

微信小程序之天气预报

1801210380
刘玄

一、注册微信小程序

首先进入微信公众平台注册登录一个小程序账号：
<https://developers.weixin.qq.com/miniprogram/introduction/index.html>
具体操作如上链接，注册成功登录之后进入小程序管理页面。
在设置里添加小程序的基本信息，如头像，名称，服务类目等，如图 1 所示。

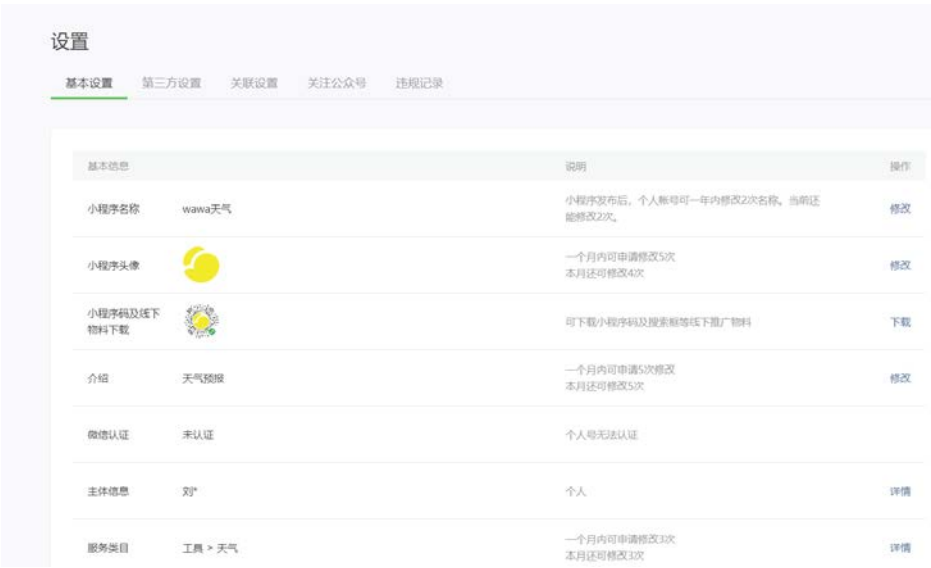


图 1

在开发页面可以获取小程序的 AppID，用来开发小程序使用。



图 2

下载微信小程序开发工具-微信开发者平台，链接如下：
<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html?t=181211>

二、微信小程序框架简介

新建项目

打开微信开发者平台，创建一个天气预报项目，AppID 输入自己注册的 ID，如果没有，可以点击或使用测试号：小程序，会随机生成一个测试 ID，测试 ID 不可以上传到微信公众平台审核和发布，选择建立普通快速启动模板。

