

微信小程序

概述

小程序是一种不需要下载安装即可使用的应用，而传统的 APP（Android/iOS），则需要在手机的应用商店中，下载之后，才能使用。小程序实现了应用“触手可及”的梦想，用户扫一扫或搜一下即可打开应用。

小程序无需下载，在宿主APP中可以直接使用。宿主APP可以给小程序带来流量。所以，小程序的推广成本低（相对与原生APP来说）。这是小程序最大的优势。

目前各大平台，都推出了自己的小程序。例如：微信，支付宝，百度，今日头条等。但最具影响力的，还是微信小程序。**后续章节，如无特殊说明。所有演示，都已微信小程序为主。**

微信小程序简介

借助微信，是一个不需要下载安装就可以使用的应用，用户只要扫一扫（小程序二维码）或者搜一搜（小程序应用关键词）就可以使用的应用，实现应用触手可及的梦想。比较出名的小程序：跳一跳

微信小程序的重要时间节点：

1. 2016年1月9日，正式立项
2. 2016年9月21日，开始内测
3. 2017年1月9日，正式上线

微信小程序入口



注册小程序账号

使用小程序之前，先要注册小程序账号。

前往: <https://mp.weixin.qq.com/wxopen/waregister?action=step1>

或参考: [注册微信小程序.md](#)

注册完成后，我们点击左侧导航中的“设置”，我们查看小程序信息。

其中，AppID 是当前小程序的ID，后续章节会多次使用（注意保密-不要随意暴露）

帐号信息

AppID(小程序ID)

wx

登录邮箱

@163.com

一个月内可申请修改1次
本月还可修改1次

[修改](#)

登录密码

用管理员微信扫码验证后，可修改你的密码

[修改](#)

原始ID

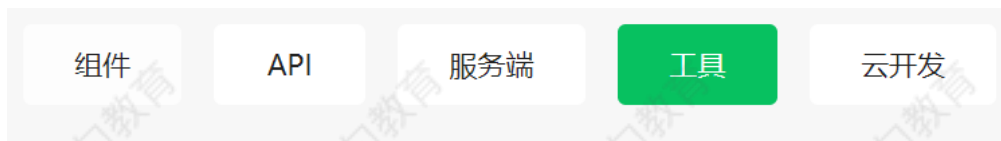
gh_

[帐号注销](#)

