

疯狂早茶：初识微信小程序：结构概览，app.js 解析

一：初识微信小程序

获取 AppID

如果你是开发者，可以登录 <https://mp.weixin.qq.com>，就可以在网站的“设置” - “开发者设置”中，查看到微信小程序的 AppID 了，注意不可直接使用服务号 或订阅号的 AppID。

上述内容来自微信的官方教程。不过不用担心，AppID 只在我们需要将程序上传到真机时才会有影响，大部分情况下，它并不会影响我们学习开发微信小程序。

获取开发工具

微信专门为开发微信小程序发布了一款开发工具，大家可以从微信开发工具下载。微信发布了 windows64，windows32，mac3 个版本，下载对应版本后安装即可。博主安装的 window64 版本，后续的教程都将会依据此版本。

安装完成后首次打开界面如下：



微信开发工具

扫码登录

启动工具时，开发者需要使用已在后台绑定成功的微信号扫描二维码登录，后续所有的操作都会基于这

个 微信帐号

首次打开需要使用微信扫码登录，登录后便可以创建项目了。

[返回](#)

添加项目

AppID

填写小程序AppID

无 AppID

项目名称

项目目录

选择

取消

添加项目

创建项目

APPID 这里选择"无 AppID"，填写项目名称，选择项目目录，点击添加项目。

注意这里的项目名称不是小程序的名称哦

之后便是开发界面。可以看到开发工具设计的非常简洁，但是功能却很强大，包含了代码编辑，调试，发布(需要 AppID)等功能。一切顺利的话，点击调试按钮，在工具左上区域会看到一个 Hello World 欢迎界面:



iPhone 6

wifi



编辑



调试



项目



重启

WeChat



张俊轩

Hello World

第一个界面

如果不能正常显示此页面，选择**设置->不使用任何代理，勾选后直连网络**，确定后点击左侧**重启**按钮，应该就能正常显示了。

二：结构概览

在上一章教程的最后，我们生成了一个类似"Hello World"的小程序，这个过程中没有编写任何一行代码。

在新建一个项目后，微信小程序会生成一个默认的程序框架，后续程序的开发工作都在这个框架上进行。这个默认框架包含下面几部分：

app.wx

每个微信小程序都会包含 app.js, app.json, app.wxss 三个文件，其中 app.js 文件包含了程序的逻辑实现代码，app.json 是一个全局配置文件，app.wxss 是一个全局样式文件。后面会详细介绍每个文件的内容和作用。

pages 目录

pages 目录包含了程序当前的页面文件，以默认生成的程序为例，该目录下包含了 index，logs 两个目录，说明程序包括 index 和 logs 两个页面。

以 index 为例，该目录下包含 index.js，index.wxss，index.wxml3 个文件，其中 js 是代码文件，wxss 是样式文件，wxml 是页面结构描述文件。

熟悉 web 开发的同学应该会感觉很熟悉。微信小程序的开发模式确实和 web 开发很相似。目前逻辑部分仅支持 javascript 语言，并使用 wxml(类似 html)和 wxss(类似 css)来描述页面的结构和样式。此处的 javascript 和 web 中是完全一样的，但因为不是运行在浏览器环境中，因此无法使用 windows，document 等对象，自然也无法使用 jquery 等第三方库。而 wxml，wxss 的语法和 html，css 也是十分相近的。

页面也同样可以包含一个 index.json 文件用于配置，不过这不是必须的。

通常一个完整的微信小程序包含上面两部分，当然我们也可以定义自己的目录用于存放公共代码和程序需要的其它文件。

app.json

打开 app.json 文件,可以看到如下代码

```
{  
  "pages":[  
    "pages/index/index",  
    "pages/logs/logs"
```

```
],  
  
"window":{  
  
  "backgroundTextStyle":"light",  
  
  "navigationBarBackgroundColor": "#fff",  
  
  "navigationBarTitleText": "WeChat",  
  
  "navigationBarTextStyle":"black"  
  
}  
  
}
```

其中 pages 部分包含了程序的页面，这样框架就能知道从哪里找到页面文件。而 window 部分包含了程序窗口的一些配置。详细的配置项可以参考[配置 小程序](#)

app.wxss

app.wxss 文件包含了全局的样式信息，在默认生成的程序中，只有一个类选择器.container，该类型在 index.wxml 和 logs.wxml 中有使用到。

app.js

该文件包含了程序的主体流程代码实现，关于该部分的分析请见下一章教程。

三：app.js 解析

上一篇教程中写道，开发工具会生成一个默认的程序框架，其中程序的主流程代码包含在 app.js 中。默认实现中，该部

分功能比较简单，不过对于学研究小程序开发还是比较有价值的。

由于代码行数不多，下面一次性贴出来后进行讲解

```
//app.js
```

```
App({
```

```
  onLaunch: function () {
```

```
    //调用 API 从本地缓存中获取数据
```

```
    var logs = wx.getStorageSync('logs') || []
```

```
    logs.unshift(Date.now())
```

```
    wx.setStorageSync('logs', logs)
```

```
  },
```

```
  getUserInfo:function(cb){
```

```
    var that = this
```

```
    if(this.globalData.userInfo){
```

```
      typeof cb == "function" && cb(this.globalData.userInfo)
```

```
}else{  
  //调用登录接口  
  wx.login({  
    success: function () {  
      wx.getUserInfo({  
        success: function (res) {  
          that.globalData.userInfo = res.userInfo  
          typeof cb == "function" && cb(that.globalData.userInfo)  
        }  
      })  
    }  
  })  
}  
},  
globalData:{  
  userInfo:null
```

```
}  
  
})
```

代码中定义了一个 App 对象，该对象包含 onLaunch, getUserInfo 两个方法和.globalData 成员变量。其中 onLaunch 方法会在程序启动时由框架来进行调用。在该方法中，程序通过微信提供的本地存储接口获取了一个 logs 变量，第一次获取时会返回空，这时将变量初始化为空数组。之后在数组头部添加一个格式化的时间字符串，之后将数组变量存储在本地。

微信小程序框架提供了一系列 api 来帮助我们进行本地数据存储，上面的代码中使用到了

wx.getStorageSync 和 wx.setStorageSync 两个 api,更多相关 api 可以参考[这里](#)

getUserInfo 方法很容易理解，会执行获取用户信息的功能。不过该方法不同于 onLaunch,它不是框架预设的函数，因此不会自动触发调用。该方法是在 index.js 的 onload 方法中调用的(关于 index.js 部分会在后续教程中讲解)。该方法需要调用者传递一个名为 cb 的参数，首先判断是否获取过用户信息，如果获取过并且 cb 是函数的话，就会调用 cb 函数并将.globalData 成员变量传递进去，否则将调用 wx.login 接口来获取用户信息。

对 js 不熟悉或者很少接触脚本语言的同学可能会对上述代码的语法产生困惑，这几行代码涉及到了函数回调，匿名函数，闭包等概念，我后续会针对这部分单独写一篇教程。

代码最后定义了.globalData 成员变量，该成员包含 userInfo 变量，用来保存用户信息。