

助农商城

made by 文雨晨 1913143

小程序说明

本小程序为2022年微信小程序大赛参赛作品，选题为农产品销售信息平台，本文档适用于描述迭代3(v1.2.2)，小程序的完整文档请见迭代4文档。

应用场景

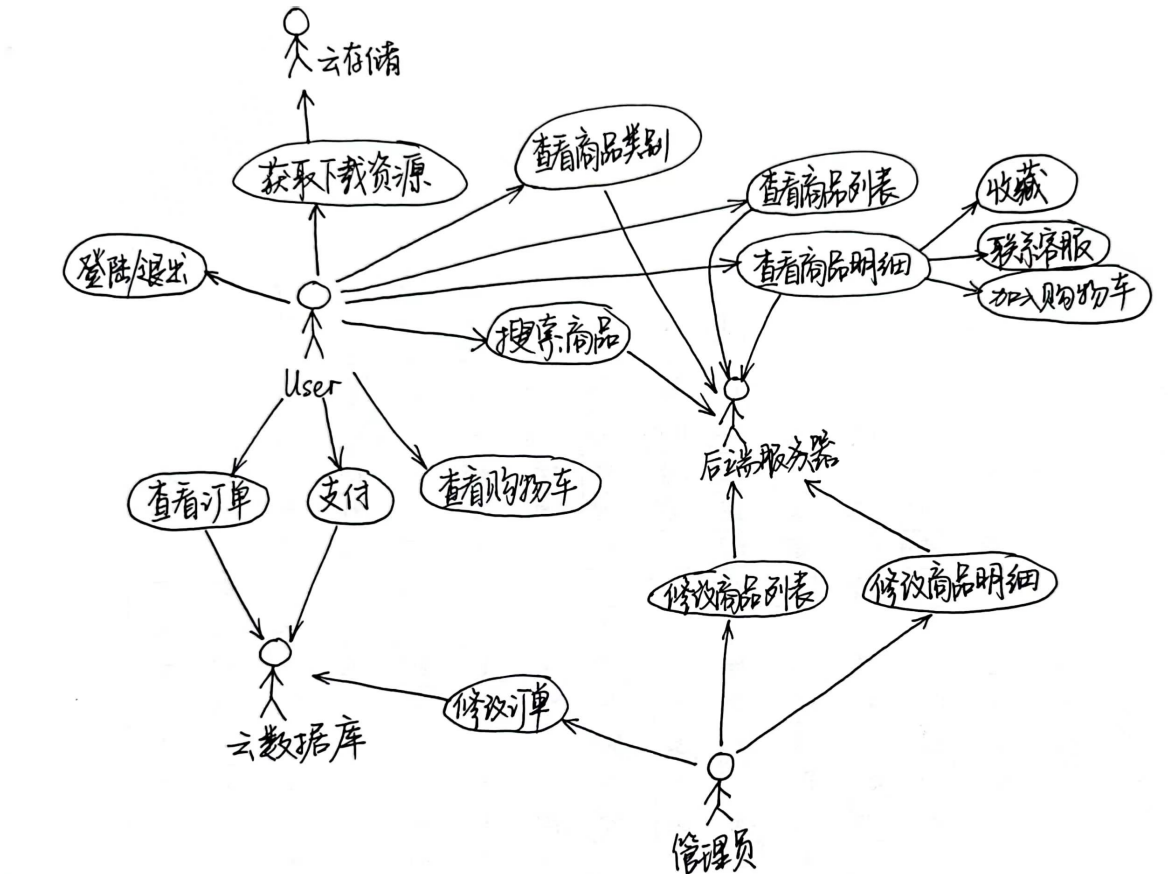
本小程序解决农村购物难、农产品难销售的电商问题，借助微信小程序和电商产业帮助农民销售各类农产品，同时帮助城镇居民方便快捷购买农产品。

实际问题背景

随着手机和移动支付的普及，以及微信小程序的推广，电商越发成为农业农村发展的新机遇。目前的农民收入中，经营性收入占30%多一点，务工收入占45%左右，家庭经营收入主要靠农产品销售，通过电商销售农产品成为农民增收的一个亮点。在实际电商销售过程中，对于农产品的规模、物流、品控都有较高的标准和要求，这对于一般农户、或者刚起步的创业者来说很困难。希望有一些结合当地旅游、美食相关的小程序，让游客查询到当地土特农产品，去到原产地边游玩边购买；也可以填写在当地临时住址，方便农户们送货上门。

