

智能移动开发小程序说明文档

软件学院 1813008 刘宇鑫

1.概述

1.1开发目标与意义

在疫情防控常态化、实体行业不景气、电子商务持续发展的当下，结合本次2022微信小程序应用开发赛所选的“用科技创造社会价值与助力乡村振兴”的主题，我决定开发一款可以专注于解除乡村特产等商品所受到的时空局限性、打通城市乡村之间销售渠道的线上商城小程序，创造社会价值、助力经济发展。

1.2开发环境

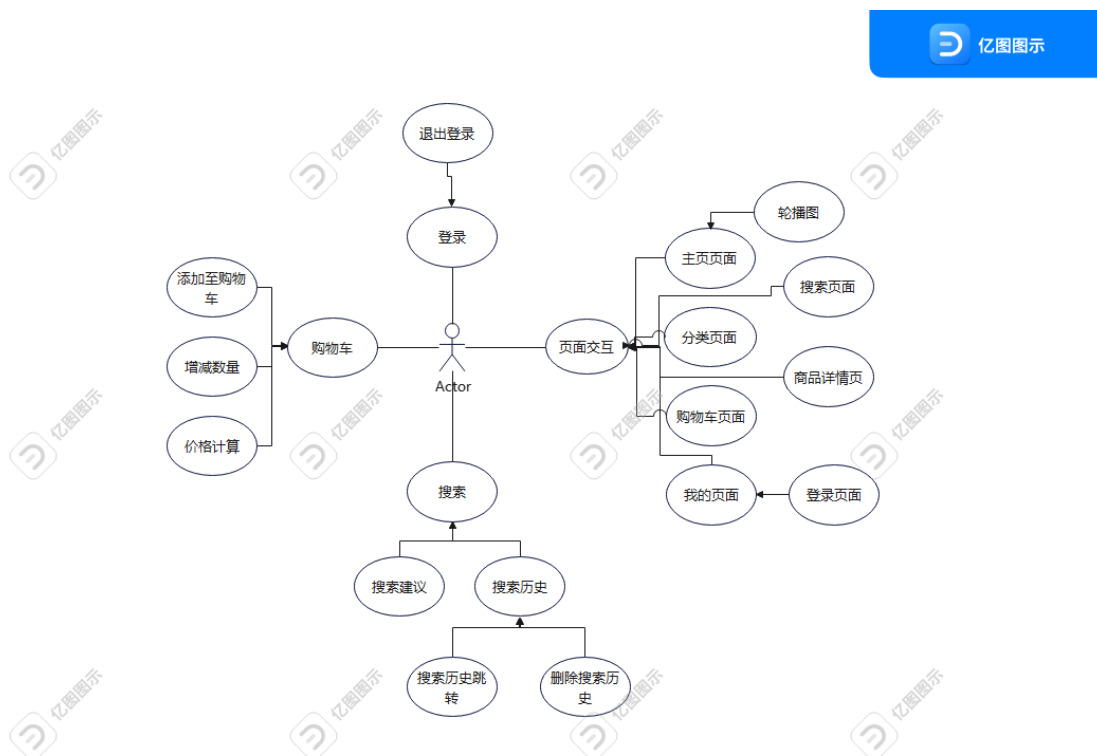
前端框架：uni-app

开发平台：HBuilderX、微信开发者工具

系统环境：Windows 10 专业版 19044.1706

部分素材来源接口：<https://api-ugo-web.itheima.net/>

1.3功能与需求（UML图展示）

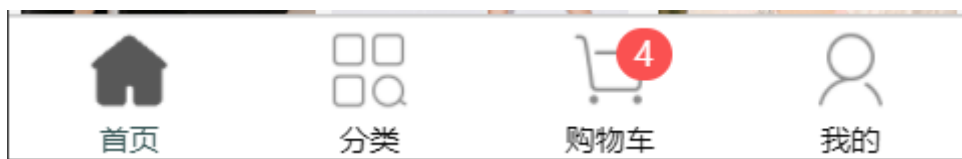


2.迭代一

1.tabBar的实现

在 pages 目录中，创建了首页(home)、分类(cate)、购物车(cart)、我的(my) 4 个页面；

修改pages.json配置文件，新增配置节点并修改globalStyle节点以修改导航条的样式效果：



2.轮播图区域的实现

在 data 中定义轮播图的数组swiperList: [];