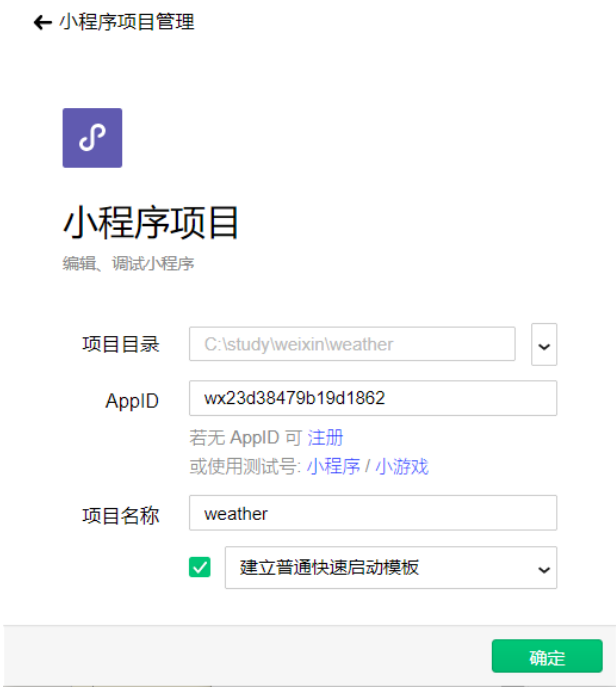


图 3

小程序文件简介

项目创建成功后如图 4 所示

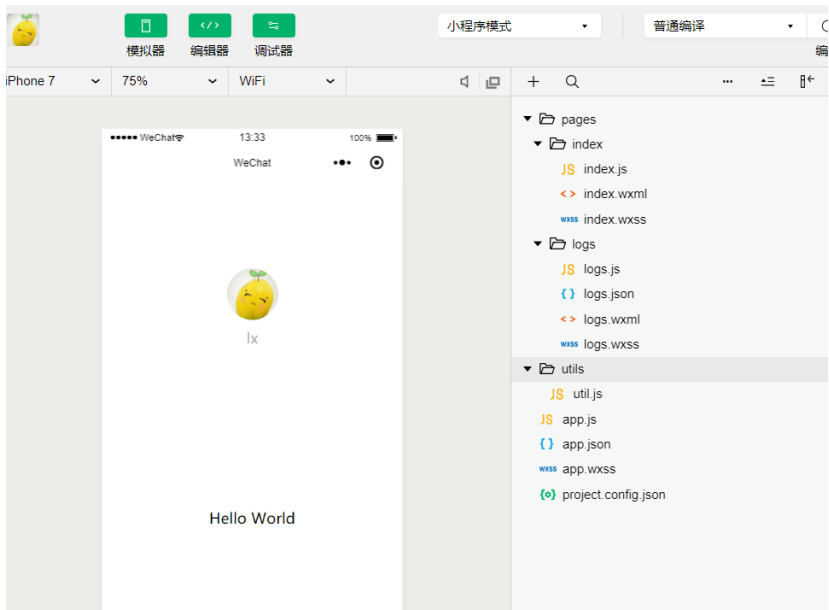


图 4

小程序文件包含了 pages 文件夹，pages 文件夹用来存放页面的文件。里面有 index 页面文件和 logs 页面文件，每一个页面文件夹有四个文件：

- .json 后缀的 JSON 配置文件；
- .wxml 后缀的 WXML 模板文件，对于前端的 html；
- .wxss 后缀的 WXSS 样式文件，对于前端的 CSS；
- .js 后缀的 JS 脚本逻辑文件。

小程序开发有三个模块，结构，表现，行为。并且编写程序时希望三个模块分离。

其中.wxml 代表的是结构，负责整个页面的结构表现，例如把一个页面分成很多块，相当于盖楼搭楼的架构。

.wxss 代表的是表现，负责给结构处理，例如某一块的颜色，背景，大小，居中等，相当于给楼房装修。

.js 代表的是行为，用来负责小程序的逻辑交互，可以获取后台数据，也可以与.wxml 打交道。

Utils 文件是公共文件，一般把常用到的函数 打包成模块，放到相应的文件里，方便调用，实现模块化设计。

app.json 是当前小程序的全局配置，包括了小程序的所有页面路径、界面表现、网络超时时间、底部 tab 等。

工具配置 project.config.json

通常大家在使用一个工具的时候，都会针对各自喜好做一些个性化配置，例如界面颜色、编译配置等等，当你换了另外一台电脑重新安装工具的时候，你还要重新配置。

考虑到这点，小程序开发者工具在每个项目的根目录都会生成一个 project.config.json，你在工具上做的任何配置都会写入到这个文件，当你重新安装工具或者换电脑工作时，你只要载入同一个项目的代码包，开发者工具就自动会帮你恢复到当时你开发项目时的个性化配置，其中会包括编辑器的颜色、代码上传时自动压缩等一系列选项。

具体如微信公众平台开发文档：

<https://developers.weixin.qq.com/miniprogram/dev/quickstart/basic/file.html>

三、编写天气预报

本文参考文档：<http://www.demodashi.com/demo/10634.html>

新建一个天气页面， pages 文件下面，创建一个 weather 文件夹，里面在创建 weather.json，weather.wxml，weather.wxss，weather.js，四个文件。

