# 注册微信小程序

地址：<https://mp.weixin.qq.com>

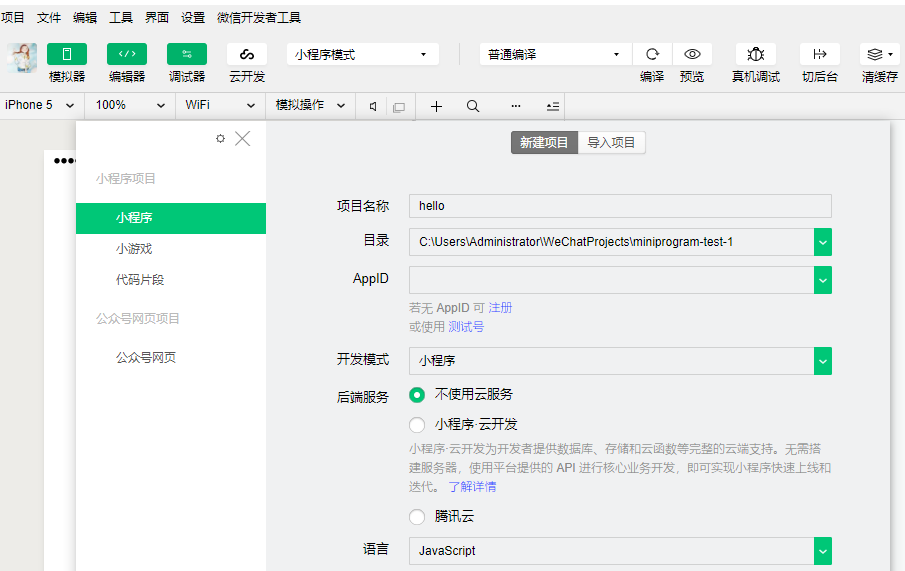
AppID(小程序ID) wx96056a86ac2ae3c1

# 搭建开发环境

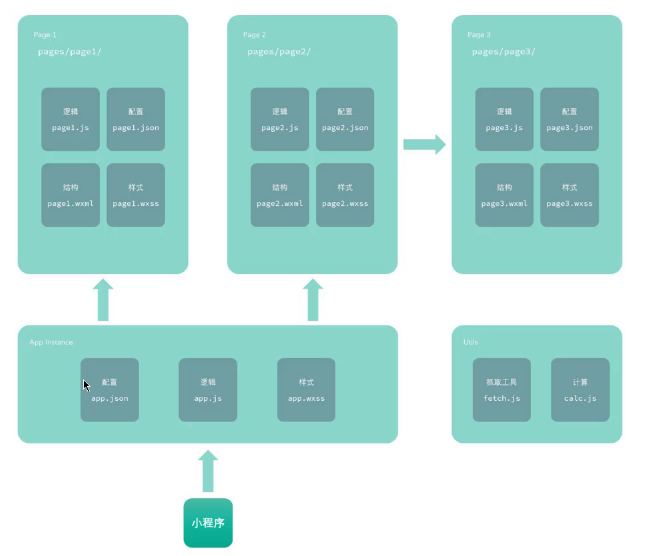








# 项目结构





# 页面结构

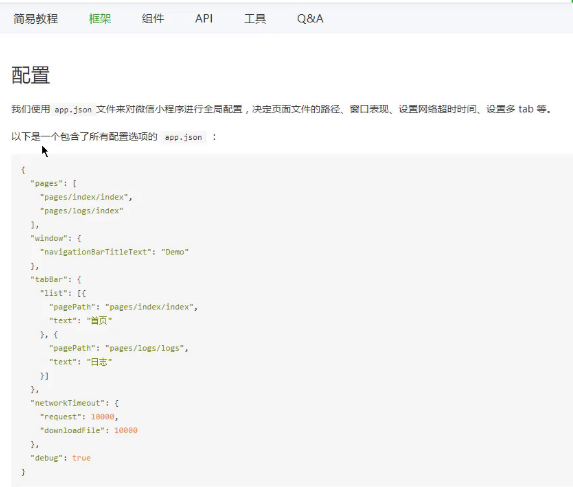


# 小程序的配置文件









# 6、标签栏配置

"tabBar": {

"color": "pink",

"selectedColor": "red",

"list": [

//至少有两个list

{

"pagePath": "pages/bear/bear",

"text": "小熊",

"iconPath": "img/grid-01.png",

"selectedIconPath": "img/grid-03.png"