在 methods 中定义获取轮播图数据的方法getSwiperList(), 发起请求并在请求成功后为 data 中的数据赋值；

在 onLoad 生命周期函数中调用getSwiperList();

渲染轮播图的 UI 结构，使用v-for循环渲染轮播图的item 项，动态绑定图片的 src 属性；

将swiper-item节点内的view 组件，改造为 navigator 导航组件，并动态绑定 url 属性的值，为之后点击跳转至商品详情页做准备：

```
<navigator class="swiper-item" :url="'/subpkg/goods_detail/goods_detail?
goods_id=' + item.goods_id">
  <image :src="item.image_src"></image>
</navigator>
```

3.分类导航区域的实现

在 data 中定义分类导航的数据列表navList: [];

在 methods 中定义获取数据的方法getNavList();

在 onLoad 生命周期函数中调用getSwiperList();

定义 navClickHandler 事件处理函数，并为 nav-item 绑定，实现点击第一项，切换到分类页面

```
// nav-item 项被点击时候的事件处理函数
navClickHandler(item) {
  // 判断点击的是哪个 nav
  if (item.name === '分类') {
    uni.switchTab({
      url: '/pages/cate/cate'
    })
  }
}
```



3.迭代二

1.分类页面

1.1基本结构

定义左侧的滚动视图区域"left-scroll-view"和右侧的滚动视图区域"right-scroll-view";

动态计算剩余高度以适应窗口

```
// 获取当前系统的信息
const sysInfo = uni.getSystemInfosSync()
// 为 wh 窗口可用高度动态赋值
this.wh = sysInfo.windowHeight
```

1.2获取分类数据

在 data 中定义分类数据列表cateList: [];

在 methods 中定义获取数据的方法getCateList();

在 onLoad 生命周期函数中调用getCateList();

1.3左侧一级分类

循环渲染结构时, 为选中项动态添加 .active 类名i === active ? 'active' : "", 为一级分类的 Item 项绑定点击事件处理函数 activeChanged, 实现动态修改选中项的索引

```
activeChanged(i) {
  this.active = i
}
```

在 data 中定义 滚动条距离顶部的距离scrollTop: 0, 动态为右侧的 组件绑定 scroll-top 属性的值, 修改 activeChanged 方法以实现切换一级分类后重置滚动条的位置

```
this.scrollTop = this.scrollTop ? 0 : 1
```

1.4右侧二级分类

定义二级分类列表cateLevel2: [];

分别修改 getCateList 方法实现为二级分类列表数据赋值， activeChanged 方法实现在一级分类选中项改变之后，为二级分类列表数据重新赋值

```
// 为二级分类赋值
this.cateLevel2 = res.message[0].children
// 为二级分类列表重新赋值
this.cateLevel2 = this.cateList[i].children
```

苏宁房产	/ 电脑整机 /		
手机相机			
电脑办公	轻薄本	平板电脑	台式机
厨卫电器	一体机	组装机	电脑清洗
食品酒水			
	电脑维修		
居家生活	/ 电脑外设 /		
厨房电器			

2.搜索功能

1.自定义搜索组件

自定义uni-app组件 my-search;

实现input 输入框的ui

```
<view class="my-search-box">
  <uni-icons type="search" size="17"></uni-icons>
  <text class="placeholder">搜索</text>
</view>
```

