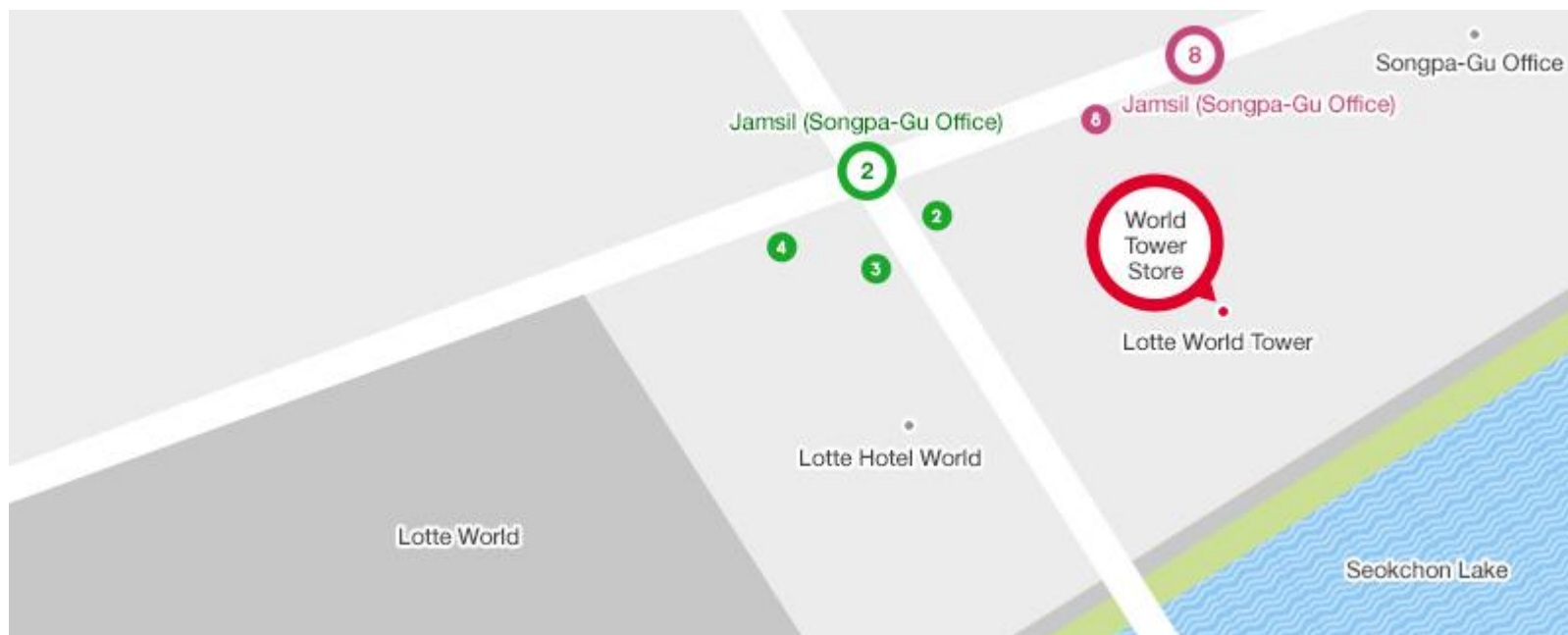


## 阿东入门系列《五》仿找事吧 APP 附近三公里 Demo



源码下载地址：[Github](#)

效果图如下：



分析一下页面，主要内容分为顶部轮播，中间 10 个分类图标的排版和单击事件，下部列表下拉刷新上拉加载更多。大部分知识点前面都讲过。这里主要说一下微信小程序中的数据绑定，前后台传值以及加载更多时的数据合并。

## 1. 数据绑定和前后台传值

中间分类图标的布局文件：

```
<view class="items" wx:for="{{array}}" wx:for-item="item" bindtap="typeclick" data-code="{{item.code}}" data-text="{{item.text}}" >
```

```
<image class="item-img" mode="aspectFit" src="{{item.src}}">image>

<view class="item-text">{{item.text}}view>

view>
```

