/api/users/captcha
不需要传递参数

7. 异步请求服务器实现用户的自动登录 get 请求
请求 URL：http://localhost:6000/api/users/getuserInfo

参数	是否必选	类型	说明
userName	Y	string	用户名

8. 把购物车数据保存在数据库中 post 请求
请求 URL：http://localhost:6000/api/users/saveCartDataToDataBase
参数：dataObj 为一个对象
dataObj 包含的字段为：

```
dataObj = {
    user_id:"", // 保存用户 ID 唯一标识
    foodName:"", // 商品名称
    desc:"", // 商品描述
    price:"", // 商品价格
    discount,"", // 商品折扣
    fee:"", // 配送费
    rate:"", // 好评率
    order:"", // 订单数量
    foodinfo;"", // 食物信息
    pic:"", // 美食图片
    choiceFlag:"", // 是否被选中
    isSale:"" // 是否已付款
}
```

9. 获取购物车的数据展示在前端页面 get 请求
请求 URL：http://localhost:6000/api/users/getCartDataFromDataBase

参数	是否必选	类型	说明
user_is	Y	string	保存用户 ID 唯一标识(数据提前保存在 localStorage 中)

10. 修改购物车数据 post 请求
请求 URL：http://localhost:6000/api/users/updateCartData
参数为 dataObj 是一个对象

```
dataObj = {
    _id:"", // mongoodb 数据库中唯一标识
    user_id:"", // 保存用户 ID 唯一标识
    foodName:"", // 商品名称
    desc:"", // 商品描述
    price:"", // 商品价格
    discount,"", // 商品折扣
```

fee:"", // 配送费
rate:"", // 好评率
order:"", // 订单数量
foodinfo;"", // 食物信息
pic:"", // 美食图片
choiceFlag:"", // 是否被选中
isSale:"" // 是否已付款
}

11. 购物车删除数据库中单条数据记录 delete 请求
请求 URL：http://localhost:6000/api/users/delsignalCartData/${_id}
|参数 |是否必选 |类型 |说明
|_id | Y |string |mongoodb 数据库中唯一标识

12. 删除当前用户下数据库中所有购物车数据记录 delete 请求
请求 URL：http://localhost:6000/api/users/delsignalCartData/${user_id}
|参数 |是否必选 |类型 |说明
|user_id |Y |string |当前用户的唯一标识

后台管理系统相关 API

1. 后台管理系统用户注册接口 post 请求
请求 URL：http://localhost:6000/api/users/admin_register
传递参数为一个对象
registerUser = {
name："", // 用户名
email:"", // email
password:"", // 密码
identity:"" // 用户身份(管理员/普通员工)
}

2. 后台管理系统用户登录接口 post 请求
请求 URL：http://localhost:6000/api/users/admin_login
传递参数为一个对象
loginUser = {
email:"", // email 邮箱
password:"", // 登录密码
}

3. 往数据库中增加食品分类(增加单条数据信息)post 请求
请求 URL：http://localhost:6000/api/users/admin_addClassify
传递参数为一个对象
data = {
ClassifyNo:"", // 分类编号

ClassifyName:"", // 分类名称
foodName:"",
description:"",
pic:"",
price:"", // 食物价格
discount:"", // 折扣价格
fee:"", // 配送费
rate:"" , // 好评得分
order:"", // 月售订单数量
foodinfo:"" // 食物组成信息
}

4. 获取食品分类列表后台管理系统数据(所有的数据信息)get 请求
请求 URL：http://localhost:6000/api/users/getClassifyDataBase
不需要传递参数

5. 编辑食品分类后台管理系统数据 (编辑修改单条信息)post 请求
请求 URL：http://localhost:6000/api/users/admin_editClassify
传递参数为一个对象
data = {
ClassifyNo:"", // 分类编号
ClassifyName:"", // 分类名称
foodName:"",
description:"",
pic:"",
price:"", // 食物价格
discount:"", // 折扣价格
fee:"", // 配送费
rate:"" , // 好评得分
order:"", // 月售订单数量
foodinfo:"" // 食物组成信息
}

6. 获取 食品分类 后台管理系统数据（单条数据信息）get 请求
请求 URL：http://localhost:6000/api/users/getClassifyInfo
|参数 |是否必选 |类型 |说明
|id |Y |string |保存用户 ID 唯一标识(数据提前保存在 localStorage 中)

7. 删除后台 食品分类 管理系统单条数据 (删除单条数据信息) delete 请求
请求 URL：http://localhost:6000/api/users/admin_DelClassify/${id}
|参数 |是否必选 |类型 |说明
|id |Y |string |数据库该条记录唯一 id

在路由相关的代码中增加路由守卫

```
// 实现路由守卫

router.beforeEach((to,from,next)=>{
  // 判断 localStorage 中是否拥有 token
  const isLogin = localStorage.admin_login_token ? true : false
  if(to.path === "/login" || to.path === "/register"){
    next();
  }else{
    isLogin ? next() : next("/login")
  }
})
```