

# 1. 分类页面概述

效果

scroll-view 组件

scroll-view 组件



业务逻辑

1. 加载分类页面数据
2. 点击左侧菜单，右侧数据动态渲染

接口

分类页面数据

<https://api-hmugo-web.itheima.net/api/public/v1/categories>

## 关键技术

scrollview 组件

es6的Promise网络请求

## 2. 分类页面数据获取

### 使用Promise优化网络请求

```
// 在request目录中编写index.js文件
// Promise 详解 https://www.jianshu.com/p/459a856c476f
// Promise是JS异步编程中的重要概念，异步抽象处理对象，是目前比较流行Javascript异步编程解决方案之一
export const request=(params)=>{
    return new Promise((resolve,reject)=>{
        wx.request({
            ...params,
            success:(result)=>{
                resolve(result);
            },
            fail:(err)=>{
                reject(err);
            }
        });
    })
}
```

### 在分类页面编写index.js

```
import { request } from "../../request/index.js";
Page({
    data: {
        // 左侧的菜单数据
        leftMenuList: [],
        // 右侧的商品数据
        rightContent: [],
    },
    // 接口的返回数据
    Cates: [],
})
```

```
onLoad: function (options) {
    this.getCates();
},
// 获取分类数据
getCates() {
    request({
        url: "https://api-hmugo-web.itheima.net/api/public/v1/categories"
    })
    .then(res => {
        this.Cates = res.data.message;
        // 构造左侧的大菜单数据
        let leftMenuList = this.Cates.map(v => v.cat_name);
        // 构造右侧的商品数据
        let rightContent = this.Cates[0].children;
        this.setData({
            leftMenuList,
            rightContent
        })
    })
}
})
```

### 3. 分类 - 页面布局

#### index.json 页面配置文件

```
// index.json 页面配置文件
{
    "usingComponents": {
        "SearchInput": "../../components/SearchInput/SearchInput"
    },
    "navigationBarTitleText": "商品分类"
}
```

#### index.wxml

```
<!-- 通过官网介绍scroll-view -->

<view class="tg_cates">
    <SearchInput></SearchInput>
    <view class="cates_container">
```

```

<!-- 左侧菜单 -->
<scroll-view scroll-y class="left_menu">
    <view class="menu_item {{index==currentIndex?'active':''}}" wx:for="{{leftMenuList}}"
        wx:key="*this">
        {{item}}
    </view>
</scroll-view>

<!-- 右侧商品内容 -->
<scroll-view scroll-top="{{scrollTop}} scroll-y class="right_content">
    <view class="goods_group" wx:for="{{rightContent}}" wx:for-index="index1" wx:for-item="item1">
        <view class="goods_title">
            <text class="delimiter"/></text>
            <text class="title">{{item1.cat_name}}</text>
            <text class="delimiter"/></text>
        </view>

        <view class="goods_list">
            <navigator wx:for="{{item1.children}}" wx:for-index="index2" wx:for-item="item2" wx:key="cat_id" url="/pages/goods_list/index?cid={{item2.cat_id}}">
                <image mode="widthFix" src="{{item2.cat_icon}}></image>
                <view class="goods_name">{{item2.cat_name}}</view>
            </navigator>
        </view>
    </view>
</scroll-view>
</view>
</view>

```

## index.less样式设置

```

page{
    height: 100%;
}

.tg_cates{
    height: 100%;
    .cates_container{
        /* less中使用calc的时候要注意 */
        height: ~'calc( 100vh - 90rpx )';
        display: flex;
        .left_menu{
            /* 子项 高度 100% flex */
            flex: 2;
            .menu_item{
                height: 80rpx;
                display: flex;
                justify-content: center;
                align-items: center;
                font-size: 30rpx;
            }
            .active{
                color: var(--themeColor);
            }
        }
    }
}

```

```
        border-left: 5rpx solid currentColor;
    }
}

.right_content{
    /* 子项 高度 100% flex */
    flex: 5;
    .goods_group{
        .goods_title{
            height: 80rpx;
            display: flex;
            justify-content: center;
            align-items: center;
            .delimiter{
                color: #ccc;
                padding: 0 10rpx;
            }
            .title{}
        }
        .goods_list{
            display: flex;
            flex-wrap: wrap;
            navigator{
                width: 33.33%;
                text-align: center;
                flex-direction: column;
                image{
                    width: 50%;
                }
                .goods_name{}
            }
        }
    }
}
}
```

/\* 提示1： app.wxss全局样式设置 \*/

```
image {
    width: 100%;
}
```

/\* 提示2： index.js 添加变量currentIndex \*/

```
data: {
    // 左侧的菜单数据
    leftMenuList: [],
    // 右侧的商品数据
    rightContent: [],
    // 被点击的左侧菜单
    currentIndex: 0
},
```

# 4. 分类-点击左侧菜单-商品联动

## 左侧菜单项绑定点击事件

```
<!--index.wxml-->

<view class="cates_container">
    <!-- 左侧菜单 -->
    <scroll-view scroll-y class="left_menu">
        <view class="menu_item {{index==currentIndex?'active':''}}" wx:for="{{leftMenuList}}"
wx:key="*this" bindtap="handleItemTap" data-index="{{index}}>
            {{item}}
        </view>
    </scroll-view>
```

## 编写点击事件

```
// index.js

// 左侧菜单的点击事件
handleItemTap(e) {
    /*
    1 获取被点击的标题身上的索引
    2 给data中的currentIndex赋值就可以了
    3 根据不同的索引来渲染右侧的商品内容
    */
    const { index } = e.currentTarget.dataset;
    let rightContent = this.Cates[index].children;
    this.setData({
        currentIndex: index,
        rightContent,
    })
}
```

## 点击菜单，右侧列表置顶

查阅官方文档: `scroll-top`  
`index.wxml`

```
<scroll-view scroll-top="{{scrollTop}}" scroll-y class="right_content">
```

在`index.js`，每次点击菜单项 `scrollTop`变量值都赋值为0

## 定义公共url，优化请求路径

```
// request/index.js

export const request=(params)=>{
    // 定义公共的url
    const baseURL="https://api-hmugo-web.itheima.net/api/public/v1";
    return new Promise((resolve,reject)=>{
        wx.request({
            ...params,
            url:baseURL+params.url,
            success:(result)=>{
                resolve(result);
            },
            fail:(err)=>{
                reject(err);
            }
        });
    })
}
```

```
// 修改分类页面的index.js

request({
    // url: "https://api-hmugo-web.itheima.net/api/public/v1/categories"
    url: "/categories"
})
```