},

{

"pagePath": "pages/index/index",

"text": "小熊",

"iconPath": "img/grid-02.png",

"selectedIconPath": "img/grid-03.png"

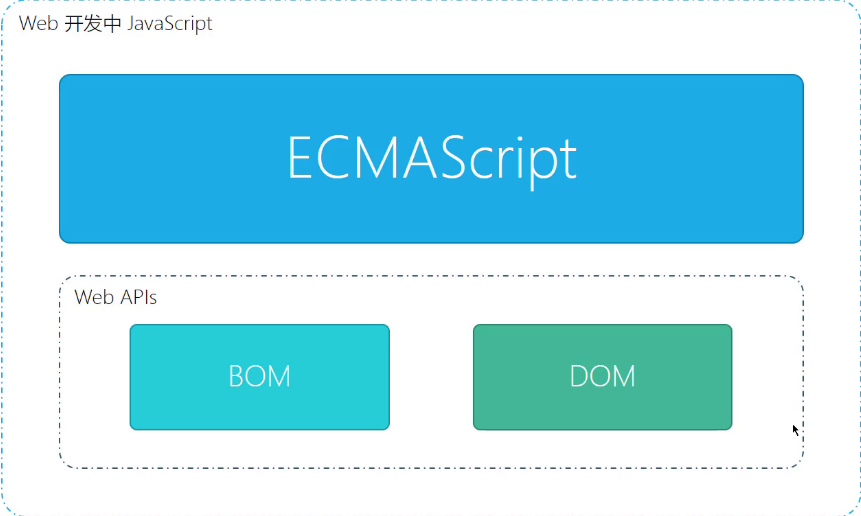
}

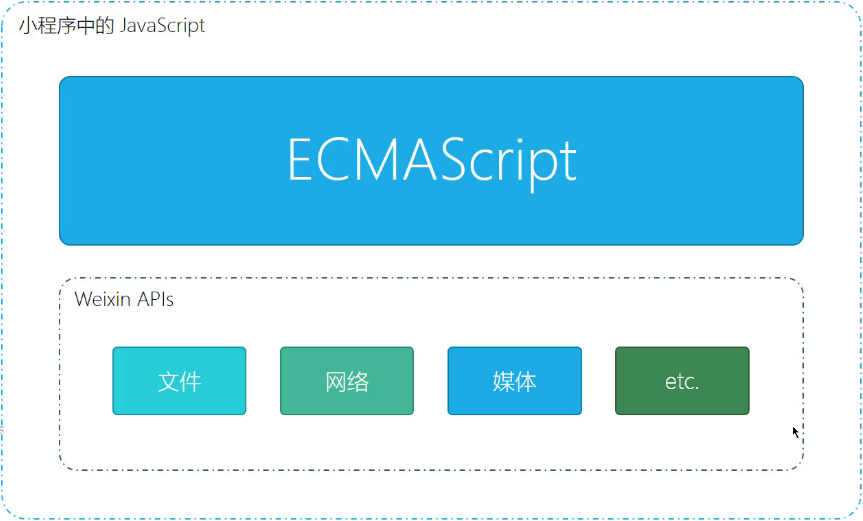
]},



# 7、逻辑层

Web开发中JavaScript





# 应用的生命周期

全局生命周期

1, onLaunch: 在整个应用启动时触发，只触发一次,可有参数

2, onShow: 应用程序显示到屏幕上时触发，每次成为焦点状态时都会触发，可有参数

3, onHide: 应用隐藏到后台时触发

4, onError: 只能捕获到运行阶段的异常

app.js

App({

onLaunch: function( {

console.log(‘应用启动了’)

},

onShow: function(options) {

console.log(options)

},

onHide: function() {

console.log(‘应用隐藏了’)

},

onError: function(msg) {

console.log(msg)

},

//除了生命周期里约定好的钩子函数，还可以定义任何成员，定义在App()里的成员可以在后续每一个页面中共享，但在这外边定义的，就不能共享，和node.js里一样，每个js文件都有单独的作用域。

foo: ‘haha’,

say() {

console.log(‘enen’)

}

})

局部生命周期

index.js

//获取应用实例

const app = getApp(

Page({

Data: {

},

// 生命周期--监听页面加载

onLoad: function() {

console.log(app.foo)

app.say()

}

})