环境搭建

安装微信开发者工具

想要开发微信小程序，必须去微信公众平台，先[下载](#)微信开发者工具，下载之后，一路 **下一步** 安装。



Windows 仅支持 Windows 7 及以上版本

稳定版 **Stable Build (1.03.2010240)**

测试版缺陷收敛后转为稳定版

Windows 64 、 Windows 32 、 macOS

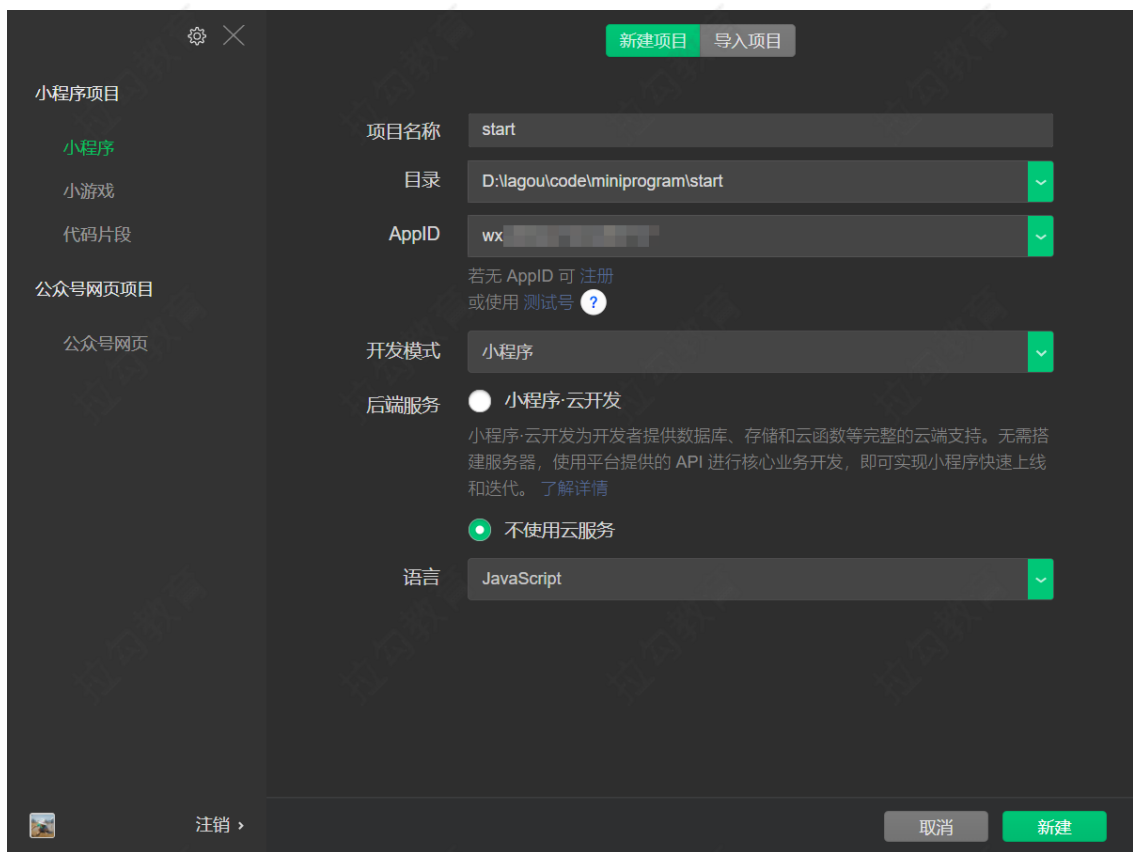
初始化小程序

1. 双击打开**微信开发者工具**

2. 填写项目信息

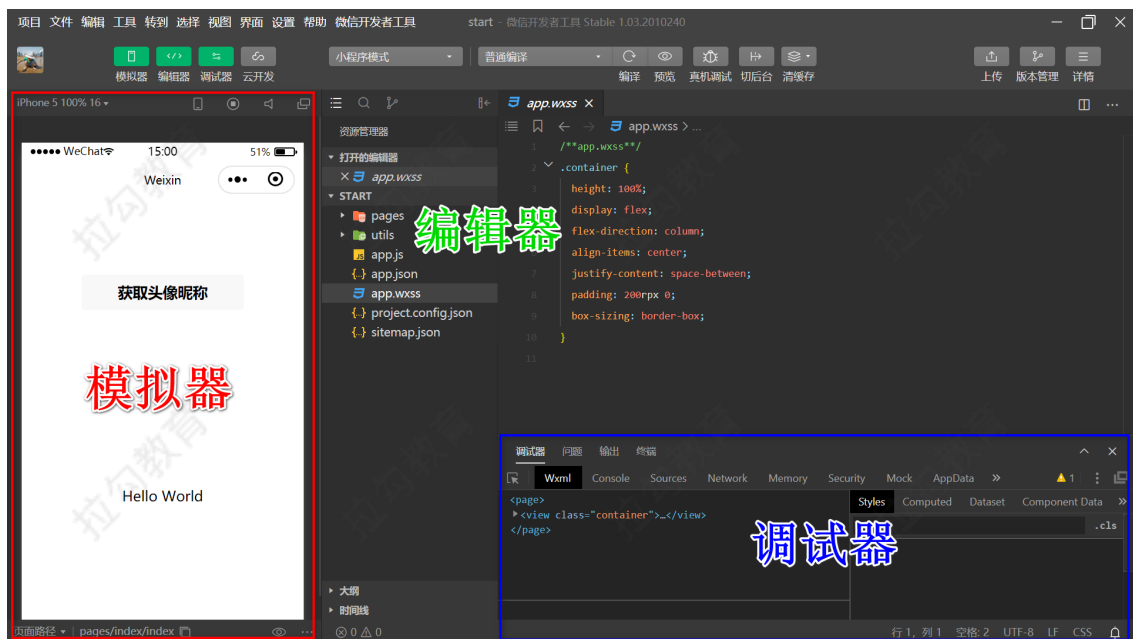
填写小程序名称、存放目录、AppID，填写完成后，点击**新建**

- 项目名称: 可以任意设置
- 目录: 设置本地空目录
- AppID: 注册小程序账号之后, 点击左侧导航中的“**设置**”，查看**账号信息**
- 开发模式: 小程序
- 后端服务: 选择“不使用云服务”（不会创建服务器端目录或者文件）



3. 进入项目后，熟悉微信开发者工具

- 模拟器: 模拟小程序项目在真机上的运行效果
- 编辑器: 修改/添加项目结构和代码
 - 代码目录
 - 代码编辑窗口
- 调试器: 调试 JavaScript 逻辑, 调试CSS样式等
 - Wxml: 查看页面标签结构, 调试CSS样式
 - Console: 测试 JavaScript 代码
 - Sources: 添加断点 / 断点调试



报错: 当前开发者未绑定此 appid,请到小程序管理

- 解决：1. 确保你使用的是小程序的 appid（而不是微信公众号的 appid）
2. 去小程序管理后台绑定开发者（如第一步已解决，请忽略第二步）

进入小程序

点击**获取头像昵称** -> **允许授权** -> 进入小程序（看到用户头像）



基础

储备知识

掌握 XML 基础

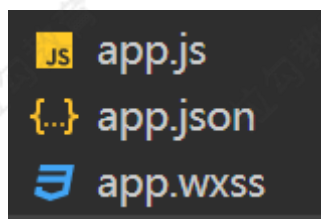
掌握 CSS（例如：弹性布局）

掌握 JavaScript 基础以及 ES6+ 新特性

目录结构

小程序包含一个描述整体程序的 `app` 和多个描述各自页面的 `page`。

全局文件



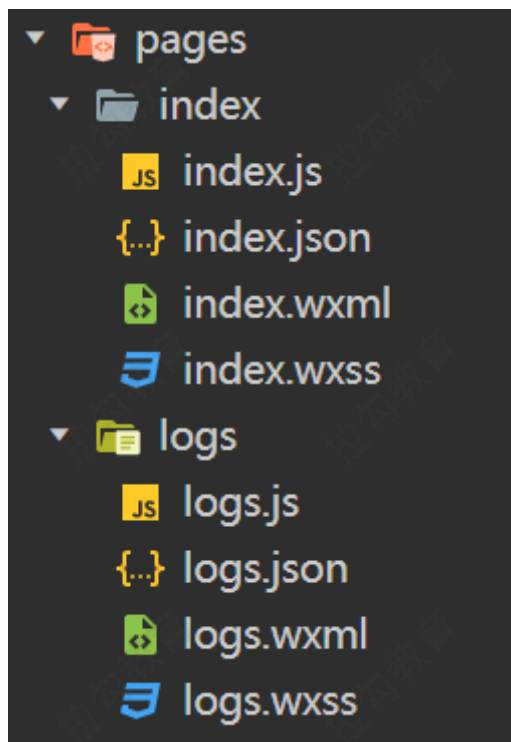
上述三个文件，在项目的根目录下，**文件位置和文件名（包括后缀名）是固定的（都不能改）。**

其中：

- `app.js` 是小程序启动后，加载的入口文件

- app.json 是整个小程序项目的配置文件
- app.wxss 是整个小程序项目的全局样式文件

页面文件



在小程序中，每个具体的页面都由四个文件组成。页面的四个文件必须具有相同的路径与文件名

- page-name.js 当前页面的入口文件
- page-name.json 当前页面的配置文件
- page-name.wxml 当前页面的内容（相当于浏览器端的 HTML）

WXML 语法: <https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/>
<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/>

- page-name.wxss 当前页面的样式文件，你可以在这里写 CSS 代码

WXSS 语法: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxs.html>

如何生成页面文件

与其他项目不同，小程序中的页面文件无需自己创建。

如果要新建页面文件，只需要在 app.json 中的 pages 字段中指定，然后保存 app.json。

这样页面文件就会自动创建出来。例如：下方的 page/me/my-page 就是通过这种方式创建的



生命周期

小程序的生命周期是指小程序生命不同阶段。这里的周期主要指两个循环状态：**前台到后台**；**初始化到销毁**。

在小程序生命不同阶段，都会自动触发函数，可以在生命周期函数中，添加相应业务逻辑。小程序的生命周期可以分为两类：

- 项目生命周期：描述小程序整个项目的不同阶段
- 页面生命周期：描述页面的不同阶段

项目生命周期

官方文档：<https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/app.html>

```
App({
  onLaunch (options) {
    // Do something initial when launch.
  },
  onShow (options) {
    // Do something when show.
  },
  onHide () {
    // Do something when hide.
  },
  onError (msg) {
    console.log(msg)
  },
 .globalData: 'I am global data'
})
```

- `onLaunch`: 全局只调用一次（逻辑只执行一次）
- `onShow`: 应用初始化完成；从后台切换到前台
- `onHide`: 从前台切换到后台（小程序应用转到别的应用）

