

初探微信小程序

最近闲下来了，就准备了解下微信小程序的内容。首先从目录结构开始吧。

摘要

最近闲下来了，就准备了解下微信小程序的内容。首先从目录结构开始吧。

创建项目

工具下载：<https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html?t=20161107>

下载安装之后，运行



微信开发者工具



请使用微信扫描二维码登录
“微信web开发者工具”

扫描登录

调试类型

本地小程序项目
用于调试本地小程序项目

公众号网页开发
用于公众号网页开发

登录后你会看到两种项目类型，这里我们添加一个本地小程序的项目

返回

添加项目

AppID

填写小程序AppID

无 AppID

项目名称

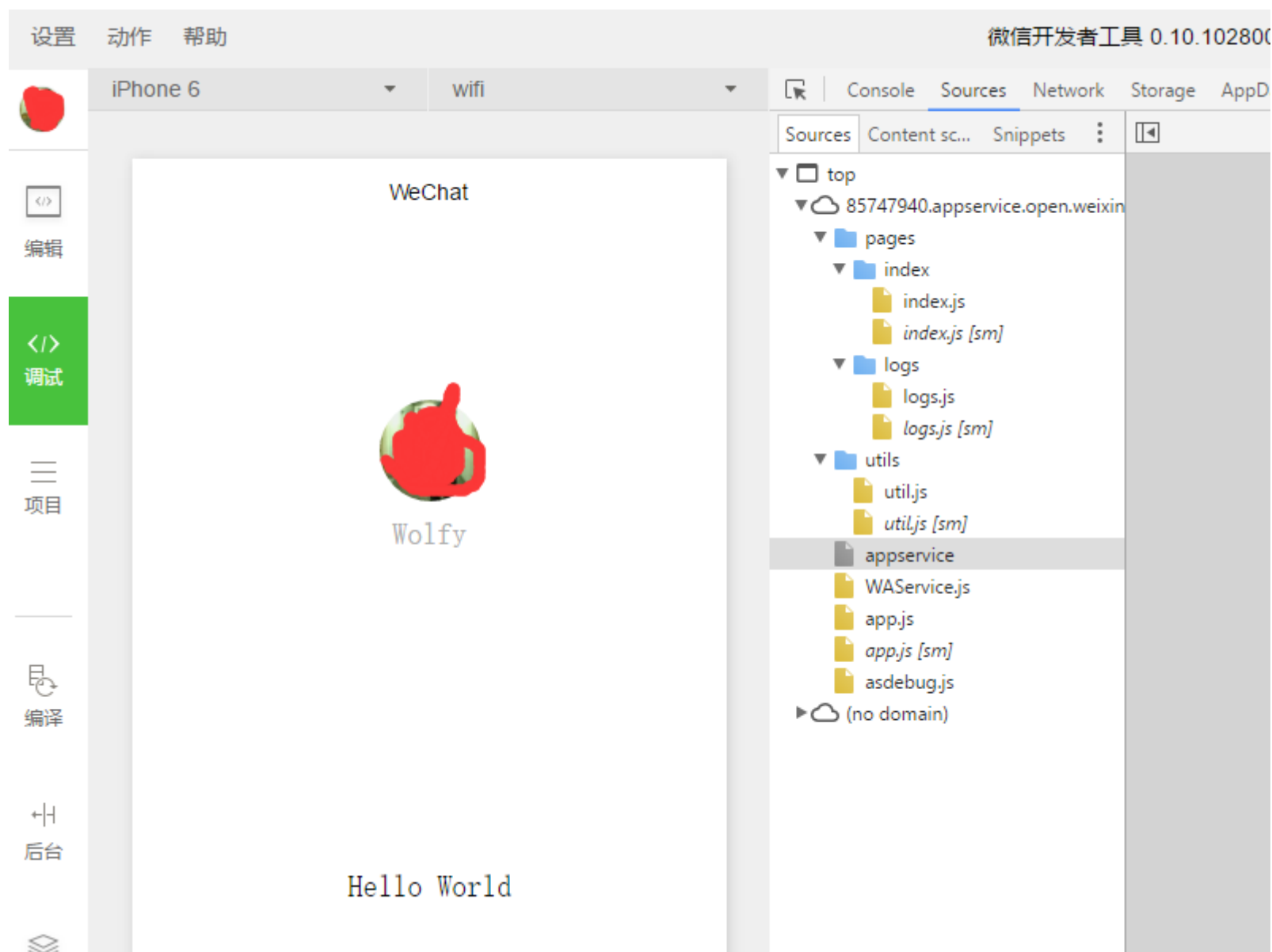
项目目录

选择

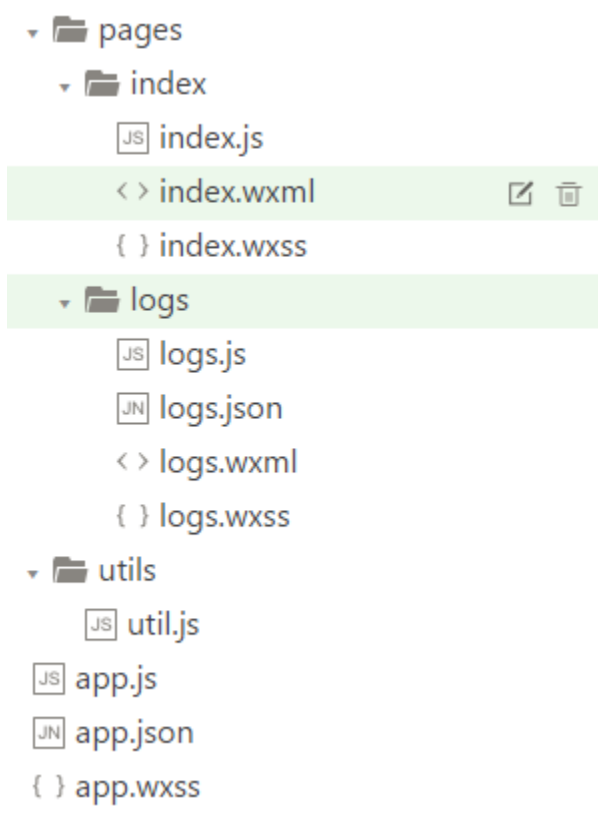
取消

添加项目

目前没有 AppID，可以选择无 AppID。填写好相关信息后，单击“添加项目”开发工具会自动生成一个 demo 项目



选择开发者工具左侧栏“编辑”，可以看到这个 demo 项目的目录，以及文件树结构



其中，app.js、app.json、app.wxss 这三个文件是必不可少的文件。.js 后缀的是脚本文件，.json 后缀的文件是配置文件，.wxss 后缀的是样式表文件。微信小程序会读取这些文件，并生成小程序实例。

app.js 是小程序的脚本代码。我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。



```
1 //app.js
2 App({
3   onLaunch: function () {
```

```
4    //调用 API 从本地缓存中获取数据
5    var logs = wx.getStorageSync('logs') || []
6    logs.unshift(Date.now())
7    wx.setStorageSync('logs', logs)
8  },
9  getUserInfo:function(cb){
10    var that = this
11    if(this.globalData.userInfo){
12      typeof cb == "function" && cb(this.globalData.userInfo)