需求和功能说明

本小程序的用例图如下：



用例说明如下：

- 用户：包括User、管理员、后端服务器和云存储、云数据库

- 登录/退出：用户使用时可以选择使用微信账号登录/退出小程序账号
- 获取下载资源：一些较大图片、音视频资源存放于云存储中，客户需要获得相应资源时，需要从云存储中下载
- 查看商品类别：用户在商品分类页面可以查看全部商品大类和每个大类中的各小类商品，相应数据需要向后端服务器请求获得
- 查看商品列表：用户可查看一个列表中的全部商品，相应数据需要向后端服务器请求获得
- 查看商品明细：用户可以查看某一商品的价格、图文详情等了解商品，并在此基础上使用收藏、联系客服、加入购物车功能
- 收藏：用户将自己喜爱的商品放入收藏夹
- 联系客服：用户可以通过微信与客服进行文本、语音、图片等形式的对话
- 加入购物车：用户将想购买的物品放入购物车
- 搜索商品：用户通过输入关键字，可以搜索到所有相关的商品信息，相应数据需要向后端服务器请求获得
- 查看购物车：用户可以查看购物车中已有的全部商品信息，并调整购买数量、增删商品
- 支付：用户可以勾选想购买的物品并使用微信支付模块进行支付，支付信息和订单信息存储在云数据库中
- 查看订单：用户可以查询近期全部订单的编号、总价、创建时间，支付信息和订单信息存储在云数据库中
- 修改商品列表：管理员可以通过修改后端服务器数据，实现更改商品列表中包含的商品
- 修改商品明细：管理员可以通过修改后端服务器数据，实现更改某一商品的明细，如价格、图文详情等
- 修改订单：管理员通过更改云数据库中的信息，更改某一订单的完成状态、属性，模拟货物的发货、运输、签收状态的变更

本次迭代已实现功能点

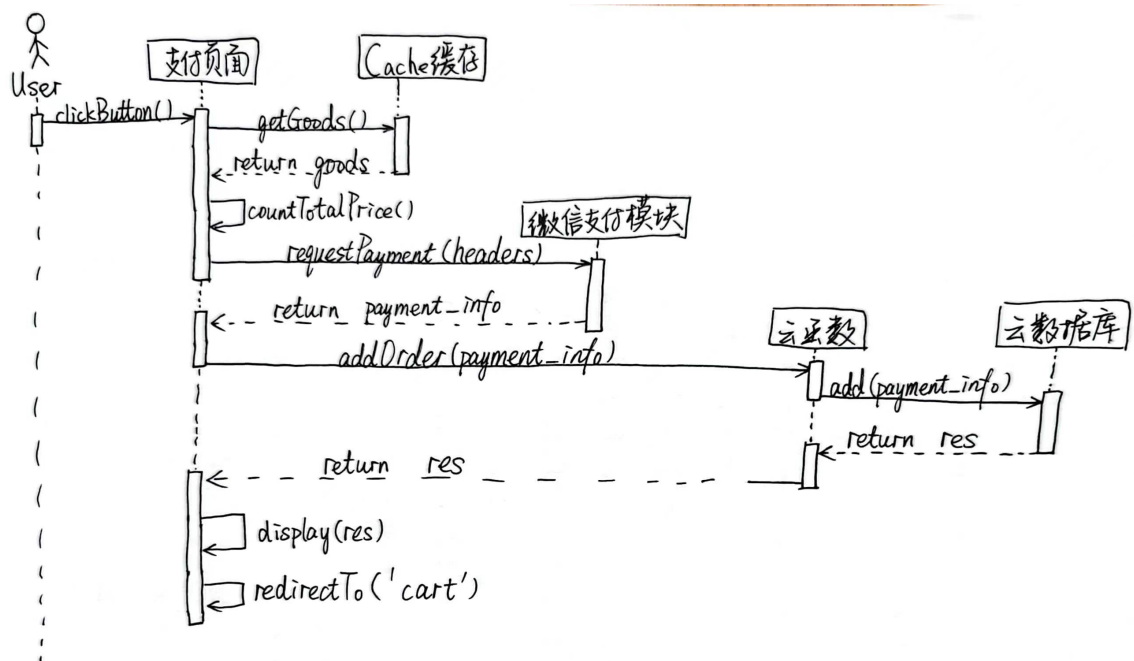
1. 商品分类和商品列表展示
2. 商品明细展示（价格、详细图文）
3. 商品收藏、取消收藏、加入购物车、分享
4. 客服交流（文字、图片、视频等）
5. 购物车相关功能（添加购物车、增删商品、结算）
6. 支付功能
7. 搜索和排序功能（待后续迭代完善）
8. 页面ui设计（配色、菜单栏、页面跳转等）
9. 云开发（云存储等）