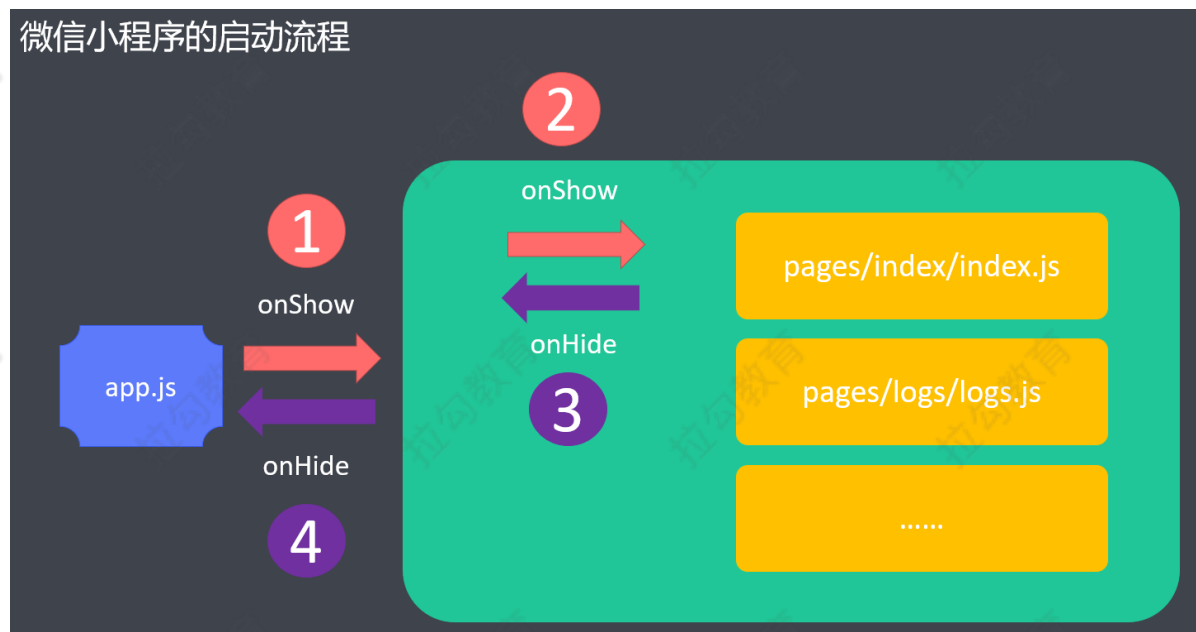
页面生命周期

官网文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/page.html>

```
//index.js
Page({
  data: {
    text: "This is page data."
  },
  onLoad: function(options) {
    // 页面创建时执行
  },
  onShow: function() {
    // 页面出现在前台时执行
  },
  onReady: function() {
    // 页面首次渲染完毕时执行
  },
  onHide: function() {
    // 页面从前台变为后台时执行
  },
  onUnload: function() {
    // 页面销毁时执行
  }
})
```

- onLoad: 小程序注册完成后, 加载页面, 一个页面只会调用一次
- onShow: 页面载入后, 显示页面, 每次打开页面都会调用一次
- onReady: 首次显示页面, 会触发onReady方法, 渲染页面元素和样式, 一个页面只会调用一次
- onHide: 小程序后台运行或跳转到其他页面时, 触发onHide方法
- onUnload: 当使用重定向方法wx.redirectTo(OBJECT)或关闭当前页返回上一页wx.navigateBack(), 触发onUnload, 页面卸载 (内存释放)

小程序的启动流程



- 1、小程序初始化完成后，页面首次加载触发onLoad，只会触发一次。
- 2、当小程序进入到后台，先执行**页面onHide方法**再执行**应用onHide方法**。
- 3、当小程序从后台进入到前台，先执行**应用onShow方法**再执行**页面onShow方法**。

解读初始化项目

app 代码

一个小程序主体部分由三个文件组成，必须放在项目的根目录，如下：

文件	必需	作用
app.js	是	小程序逻辑
app.json	是	小程序公共配置
app.wxss	否	小程序公共样式表

1. app.js

1. 本地文件存储

可以将全局应用的数据，存储到本地缓存中。操作本地缓存数据，可以通过 云API 实现

设置本地缓存数据：wx.setStorageSync(key, value)

获取本地缓存数据：wx.getStorageSync(key)

2. 获取用户信息

wx.getUserInfo() 可以返回用户的昵称、头像、所在城市等。

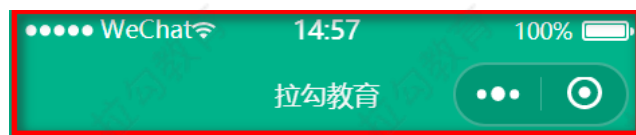
```
app.js? [sm]:23
{errMsg: "getUserInfo:ok", rawData:
  ("nickName": "初六", "gender": 1, "language": "zh_CN", "city": "Qingdao", "province": "Shandong", "country": "China", "avatarUrl": "https://thirdwx.qlogo.cn/mmopen/vi_32/0Qbiaey0otm9ric6ma821TmOp5N...4o8Mzd...
  signature: "07ee6065aba64a9bc091c921d5a493c0127811f9", encryptedData:
    "Mjx+nGSUvaG0y3JxLJMHBmMvIKyZSK1W3BMDn8eTK0iBymo4EZ...8an3BBOAdcvLOyrMPC
    encryptedData: "Mjx+nGSUvaG0y3JxLJMHBmMvIKyZSK1W3BMDn8eTK0iBymo4EZ52F/Mhx7nqa4kNgDKJZaH/3J8l...4o8Mzd...
    errMsg: "getUserInfo:ok"
    iv: "Gbdgj2oZ0qr7nUEQXNu8A=="
    rawData: "{ \"nickName\": \"初六\", \"gender\": 1, \"language\": \"zh_CN\", \"city\": \"Qingdao\", \"province\": \"Shandong\", \"country\": \"China\", \"avatarUrl\": \"https://thirdwx.qlogo.cn/mmopen/vi_32/0Qbiaey0otm9ric6ma821TmOp5N...4o8Mzd...
    signature: \"07ee6065aba64a9bc091c921d5a493c0127811f9\" }"
  }
  userInfo:
    avatarUrl: "https://thirdwx.qlogo.cn/mmopen/vi_32/0Qbiaey0otm9ric6ma821TmOp5N...4o8Mzd..."
    city: "Qingdao"
    country: "China"
    gender: 1
    language: "zh_CN"
    nickName: "初六"
    province: "Shandong"
    __proto__: Object
  }
  __proto__: Object
```

2. app.json

app.json 负责全局配置，例如：设置小程序的导航条、标题、窗口背景色

```
"window": {
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#00b38a",
  "navigationBarTitleText": "拉勾教育",
  "navigationBarTextStyle": "white"
},
```

效果图：



初六

全局配置: <https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html>

3. app.wxss

app.wxss 负责全局样式, 例如: .container 中使用的弹性布局。

与 CSS 相比, WXSS 扩展的特性有:

详情查看: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxss.html>

云 API

云 API, 是微信公众平台提供的, 供小程序调用的接口。在示例程序中, 用到的云 API 有:

云 API	作用
wx.getUserInfo	获取用户信息
wx.login	调用接口获取登录凭证 (code)
wx.getSetting	获取用户当前设置
wx.setStorageSync	将数据同步存储在本地缓存中
wx.getStorageSync	同步获取本地缓存中的数据
wx.navigateTo	保留当前页面, 跳转到应用内的某个页面
.....