13    }else{
14      //调用登录接口
15      wx.login({
16        success: function () {
17          wx.getUserInfo({
18            success: function (res) {
19              that.globalData.userInfo = res.userInfo
```

```
20         typeof cb == "function" && cb(that.globalData.userInfo)
21     }
22 })
23 }
24 })
25 }
26 },
27 globalData:{
28     userInfo:null
29 }
30 })
```



app.json 是对整个小程序的全局配置。我们可以在这个文件中配置小程序是由哪些页面组成，配置小程序的窗口背景色，配置导航条样式，配置默认标题。

注意

该文件不可添加任何注释。




```
1 {
2   "pages":[
3     "pages/index/index",
4     "pages/logs/logs"
5   ],
6   "window":{
7     "backgroundTextStyle":"light",
8     "navigationBarBackgroundColor": "#fff",
9     "navigationBarTitleText": "WeChat",
10    "navigationBarTextStyle":"black"
11  }
12 }
```



app.wxss 是整个小程序的公共样式表。我们可以在页面组件的 class 属性上直接使用 app.wxss 中声明的样式规则。



```
1 /**app.wxss**/
```

```
2 .container {  
3   height: 100%;  
4   display: flex;  
5   flex-direction: column;  
6   align-items: center;  
7   justify-content: space-between;  
8   padding: 200rpx 0;  
9   box-sizing: border-box;  
10 }
```