技术实现说明

1.前后端数据交互（云函数、云数据库、request）

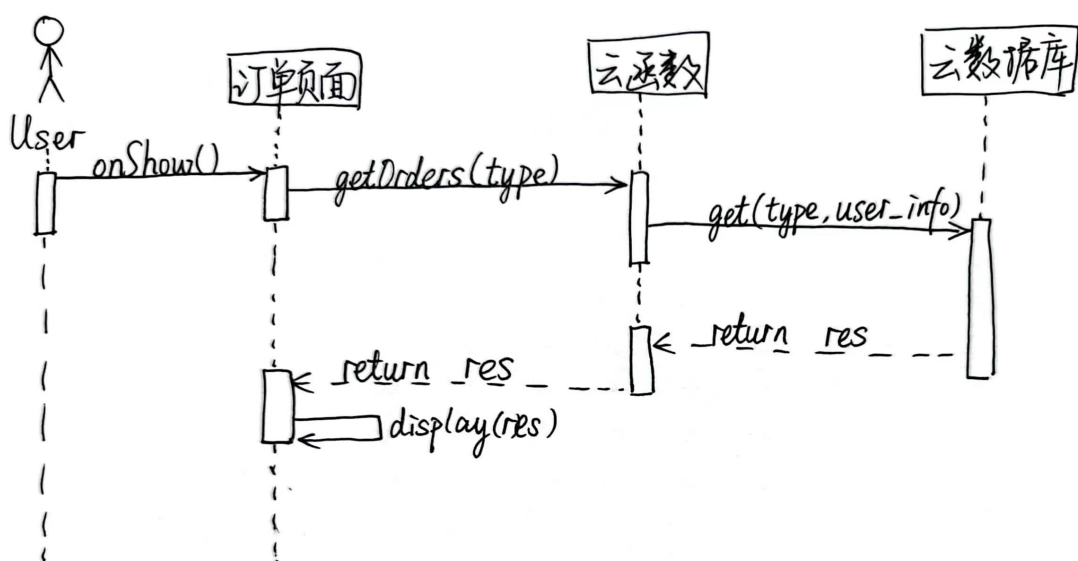
购物类小程序的重点技术是前后端数据交互，主要应用于商品类别获取、商品列表获取、商品详情获取、订单生成和获取等等。本小程序尝试使用两种方式实现前后端数据的交互，即微信云开发和request请求后端服务器。下面，以生成和获取订单为例，展示微信云开发的技术实现；以获取商品明细为例，展示request请求的技术实现。