其他云 API, 请参考官方文档

官方文档: <https://developers.weixin.qq.com/miniprogram/dev/api/>

页面跳转

在小程序中做页面跳转，主要有以下三种方式。

- 链接方式

通过 navigator 组件来实现

```
# 相对地址
<navigator url="/pages/me/my-page">my-page</navigator>

# URL地址
<navigator url="https://www.baidu.com/">百度</navigator>
```

链接详情: <https://developers.weixin.qq.com/miniprogram/dev/component/navigator.html>

- 事件方式

1. 在 .wxml 中，通过 bindtap 属性绑定 **bindViewTap** 事件

```
<image bindtap="bindViewTap" class="userinfo-avatar" src="{{userInfo.avatarUrl}}" mode="cover"></image>
```

2. 在 .js 文件中声明事件

```
Page({
  bindViewTap: function(event) {
    // console.log(event)
    wx.navigateTo({
      url: '../logs/logs'
    })
  }
})
```

- tabBar

在 app.json 中，通过 tabBar 字段指定。

详情查看: <https://developers.weixin.qq.com/miniprogram/dev/framework/ability/custom-tabbar.html>

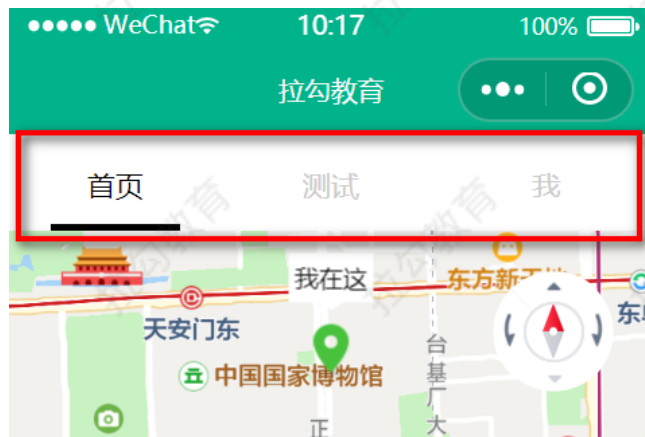
```
"tabBar": {
  "color": "#ccc",
  "selectedColor": "#000",
  "position": "top",
  "list": [
    {
      "text": "首页",
      "iconPath": "static/images/featured.png",
      "selectedIconPath": "static/images/featured-actived.png",
      "pagePath": "pages/qweather/index"
    },
    {
      "text": "测试",
      "iconPath": "static/images/search.png",
      "selectedIconPath": "static/images/search-actived.png",
      "pagePath": "pages/test/test"
    }
  ]
}
```

```

{
  "text": "我",
  "iconPath": "static/images/profile.png",
  "selectedIconPath": "static/images/profile-actived.png",
  "pagePath": "pages/index/index"
}
],
},

```

tabBar 效果



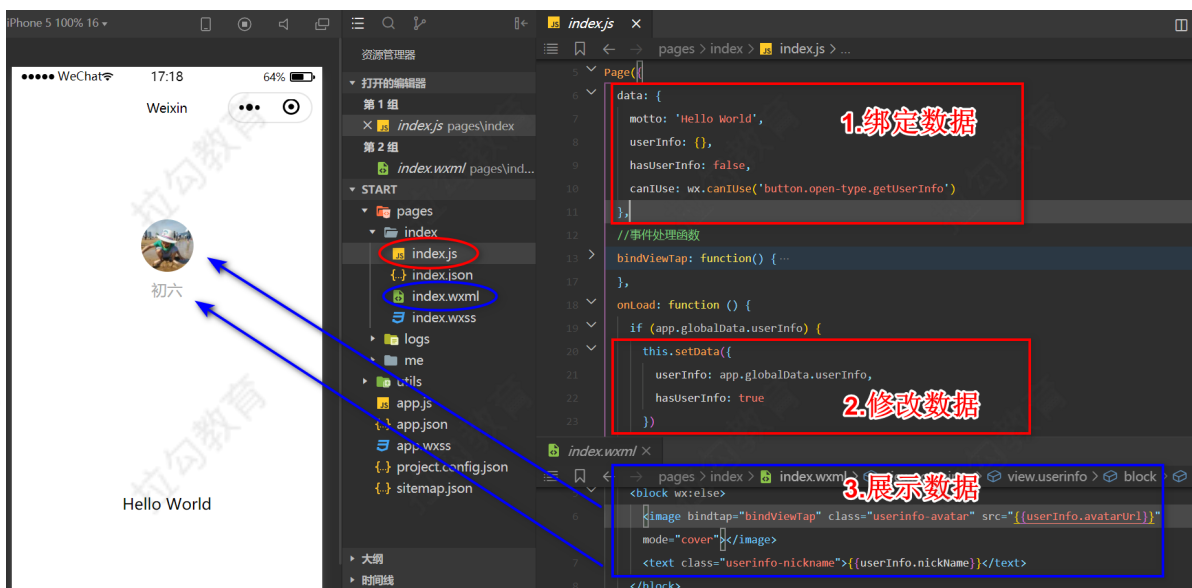
数据绑定

1. 在 .js 文件中绑定数据

- o data 字段设置数据的初始值
- o this.setData 函数可以更新 data 中绑定的数据

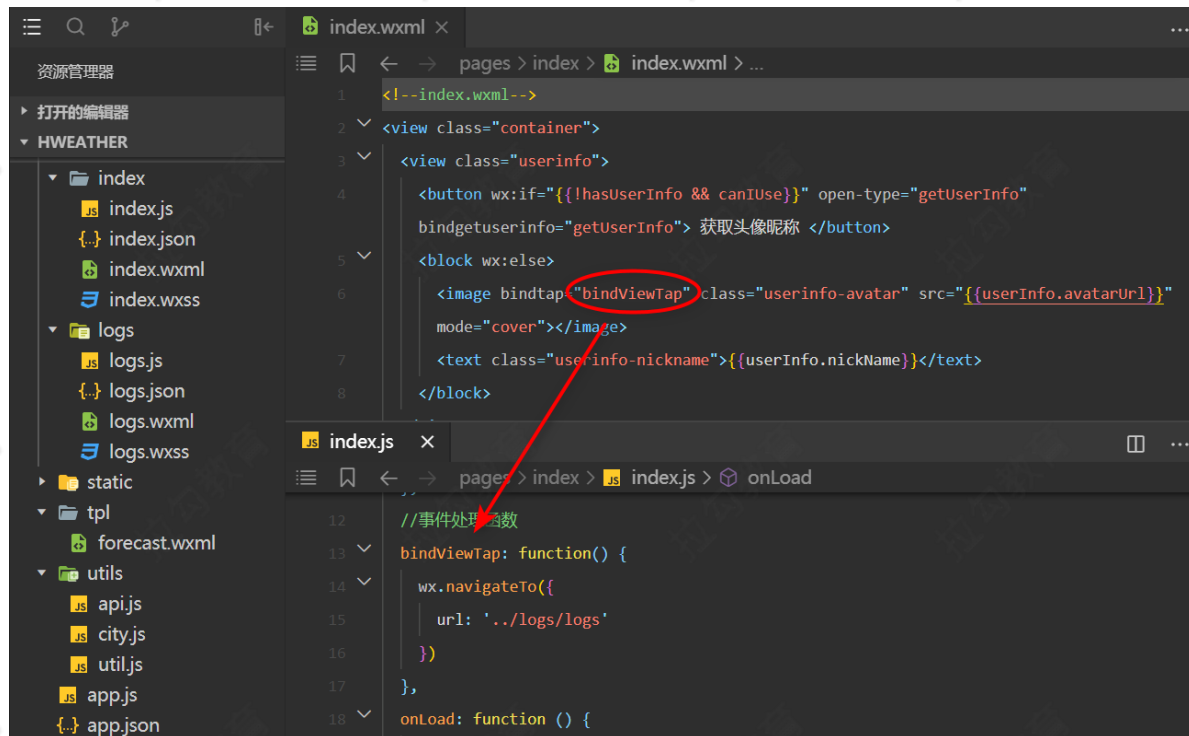
2. 在 .wxml 中展示绑定的数据

- o 展示语法是 {{变量名}}
- o WXML语法: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/>



