

微信小程序学习笔记《一》：项目路径，使用页面，新建多一页面 ...

微信的小程序已经推出了，相继的教程也有。官方网页推出 API 和接口说明，目前来讲暂时足够。

可以进入该链接查看。[微信官网](#)

今天尝试一下下载该 web 的开发工具，从目前来讲，使用了一个下午。目前版本给人的感觉还有待继续提升的恐惧。从编码的体验来讲，对一个重度用户来讲并不算太好用。但足够对付一个轻量级的程序。在调试的过程，由于经常切换显得不是太舒服。

1.注意项目路径

由于第一次初始的时候，下载的 demo，由于没有注意 app.json 的放置的文件夹位置，导致了多了一层，导致老是报一个错误。经同事指点后，将文件和项目文件夹放置同一级处，这样就可以顺利调试到。由于 appid 申请不到，在公众号那里需要公测，因此只能有限制使用。

一个下午经过调试了一下临时搭建了一个没有功能的界面。



iPhone 5

wifi



Console

Sources

Network

Storage

AppData

Wxml



编辑



调试



项目



重启



后台



缓存



关闭

初心暖暖心意



荣蒸奶香蛋糕整箱1kg
\$12元

购买



三只松鼠
\$13元

购买



苹果ipad 2
\$399元

购买



飞利浦台灯
\$499元

购买



小米充电宝
\$69元

购买



苹果ipad pro



首页



我的

> <page>...</page>

2.使用组件

微信这里提供了很多组件，这个和之前的微信样式有点类似。通过组件的组合可以拼接一些界面使用。这个有点像 css 那样，据官方说 wxss 文件和 css 有点相似，区别和限制。

配置一个 Tabbar，一个 Tabbar 最多是 5 个，相应的字段需要对其进行配置，提供了图片，点击的关联的页面。选择的图标状态。

官方的代码如下：

```
"tabBar": {  
  "color": "#ddddd",  
  "selectedColor": "#3cc51f",  
  "borderStyle": "black",  
  "backgroundColor": "#ffffff",  
  "list": [{  
    "pagePath": "pages/index/index",  
    "iconPath": "image/icon_component.png",  
    "selectedIconPath": "image/icon_component_HL.png",  
    "text": "首页"
```

```
}, {  
  "pagePath": "pages/logs/logs",  
  "iconPath": "image/icon_API.png",  
  "selectedIconPath": "image/icon_API_HL.png",  
  "text": "我的"  
}]
```

使用视图，图片，文本，按钮

```
<view class="card">  
  <view class="shopitem">  
    <image class="shopimg" src="http://img12.360buyimg.com/n5/jfs/t1174/338/865120803/45181/e564982f/5563cb4cNa42cf3  
2c.jpg"></image>  
    <text class="title">小米充电宝</text>  
    <text class="price">$69 元</text>  
    <button class="buy" type="primary">购买</button>  
  </view>  
</view>
```

按钮提供了一些使用的样式类型。

3.使用页面

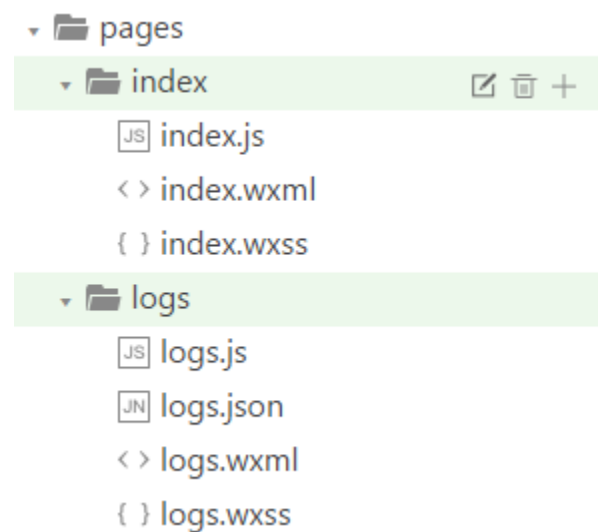
创建 Tabbar 按钮，需要关联一些页面，页面的命名方式需要保持一致。

如文件夹为 index，里面 js wxml 和 wxss 保持 前缀一样。

index.js

index.wxml

index.wxss



切换页面的时候，可以在 js 里面控制一些行为。下面可以看到，通过 app 来获取用户的基本信息。

定义数据

```
data: {  
  
  motto: 'Hello World',  
  
  userInfo: {}//用户的基本信息，微信用户的信息  
  
},
```