- 1
- 2
- 3
- 4

可以看出是以 控制属性 `wx:for` 绑定数据 `array` 来循环渲染布局，并对 `view` 绑定了单击事件

`bindtap="typeclick"`。因为每一个分类点击都会刷新下部列表，所以需要在事件中获得当前分类数据的 `code`。小程序

中提供自定义标签 `data-xxx`，供开发者使用来绑定数据，`xxx` 可以随意取名，这里我们用 `data-`

`code="{{item.code}}" data-text="{{item.text}}"`把每条数据的 `code` 和 `text` 传给 `function typeclick`。

然后在 `js` 中的 `typeclick` 函数中，我们可以通过 `event` 拿到绑定的数据。

```
// 分类 item 单击事件
```

```
typeclick: function (e) {
```

```
    total = 0;
```

```
    code = e.currentTarget.dataset.code + "";
```

```
var name = e.currentTarget.dataset.text + " ";  
  
this.data.dataArray = [];  
  
this.setData({  
  title: "附近三公里: " + name  
})  
  
this.periphery();  
},
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

- 10
- 11
- 12
- 13

`e.currentTarget.dataset.code` 后边的 `code` 就是我们在布局文件中定义的 `data-xxx` 中的 `xxx`，这里需要注意一下，因为 js 的机制，有时候我们拿到的数据类型可能不对，需要自己处理一下。

## 2. 加载更多时的数据合并

**// 网络请求**

```
periphery: function () {  
  
    var that = this  
  
    //sliderList  
    wx.request({  
        url: 'http://xxx',  
  
        method: 'POST',  
  
        data: {  
            city: "深圳",  
            code: code,  
            count: count + "",
```

```

        total: total + "",

        lat: app.globalData.latitude + "",

        lng: app.globalData.longitude + ""
    },
    header: {
        'Accept': 'application/json'
    },
    success: function (res) {
        that.data.dataArray = that.data.dataArray.concat(res.data.data.list)
        that.setData({
            dataArray: that.data.dataArray
        })

        setTimeout(function () {
            that.setData({
                loadingHidden: true
            })
        }, 1000)
    }
})
},

```

- 1

- 2

- 3

- 4

- 5

- 6

- 7

- 8

- 9

- 10

- 11

- 12

- 13

- 14

- 15

- 16

- 17

- 18

- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32

因为列表有上拉刷新和下拉加载更多功能。所以每次的网络请求通过 `total` 和 `count` 控制每次请求的数据的页码，  
然后在 `success` 回调中把数据拼接到原来的数据集合上。



首先注意一点。在 `wx.request` 的回调中，我们不能直接用 `this.data.dataArray` 来取 `data` 标签下的 `dataArray`，因为这里的 `this` 代表的并不是 `js` 的全局上下文对象，他对应的是这个 `function` 的上下文。所以我们需要在 `wx.request` 的外部，通过一个变量来保存 `js` 的全局上下文对象，`var that = this`，然后在回调中用 `that.data.dataArray` 然后说数据拼接，需要用到 `concat` 关键字，他可以把其参数拼接调用者身上。

`that.data.dataArray.concat(res.data.data.list)` 这里需要注意请求返回的数据格式，`res.data` 代表的是返回的 `json`，然后自己根据数据格式拼接，直到取到数据集合。

其次因为上拉和下拉的性质不同，其处理方式也不同，下拉需要把数据集合置为空并从头开始去数据。上拉需要处理 `total`，来取下一个 `count` 条数的数据。代码如下：

#### // 下拉刷新回调接口

```
onPullDownRefresh: function () {  
  
    total = 0;  
  
    this.data.dataArray = [];  
  
    this.periphery();  
    wx.stopPullDownRefresh;  
},
```

#### // 上拉加载回调接口

```
onReachBottom: function () {
```

```
    total += count;
```

```
    this.periphery();
```

```
},
```

- 1

- 2

- 3

- 4

- 5

- 6

- 7

- 8

- 9

- 10

- 11

- 12

- 13
- 14
- 15

下面附上完整的代码：

```
<view>
```

```
  <swiper class="swiper_box" indicator-dots="{{indicatorDots}}" vertical="{{vertical}}"
```

```
    autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}" bindchange="swiper
```

```
change">
```

```
  <block wx:for="{{images}}">
```

```
    <swiper-item bindtap="itemclick" data-id="{{item.img}}" data-name="{{item.name}}">
```