事件触发

通过组件的 bindtap 属性，可以绑定触碰事件。如下图：



事件详情：<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>

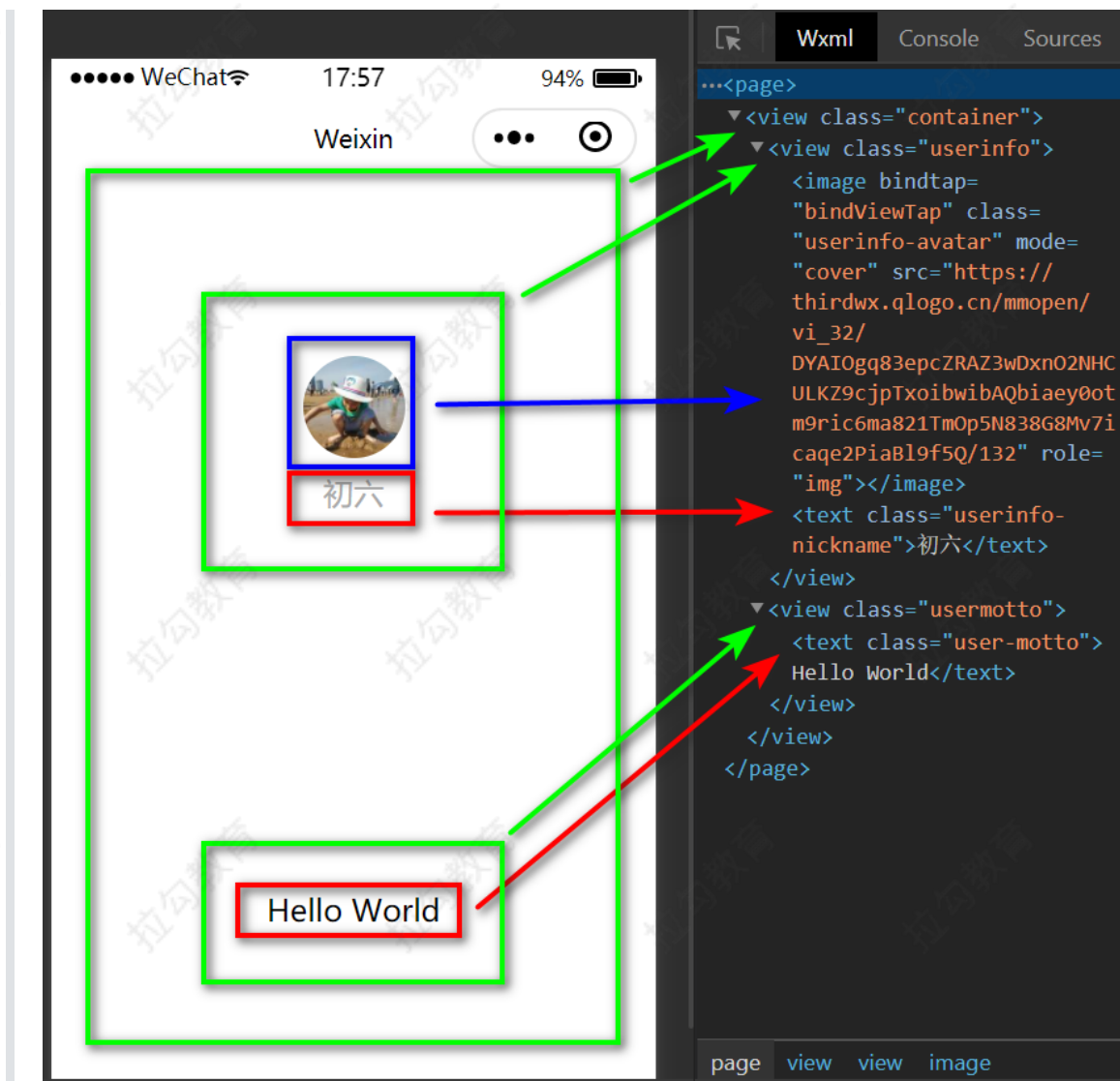
组件

小程序中的组件，对应HTML中的标签。

小程序不是运行在浏览器上，所以 HTML 标签不能使用。小程序中，展示页面内容的标签，称之为组件。

作用	小程序组件	HTML 标签
展示区块	view	div
展示图片	image	img
展示文本	text	p
链接导航	navigator	a
.....

组件在 .wxml 文件中的使用



官网提供了功能丰富的组件。详情查看官方文档

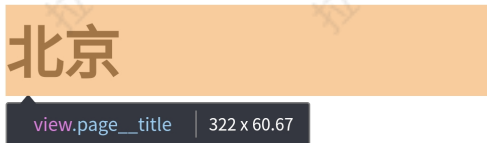
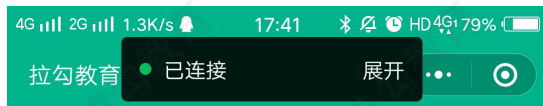
官方文档: <https://developers.weixin.qq.com/miniprogram/dev/component/>

调试

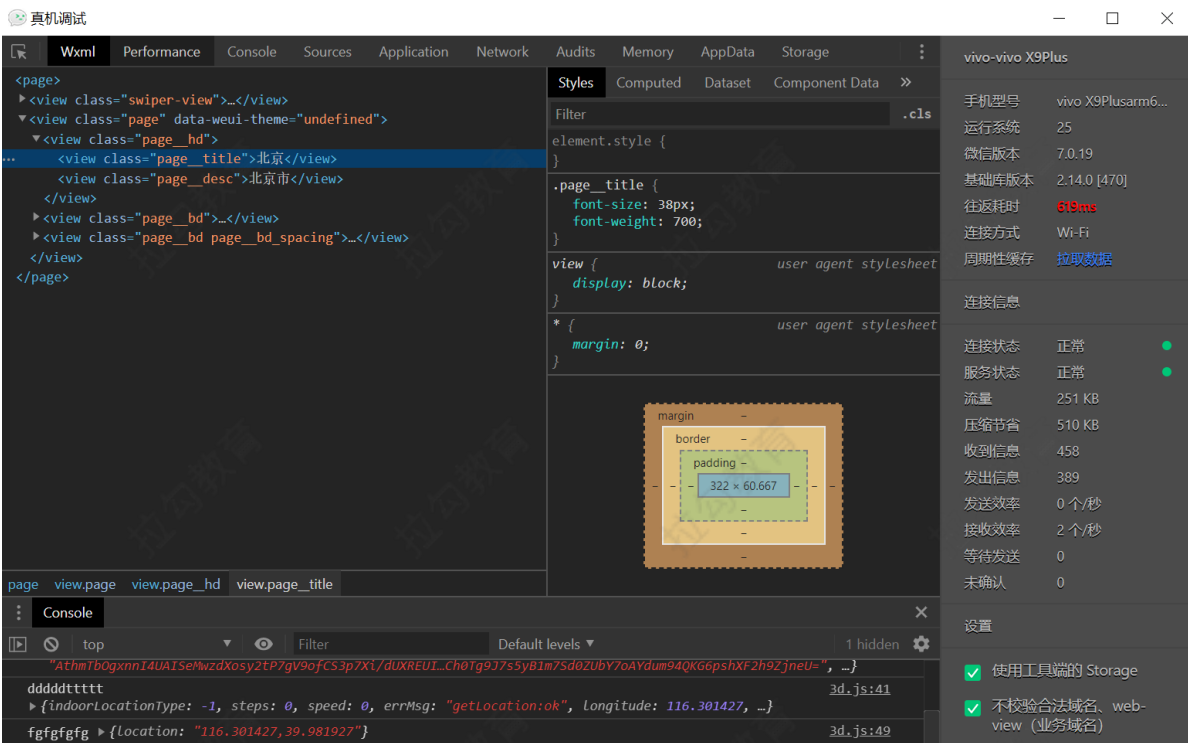
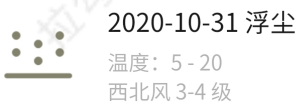