//官方案例

//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
        })
    })
}
})
```

数据绑定

通过加载的数据，获取到了用户的基本信息 userInfo。通过数据绑定来显示微信的头像和名字。

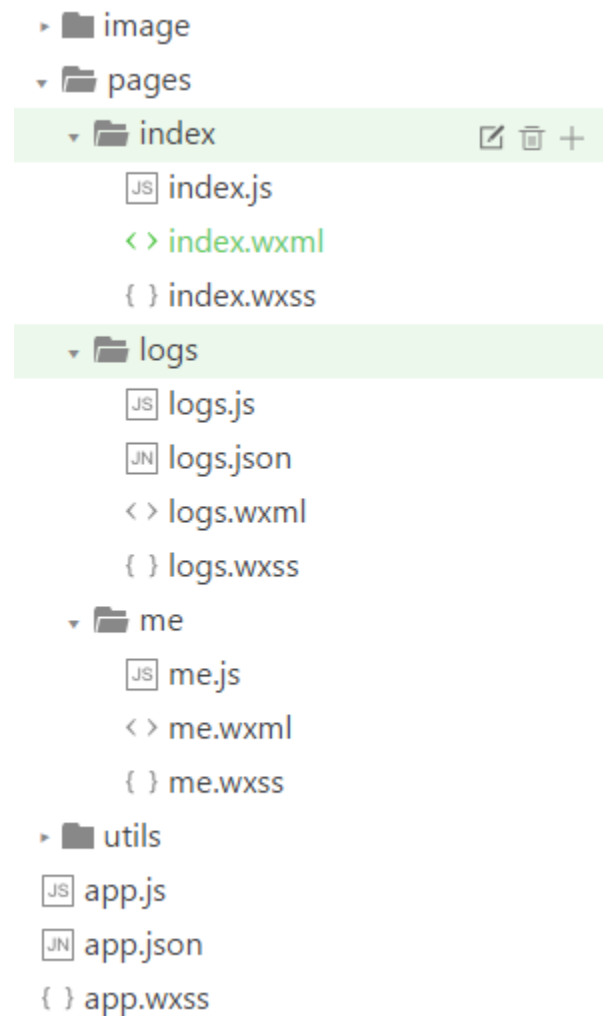
```
<view bindtap="bindViewTap" class="userinfo">
```

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

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

```
</view>
```


4.新建多一页面



编辑器里面新增一个页面，并且在 app.json 里面配置多一页面，把相应的页面配置一下。

在 pages 里面进行配置新增多一页面。并在 tabBar 里面新增一个选项指向我们刚创建的页面。

```
{  
  "pages": [  
    "pages/index/index",  
    "pages/logs/logs",  
    "pages/me/me"  
  ],  
  "window": {  
    "navigationBarTitleText": "初心暖暖心意",  
    "navigationBarBackgroundColor": "#000",  
    "navigationBarTextStyle": "white"  
  },  
  "tabBar": {  
    "color": "#dddddd",  
    "selectedColor": "#3cc51f",  
    "borderStyle": "black",  
  }  
}
```

```
"backgroundColor": "#ffffff",  
  
"list": [{  
  
  "pagePath": "pages/index/index",  
  
  "iconPath": "image/icon_component.png",  
  
  "selectedIconPath": "image/icon_component_HL.png",  
  
  "text": "首页"  
  
}, {  
  
  "pagePath": "pages/logs/logs",  
  
  "iconPath": "image/icon_API.png",  
  
  "selectedIconPath": "image/icon_API_HL.png",  
  
  "text": "日志"  
  