```
      <image src="{{item.img}}" class="slide-image"/>
```

```
    </swiper-item>
```

```
  </block>
```

```
</swiper>
```

```
</view>
```

```
<scroll-view class="sv" scroll-y="true">
```

```

<view style="overflow:hidden;">
    <view class="items" wx:for="{{array}}" wx:for-item="item" bindtap="typeclick" data-code
= "{{item.code}}" data-text="{{item.text}}" >

        <image class="item-img" mode="aspectFit" src="{{item.src}}">image>

        <view class="item-text">{{item.text}}view>
view>
view>
<view class="data">

    <text class="data-title">{{title}}text>

    <view style="overflow:hidden;">

        <view class="data-items" wx:for="{{dataArray}}" wx:for-item="item" wx:key="id" bindta
p="openmap"

            data-lat="{{item.lat}}" data-lng="{{item.lng}}" data-name="{{item.name}}" data-address=
"{{item.address}}">

                <image class="data-item-img" mode="aspectFit" src="{{item.img}}">image>
                <view class="data-item-text">

                    <view style="width:100%; font-size: 30rpx; padding:2rpx;">{{item.name}}view>

                    <view style="width:100%; font-size: 25rpx; padding:2rpx;">{{item.address}}vie
w>

```

```
<view style="width:100%; font-size: 25rpx; padding:2rpx;">{{item.phone}}vie
```

W>

```
view>
```

```
view>
```

```
view>
```

```
view>
```

```
scroll-view>
```

```
<loading hidden="{{loadingHidden}}">
```

加载中...

loading>

• 1

• 2

• 3

• 4

• 5

• 6

• 7

• 8

- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 25

- 26

- 27

- 28

- 29

- 30

- 31

- 32

- 33

- 34

- 35

- 36

- 37

- 38

- 39

- 40

```
/**main.wxss**/
```

```
.swiper_box {
```

```
    width: 100%;
```

```
}
```

```
swiper-item image {
```

```
    width: 100%;
```

```
    display: inline-block;
```

```
    overflow: hidden;
```

```
}
```

```
.sv{
```

```
    background-color:#efff4;
```

```
    margin-top: 10rpx
```

```
}
```

```
.items{
```

```
    float:left;
```



```
width: 20%;  
background-color:#fff;  
}
```

```
.item-img{  
width: 100%;  
height: 60rpx;  
}
```

```
.item-text{  
width: 100%;  
height: 60rpx;  
font-size: 25rpx;  
text-align:center;  
}
```

```
.data{  
margin-top: 10rpx;  
background-color:#fff;
```

```
padding: 10rpx;
}

.data-title{
padding-left: 10rpx;
padding-top: 15rpx;
}

.data-items{
width: 100%;
margin-top: 10rpx;
margin-bottom: 10rpx;
overflow: hidden;
}

.data-item-img{
width: 20%;
height:120rpx;
float:left;
```

```
}
```

```
.data-item-text{
```

```
width: 75%;
```

```
padding: 5rpx;
```

```
height:120rpx;
```

```
float:left;
```

```
}
```

- 1

- 2

- 3

- 4

- 5

- 6

- 7

- 8

- 9

- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

- 26

- 27

- 28

- 29

- 30

- 31

- 32

- 33

- 34

- 35

- 36

- 37

- 38

- 39

- 40

- 41

- 42

- 43

- 44

- 45

- 46

- 47

- 48

- 49

- 50

- 51

- 52

- 53

- 54

- 55

- 56

- 57

```
//main.js
```

```
//获取应用实例
```

```
var app = getApp()
```

```
var count = 10;
```

```
var total = 0;
```

```
var code = "2";
```

```
Page({  
  data: {  
    title: "附近三公里",  
    indicatorDots: true,  
    vertical: false,  
    autoplay: true,  
    interval: 3000,  
    duration: 1000,  
    loadingHidden: false, // loading  
    array: [{  
      code: '1',  
      id: 'icon_1',
```