点击 预览, 查看小程序的效果。

点击 真机调试, 预览只查看效果, 真机调试会伴随调试窗口



未来三天的天气情况



项目实践

项目需求

1. 进入小程序首页时，获取当前地理位置信息

通过 [wx.getLocation](#) 可以获取当前位置信息



2. 根据定位信息，查询当前位置的天气预报

目前有很多查询天气预报的第三方接口，本节使用和风天气 API



数据接口

申请接口 - 和风天气 API

和风天气提供了免费的天气预报相关的 API。但是，使用之前，需要注册申请



1.注册账号



2.创建应用和KEY



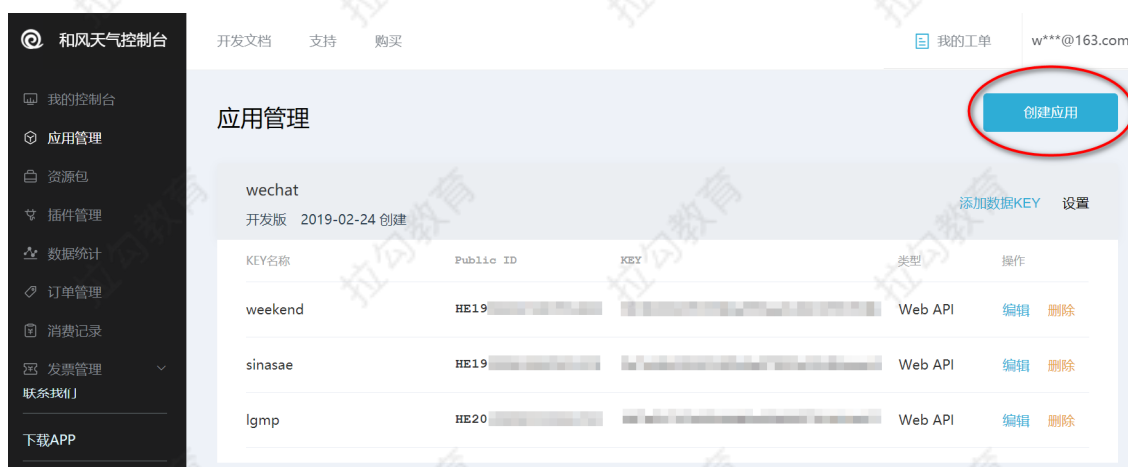
3.开发集成

1. 注册和风天气账号

<https://id.qweather.com/#/register>

2. 创建应用，获取 key

登陆账号，进入控制台



3. 开发集成

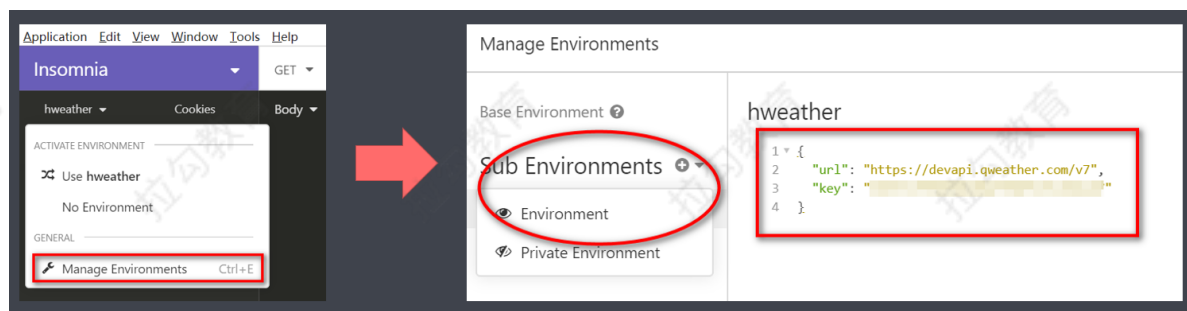
<https://dev.qweather.com/docs/start/#documentation>

调试接口 - API 调试工具

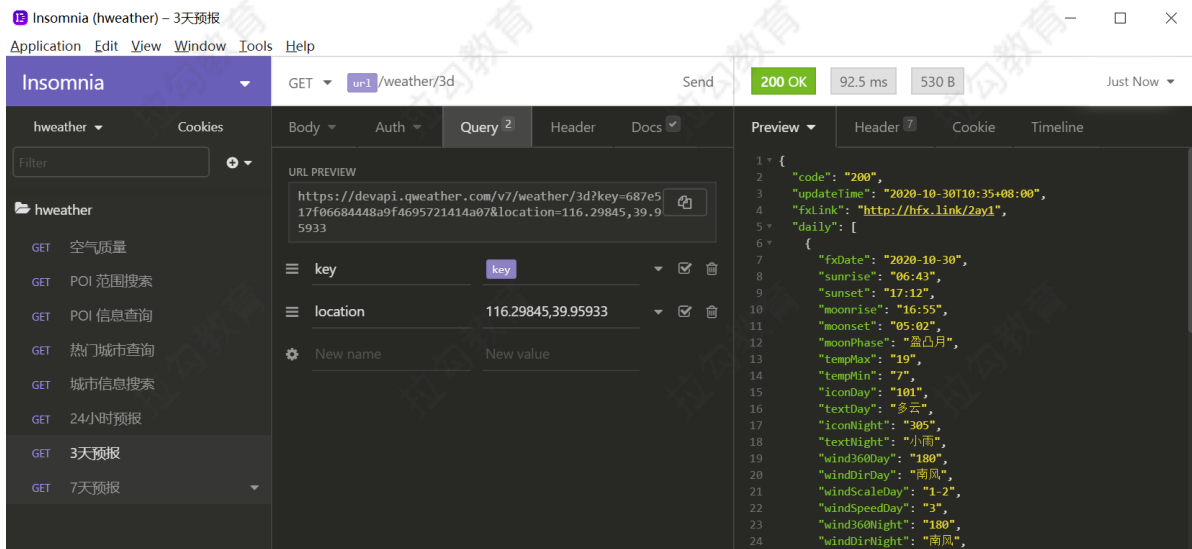
API 调试工具是用来对 API 接口进行调试的。比较出名的有 Postman，Insomnia 等。本节以 Insomnia 为例。