但一个一个新建太麻烦了。

快捷方式：在 app.json 里写入如下

"pages/weather/weather",注册页面代码，自动会生成上述所有文件。

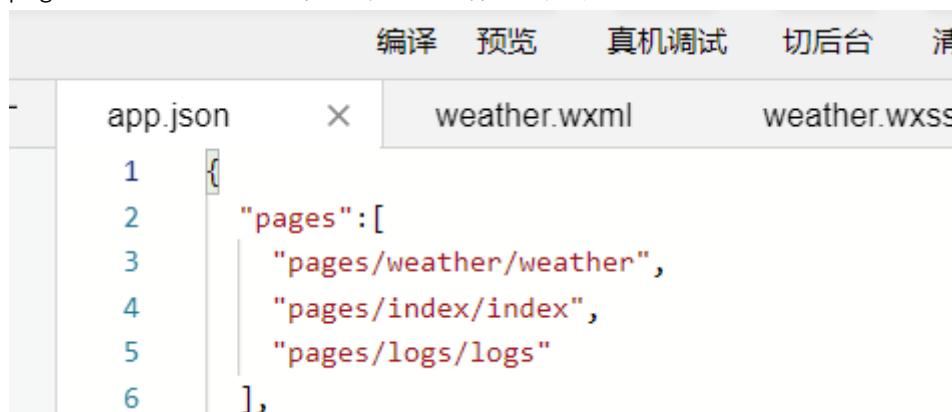


图 5

天气预报结构 weather.wxml 编写：
结构图如下：
黄色代表背景，其他代表分块。



图 6

根据结构图代码如下：
Class 相当于这个块的名字，利用名字可以在 wxss 里给不同的块给与不同的表现形式。

```

<image src="../../images/bg5.jpg" class="bg"></image>
<view class="container">

  <view class="nowWeather">
    <view class="temp w ">{{tmp}}°</view>
    <view class="city w">{{city}} {{district}} {{street}}</view>
    <view class="weather w">{{txt}}</view>
  </view>

  <view class="pm25 w">空气{{qlty}} {{pm25}}</view>

  <view class="weahterDetail">
    <view class="">
      <view class="w center">{{dir}}</view>
      <view wx:if="{{sc == '微风'}}" class="w b center f50">微风</view>
      <view wx:else class="w b center f50">{{sc}}级</view>
    </view>
    <view class="l"></view>
    <view class="">
      <view class="w center">相对湿度</view>
      <view class="w b center f50">{{hum}}%</view>
    </view>
    <view class="l"></view>
    <view class="">
      <view class="w center">体感温度</view>
      <view class="w b center f50">{{fl}}°</view>
    </view>
  </view>
  <view wx:for="{{daily_forecast}}" wx:for-index="i" wx:for-item="item">

    <view class="hor forecast">
      <view class="center">{{day[i]}}</view>
      <view class="hor">
        <view class="center">{{item.cond_txt_d}}|{{item.wind_dir}}</view>
      </view>
      <view class="center">{{item.tmp_min}}° / {{item.tmp_max}}°</view>
    </view>
  </view>
</view>

```

项目里建立 images 文件夹，里面存放图标图和背景图，如图 7 所示。

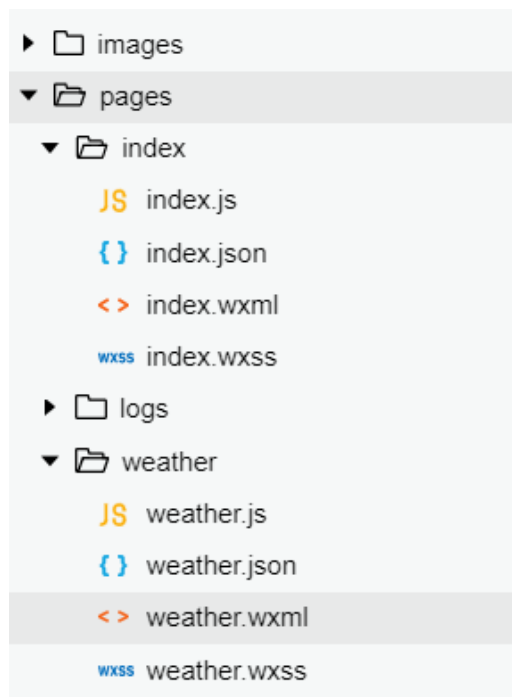


图 7

天气预报表现 weather.wxss 编写：

根据自己的喜好，给不同的结构展现不同的样式。

代码如下：

```
.w{
  color: white;
}

.b{
  font-weight: bold;
}

.l{
  border: 1rpx solid #fff;
}

.center{
  text-align: center;
  margin: auto 0;
}

.hor{
  display: flex;
}
```

```
.bg {  
  width: 100%;  
  height: 1800rpx;  
  
}  
  
.temp{  
  font-size: 170rpx;  
}  
  
.container {  
  position: absolute;  
  left: 0;  
  top: 0;  
  width: 100%;  
  padding: 0;  
  margin: 0;  
  height: 1800rpx;  
  display: block;  
}  
  
.nowWeather{  
  padding: 60rpx;  
}  
  
.pm25{  
  padding: 200px 60px 0px 30px;  
}  
  
.weahterDetail{  
  width: 100%;  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
  padding: 10px 0px 100px 10px;  
}  
  
.forecast{  
  padding: 50px 0px 0px 0px;  
  margin-left: 10rpx;  
  margin-right: 10rpx;  
  font-size: 40rpx;  
  border-bottom: 1rpx solid rgb(241, 229, 229);  
  color: white;
```

```

justify-content: space-around;
}

