

石头记的新手进阶教程：开发环境搭建及目录结构分析

一：开发之环境搭建

摘要: 微信小程序开发环境主要利用官方提供的开发包

1、下载官方提供的开发者工具，它集成了开发调试、代码编辑及程序发布等功能。当前为（0.10.101100）

下载地址：<https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html?t=1476434678461>

2、安装并且运行开发工具，第一次启动如下图：



3、点击添加项目：

< 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

4、由于暂时没有内测帐号，所有没有 appId（与订阅号和服务号 appId 不一样），输入项目名称 helloWorld, 选择目录为上面下载的官方 demo

[返回](#)

添加项目

AppID

无 AppID 部分功能受限

返回填写小程序AppID

项目名称

helloWorld

项目目录

D:\wx\helloworld

选择

取消

添加项目

5、单击添加项目，项目创建成功，看到如下页面表示项目配置成功。



环境配置还是很简单的，按照步骤来做应该不会出现问題。

二：目录结构分析

学习一个框架，首先应该了解一下他的目录结构，小程序的目录接口如下图：



helloWorld



编辑



调试



项目



编译



关闭

pages

index

index.js

index.wxml

index.wxss

logs

logs.js

logs.json

logs.wxml

logs.wxss

utils

util.js

app.js

app.json

app.wxss

从 demo 中可以看到主要分为两部分

1、系统文件或者公用文件

文件	必填	作用
app.js	是	小程序逻辑，app 启动首先会执行该文件内的部分方法
app.json	是	小程序公共设置，整个 app 的系统配置
app.wxss	否	小程序公共样式表，可以理解为前端中的 base.css

2、页面文件（page 文件夹）

文件类型	必填	作用
js	是	页面逻辑，可以理解为 php 或 java 的 controller，主要来处理业务逻辑
wxml	是	页面结构，可以理解为视图层 html
wxss	否	页面样式表，可以理解为 css 文件
json	否	页面配置，当前页面配置文件

下面分别来学习每个文件的具体用法及作用

app.js

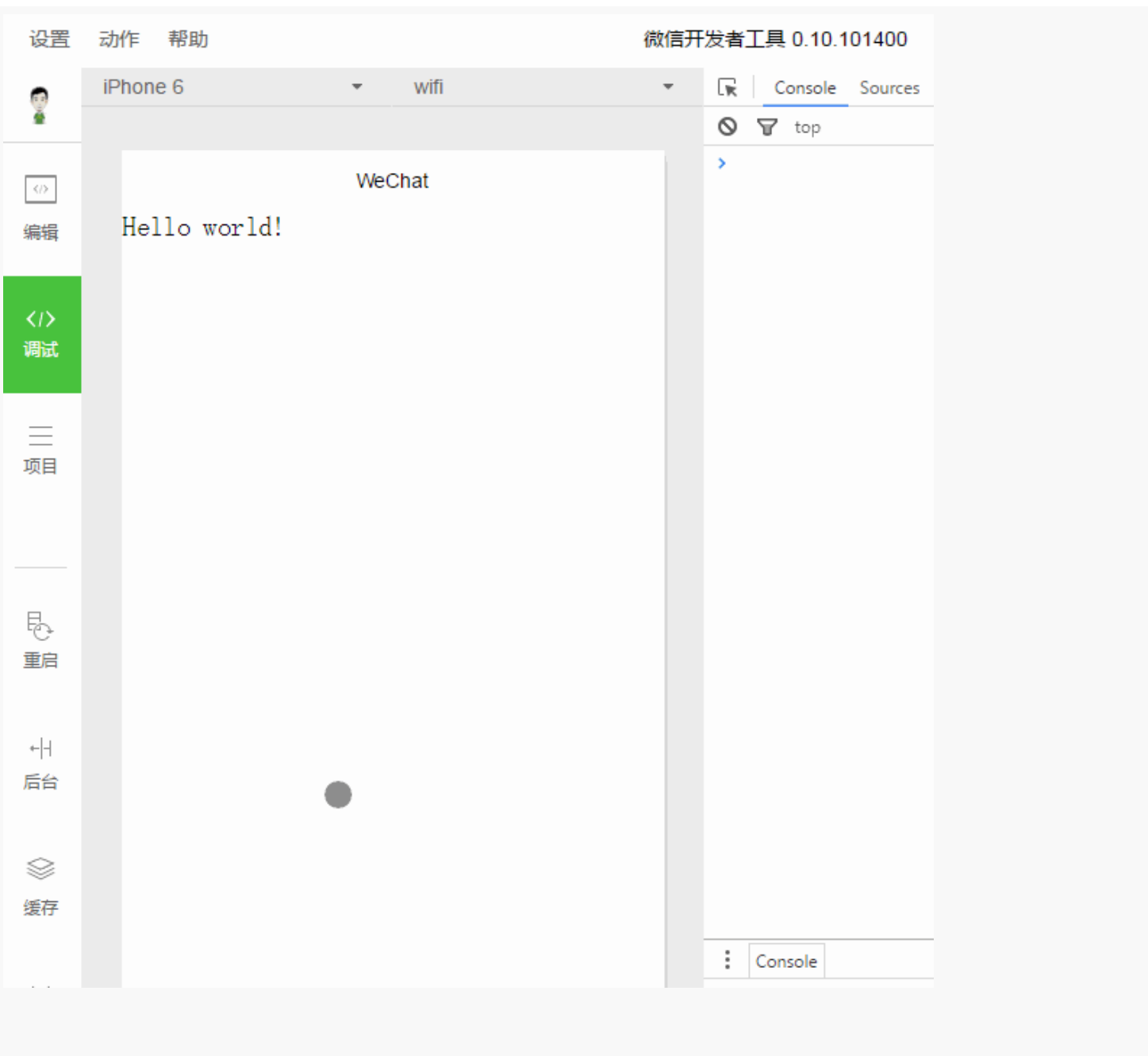
小程序启动先回执行此文件中的方法，此文件为空，程序也可以正常执行，但通常不会这么做，一般用于执行 App 方法返回 app 对象。

```
App({  
  onLaunch: function () {  
    console.log('onLaunch')  
  },  
  onShow: function () {  
    console.log('onShow')  
  },  
  onHide: function () {  
    console.log('onHide')  
  },  
  other:function(){  
  
  }  
})
```