为实现点击功能，在组件内部，给my-search-box 的 view 绑定 click 事件处理函数
@click="searchBoxHandler"

```
searchBoxHandler() {
  this.$emit('click')
}
```

在分类页面中使用搜索组件时，为其绑定点击事件处理函数@click="gotoSearch"跳转到分包中的搜索页面

```
gotoSearch() {  
  uni.navigateTo({  
    url: '/subpkg/search/search'  
  })  
}
```

2.搜索建议

使用 uni-ui 提供的搜索组件uni-search-bar，定义输入事件处理函数input(e);

为准确记录搜索历史，需实现搜索框的防抖处理，定义防抖的延时器 timerId，修改input 事件处理函数

```
// 清除延时器  
clearTimeout(this.timer)  
// 重新启动一个延时器，并把 timerId 赋值给 this.timer  
this.timer = setTimeout(() => {  
  // 如果 500 毫秒内，没有触发新的输入事件，则为搜索关键词赋值  
  this.kw = e  
  console.log(this.kw)  
}, 500)
```

为根据关键词查询搜索建议列表，定义搜索关键词kw和搜索结果列表searchResults: []，定义getSearchList方法获取搜索建议列表

```
// 判断关键词是否为空  
if (this.kw === '') {  
  this.searchResults = []  
  return  
}  
// 发起请求，获取搜索建议列表  
const { data: res } = await uni.$http.get('qsearch', { query: this.kw })  
if (res.meta.status !== 200) return uni.$showMsg()  
this.searchResults = res.message
```

为实现点击搜索建议的商品跳转到商品详情页面，定义事件处理函数gotoDetail()

```
gotoDetail(goods_id) {  
  uni.navigateTo({  
    url: '/subpkg/goods_detail/goods_detail?goods_id=' + goods_id  
  })  
}
```

3.搜索历史

定义搜索关键词的历史记录historyList: [];

当搜索结果列表的长度不为 0 的时候，需要展示搜索建议区域，隐藏搜索历史区域；

当搜索结果列表的长度等于 0 的时候，则需要隐藏搜索建议区域，展示搜索历史区域

使用 v-if 和 v-else实现搜索建议和搜索历史的按需展示

```
//搜索建议列表
<view class="sugg-list" v-if="searchResults.length !== 0"></view>
//搜索历史
<view class="history-box" v-else></view>
```

为将搜索关键词存入 historyList，定义保存搜索关键词的方法saveSearchHistory()，但直接push有两个问题，一是关键词前后顺序的问题，二是关键词重复的问题

```
saveSearchHistory() {
  this.historyList.push(this.kw)
}
```

定义一个计算属性 historys，其值为historyList 数组 reverse后以解决关键字前后顺序的问题

```
computed: {
  historys() {
    return [...this.historyList].reverse()
  }
}
```

使用 Set 对象解决关键词重复的问题，修改 saveSearchHistory 方法

```
//Array 数组转化为 Set 对象
const set = new Set(this.historyList)
//调用 Set 对象的 delete 方法，移除对应的元素
set.delete(this.kw)
//调用 Set 对象的 add 方法，向 Set 中添加元素
set.add(this.kw)
//将 Set 对象转化为 Array 数组
this.historyList = Array.from(set)
```

实现清空搜索历史记录功能，定义 cleanHistory 处理函数

```
cleanHistory() {
  // 清空 data 中保存的搜索历史
  this.historyList = []
  // 清空本地存储中记录的搜索历史
  uni.setStorageSync('kw', '[]')
}
```

实现点击搜索历史跳转到商品列表页面，定义 gotoGoodsList 处理函数

```
gotoGoodsList(kw) {
  uni.navigateTo({
    url: '/subpkg/goods_list/goods_list?query=' + kw
  })
}
```

🔍 a



【海外购自营】黎珏(ReFa) MTG日本 CARAT铂金微... >

沿途 (yantu) 车载充电器车充一拖二usb转接口手机... >

搜索历史



a

电视