生成和获取订单：



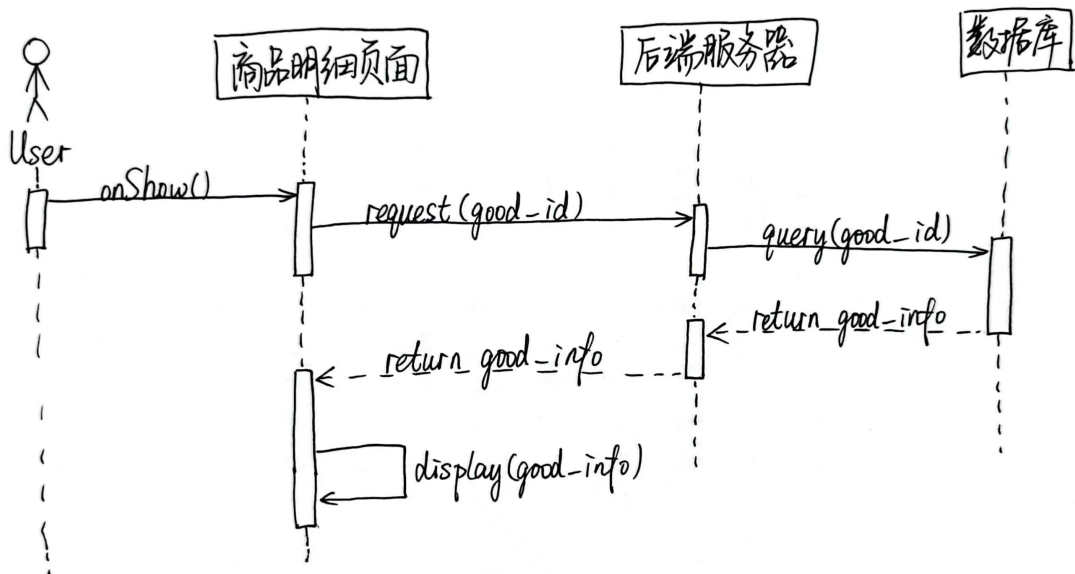
使用云函数，可以直接通过wxContext对象获得用户的openid等用户信息，并且能方便操纵云数据库进行增删改查操作。因此，使用获得的用户openid信息、再加上支付的相关信息（总价格、创建时间、订单状态），就可以创建一个完整的用户订单并存入云数据库中。

