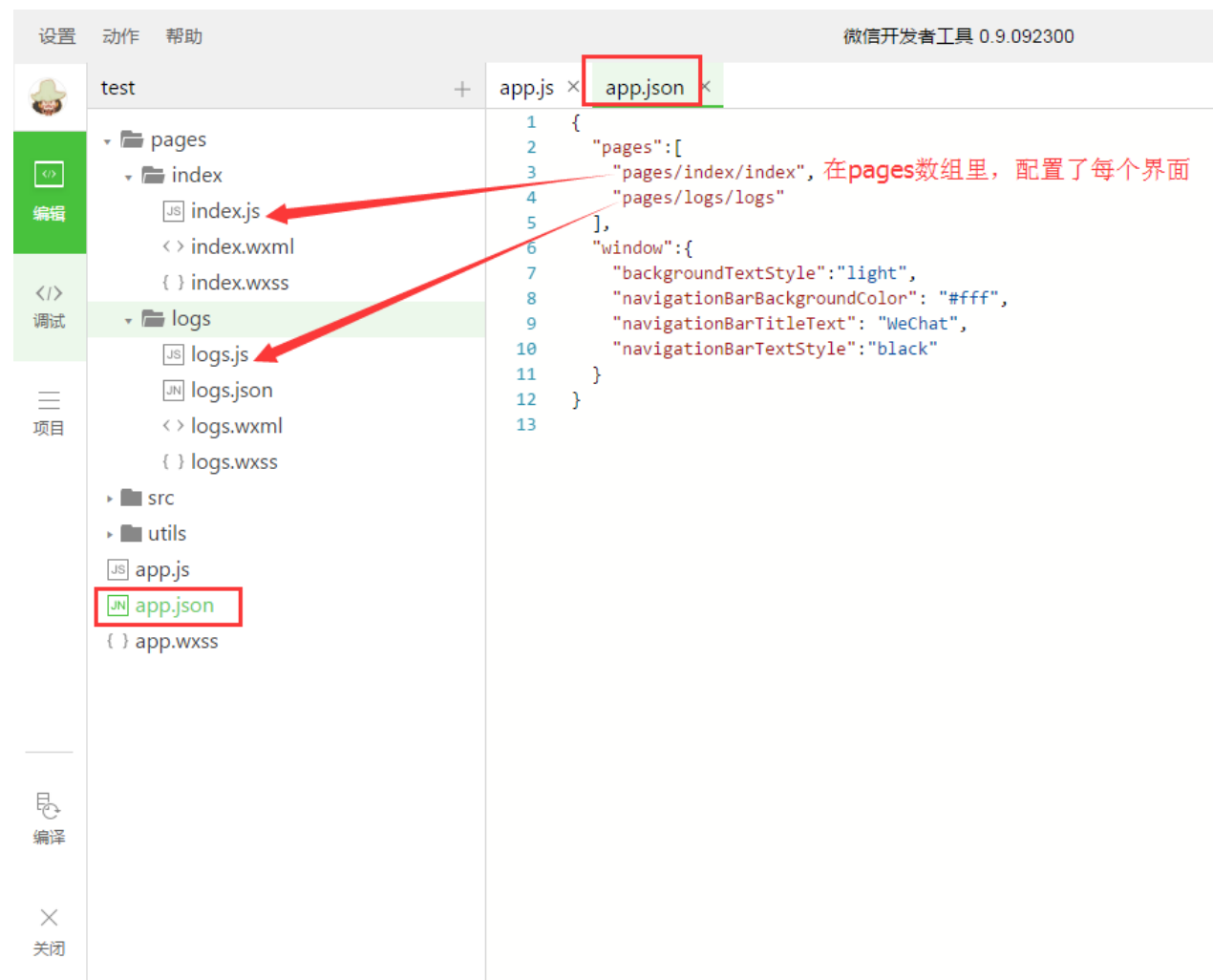


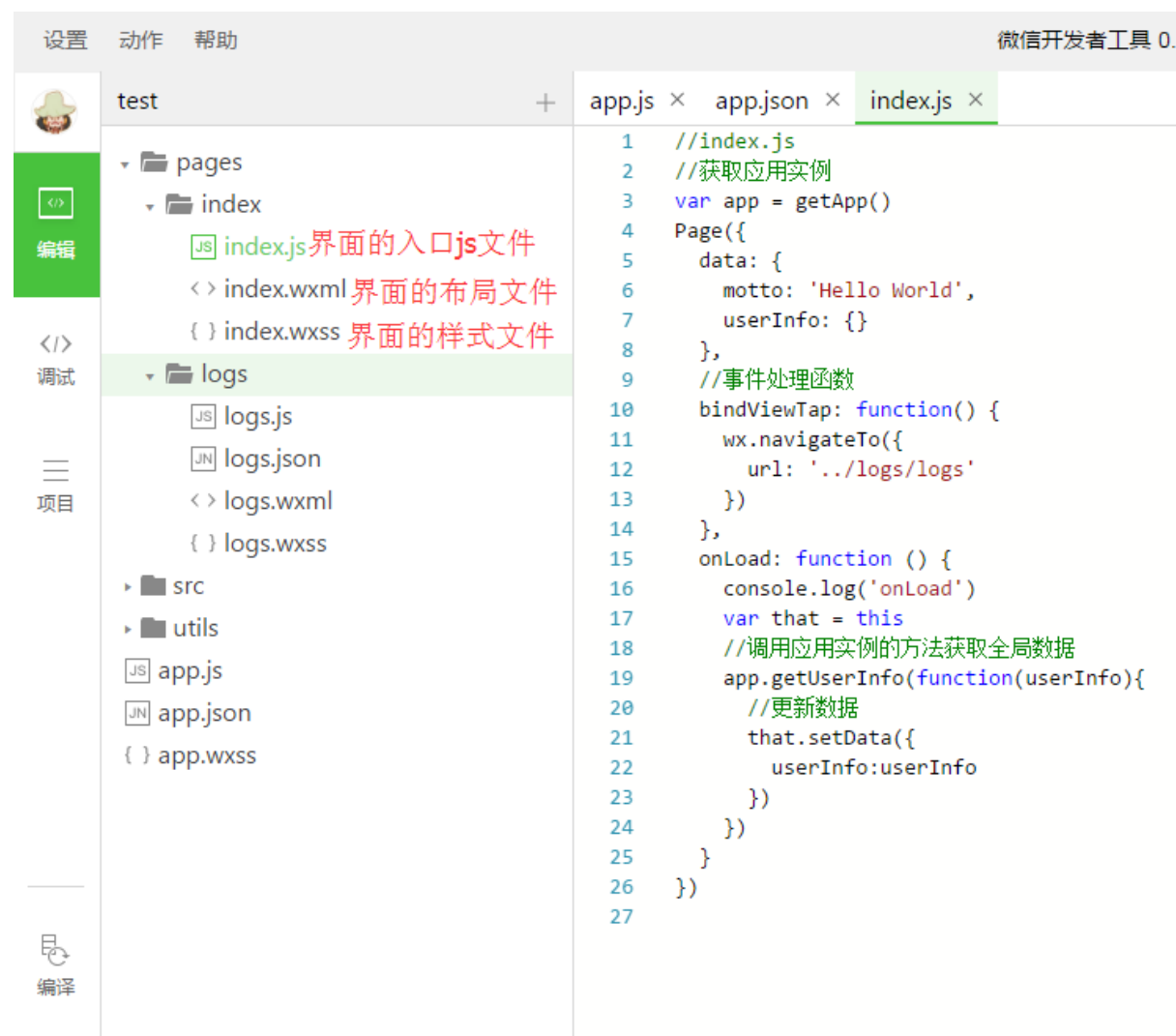
