# 有渔微信小程序系统概述《二》了解.js 文件及.json 文件

## 了解.js 文件

### 1、app.js

app.js 是整个小程序的入口文件，也是控制整个小程序生命周期的文件。App.js 用 App()来实现对整个程序的注册，同时 App()里面还实现了对小程序生命周期的监控函数(onLaunch, onShow, onHide)。

APP()内部参数说明：

| 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 | |

我们来看一个 app.js 的代码：

```
//app.js
App({
  onLaunch: function () {
    ... ...
  },
  getUserInfo:function(cb){
    ... ...
  },
```

```
  globalData:{
    userInfo:null
  },
  onShow: function(){
    ... ...
  },
  onHide: function(){
    ... ...
  }
})
```

小程序前台、后台定义： 当用户点击左上角关闭，或者按了设备 Home 键离开微信，小程序并没有直接销毁，而是进入了后台；当再次进入微信或再次打开小程序，又会从后台进入前台。只有当小程序进入后台一定时间，或者系统资源占用过高，才会被真正的销毁。简单地说，前台就是你在操作小程序时；后台就是你离开了小程序，去玩手机的其他东西去了，这时小程序就隐藏到后面去了。

**2、index.js**

小程序中每一个页面可以放在一个文件夹中，这个文件夹中一般包括 4 个文件：

.js, .json, .wxml, .wxss。官方建议这四个文件的名字最好和文件夹的名字一致，这样便于框架自动查找。

每个页面都需要注册，index.js 用 Page()这个函数来注册一个页面，它接受一个 object 参数，用这个参数来指定页面的初始数据，生命周期函数，事件处理函数。

Page()内部参数说明：

| 属性 | 类型 | 描述 |
| --- | --- | --- |
| data | Object | 页面的初始数据 |
| onLoad | Function | 生命周期函数--监听页面加载 |
| onReady | Function | 生命周期函数--监听页面初次渲染完成 |
| onShow | Function | 生命周期函数--监听页面显示 |
| onHide | Function | 生命周期函数--监听页面隐藏 |
| onUnload | Function | 生命周期函数--监听页面卸载 |
| onPullDownRefreash | Function | 页面相关事件处理函数--监听用户下拉 |

| | | |
|---|---|---|
| | | 动作 |
| 其他 | Any | 开发者可以添加任意的函数或数据<br>到 object 参数中，用 this 可以访问 |

我们来看一个 index.js 的代码：

```
//index.js
//获取应用实例
var app = getApp()
Page({
  data: {
    motto: '点击上面 View 跳转'
  },
  //事件处理函数
  bindViewTap: function() {
    //通过调用 API 进行跳转
    wx.navigateTo({
```

```
    url: '../logs/logs'
  })
},
onLoad: function () {
  // this 指的就是本页面对象
    // var that = this
    ... ...
},
onShow: function() {
  ... ...
},
onReady: function() {
  ... ...
},
onHide: function() {
  ... ...
```

```
  },
  onUnload: function() {
    ... ...
  }
})
```

## 了解.json 文件

### 1、app.json 的配置项

下面是微信官方给出的一个 json 配置文件。

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/index"
  ],
  "window": {
    "navigationBarTitleText": "Demo"
```

```
  },
  "tabBar": {
    "list": [{
      "pagePath": "pages/index/index",
      "text": "首页"
    }, {
      "pagePath": "pages/logs/logs",
      "text": "日志"
    }]
  },
  "networkTimeout": {
    "request": 10000,
    "downloadFile": 10000
  },
  "debug": true
}
```

app.json 配置项列表：

| 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  开启 debug 模式 |

(1) pages

它的作用是配置小程序的页面,这个配置项是必填的,它接受一个数组,里面的每一项都是字符串,从上面给出的代码：

```
"pages": [

    "pages/index/index",
```

```
    "pages/logs/logs"
  ]
```

pages 里每一项分别对应的都是文件的路径以及文件名。

(2) window

window 配置项是用来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 |

注：HexColor（十六进制颜色值），如"#ff00ff"

(3) tabBar

这个配置项是用来配置页面底部的 tab 栏。

| 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 |

(4) 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 |

(5) 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。

**2、例子**

看千遍不如做一遍。下面让我们做几个例子，加深对 app.json 配置项的理解。

**Example1：window 配置项**

我们把 window 配置项修改成下面的代码，然后看界面的显示。

```
  "window":{
    "navigationBarBackgroundColor": "#ffffff",
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "微信接口功能演示",
    "backgroundColor": "#eeeeee",
    "backgroundTextStyle": "light"
  }
```

点击微信开发者工具左边的“编译”后，弹出如下界面：

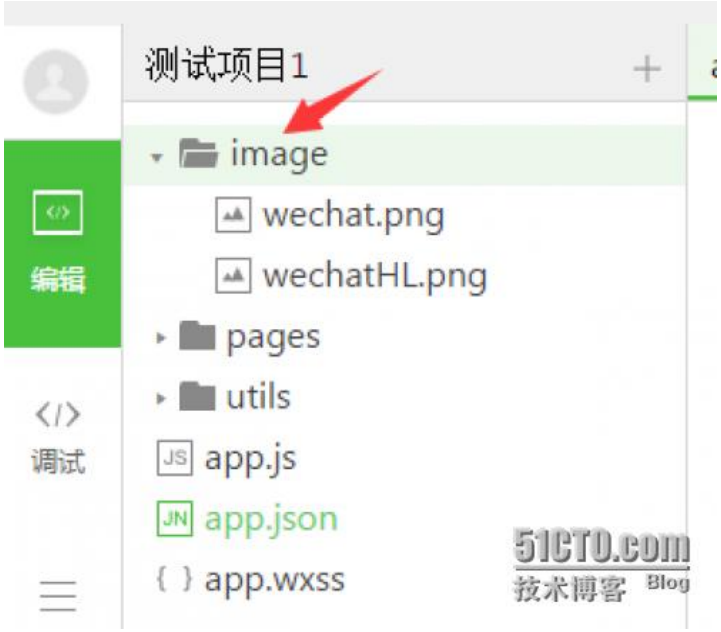
**Example2：tabbar 配置项**

(1) app.json 中添加如下代码：

```
"tabBar":{
    "color":"#dddddd",
    "selectedColor":"#3cc51f",
    "borderStyle":"white",
    "backgroundColor":"#ffffff",
    "list":[{
      "pagePath":"pages/index/index",
      "iconPath":"image/wechat.png",
      "selectedIconPath":"image/wechatHL.png",
      "text":"首页"
    },{
      "pagePath":"pages/logs/logs",
      "iconPath":"image/wechat.png",
      "selectedIconPath":"image/wechatHL.png",
```

```
    "text":"日志"
  }]
}
```

(2) 创建 image 目录，并把图片放到这里目录里




其中的 2 个图片文件如下：

wechat.png

wechatHL.png

(3) 点击微信开发者工具左边的“编译”后，弹出如下界面：


iPhone 4
wifi
编辑
调试
项目
重启
后台
微信接口功能演示
有渔@蔡
borderStyle
backgroundColor
selectedIconPath
Hello World
iconPath
selectedColor
首页
51CTO.com
技术博客
Blog