实际操作时，由于微信云函数可以直接获得用户openid信息而不需授权、操作数据库更加方便，因此，直接调用云函数，由云函数再操纵云数据库添加记录。



原理同上，并且由于直接查询云数据库有20条记录的限制，而云函数查询则无限制，因此采用这种方式获得全部订单。

获取商品明细：



采用request方式是因为，商品类别、明细通常较为固定，基本不会有增删改的操作，因此，直接采用后端接受请求、查询数据库的方式，减轻云数据库的工作量。

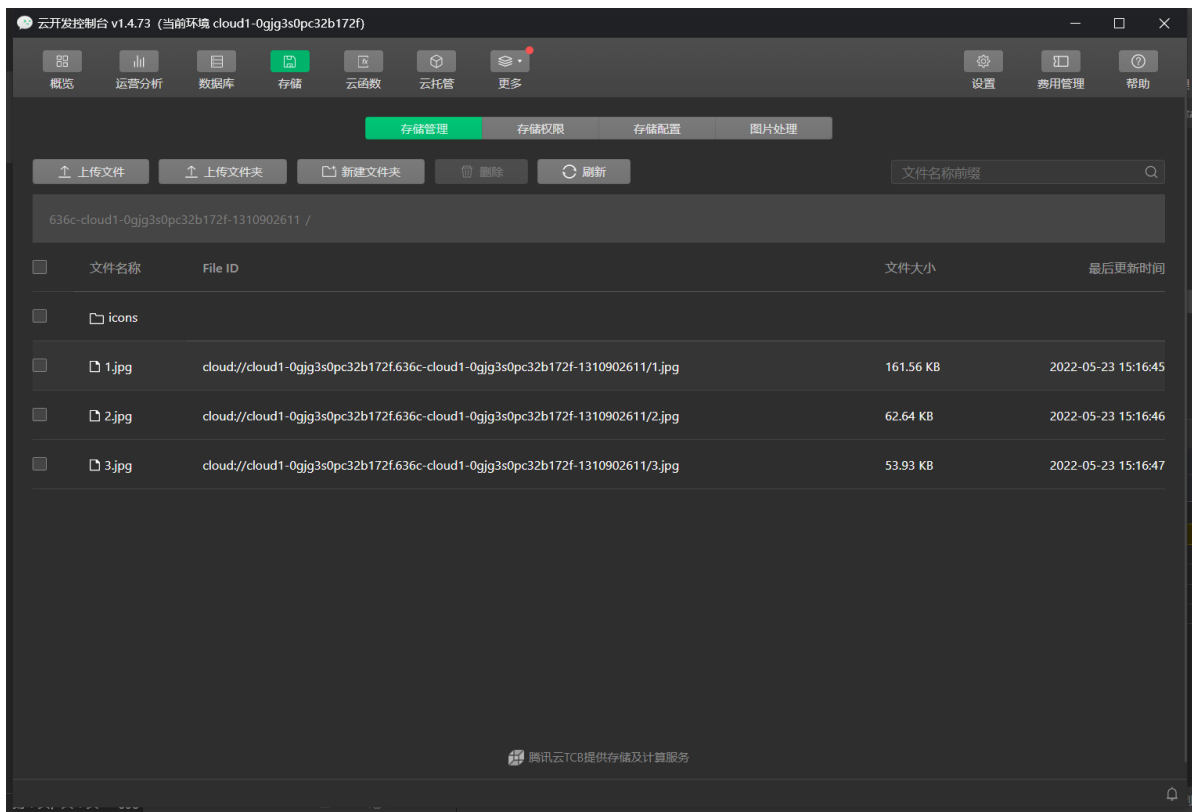
2.商品数据的爬取和预处理

本小程序数据采集自苏宁易购，并存储于后端数据库，主要采集的内容是商品类别、商品明细。需要使用时，向后端服务器发送请求即可。为实现商品搜索功能，对每个商品明细进行了分词、词频统计，结果存储在数据库中。搜索关键字时，按词频高低倒序返回结果。



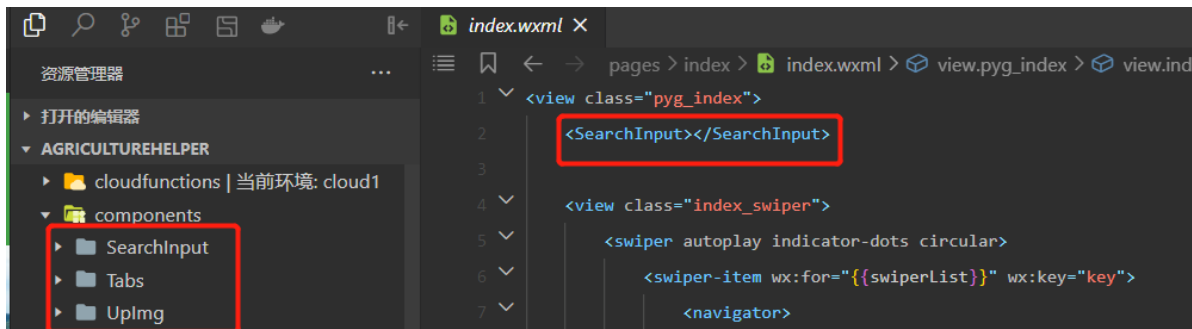
3.云存储

由于应用中含有一些较大的图片文件，直接放在小程序目录中会显著增加小程序体积、延缓小程序进入时间，因此将这些文件放入云存储中，用户浏览到相应页面后再异步加载。考虑到本小程序的应用场景，没有需要用户上传数据、文件的功能需求，因此没有使用云存储的上传功能。