```
src: 'http://xxx',  
  
text: '家政'  
}, {  
  
  code: '2',  
  
  id: 'icon_2',  
  
  src: 'http://xxx',  
  
  text: '药店'  
}, {  
  
  code: '3',  
  
  id: 'icon_3',  
  
  src: 'http://xxx',  
  
  text: '银行'  
}, {  
  
  code: '4',  
  
  id: 'icon_4',  
  
  src: 'http://xxx',  
  
  text: '维修'  
}, {
```



```
code: '5',  
  
id: 'icon_5',  
  
src: 'http://xxx',  
  
text: '公厕'  
}, {  
  
code: '6',  
  
id: 'icon_6',  
  
src: 'http://xxx',  
  
text: '医院'  
}, {  
  
code: '7',  
  
id: 'icon_7',  
  
src: 'http://xxx',  
  
text: '加油站'  
}, {  
  
code: '8',  
  
id: 'icon_8',  
  
src: 'http://xxx',
```

```
        text: '汽车洗护'
      }, {
        code: '9',

        id: 'icon_9',

        src: 'http://xxx',

        text: '营业厅'
      }, {
        code: '10',

        id: 'icon_10',

        src: 'http://xxx',

        text: '停车场'
      }
    ],
    dataArray: []
  },
```

//事件处理函数

```
swiperchange: function (e) {

  // 此处写 轮播 改变时会触发的 change 事件

},
```

// 轮播 item 点击事件

```
itemclick: function (e) {  
  wx.showToast({  
    title: e.currentTarget.dataset.id + "",  
    icon: 'success',  
    duration: 2000  
  })  
},
```

// 分类 item 单击事件

```
typeclick: function (e) {  
  total = 0;  
  
  code = e.currentTarget.dataset.code + "";  
  
  var name = e.currentTarget.dataset.text + "";  
  
  this.data.dataArray = [];  
  
  this.setData({  
    title: "附近三公里: " + name  
  })  
}
```

```
    this.periphery();  
  },
```

```
onLoad: function () {  
  
    console.log('onLoad')  
  
    var that = this  
  
  
    count = 10;  
  
    total = 0;
```

//sliderList

```
wx.request({  
    url: 'http://xxx',  
  
    method: 'POST',  
    data: {  
        type: "1"  
    },  
    header: {  
        'Accept': 'application/json'  
    },  
    success: function (res) {
```

```
        that.setData({
            images: res.data.data.guanggao
        })
    }
})
```

```
    this.periphery();
},
```

// 网络请求

```
periphery: function () {
```

```
    var that = this
```

//sliderList

```
    wx.request({
```

```
        url: 'http://xxx',
```

```
        method: 'POST',
```

```
        data: {
```

```
            city: "深圳",
```

```
            code: code,
```

```
            count: count + "",
```

```
            total: total + "",
```

```
            lat: app.globalData.latitude + "",
```

```
        lng: app.globalData.longitude + ""
    },
    header: {
        'Accept': 'application/json'
    },
    success: function (res) {
        that.data.dataArray = that.data.dataArray.concat(res.data.data.list)
        that.setData({
            dataArray: that.data.dataArray
        })

        setTimeout(function () {
            that.setData({
                loadingHidden: true
            })
        }, 1000)
    }
})
},
```

// 下拉刷新回调接口

```
onPullDownRefresh: function () {

    total = 0;
```

```
    this.data.dataArray = [];  
  
    this.periphery();  
    wx.stopPullDownRefresh;  
  },
```

// 上拉加载回调接口

```
onReachBottom: function () {
```

```
    total += count;
```

```
    this.periphery();
```

```
  },
```

```
openmap: function (e) {
```

```
    wx.openLocation({
```

```
        latitude: e.currentTarget.dataset.lat , // 纬度 , 范围为-90~90 , 负数表示南纬
```

```
        longitude: e.currentTarget.dataset.lng, // 经度 , 范围为-180~180 , 负数表示西经
```

```
        scale: 28, // 缩放比例
```

```
        name: e.currentTarget.dataset.name, // 位置名
```

```
        address: e.currentTarget.dataset.address, // 地址的详细说明
```

```
    success: function(res){  
        // success  
    },  
    fail: function() {  
        // fail  
    },  
    complete: function() {  
        // complete  
    }  
    })  
    },  
    })
```

main.json

```
{  
    "enablePullDownRefresh": true  
}
```