# 页面的生命周期

Index.js

// 创建一个页面实例对象

Page({

data: { },

// 页面加载，这个页面即将要工作了，适合去做数据初始化

onLoad() {

// 动态添加数据

this.setData({ foo: ‘hello world’ })

console.log(‘开始工作了’)

},

// 页面进入焦点状态（前台展示）

onShow() {

console.log(‘页面展示了’)

},

// 当页面准备好了，页面渲染完成

onReady() {

// wx微信的全局对象，可以设置标题，但只有在页面加载完成之后，才可以设置标题

wx.setNavigationBarTitle({

Title: ‘小熊’

}

console.log(‘页面渲染完成了’)

},

// 当前页面失去焦点时

onHide() {

console.log(‘页面切换隐藏了’)

},

// 页面卸载时触发，当前页面被销毁，可以用于在页面卸载之前保存页面状态

onUnload() {

console.log(‘页面被销毁了’)

}

)}

Index.wxml

<view class=”container”>

// mustache小胡子语法 专门用于输出逻辑层暴露的数据

<text>{{ foo }}</text>

</view>

# 数据绑定和WXS

mustache小胡子语法{{}}

data:{}里面的数据，可以直接引用，但方法不可以,但可以如下：

index.js

Page({

data: {

message: “hello”

}

})

Index.wxml

<view class=”container”>

<text>{{ foo.addPostfix(message) }}</text>

</view>

<wxs module=”foo”>

// 这里可以到处一个对象，这个对象可以在界面上使用

module.exports = {

addPostfix: function(msg) {

Return msg + ‘1’

}

}

}

</mxs>

# 控制属性（条件渲染）

wx:if=””: 如果条件是true，则显示。wx:else: 如果条件不满足时

Index.js

Pages({

data: {

flag: true

},

onReady() {

setTimeout(() => {

this.setData(flag: flase)

})

}

}

Index.wxml

<view wx:if=”{{ flag }}”>

<text>loading....</text>

</view>

<view wx:else>

<text>load done</text>

</view>

// 对于频繁需要被切换显示的元素，不应该使用wx:if,应该使用

<view hidden=”{{ !flag }}”>

<text>loading....</text>

</view>

<view hidden=”{{ flag }}”>

<text>load done</text>

</view>

// 中间元素不受影响

<view>

<block wx:if=”{{ flag }}”>

<text>hello world</text>

</block>

<text>不要隐藏我</text>

</view>

# 控制属性（列表渲染）

wx:for=””: 循环数据。取值方式：item.\*\*;

wx:for-item=”s”: 改变取值名称。取值方式：s.\*\*;

wx:for-index=”i”: 改变索引名称。取值{{ i }};

wx:key=””: 添加唯一标识符，属性值写得是遍历的数据单项属性名称,如果数据是数组[1,2,3,4]这类,属性值 使用\*this去保留，wx:key=”\*this”

Index.wxml

<view>

<view wx:for=”{{ students }}” wx:for-name=”s” wx:for-index=”i” wx:key=”id”>

<text>{{ i }}--</text>

<text>name: {{ s.name }}</text>

<text>age: {{ s.age }}</text>

</view>

</view>

Index.js

Page({

data: {

students: [

{id: 1, name: “zhangsan”, age: 18},

{id: 2, name: “lisi”, age: 19},

{id: 3, name: “wangwu”, age: 20},

{id: 4, name: “zhaoliu”, age: 21}

]

}

})

# 事件处理（以bind开头）

Index.wxml

<button bindtap=”tapHandle”>clickme</button>

Index.js

Page({

tapHandle(e) {

console.log(e)

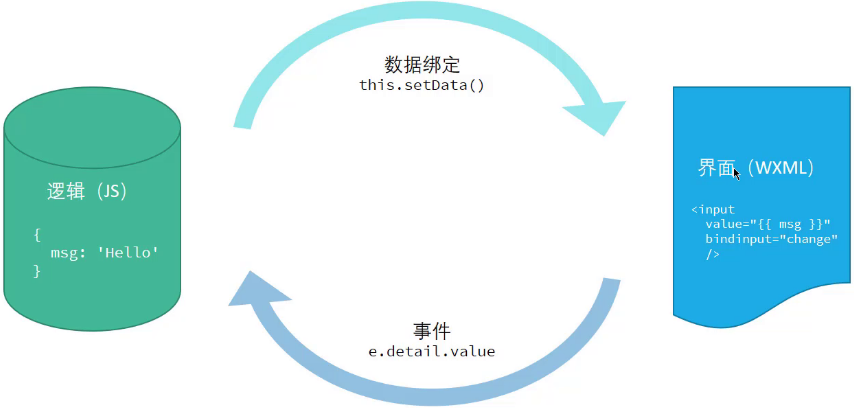
}

})