首先，需要跳转到 Insomnia 官网，下载（Insomnia Core）：<https://insomnia.rest/download> 然后执行安装。

然后，配置开发环境。



配置完之后，就可以根据和风天气接口的使用规则，在 Insomnia 中调试接口了。使用的效果图如下：



调用接口

1. 通过云 API，获取当前位置信息

1. 调用获取地理位置接口

```
wx.getLocation({
  type: "wgs84",
  success (res) {
    console.log(res)
  }
})
```

2. 在 app.json 里声明 permission 字段



```
"permission": {
  "scope.userLocation": {
    "desc": "您的服务需要位置信息"
  }
}
```

3. 查看获取结果

```
{
  accuracy: 65
  errMsg: "getLocation:ok"
  horizontalAccuracy: 65
  latitude: 39.95933 # 纬度
  longitude: 116.29845 # 经度
  speed: -1
  verticalAccuracy: 65
}
```

2. 通过 wx.request 调用和风天气的接口。

```
# 接口调用示例
wx.request({
  method: 'get',
  url: 'https://devapi.qweather.com/v7/indices/1d', // 接口地址
  data: {
    key: '687e517f0668444*****414a07', // 和风天气 key
    location: '116.29845,39.95933', // 位置信息（经度,纬度）
    type: '1,2' // 1:洗车指数; 2:运动指数
  },
  header: {
    'content-type': 'application/json' // 默认返回数据格式是 json
  },
  success (res) {
    console.log(res.data)
  }
})
```

返回结果中各个字段的含义，可以查看和风天气官网获取。

此处调用的是 天气生活指数接口

天气生活指数接口: <https://dev.qweather.com/docs/api/indices/>

```
# 返回数据示例
{
  "code": "200",
  "updateTime": "2020-10-31T15:36+08:00",
  "fxLink": "http://hfx.link/2ay2",
  "daily": [
    {
      "date": "2020-10-31",
      "type": "2",
      "name": "洗车指数",
      "level": "4",
      "category": "不宜",
      "text": "不宜洗车，未来24小时内有扬沙或浮尘，如果在此期间洗车，极易很快蒙上新的灰尘。"
    },
    {
      "date": "2020-10-31",
      "type": "1",
      "name": "运动指数",
      "level": "3",
      "category": "较不宜",

```

```
"text": "有扬沙或浮尘, 建议适当停止户外运动, 选择在室内进行运动, 以避免吸入更多沙尘, 有损健康。"
},
],
"refer": {
  "sources": [
    "weather China"
  ],
  "license": [
    "no commercial use"
  ]
}
}
```

处理调用报错: 发送接口请求时, 请求域名必须是https协议, 而且必须绑定到小程序管理平台。

✖ ▶ <https://devapi.qweather.com> 不在以下 request 合法域名列表中, 请参考文档: <https://developer.weixin.qq.com/miniprogram/dev/framework/ability/network.html>

上线阶段解决方案: 把报错域名 (<https://devapi.qweather.com>), 添加到微信公众平台 (小程序账号)

开发 -> 开发设置 -> 服务器域名



开发阶段解决方案: 不检查域名是否合法, 点击微信开发者工具右上角的 **详情** 菜单



第三方组件

成功获取数据之后，下一步就是在小程序中展示数据。展示数据时，可以使用官网提供的组件，也可以使用一些第三方的组件（**官网组件更基础，第三方组件更高效**）。这里我们选中 WeUI。

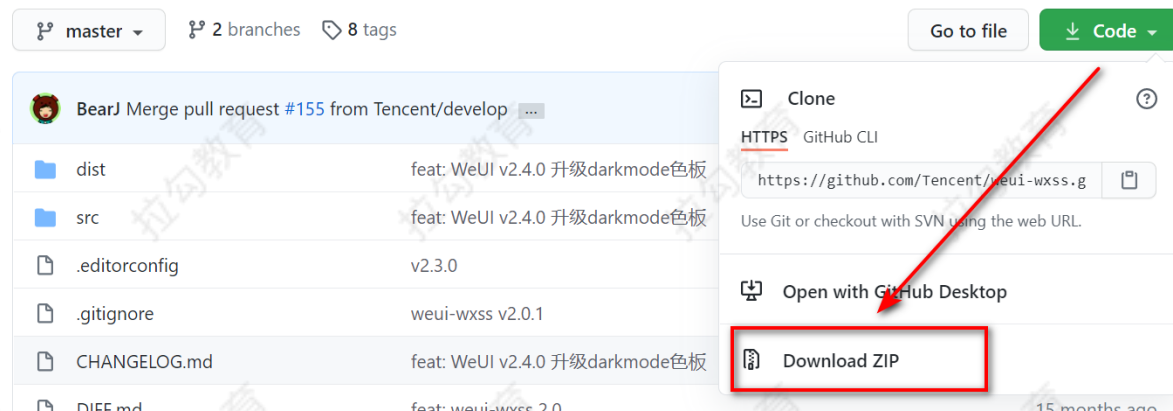
WeUI: <https://weui.io/>



WeUI 是微信团队推出的，针对小程序开发的 UI 组件库。下面我们来看一下，如何在小程序中使用 WeUI。

下载 WeUI

进入下载地址：<https://github.com/Tencent/weui-wxss>



引入 WeUI

1. 解压下载后的包，找到 dist/style/ 目录下的 weui.wxss 文件



2. 将 weui.wxss 文件，复制到小程序项目的根目录中

program (D:) > lagou > code > miniprogram > hweather

名称	类型
pages	文件夹
utils	文件夹
app.js	JavaScript 文件
app.json	JSON 文件
app.wxss	WXSS 文件
project.config.json	JSON 文件
sitemap.json	JSON 文件
weui.wxss	WXSS 文件

3. 在全局 app.wxss 中加入 @import "weui.wxss", 至此完成引入

```
app.wxss x
1  /**app.wxss**/
2  @import "weui.wxss";
3
4  .container {
5    height: 100%;
6    display: flex;
7    flex-direction: column;
8    align-items: center;
9    justify-content: space-between;
10   padding: 200px 0;
11   box-sizing: border-box;
12 }
```