微信小程序学习《一》：目录文件详解，视图渲染详解

一、目录文件详解

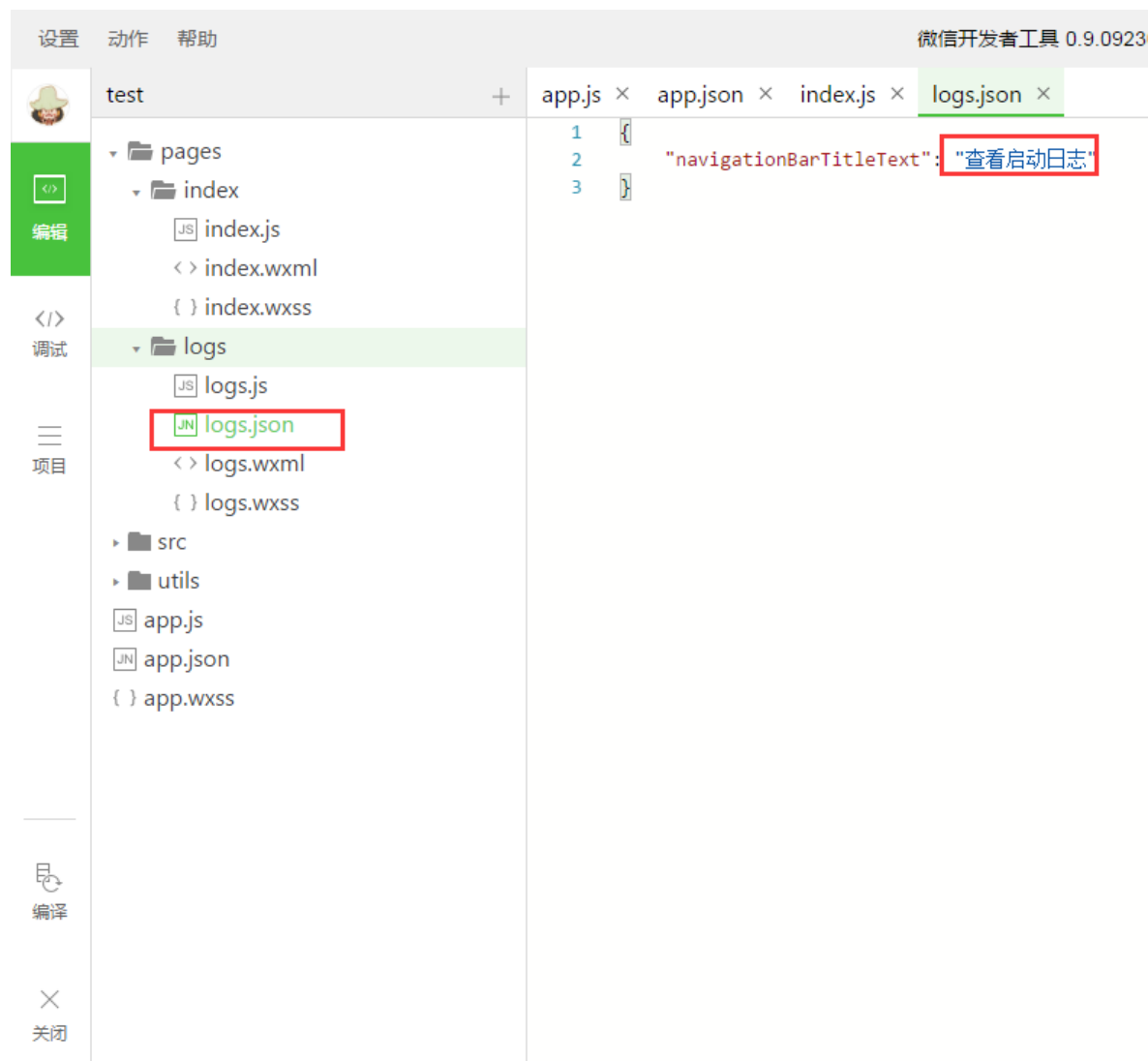
1.首先看下根目录下的 app.json 的文件，可以看到在“pages”的数组里，里面配置了每个界面，且包含了每个界面文件的目录