注意：catchtap触发的事件不会发生冒泡行为。bindtap不能直接通过bindtap=”remove(i)”的形式传参，可 以通过data-xx属性，给事件处理函数传 递额外的参数<button bindtap=”tapHandle” data-id=” 1”>clickme</button>



# 14、单向数据流



Index.wxml

<!-- 单向数据流 -->

<view>

<text>{{ foo1 }}</text>

<input value='{{foo1}}' bindinput='inputChange'></input>

</view>

Index.js

inputChange(e) {

console.log(e.detail.value)

this.setData({foo1: e.detail.value})

}

# 15、wxss和css

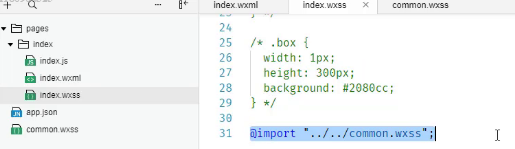
wxss是用来决定wxml的组件应该如何解释，是对css的扩展和修改。与css相比，扩展的特性有：

* 尺寸单位
* 样式导入

尺寸单位：规定任何屏幕宽度为750rpx.

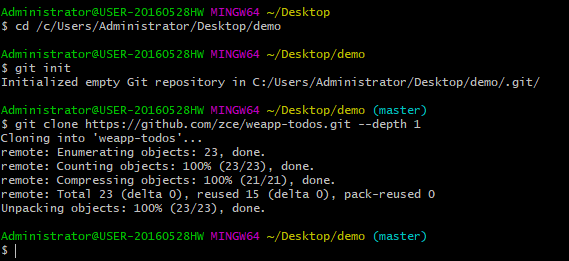
样式导入：微信有个ui框架WeUI,地址为：https://weui.io

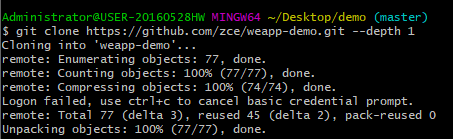


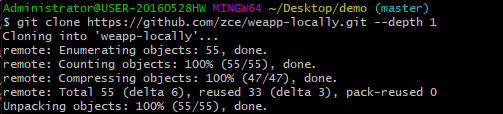


# 16、从GitHub上克隆文件

先建一个文件夹，然后打开Git Bash Here小黑框







# locally\_life小程序案例

## 1、外观及标签栏的配置

#### app.json

{

"pages": [

"pages/index/index",

"pages/contact/contact",

"pages/message/message",

"pages/profile/profile",

"pages/logs/logs"

],

"window": {

"backgroundTextStyle": "light",

"navigationBarBackgroundColor": "#3a4861",

"navigationBarTitleText": "本地生活",

"backgroundColor": "#bcc0c9",

"navigationBarTextStyle": "white",

"enablePullDownRefresh": false

},

"tabBar": {

"list": [

{

"color": "#999",

"selectedColor": "#444",

"backgroundColor": "#fff",

"borderStyle": "black",

"pagePath": "pages/index/index",

"text": "首页",

"iconPath": "assets/home.png",

"selectedIconPath": "assets/home-active.png"

},

{

"pagePath": "pages/message/message",

"text": "消息",

"iconPath": "assets/message.png",

"selectedIconPath": "assets/message-active.png"

},

{

"pagePath": "pages/contact/contact",

"text": "联系",

"iconPath": "assets/contact.png",

"selectedIconPath": "assets/contact-active.png"

},

{

"pagePath": "pages/profile/profile",

"text": "我的",

"iconPath": "assets/profile.png",

"selectedIconPath": "assets/profile-active.png"

}

]

},

"sitemapLocation": "sitemap.json"

}

#### app.wxss

page {

background-color: #f0f0f0;