4.自定义组件的开发

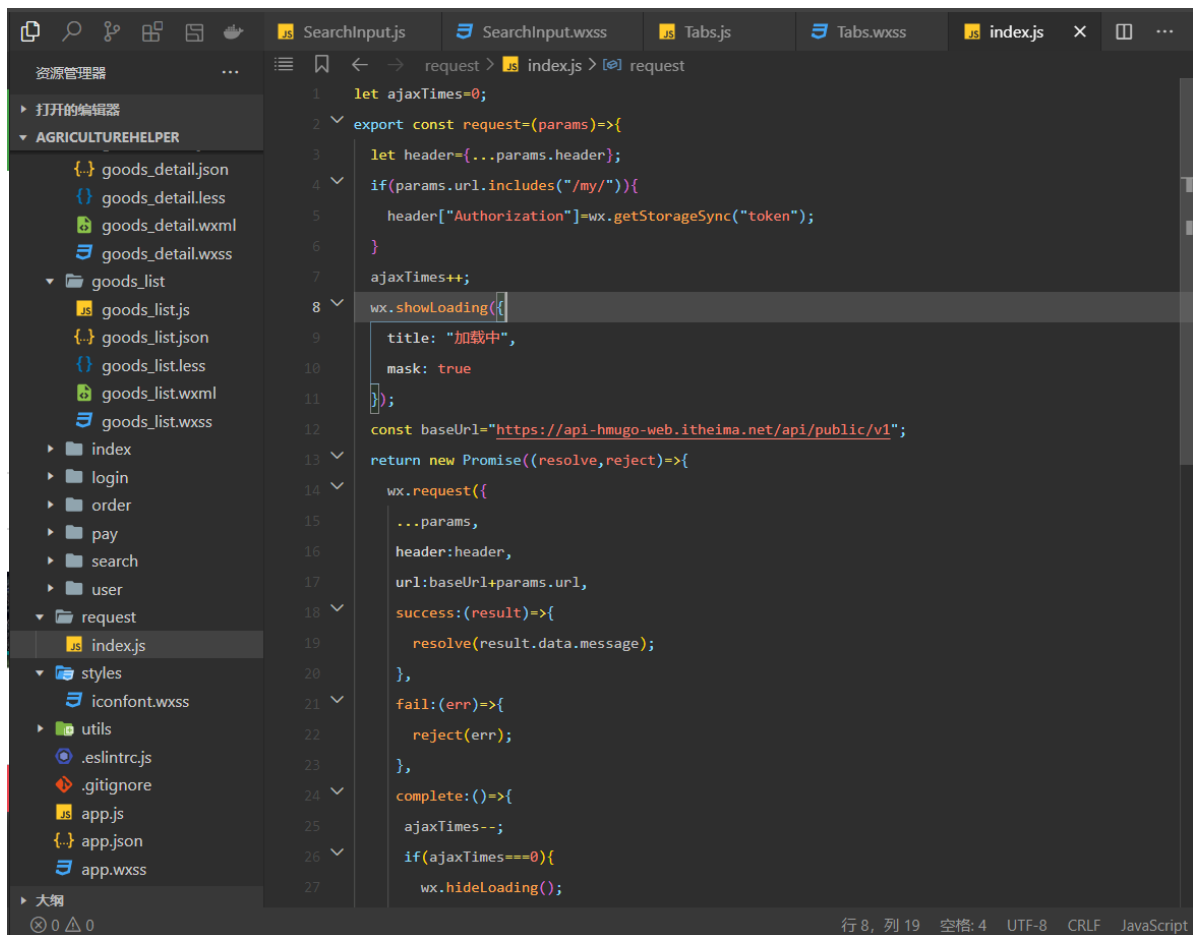
对小程序中反复出现的组件，包括搜索栏、菜单选择栏、商品缩略图，进行了封装，便于重复使用。



产品实现说明

1.request的统一封装

由于服务器地址不变，发送不同请求时只需要输入不同路径，因此将整个发送请求、接受请求和返回数据的过程封装在request.js，请求和相应的过程中，统一显示“加载中”字样。