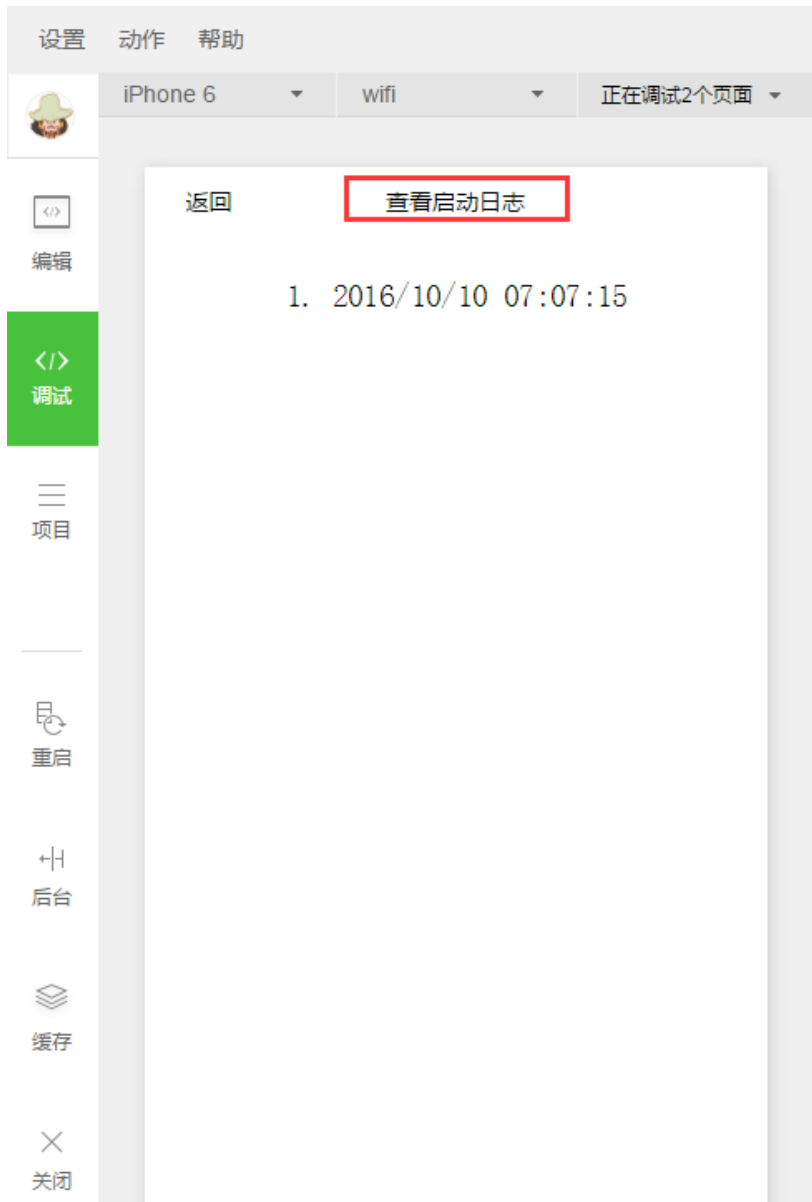


2.可以看到每个页面需要包含两个文件，一个是js 文件，是每个界面的入口，wxml 的文件是每个界面的布局文件，那 wxss 是什么呢？它是每个界面的样式文件。比如：看下图可以看到，创建新的项目有两个界面，一个是 index.js 界面，一个是 logs.js 的界面，且都有对应的布局文件(.wxml) 和样式文件(.wxss)