}

## 2、在vsCode中打开项目

下载mnapp插件，可以让其中间的标签语句显示不同的颜色

## 3、首页index

### 3.1 首页布局

#### Index.wxml

<!-- 微信内部提供了轮播图插件 -->

<swiper class="slides">

<swiper-item >

<!-- mode="aspectFill"较短边充满 -->

<image src="../../assets/img/banner-01.png" mode="aspectFill"/>

</swiper-item>

<swiper-item >

<image src="../../assets/img/banner-02.png" mode="aspectFill"/>

</swiper-item>

</swiper>

<!-- 九宫格部分 -->

<view class="grids">

<view class="item">

<image src="/assets/icons/grid-01.png"/>

<text>美食</text>

</view>

<view class="item">

<image src="/assets/icons/grid-02.png"/>

<text>洗浴足疗</text>

</view>

<view class="item">

<image src="/assets/icons/grid-03.png"/>

<text>结婚</text>

</view>

<view class="item">

<image src="/assets/icons/grid-04.png"/>

<text>KTV</text>

</view>

<view class="item">

<image src="/assets/icons/grid-05.png"/>

<text>找工作</text>

</view>

<view class="item">

<image src="/assets/icons/grid-06.png"/>

<text>辅导</text>

</view>

<view class="item">

<image src="/assets/icons/grid-07.png"/>

<text>汽车保养</text>

</view>

<view class="item">

<image src="/assets/icons/grid-08.png"/>

<text>租房</text>

</view>

<view class="item">

<image src="/assets/icons/grid-09.png"/>

<text>装修</text>

</view>

</view>

<!-- 下部链接部分 -->

<view class="links">

<image src="/assets/img/link-01.png" />

<image src="/assets/img/link-02.png" />

</view>

#### Index.wxss

.slides {

height: 380rpx;

}

.slides image {

width: 100%;

height: 100%;

}

.grids {

display: flex;

/\* 换行 \*/

flex-wrap: wrap;

background-color: #fff;

color: #888;

font-size: 28rpx;

border-top: 1px solid #eee;

border-left: 1px solid #eee;

}

.grids .item {

display: flex;

/\* 垂直排列 \*/

flex-direction: column;

/\* 垂直居中 \*/

justify-content: center;

/\* 水平居中 \*/

align-items: center;

width: 33.3333%;

height: 250rpx;

border-right: 1px solid #eee;

border-bottom: 1px solid #eee;

box-sizing: border-box;

}

.grids .item image {

width: 70rpx;

height: 70rpx;

}

.grids .item text {

margin-top: 20rpx;

display: block;

}

.links image {

width: 360rpx;

height: 200rpx;

margin-left: 10rpx;

}

### 3.2 动态渲染数据

#### Index.js

const fetch = require('../../utils/util.js')

Page({

data: {

slides: [],

categories: []

},

onLoad: function() {

// 微信提供的请求方式，这里没有跨域概念

fetch('slides').then(res => {

this.setData({ slides: res.data })

})

fetch('categories').then(res => {

this.setData({ categories: res.data })

})

// 多了就不利于维护和优化了，故在公共utils中定义一个公共格式

// wx.request({

// url: 'https://locally.uieee.com/categories',

// header: {

// 'ContentType': 'json'

// },

// success: (res) => {

// this.setData({ categories: res.data })

// // console.log(this.data.categories)

// }

// })

}

})

#### util.js

module.exports = (url, dathttps:a) => {

return new Promise((resolve, reject) => {

wx.request({

url: `//locally.uieee.com/${url}`,

data: data,

// header: {

// 'ContentType': 'json'

// },

success: resolve,

fail: reject

})

})

}

#### Index.wxml

<!-- 微信内部提供了轮播图插件 -->

<swiper class="slides">

<swiper-item wx:for="{{slides}}" wx:key="id" >

<!-- mode="aspectFill"较短边充满 -->

<image src="{{item.image}}"/>

</swiper-item>

</swiper>

<!-- 九宫格部分 -->

<view class="grids">

<view class="item" wx:for="{{categories}}" wx:key="id">

<image src="{{item.icon}}"/>

<text>{{item.name}}</text>

</view>

</view>

## 4、列表页面list

### 4.1 页面九宫格跳转

#### Index.wxml

<!-- 九宫格部分 -->

<view class="grids">

<!-- 把view改为navigator类似于vue中的router-link，不过属性不是to，而是url-->

<navigator class="item" wx:for="{{categories}}" url="/pages/list/list?cat={{item.id}}" wx:key="id">

<image src="{{item.icon}}"/>

<text>{{item.name}}</text>

</navigator>

</view>

### 4.2 跳转到list页面

#### list.wxml

<view class="search">

<input placeholder="搜索"/>

<icon type="search" size="15"/>

</view>

<view class="cells">

<navigator class="item" wx:for="{{shop}}" wx:key="id" url="/pages/detail/detail?item={{item.id}}">