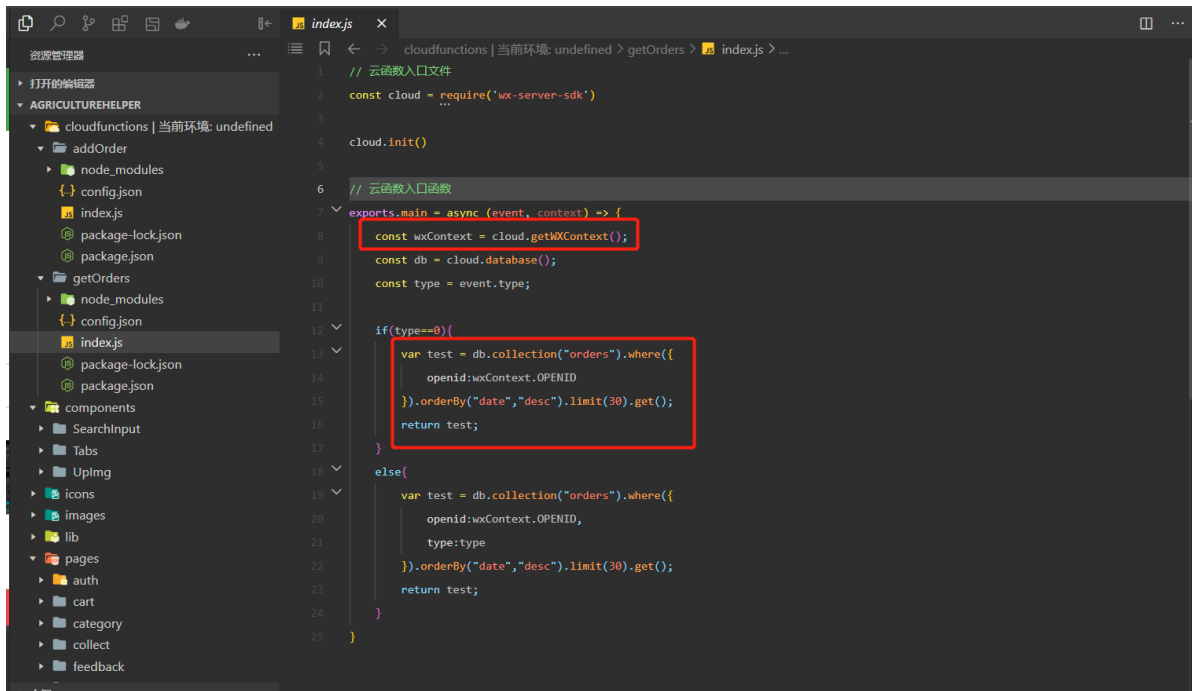
2.设置页面的请求参数，实现不同效果

例如，将商品id作为参数，发送请求给后端，服务器根据id向数据库发起查询，将返回结果渲染到页面上，即实现了不同商品的填充。



3.云函数的使用，充分利用云开发的优点

在与云数据库进行交互时，云函数调用的方式可以便捷地获得当前环境下的数据库，并且返回值不限于30条。云函数可以直接编写，微信云开发省去了复杂的后端配置环节，开发者直接编写云函数即可，非常方便。此外，云函数可直接通过cloud.getWXContext()快速获得用户的openid等信息。



4.数据动态加载

通过双滚动栏和异步请求的方式，实现了各类商品数据的动态加载。包括商品列表中各种商品的加载，每当检测到用户划到列表底部，就再尝试获取新的数据填充列表，实现动态加载的效果。

商品分类



搜索

大家电

/ 大牌推荐 /

热门推荐







海外购

荣耀

华为

三星

苏宁房产


魅族手机





手机相机

魅族

OPPO

vivo







电脑办公

努比亚

美图

诺基亚

厨卫电器



食品酒水

飞利浦

居家生活

/ 搞机达人 /

厨房电器



拍照手机

游戏手机

好评过万

生活电器



个护健康

全面屏手机

大屏手机

老人机



商城首页



商品分类



购物车



我的

```

47  ✓   handleTabsItemChange(e){
48      const {index}=e.detail;
49      let {tabs}=this.data;
50      tabs.forEach((v,i)=>i===index?v.isActive=true:v.isActive=false);
51  ✓   this.setData({
52       |   tabs
53       | })
54  },
55  ✓   onReachBottom(){
56  ✓   if(this.QueryParams.pagenum>=this.totalPages){
57       |   wx.showToast({ title: '没有下一页数据' });
58       |
59  ✓   }else{
60       |   this.QueryParams.pagenum++;
61       |   this.getGoodsList();
62       |   }
63  },

```

5. 下拉刷新页面

使用onPullDownRefresh()函数，每次下拉时，即重新加载一次数据，达到刷新的效果。

```

64  ✓   onPullDownRefresh(){
65  ✓   this.setData({
66       |   goodsList:[]
67       | })
68       |   this.QueryParams.pagenum=1;
69       |   this.getGoodsList();
70       |   }
71  })