代码中的几个方法，非常类似原生 app 开发中的生存周期函数，详细解释如下：

属性	类型	描述	触发时机
onLaunch	Function	生命周期函数--监听小程序初始化	当小程序初始化完成时，会触发 onLaunch (全局只触发一次)
onShow	Function	生命周期函数--监听小程序显示	当小程序启动，或从后台进入前台显示，会触发 onShow
onHide	Function	生命周期函数--监听小程序隐藏	当小程序从前台进入后台，会触发 onHide
其他	Any	开发者可以添加任意的函数或数据到 Object 参数中，用 this 可以访问	

当程序启动时，首先会执行 onLauch 函数，启动成功后会执行 onShow,当程序退出活动窗口，会执行 onHide 函数，程序再次切换到活动窗口会执行 onShow。



注意：

App() 必须在 app.js 中注册，且不能注册多个。

不要在定义于 App() 内的函数中调用 getApp()，使用 this 就可以拿到 app 实例。

不要在 onLaunch 的时候调用 getCurrentPage()，此时 page 还没有生成。

通过 getApp() 获取实例之后，不要私自调用生命周期函数。

app.json

我们使用 app.json 文件来对微信小程序进行全局配置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等，简单的理解就是配置文件了。

配置项表：

属性	类型	必填	描述
pages	Array	是	设置页面路径
window	Object	否	设置默认页面的窗口表现
tabBar	Object	否	设置底部 tab 的表现
networkTimeout	Object	否	设置网络超时时间
debug	Boolean	否	设置是否开启 debug 模式

pages 属性

pages 属性的第一个值为系统启动页面，如下图配置项，那么小程序启动后，会先运行 logs 页面

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

window 属性

windows 属性用于设置小程序的状态栏、导航条、标题、窗口背景色。

属性	类型	默认值	描述
navigationBarBackgroundColor	HexColor	#000000	导航栏背景颜色，如"#000000"
navigationBarTextStyle	String	white	导航栏标题颜色，仅支持 black/white
navigationBarTitleText	String		导航栏标题文字内容

backgroundColor	HexColor	#ffffff	窗口的背景色
backgroundTextStyle	String	dark	下拉背景字体、loading 图的样式，仅支持 dark/light
enablePullDownRefresh	Boolean	false	是否开启下拉刷新，详见页面相关事件处理函数。

tabBar

如果我们的程序是一个多 tab 应用（客户端窗口的底部有 tab 栏可以切换页面），那么我们可以通过 tabBar 配置项指定 tab 栏的表现，以及 tab 切换时显示的对应页面。

tabBar 是一个数组，只能配置最少 2 个、最多 5 个 tab，tab 按数组的顺序排序。

属性	类型	必填	默认值	描述
color	HexColor	是		tab 上的文字默认颜色
selectedColor	HexColor	是		tab 上的文字选中时的颜色
backgroundColor	HexColor	是		tab 的背景色
borderStyle	String	否	black	tabbar 上边框的颜色， 仅支持 black/white
list	Array	是		tab 的列表，详见 list 属性说明，最少 2 个、最多 5 个 tab

其中 list 接受一个数组，数组中的每个项都是一个对象，其属性值如下：

属性	类型	必填	说明
pagePath	String	是	页面路径，必须在 pages 中先定义
text	String	是	tab 上按钮文字
iconPath	String	是	图片路径，icon 大小限制为 40kb
selectedIconPath	String	是	选中时的图片路径，icon 大小限制为 40kb

networkTimeout

可以设置各种网络请求的超时时间。

属性	类型	必填	说明
request	Number	否	wx.request 的超时时间，单位毫秒
connectSocket	Number	否	wx.connectSocket 的超时时间，单位毫秒
uploadFile	Number	否	wx.uploadFile 的超时时间，单位毫秒
downloadFile	Number	否	wx.downloadFile 的超时时间，单位毫秒

debug

可以在开发者工具中开启 debug 模式，在开发者工具的控制台面板，调试信息以 info 的形式给出，其信息有 Page 的注册，

页面路由，数据更新，事件触发。可以帮助开发者快速定位一些常见的问题。

欢迎加我微信，交流探讨！

