

微信小程序学习记录《一》：目录结构介绍，开发工具菜单介绍

第一部分：目录结构介绍

1.微信小程序目录结构



微信小程序默认的目录结构，下面——介绍：

1.1 小程序的文件格式介绍

在项目中我们可以看到四种文件类型：

.js 后缀的文件是脚本文件，页面的交互等代码在这里实现；

.json 后缀的文件是配置文件，主要是 json 数据格式存放，用于设置程序的配置效果；

.wxss 后缀的是样式表文件，类似于前端中的 css,用于对界面进行美化；

.wxml 后缀的文件是页面结构文件，用于构建页面，在页面上增加控件。

1.1 pages 目录介绍

1pages:主要存放小程序的页面文件，其中每个文件夹为一个页面，每个页面包含四个文件，.wxml 文件是界面文件，.js 是事件交互文件，用于处理界面的点击事件等功能；.wxss 为界面美化文件，让界面显示的更加美观；.json 为配置文件，用于修改导航栏显示样式等，小程序每个页面必须有.wxml 和.js 文件，其他两种类型的文件可以不需要。

注意：文件名称必须与页面的文件夹名称相同，如 index 文件夹，文件只能是 index.wxml、index.wxss、index.js 和 index.json。

1.2 utils

该文件件主要用于存放全局的一些.js 文件，公共用到的一些事件处理代码文件可以放到该文件夹下，用于全局调用。

[javascript] [view plain copy](#)

```
1. module.exports = {  
2.   formatTime: formatTime
```

```
3. }
```

对于允许外部调用的方法，用 `module.exports` 进行声明，才能在其他 js 文件中用一下代码引入

[html] [view plain copy](#)

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

然后就可以调用该方法。

1.3 app.js、app.json、app.wxss

app.js：系统的方法处理文件，主要处理程序的声明周期的一些方法；例如：程序刚开始运行时事件处理等

app.json：系统全局配置文件，设置导航头的颜色，字体大小，下面有没有 tabbar 等功能，具体页面的配置在页面的 json 文件中单独修改；

app.wxss：全局的界面美化代码

第二部分：开发工具菜单介绍

1.开发工具界面

iPhone 6 wifi

WeChat

swlh

运行界面显示

Hello World

test

- 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

app.js × app.json × app.w

```
1 //app.js
2 App({
3   onLaunch: function () {
4     //调用API从本地缓存中获取
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"
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"
21            }
22          })
23        }
24      })
25    }
26  },
27  globalData:{
28    userInfo:null
29  }
30 })
```

2.菜单区介绍

2.1 编辑

默认为编辑状态显示，下面的编译为手动编译功能，通常情况下修改文件后保存会刷新界面显示，如果修改的是js 文件，保存时会自动进行编译和刷新界面显示

2.2 调试

iPhone 6

wifi

编辑


调试

项目

编译

后台

WeChat



swh

Hello World

微信web开发者工具 v0.10.102800

Console Sources Network Storage AppData Wxml

Sources Content sc... Snippets app.js x

top

70979751.appservice.open.weixin

pages

index

logs

utils

appservice

WAService.js

app.js

app.js [sm]

asdebug.js

(no domain)

Source Map detected.

```
1 define("app.js", function(require, module,
2
3 //app.js
4 App({
5   onLaunch: function onLaunch() {
6     //调用API从本地缓存中获取数据
7     var logs = wx.getStorageSync('logs') ||
8     logs.unshift(Date.now());
9     wx.setStorageSync('logs', logs);
10   },
11   getUserInfo: function getUserInfo(cb) {
12     var that = this;
13     if (this.globalData.userInfo) {
14       typeof cb == "function" && cb(that.g
15     } else {
16       //调用登录接口
17       wx.login({
18         success: function success() {
19           wx.getUserInfo({
20             success: function success(res)
21               that.globalData.userInfo = r
22               typeof cb == "function" && cl
23             }
24           });
25         }
26       });
27     }
28   },
29   globalData: {
30
31 }
32 }
```

Line 1, Column 1

Console

top

Preserve log

onLoad

Mon Nov 21 2016 12:52:51 GMT+0800 (CST) 无 AppID 关联

⚠ 请注意无 AppID 关联下, 调用 wx.login 是受限的, API 的返回是工具的模拟返回

Mon Nov 21 2016 12:52:51 GMT+0800 (CST) 无 AppID 关联

⚠ 请注意无 AppID 关联下, 调用 wx.operateWXData 是受限的, API 的返回是工具的模拟返回

通过界面我们可以看出编辑状态的菜单栏与 chrome 浏览器的调试界面几乎相同，包括控制台、源码、网络等，并且支持断电调试，功能几乎与**前端开发**调试类似，不再介绍，