```

6. 客服交谈

通过button按钮的contact，即可使用官方提供的客服交流模块，直接与小程序预先分配的客服进行文字、图片的交流。

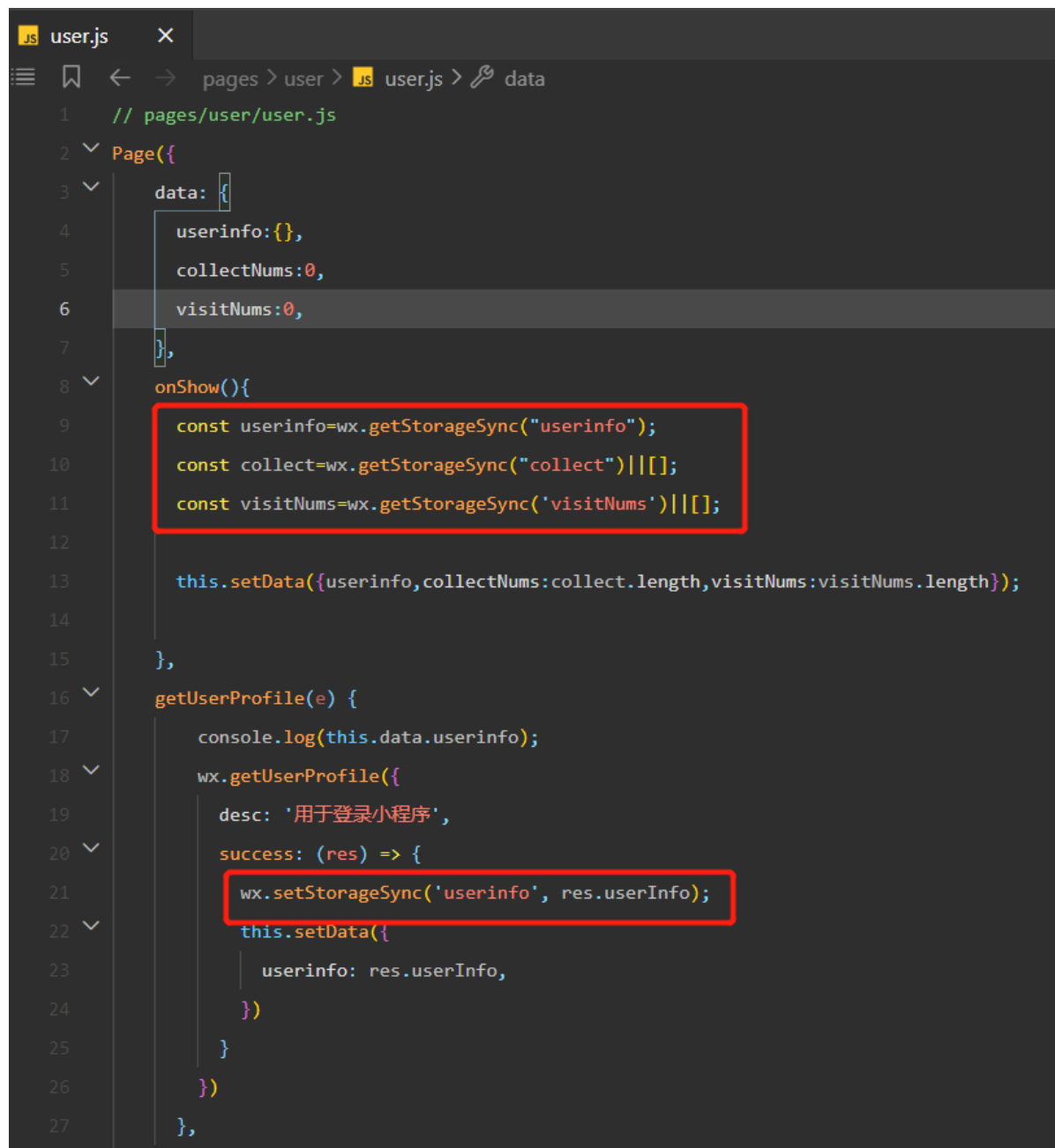
```

<view class="iconfont icon-kefu"></view>
<view>客服</view>
<button open-type="contact"></button>

```

7.通过缓存保存数据

使用者的个人信息在登陆后应当自动保存一段时间，此外，加入购物车的物品也需要在结算前一直保持，对于这样的数据，采用缓存的方式，暂存在用户的终端上，使用时再加载即可。



8.图片放大查看功能

通过wx.previewImage(), 将图片进行放大查看，方便使用者仔细查看商品。

```
handlePreviewImage(e) {  
    const urls = this.GoodsInfo.pics.map(v => v.pics_mid);  
    const current = e.currentTarget.dataset.url;  
    wx.previewImage({  
        current,  
        urls  
    });  
},
```

致谢

特别感谢《智能移动开发》这门课程，通过课程讲授、上机实践、参赛的方式，给予我机会充分学习了移动应用，包括app、小程序等流行软件的开发方式和特点，并通过这一小程序充分学习了微信小程序的开发思想、云开发的优点等，对微信小程序开发有了更加深刻的理解和认识，感受颇多。但是，我也充分认识到当前掌握的知识和技能还远远不够，小程序中仍存在很多不完善的地方，还有微信云托管技术需要学习，希望自己能在未来的学习和开发生涯中不断精进技术，成为更优秀的开发者。在此，感谢师老师、两位助教学长一学期的辛勤付出和微信小程序大赛官方提供的宝贵机会！