```

天气预报行为 weather.js 编写：

主要是获取本地位置和获取天气数据让页面显示天气信息。

获取位置：

利用微信接口获取经纬度：

//获取经纬度方法

```

getLocation: function () {
  var that = this
  wx.getLocation({
    type: 'wgs84',
    success: function (res) {
      var latitude = res.latitude
      var longitude = res.longitude
      console.log("lat:" + latitude + " lon:" + longitude);

      that.getCity(latitude, longitude);
    }
  })
},

```

利用获取到的经纬度调用百度地图 API 获取到当前城市，链接如下

<http://lbsyun.baidu.com/apiconsole/key/create>

创建应用

应用名称:

应用类型: 微信小程序

启用服务: ☒ 地点检索 ☒ 正逆地理编码 ☒ 天气查询API

APP ID:

只有使用该真实APPID的开发者才能成功发起调用

APPID是一个由大小写字母和阿拉伯数字组成的验证码，通过微信小程序微信认证获得，只有使用该真实APPID的开发者才能成功发起调用。

格式: APPID1,APPID2 多个APPID之间请用英文半角逗号隔开

如开发者还未认证成为微信小程序的开发者，APPID请填写为touristappid，为开发者模式，如需发布小程序请及时更改为真实APPID 如果不想对APPID做限制，设置为英文半角星号* (谨慎使用，AK如果泄露配额会被其用户消费，发布线上小程序请设置真实使用APPID)

提交

图 8

进入百度地图开放平台创建一个微信小程序应用，填写对应的 APPID。

在查看应用里获取自己的 AK 值。



图 9

获取城市信息代码如下：

你的百度地图 ak 别忘了换。

```

getCity: function (latitude, longitude) {
    var that = this;
    var url = "https://api.map.baidu.com/geocoder/v2/";
    var params = {
        ak: "你的百度地图 ak",
        output: "json",
        location: latitude + "," + longitude
    }
    wx.request({
        url: url,
        data: params,
        success: function (res) {
            var city = res.data.result.addressComponent.city;
            var district = res.data.result.addressComponent.district;
            var street = res.data.result.addressComponent.street;

            that.setData({
                city: city,
                district: district,
                street: street,
            })

            var descCity = city.substring(0, city.length - 1);
            that.getWeahter(descCity);
        },
        fail: function (res) { },
        complete: function (res) { },
    })
},

```

根据城市信息，获取天气信息：

注册和风天气，

<http://www.heweather.com/>

获取免费天气接口 API，获取 KEY。

普通用户

用户ID: HE1812111933241972

暂时还没有购买服务 [购买](#)

认证key: 27[REDACTED]2978

图 10

你的和风天气 key 别忘了换，和风天气 api 开发文档如下，可以参考添加自己想要的天气数据。

<https://www.heweather.com/documents>

获取天气信息代码：

```
getWeather: function (city) {
  var that = this
  var url = "https://free-api.heweather.com/s6/weather"
  var url2 = "https://free-api.heweather.com/s6/air"
  var parameters = {
    location: city,
    key: "你的和风天气 key"
  }
  wx.request({
    url: url,
    data: parameters,
    success: function (res) {
      var tmp = res.data.HeWeather6[0].now.tmp;
      var txt = res.data.HeWeather6[0].now.cond_txt;
      var code = res.data.HeWeather6[0].now.cond_code;
      var dir = res.data.HeWeather6[0].now.wind_dir;
      var sc = res.data.HeWeather6[0].now.wind_sc;
      var hum = res.data.HeWeather6[0].now.hum;
      var fl = res.data.HeWeather6[0].now.fl;
      var daily_forecast = res.data.HeWeather6[0].daily_forecast;
      that.setData({
        tmp: tmp,
        txt: txt,
        code: code,
        dir: dir,
        sc: sc,
        hum: hum,
        fl: fl,
        daily_forecast: daily_forecast
      })
    }
  })
}
```

```

        })
    },
    fail: function (res) { },
    complete: function (res) { },
})
wx.request({
    url: url2,
    data: parameters,
    success: function (res) {
        var qlty = res.data.HeWeather6[0].air_now_city.qlty;
        var pm25 = res.data.HeWeather6[0].air_now_city.pm25;
        // var air_forecast = res.data.HeWeather6[0].air_forecast;
        that.setData({
            qlty: qlty,
            pm25:pm25
            // air_forecast: air_forecast
        })
    },
    fail: function (res) { },
    complete: function (res) { },
})
}

}))