调试的时候，菜单栏下面会多出一个后台和缓存按钮：

后台：是模拟程序运行到后台（按 Home 键回到主页面，不是退出程序）时的执行情况，进入后台时再次点击会进入前台执行。

缓存：数据的本地缓存，可以在这里进行清除缓存等操作。

2.3 项目菜单



编辑



调试



项目



test

App ID: touristappid

本地开发目录 /Users/gpios001/Desktop/test

最近上传时间

上传代码，只有申请了APPID
的用户支持该功能，且目前只能上传不能发布

<http://blog.csdn.net/>

上传

最新更新时间

项目未关联AppID

预览

☒ 开启 ES6 转 ES5

☒ 监听文件变化，自动刷新开发者工具

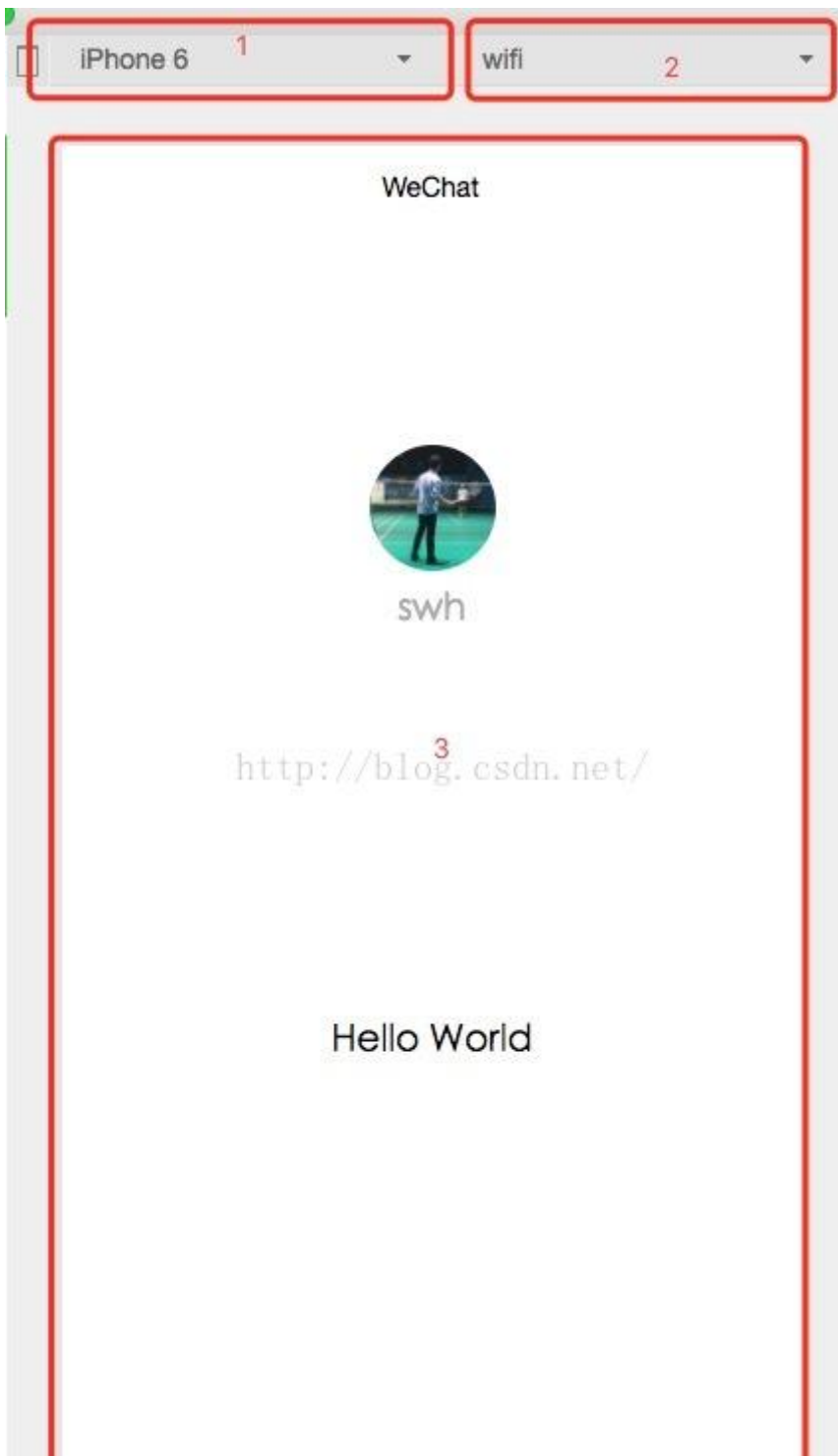
☐ 开启 代码压缩上传

开启ES6转ES5之后支持低版本浏览器编码，开启监听后，修改代码
cmd+s(Mac用户)，保存代码会自动编译运行，wind为Ctrl+ s

预览按钮，也是只有通过认证的开发者，和添加到开发者主账号中的开发者有预览功能，点击预览会生成一个用开发者账号登录的微信扫描二维码微信中查看预览效果。

删除项目

3.界面显示栏



界面显示栏包括上下两部分，上面设置运行环境，下面显示运行效果，各部分介绍：

1 选择运行的环境，上面的手机型号不具体指该手机，单指相同的分辨率，例如 iphone6 指 375 * 667 分辨率的所有机型

2 模拟网络运行环境，包括 wifi、4G、3G、2G，默认为 wifi 运行环境

3 界面效果展示区，主要显示程序的运行效果

注意：切换运行的手机分辨率时记得要手动编译以下，否则界面显示不正确

4.目录结构栏和编辑栏

目录结构已经介绍过了，不在多做介绍；

编辑区显示当前正在编辑的文件，如果是正在编辑的文件，名字显示为淡绿色，并且未保存的文件，文件名后面会有一个绿色的点，注意如果修改过文件后没有保存，程序不能正确运行



test



app.js

app.json

app.wxss



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

淡绿色背景，表示该文件
正在被编辑，后面有一个
绿色的点，表示修改后未
保存

```
//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
  }
})
```