},  
  
{  
  
  "pagePath": "pages/me/me",  
  
  "iconPath": "image/icon_API.png",  
  
  "selectedIconPath": "image/icon_API_HL.png",
```

```
"text": "我的"
```

在 me.js 里面定义一个数据

```
//index.js
```

```
//获取应用实例
```

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

```
Page({
```

```
  data: {
```

```
    name: 'weChat',
```

```
    age:"qqq"
```

```
  },
```

```
  onLoad: function () {
```

```
  }
```

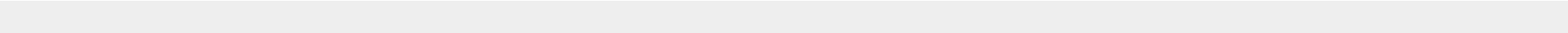
```
})
```

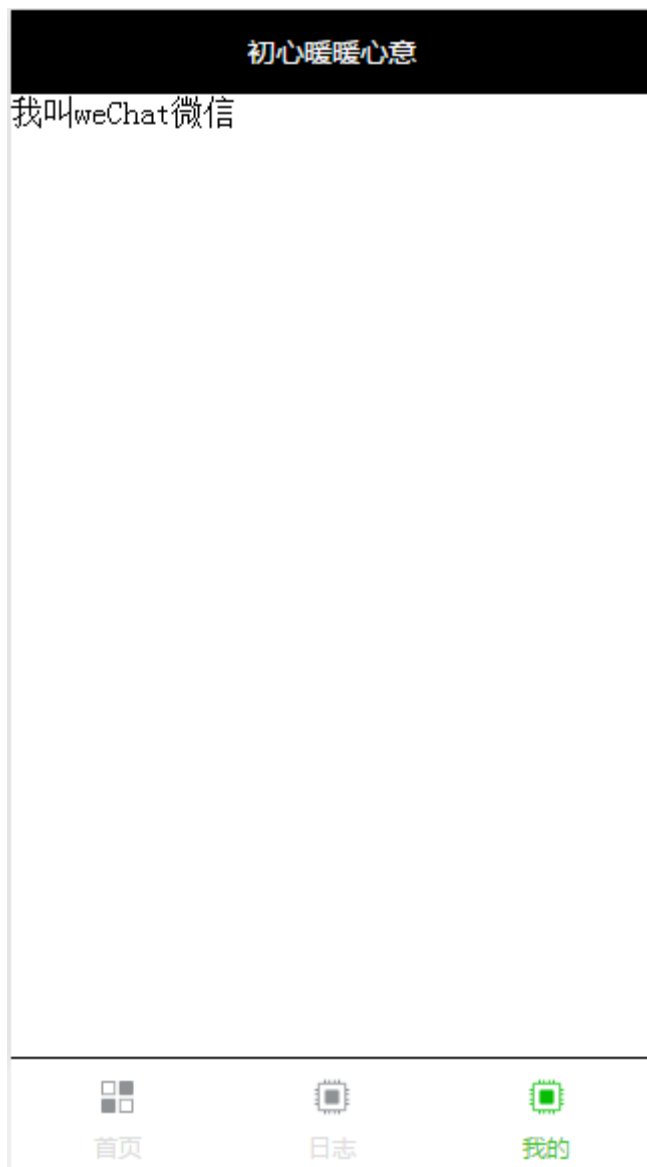
界面如下，同时绑定了我们创建的字段。

<view>

<text>我叫{{name}}微信</text>

</view>





初步的页面也实现出来了。

5.不足的地方

用起来的时候，软件启动起来十分慢，其次是部分文档出现了错误（同事发现了标签不对的地方）。

软件调试提示性一般对 wxml 支持不够好，编写起来会有点吃力。提示这块存在薄弱的地方。

（1）建立文件的时候，还需要填写后缀名，这一点让我不理解

（2）软件经常切换调试，如果能实时刷新这个效果不错。类似 ReactNative 那样

（3）启动画面比较慢

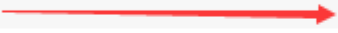
（4）提示比较薄弱，对 wxml 提示比较差。对这种重度用户而言，用起来不够智能。

（5）文档复制代码比较糟糕，选取的时候经常选取不到。加个复制按钮如何？

（6）代码区域不能拖动，编译的时候为何不在加一个按钮在编辑区域里？

数据绑定

```
<!--wxml-->  
<text> {{message}} </view>
```



```
// page.js  
Page({  
  data: {  
    message: 'Hello MINA!'  
  }  
})
```

6.app 和小程序争论焦点

凭借小程序曝光，很多媒体过度阅读这个小程序可以消灭 app 这样的说法。app 的开发门槛相对小程序要高，对于游戏而言，能不能进去还是未知之数。小程序限制了 document 和 window 对象，也不能操控 dom，因此在制作上会有所限制。app 的门槛高，这个其实 h5 和 app 争执的观点。在 app 上性能更加有所保障，对硬件底层支持更足。这也是小程序很难比较的。目前还没有公测，对小程序可以发挥到什么商业价值。这还有待观察。

我们在微信可以看到滴滴打车也就是一个一类的小程序了。对于小程序可以消灭 app 这样说法，笔者还是要持怀疑态度。