<image src="{{item.images[0]}}" mode="aspectFill"/>

<view class="meta">

<text class="name">{{item.name}}</text>

<text class="phone">电话：{{item.phone}}</text>

<text class="address">地址：{{item.address}}</text>

<text class="hours">营业时间：{{item.businessHours}}</text>

</view>

<view class="score">{{item.score}}</view>

</navigator>

</view>

#### list.js

const fetch = require('../../utils/util.js')

Page({

/\*\*页面的初始数据 \*\*/

data: {

category: {},

shop: [],

pageIndex: 0,

pageSize: 10,

},

loadMore() {

let {pageIndex,pageSize} = this.data

const params = {\_page:++pageIndex,\_limit:10}

fetch(`categories/${this.data.category.id}/shops?`,params).then(res => {

const shop = this.data.shop.concat(res.data)

this.setData({ shop: shop, pageIndex })

console.log(this.data.shop)

})

},

/\*\* 生命周期函数--监听页面加\*\*/

onLoad: function (options) {

// console.log(options)

// 加载分类信息

fetch(`categories/${options.cat}`).then(res => {

this.setData({category: res.data})

// console.log(this.data.category)

wx.setNavigationBarTitle({

title: this.data.category.name,

})

// 加载商铺信息

this.loadMore()

})

},

// 生命周期函数--监听页面初次渲染完成

onReady: function () {

if(this.data.category.name) {

wx.setNavigationBarTitle({

title: this.data.category.name,

})

}

},

// 页面上拉触底事件的处理函数

onReachBottom: function () {

this.loadMore()

}

})

list.wxss

.search {

position: relative;

}

.search icon {

position: absolute;

right: 45rpx;

top: 15rpx;

z-index: 11;

}

.search input {

margin: 15rpx 30rpx;

padding: 5rpx 20rpx;

border-radius: 10rpx;

background-color: #fff;

font-size: 30rpx;

}

.cells {

background-color: #fff;

}

.cells .item {

display: flex;

align-items: flex-start;

border-bottom: 1rpx solid #eee;

}

.cells .item image {

width: 160rpx;

height: 160rpx;

margin: 20rpx;

}

.cells .item .meta {

display: flex;

flex-direction: column;

flex: 1;

padding: 10rpx 0;

font-size: 30rpx;

}

.cells .item .meta .name {

color: #234;

font-size: 28rpx;

}

.cells .item .meta .phone,

.cells .item .meta .address {

color: #678;

font-size: 24rpx;

}

.cells .item .meta .hours {

/\*color: #ff69b4;\*/

color: #456;

font-size: 22rpx;

}

.cells .item .score {

margin: 20rpx 20rpx 0 -40rpx;

padding: 0 10rpx;

background-color: #ee523d;

border-radius: 30rpx;

color: #fff;

font-size: 24rpx;

text-align: center;

}

### 4.3 向上加载更多

#### list.wxml

<view wx:if="{{hasMore}}" class="loadmore loading">正在加载...</view>

<view wx:else class="loadmore">客官，没有更多了</view>

list.wxss

.loadmore {

color: #888;

font-size: 30rpx;

line-height: 100rpx;

text-align: center;

}

.loadmore.loading::before {

content: '';

width: 40rpx;

height: 40rpx;

margin-top: -10rpx;

margin-right: 10rpx;

display: inline-block;

vertical-align: middle;

animation: loading 1s steps(12) infinite;

background: transparent url(data:image/svg+xml;base64,) no-repeat;

background-size: 100%

}

@keyframes loading {

from {

transform: rotate(0deg)

}

to {

transform: rotate(-360deg)

}

}

#### list.js

Page({

data: {

hasMore: true

},

loadMore() {

if(!this.data.hasMore) return

const totalCount = parseInt(res.header['X-Total-Count'])

const hasMore = pageIndex \* pageSize < totalCount

this.setData({ shop: shop, pageIndex, hasMore })

},

// 页面上拉触底事件的处理函数

onReachBottom: function () {

this.loadMore()