pages

微信小程序中的每一个页面的【路径+页面名】都需要写在 app.json 的 pages 中，且 pages 中的第一个页面是小程序的首页。

在给出的 demo 中，有两个页面 index 和 logs 页面。

注意

每一个小程序页面是由同路径下同名的四个不同后缀文件的组成，如：index.js、index.wxml、index.wxss、index.json。 .js 后缀的文件是脚本文件，.json 后缀的文件是配置文件，.wxss 后缀的是样式表文件，.wxml 后缀的文件是页面结构文件。

index.wxml



```
<!--index.wxml-->
```

```
<view class="container">
```

```
  <view bindtap="bindViewTap" class="userinfo">
```

```
    <image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"> </image>
```

```
    <text class="userinfo-nickname">{{userInfo.nickName}}</text>
```

```
  </view>
```

```
  <view class="usermotto">
```

```
    <text class="user-motto">{{motto}}</text>
```

```
  </view>
```

```
</view>
```



你会发现，里面出现了一些新的标签，例如：<view/>，关于这些在以后的学习中，一一介绍。今天只介绍项目结构。

index.js 是页面的脚本文件，在这个文件中我们可以监听并处理页面的生命周期函数、获取小程序实例，声明并处理数据，响应页面交互事件等。



```
//index.js
```

```
//获取应用实例
```

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

```
Page({
```

```
  data: {
```

```
    motto: 'Hello World',
```

```
    userInfo: {}
```

```
  },
```

```
  //事件处理函数
```

```
  bindViewTap: function() {
```

```
    wx.navigateTo({
```

```
      url: '../logs/logs'
```

```
    })  
  },  
  onLoad: function () {  
    console.log('onLoad')  
    var that = this  
    //调用应用实例的方法获取全局数据  
    app.getUserInfo(function(userInfo){  
      //更新数据  
      that.setData({  
        userInfo:userInfo  
      })  
    })  
  }  
})
```



index.wxss 是页面的样式表：



```
/**index.wxss**/
```

```
.userinfo {
```

```
  display: flex;
```

```
  flex-direction: column;
```

```
  align-items: center;
```

```
}
```

```
.userinfo-avatar {
```

```
  width: 128rpx;
```

```
  height: 128rpx;
```

```
  margin: 20rpx;
```

```
  border-radius: 50%;
```

```
}
```

```
.userinfo-nickname {
```

```
  color: #aaa;
```

```
}  
  
.usermotto {  
    margin-top: 200px;  
}
```



注意

页面的样式表是非必要的。当有页面样式表时，页面的样式表中的样式规则会层叠覆盖 app.wxss 中的样式规则。如果不指定页面的样式表，也可以在页面的结构文件中直接使用 app.wxss 中指定的样式规则。

index.json 是页面的配置文件：

页面的配置文件是非必要的。当有页面的配置文件时，配置项在该页面会覆盖 app.json 的 window 中相同的配置项。如果没有指定的页面配置文件，则在该页面直接使用 app.json 中的默认配置。

pages/logs 目录

该目录下是小程序的日志页面。

logs 的页面结构

```
<!--logs.wxml-->  
  
<view class="container log-list">
```

```
<block wx:for="{{logs}}" wx:for-item="log" wx:key="*this">
  <text class="log-item">{{index + 1}}. {{log}}</text>
</block>
```

```
</view>
```



```
//logs.js
```

```
var util = require('../utils/util.js')
```

```
Page({
  data: {
    logs: []
  },
  onLoad: function () {
    this.setData({
      logs: (wx.getStorageSync('logs') || []).map(function (log) {
        return util.formatTime(new Date(log))
      })
    })
  })
})
```



```
}  
  
})
```



从 logs.js 中可以看出，小程序的日志是记录在 localStorage 中的。

运行如下：

[返回](#)

[查看启动日志](#)

1. 2016/11/21 15:28:02

总结

如果使用过 angularjs，那么对小程序的项目结构，以及每个文件的作用，应该比较了解。其中.wxml 文件为视图文件，每个页面对应的.js 文件为控制器文件，在控制器与视图进行传递的就是 mvc 中的 Model 了。

参考文章

<https://mp.weixin.qq.com/debug/wxadoc/dev/?t=20161109>