

## 豆豆尖的入门之旅：可爱的小程序《一》

本人真的是前端的入门级菜鸟，但公司的技术大大们对我都好好，给我很多锻炼机会，因此我决定好好学习总结，随时记录自己的心得，争取在几年内达到一个让自己满意的水平。小程序这个东西确实比较新，但是写起来蛮有意思，比起 Vue 这种框架好理解太多！福音哪~~~

感觉实在没法把这东西写个教程，暂且算成自己的开发心得？这里主要讲一下刚开始接触小程序需要知道的基本知识，话不多说，开搞哈哈。。

我们知道开发小程序前首先要获取微信小程序的 APPID，不然，真的就只能看文档玩儿啦~然后还要安装个微信开发者工具，就可以开始创建项目了，如下：

< 返回

## 新建项目

AppID

填写小程序AppID，可在公众平台开发设置页中查看

项目名称

本地开发目录

选择

取消

添加项目

这里提一句，如果是多人开发的话，开发目录最好是统一将代码文件放在桌面。添加完后，左侧导航栏可以选择项目进行预览，调试，编辑代码（也可以在自己本地编译器编辑），纯属吐槽一下自带的编译器是真不好用

→\_→



左侧导航条

代码部分：

微信小程序已经存在初始化的一些文件：**app.js**、**app.json**、**app.wxss**

app.js 是小程序的脚本代码，这个文件中可以监听并处理小程序的生命周期函数、声明全局变量，调用丰富的

API：

```

var api = require('config/api.js');

//app.js
App({
  onLaunch() {},

  getLoginInfo(cb) {
    var that = this;

    //调用登录接口
    wx.login({
      success(loginRes) {
        wx.getUserInfo({
          success: function(userInfoRes) {
            that.globalData.userInfo = userInfoRes.userInfo;

            //TODO:转义+和&防止转义为空
            var encodeEncrypt = userInfoRes.encryptData && userInfoRes.encryptData.replace(/\+/g, '%2B').replace(/\&/g, '%26');

            var data = {
              thirdCode: loginRes.code,
              thirdType: 4,
              scope: 'base',
              wxEncryptData: encodeEncrypt
            };

            wx.request({
              url: api.getToken.url,
              method: 'POST',
              data: data,
              header: {
                'content-type': 'application/x-www-form-urlencoded'
              },
              success: function(res) {
                if (res.data.code == 0) {
                  that.globalData.loginInfo = {
                    openId: res.data.data.openId,
                    userId: res.data.data.userId,
                    thirdToken: res.data.data.thirdToken,
                    token: res.data.data.token
                  };
                }
              }
            });
          }
        });
      }
    });
  }
});

```

app.js

可以看到，wx.login() wx.getUserInfo() 等都是开放接口，获取用户登录的相关信息。

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。但需注意该文件不可添加任何注释：

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/logs"
  ],
  "window": {
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#fff",
    "navigationBarTitleText": "WeChat",
    "navigationBarTextStyle": "black"
  }
}
```

app.json

app.wxss 是整个小程序的公共样式表:

```
/**app.wxss**/  
page{  
  height: 100%;  
  box-sizing: border-box;  
}  
.border-top-hack ,.border-bottom-hack,.border-right-hack{  
  position: relative;  
}  
.border-top-hack:after,.border-bottom-hack:after,.border-right-hack:after{  
  position: absolute;  
  content: '';  
  width: 100%;  
  left: 0;  
  height: 1px;  
  background-color: #dadada;  
  -webkit-transform: scale(1,.5);  
  transform: scale(1,.5);  
  -webkit-transform-origin: center top;  
  transform-origin: center top;  
}
```

app.wxss

**创建：**

接下来说说具体的创建页面：

我们会在 pages 下新建项目文件夹，比如 index 文件夹，里面包含 index.js，index.wxml，index.wxss，index.json（wxml 可看做 html，wxss 可看做 css）。

需要强调的是，index.wxml 里搭建页面结构时，基本用<bock>, <view>（块级）,<text>（行内）,<image>等标签组成，同时可以在上面绑定数据和交互处理函数。而 index.json 可对该页做一些配置，例如窗口 title：

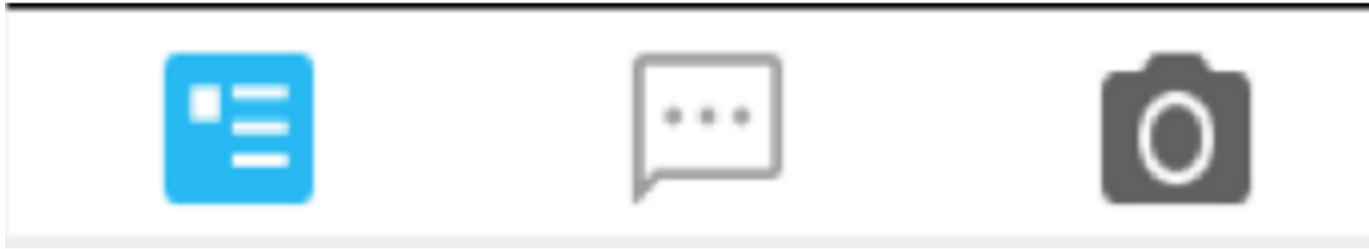
```
{  
  
"navigationBarTitleText": "首页"  
}
```

还有像下面这种底部菜单导航条的配置：



```
"tabBar": {  
  "color": "#666",  
  "selectedColor": "#268dcd",  
  "borderStyle": "white",  
  "backgroundColor": "#fafafa",  
  "list": [{  
    "pagePath": "pages/index/index",  
    "iconPath": "images/navOne.png",  
    "selectedIconPath": "images/navOne-active.png",  
    "text": "名片"  
  }, {  
    "pagePath": "pages/me/me",  
    "iconPath": "images/navTwo.png",  
    "selectedIconPath": "images/navTwo-active.png",  
    "text": "信息"  
  },  
  {  
    "pagePath": "pages/me/me",  
    "iconPath": "images/navTwo.png",  
    "selectedIconPath": "images/navTwo-active.png",  
    "text": "相册"  
  }]  
}
```

底部导航



底部导航

index.js 里，生命周期运作流程如下：

```

},
onMenuShareTimeline: function() {
  return {
    title: "首页"
  }
},
onLoad:function(options){
  // 页面初始化 options为页面跳转所带来的参数
  console.log("index-----inload")
},
onReady:function(){
  // 页面渲染完成
  console.log("index-----onReady")
},
onShow:function(){
  // 页面显示
  console.log("index-----onShow")
},
onHide:function(){
  // 页面隐藏
  console.log("index-----onHide")
},
onUnload:function(){
  // 页面关闭
  console.log("index-----onUnload")
},

```

index.js

**路由：**

开发中很核心的一个东西就是路由了，不需要我们配置，微信已经封装得棒棒的三种跳转方法：

`wx.navigateTo(object)`：保留当前页面，跳转到应用内的某个页面，使用 `wx.navigateBack` 可以返回到原页面。

`wx.redirectTo(object)`：关闭当前页面，跳转到应用内的某个页面。

`wx.navigateBack()`：关闭当前页面，回退前一页面

大家看过微信文档应该知道，微信限制了保留页面的跳转(`navigateTo`)只能有 5 层，所以当超过 5 层时，就可以考虑(`redirectTo`)即关闭当前页面的跳转了。