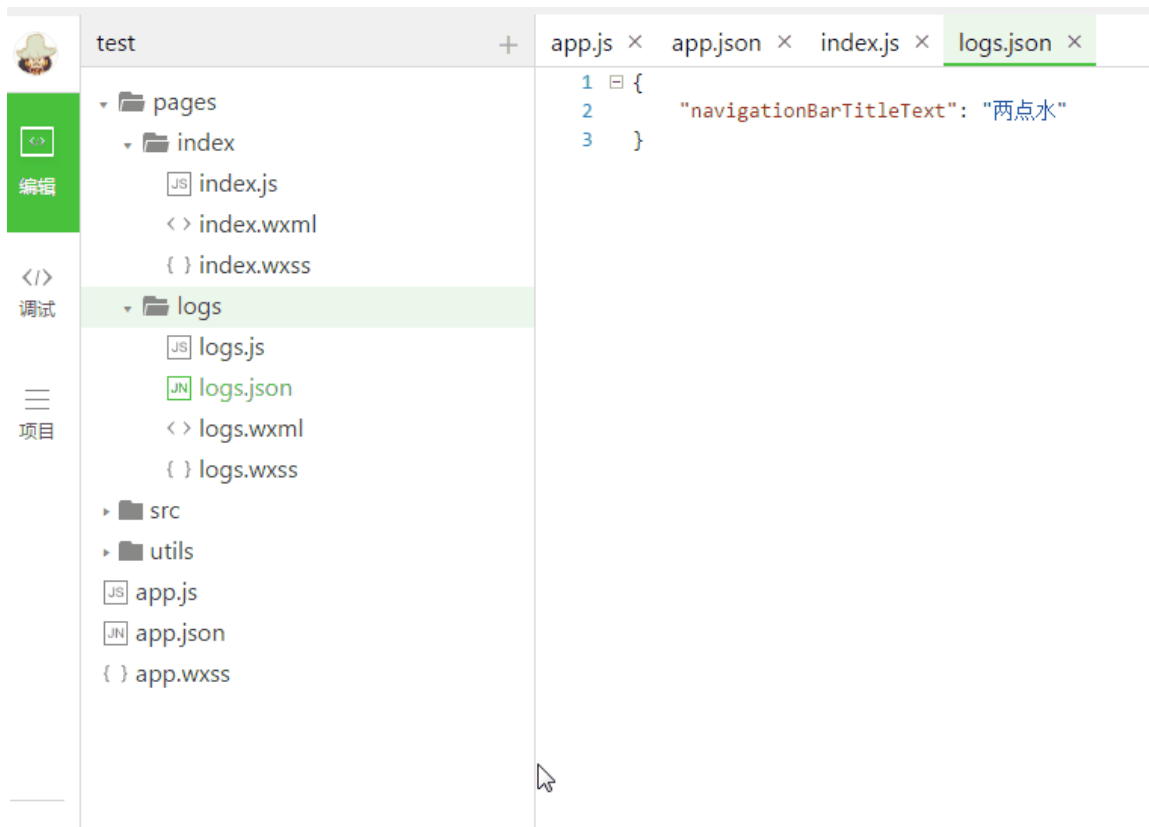


3.可以看到在 logs 文件夹中比 index 文件夹中多出了一个 logs.json 的文件，这个文件是干嘛的呢？可见，这个文件不是必须的，因为 index 界面的文件夹中就没有这个文件，logs.json 这个文件配置了一个界面的 title 的信息