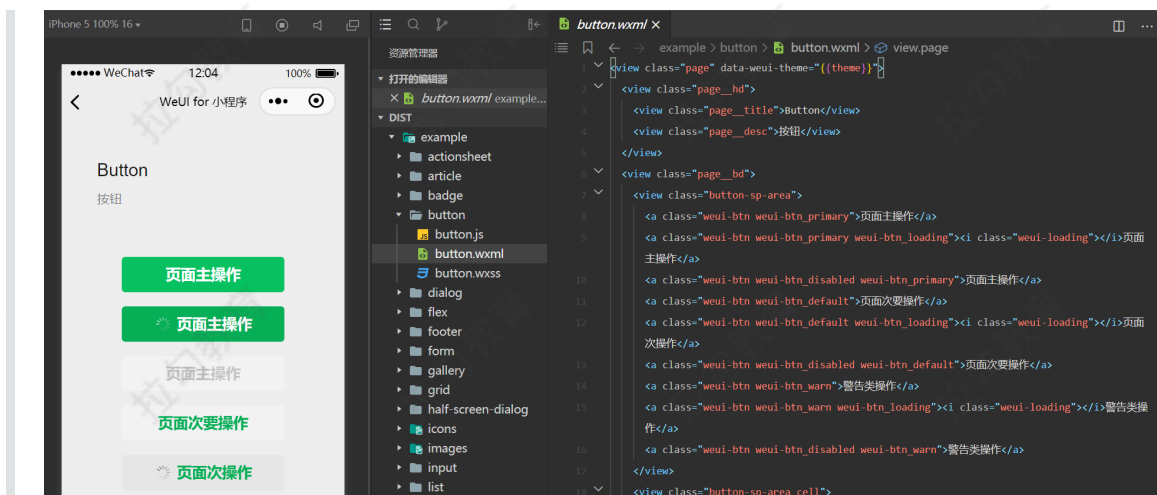
注意: 引入代码需要以分号 (;) 结尾

使用 WeUI

通过微信开发者工具打开解压后的 dist 目录, 里面 example 目录下, 给出了 WeUI 的代码示例

此电脑 > program (D:) > cliu > weui-wxss-master > dist

名称	类型
example	文件夹
libs	文件夹
mixin	文件夹
style	文件夹
app.js	JavaScript 文件
app.json	JSON 文件
app.wxss	WXSS 文件
project.config.json	JSON 文件
sitemap.json	JSON 文件

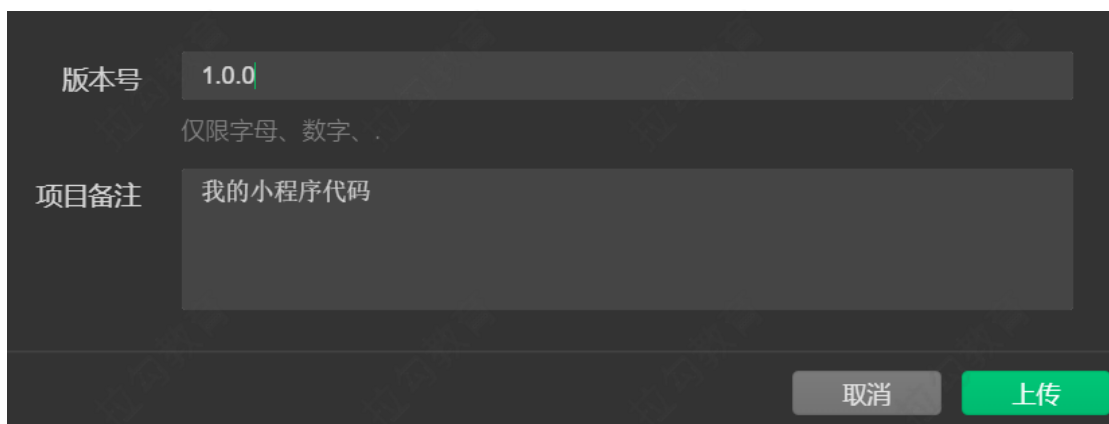
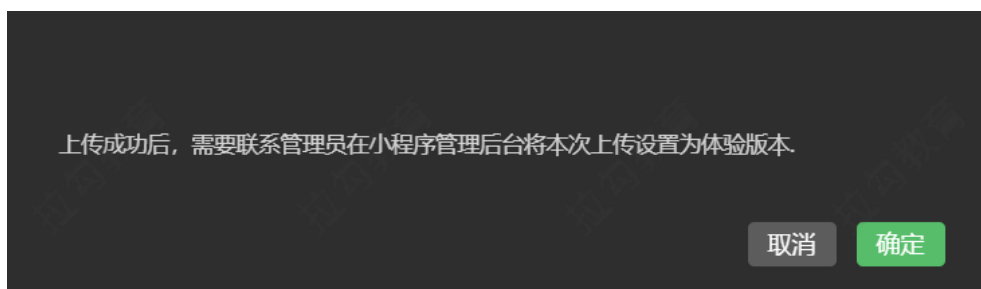


可以将需要的代码段，直接复制到你的项目中使用。

发布小程序

上传代码

在微信开发者工具的右上角，有 **上传** 菜单，之后根据提示一直向下，直到上传成功



上传成功

已生成 SourceMap 文件, [点击下载](#)

以下文件没有被打包上传:

· static/WeatherIcon-master/weather-icon-S1/color-128/LICENSE

确定

提交审核

代码上传成功之后, 跑到小程序账号后台, **管理 -> 版本管理 -> 开发版本** 下, 可以看到刚才上传的代码

开发版本

版本号	开发者	初六
1.0.0	提交时间	2020-11-11 14:11
	项目备注	我的小程序代码

提交审核



点击 **提交审核** 按钮, 依次向下, 直到提交成功



已提交审核

本次审核预计在1-7天内完成, 请耐心等待, 并留意通知中心的消息。

确定

提交成功后, 小程序就会出现在 **审核版本** 中

审核版本

版本号	开发者	初六
1.0.0	提交审核时间	2020-11-11 14:11
审核中	项目备注	我的小程序代码

详情



审核失败时, 根据官网提示, 纠正失败问题, 然后重新提交审核。

审核版本

版本号	开发者	初六
1.0.0	提交审核时间	2020-11-11 14:11
审核不通过	项目备注	我的小程序代码

删除



审核不同过的原因是小程序类目与实际内容不符。

文档

社区

工具



电话

通知

全部已读

小程序" "代码发布审核结果

你的小程序"金玉满堂莫之能守", 提审时间2020

, 版本审核未通过。

1: 你好, 你的小程序涉及提供天气查询等相关服务, 请选择: 工具-天气类目。

我之前选择的类目是 **教育 > 在线教育**

设置

基本设置

第三方设置

关联设置

关注公众号

基本设置/ 服务类目

服务类目

已添加1个, 最多5个, 本月可添加3次

添加服务类目

服务名称

所有状态

操作

教育 > 在线教育

已通过

删除

再添加一条类目 **工具 > 天气**

添加服务类目



请根据小程序自身的功能，正确选择服务类目。

服务类目

工具

天气



类目适用范围

适用于提供天气查询等服务

提交

取消

添加完成后，我们就可以看到新添加的类目了。

服务类目

已添加2个，最多5个，本月可添加2次

添加服务类目

服务名称	所有状态	操作
教育 > 在线教育	已通过	删除
工具 > 天气	已通过	删除

然后再次重新提交审核

发布版本

审核通过后，就可以发布小程序了。

版本管理



线上版本

尚未提交线上版本

审核版本

版本号	开发者	初六
1.0.1	提交审核时间	2020
审核通过待发布	项目备注	添加工具类目

提交发布



点击 **提交发布**，发布成功之后，审核版本就跑到线上版本了

版本管理



线上版本

版本号	发布者	初六
1.0.1	发布时间	2020
	项目备注	添加工具类目

详情



审核版本

你暂无提交审核的版本或者版本已发布上线