```

Weather.js 全部代码如下

```

var app = getApp()
var day = ["今天", "明天", "后天"];
Page({
    data: {
        day: day,
    },

    onLoad: function () {
        console.log('onLoad')
        var that = this
        that.getLocation();
    },

    //获取经纬度方法
    getLocation: function () {

```

```

var that = this
wx.getLocation({
  type: 'wgs84',
  success: function (res) {
    var latitude = res.latitude
    var longitude = res.longitude
    console.log("lat:" + latitude + " lon:" + longitude);

    that.getCity(latitude, longitude);
  }
})
},

//获取城市信息
getCity: function (latitude, longitude) {
  var that = this;
  var url = "https://api.map.baidu.com/geocoder/v2/";
  var params = {
    ak: "你的百度地图 ak",
    output: "json",
    location: latitude + "," + longitude
  }
  wx.request({
    url: url,
    data: params,
    success: function (res) {
      var city = res.data.result.addressComponent.city;
      var district = res.data.result.addressComponent.district;
      var street = res.data.result.addressComponent.street;

      that.setData({
        city: city,
        district: district,
        street: street,
      })

      var descCity = city.substring(0, city.length - 1);
      that.getWeahter(descCity);
    },
    fail: function (res) { },
    complete: function (res) { },
  })
},

```

//获取天气信息

```
getWeahter: function (city) {
  var that = this
  var url = "https://free-api.heweather.com/s6/weather"
  var url2 = "https://free-api.heweather.com/s6/air"
  var parameters = {
    location: city,
    key: "你的和风天气 key"
  }
  wx.request({
    url: url,
    data: parameters,
    success: function (res) {
      var tmp = res.data.HeWeather6[0].now.tmp;
      var txt = res.data.HeWeather6[0].now.cond_txt;
      var code = res.data.HeWeather6[0].now.cond_code;
      var dir = res.data.HeWeather6[0].now.wind_dir;
      var sc = res.data.HeWeather6[0].now.wind_sc;
      var hum = res.data.HeWeather6[0].now.hum;
      var fl = res.data.HeWeather6[0].now.fl;
      var daily_forecast = res.data.HeWeather6[0].daily_forecast;
      that.setData({
        tmp: tmp,
        txt: txt,
        code: code,
        dir: dir,
        sc: sc,
        hum: hum,
        fl: fl,
        daily_forecast: daily_forecast
      })
    },
    fail: function (res) { },
    complete: function (res) { },
  })
  wx.request({
    url: url2,
    data: parameters,
    success: function (res) {
      var qlty = res.data.HeWeather6[0].air_now_city.qlty;
      var pm25 = res.data.HeWeather6[0].air_now_city.pm25;
      // var air_forecast = res.data.HeWeather6[0].air_forecast;
      that.setData({
        qlty: qlty,
```

```

        pm25:pm25
        // air_forecast: air_forecast
    })
  },
  fail: function (res) { },
  complete: function (res) { },
})
}

}))

```

Weather.json 设置顶部栏样式

```

{
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#fff",
  "navigationBarTitleText": "天气",
  "navigationBarTextStyle": "black"
}

```

编译运行即可成功获取天气信息：
背景图和 tabbar 图可以去网上下载。



图 11



图 12

附加：

分享功能：在 weather.js 添加如下分享代码如下，即可。

```
onShareAppMessage: function () {  
  return {  
    path: "/pages/weather/weather",  
    success: function () {  
      wx.showToast({  
        title: '分享成功',  
        icon: "success",  
        duration: 2000  
      })  
    }  
  }  
}
```

Tabbar：

在 app.json 里添加 tabbar 代码。

```
"tabBar": {  
  "color": "#a0a0a0",  
  "selectedColor": "#ec5c30",
```

```

"backgroundColor": "#f2f2f2",
"borderStyle": "white",
"list": [
  {
    "pagePath": "pages/weather/weather",
    "text": "天气",
    "iconPath": "images/weather.png",
    "selectedIconPath": "images/weather3.png"
  },
  {
    "pagePath": "pages/index/index",
    "text": "我",
    "iconPath": "images/my.png",
    "selectedIconPath": "images/my3.png"
  }
]
}

```



图 13



图 14

然后可以选择上传，将会上传到小程序管理后台。



图 15

进入微信公众平台小程序管理后台即可提交审核，审核通过后，提交线上版本，就可以在微信里搜索到你的微信小程序 app 了。



图 16



图 17

至此，完成了小程序天气预报的开发上传工作。