}

### 4.4 下拉加载

#### list.json

{

"enablePullDownRefresh": true

}

#### list.js

loadMore() {

if(!this.data.hasMore) return

let {pageIndex,pageSize} = this.data

const params = {\_page:++pageIndex,\_limit:10}

return fetch(`categories/${this.data.category.id}/shops?`,params).then(res => {

const shop = this.data.shop.concat(res.data)

const totalCount = parseInt(res.header['X-Total-Count'])

const hasMore = pageIndex \* pageSize < totalCount

this.setData({ shop: shop, pageIndex, hasMore })

// console.log(res.header['X-Total-Count'])

})

}

// 页面相关事件处理函数--监听用户下拉动作

onPullDownRefresh: function () {

// 把数据清空，然后loadMore（）

this.setData({shop:[],pageIndex:0,hasMore:true})

this.loadMore().then(()=> {

// 默认下拉会持续3秒钟，如果加载完了，立马收回下拉样式

wx.stopPullDownRefresh()

})

}

## 5、详情页面detail

#### detail.wxml

<swiper class="slides" indicator-dots="{{shop.images.length > 1}}" indicator-color="#bcc0c9" indicator-active-color="#3a4861">

<swiper-item wx:for="{{shop.images}}" wx:key="id">

<image src="{{item}}" mode="aspectFill" bindtap="previewHandle" data-src="{{item}}"/>

</swiper-item>

</swiper>

<view class="heading" wx:if="{{shop.name}}">

<text class="name">{{shop.name}}</text>

<text class="phone">电话：{{shop.phone || 'none'}}</text>

<text class="address">地址：{{shop.address}}</text>

<text class="hours">营业时间：{{shop.businessHours}}</text>

<view class="score" wx:if="{{item.score}}">{{item.score}}</view>

</view>

<view class="introduction" wx:if="{{shop.introduction}}">

<text>{{shop.introduction}}</text>

</view>

<view class="comments" wx:if="{{shop.comments.length}}">

<view class="item" wx:for="{{shop.comments}}" wx:key="\*this">

<text class="name">{{item.name}}</text>

<text class="date">{{item.date}}</text>

<text class="rating">{{item.rating}}</text>

<text class="content">{{item.content}}</text>

<view class="images">

<image wx:for="{{item.images}}" wx:key="\*this" src="{{util.size(item)}}"/>

</view>

</view>

</view>

<wxs module="util"> //模块

// 这里不支持一些新特性,类似于过滤器，把图片的宽高设置为不超过100.100

module.exports = {

size: function (input) {

return input.replace('w.h', '100.100')

}

}

</wxs>

#### detail.wxss

.slides {

height: 580rpx;

}

.slides image {

min-width: 100%;

height: 100%;

}

.heading {

display: flex;

flex-direction: column;

padding: 20rpx 30rpx;

background-color: #fff;

color: #567;

font-size: 24rpx;

}

.heading .name {

font-size: 40rpx;

color: #234;

}

.heading .score {

position: absolute;

top: 20rpx;

right: 20rpx;

padding: 0 10rpx;

background-color: #ee523d;

border-radius: 30rpx;

color: #fff;

font-size: 24rpx;

text-align: center;

}

.introduction {

margin-top: 20rpx;

padding: 30rpx;

background-color: #fff;

color: #456;

font-size: 24rpx;

line-height: 2;

}

.comments {

margin-top: 20rpx;

padding: 20rpx;

background-color: #fff;

color: #345;

font-size: 24rpx;

}

.comments .item {

display: flex;

position: relative;

flex-direction: column;

padding: 20rpx;

border-bottom: 1rpx solid #eee;

}

.comments .item .name {

margin-top: 20rpx;

font-size: 36rpx;

}

.comments .item .date {

align-self: flex-end;

margin: -40rpx 0 20rpx;

color: #567;

}

.comments .item .rating {

position: absolute;

top: 20rpx;

right: 10rpx;

color: #ee523d;

}

.comments .item image {

width: 110rpx;

height: 110rpx;

margin: 10rpx;

}

#### detail.js

const fetch = require('../../utils/util.js')

Page({

// 页面的初始数据

data: {

shop: {}

},

// 生命周期函数--监听页面加载

onLoad(options) {

fetch(`shops/${options.item}`).then(res => {

this.setData({ shop: res.data })

wx.setNavigationBarTitle({ title: res.data.name })

})

},

// API接口，在新页面中全屏预览图片。预览的过程中用户可以进行保存图片、发送给朋友等操作。

previewHandle(e) {

wx.previewImage({

current: e.target.dataset.src,

urls: this.data.shop.images

})

}

})