具体效果可以看下 gif 图



总结

1.根目录下的 app.js 是整个程序的入口，app.json 配置整个程序的信息

2.一个微信小程序必需包含的文件只需要有 app.js 和 app.json 以及对应界面的 js 文件和布局文件（.wxml）就可以了

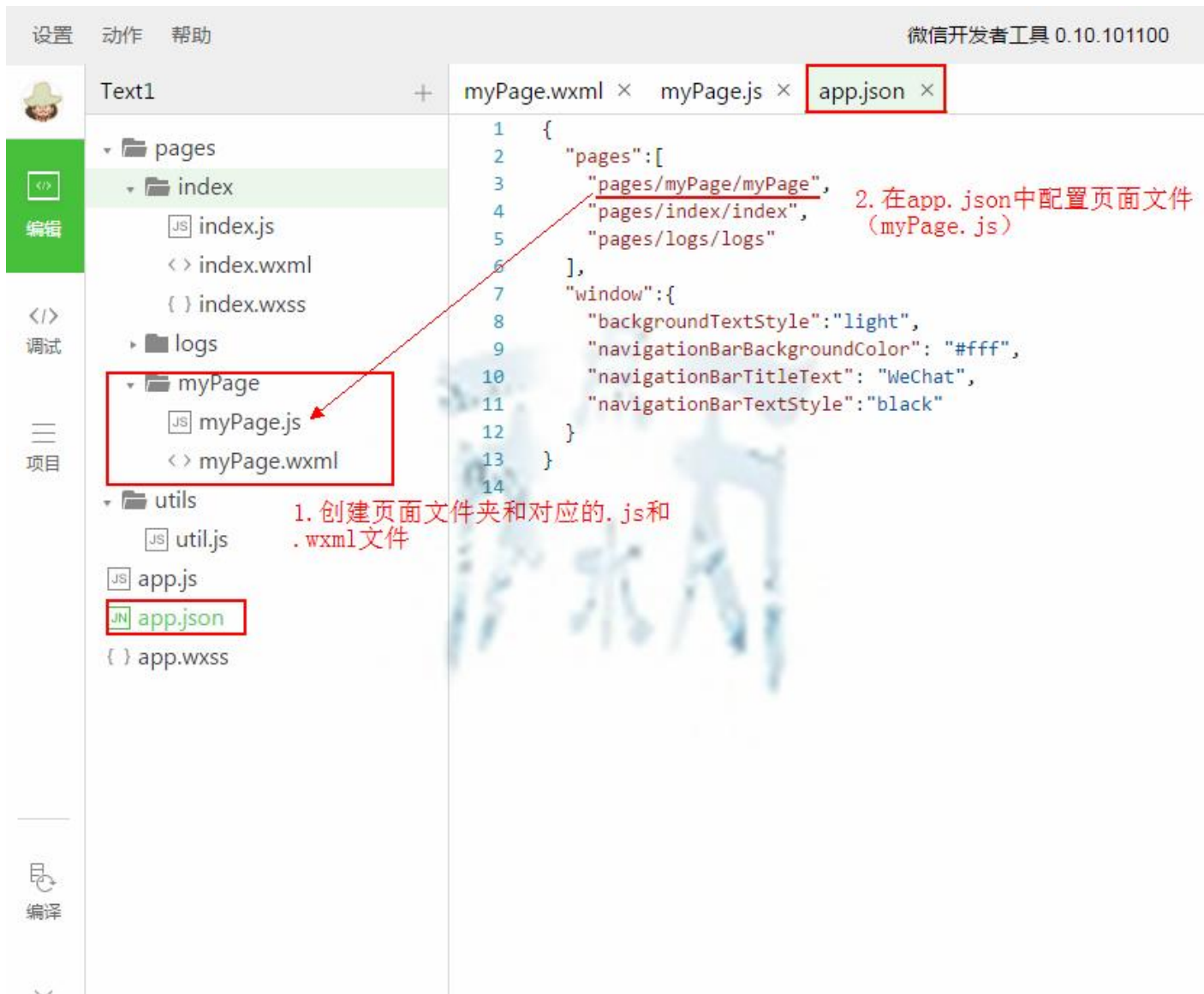
二、视图渲染

一：创建一个页面

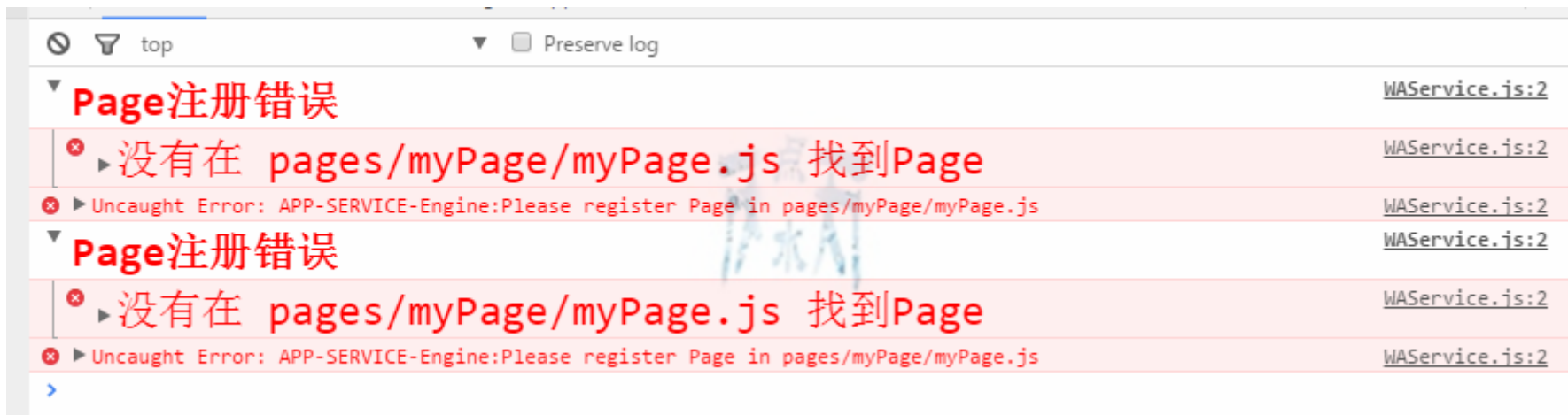
1.首先我们新建一个项目，在新建的项目中有 index 页面和 logs 页面，即欢迎页和小程序启动日志的展示页，他们都在 pages 目录下。微信小程序中的每一个页面的【路径+页面名】都需要写在 app.json 的 pages 中，且 pages 中的第一个页面是小程序的首页。

每一个小程序页面是由同路径下同名的四个不同后缀文件的组成，如：index.js、index.wxml、index.wxss、index.json。js 后缀的文件是脚本文件，json 后缀的文件是配置文件，.wxss 后缀的是样式表文件，.wxml 后缀的文件是页面结构文件。

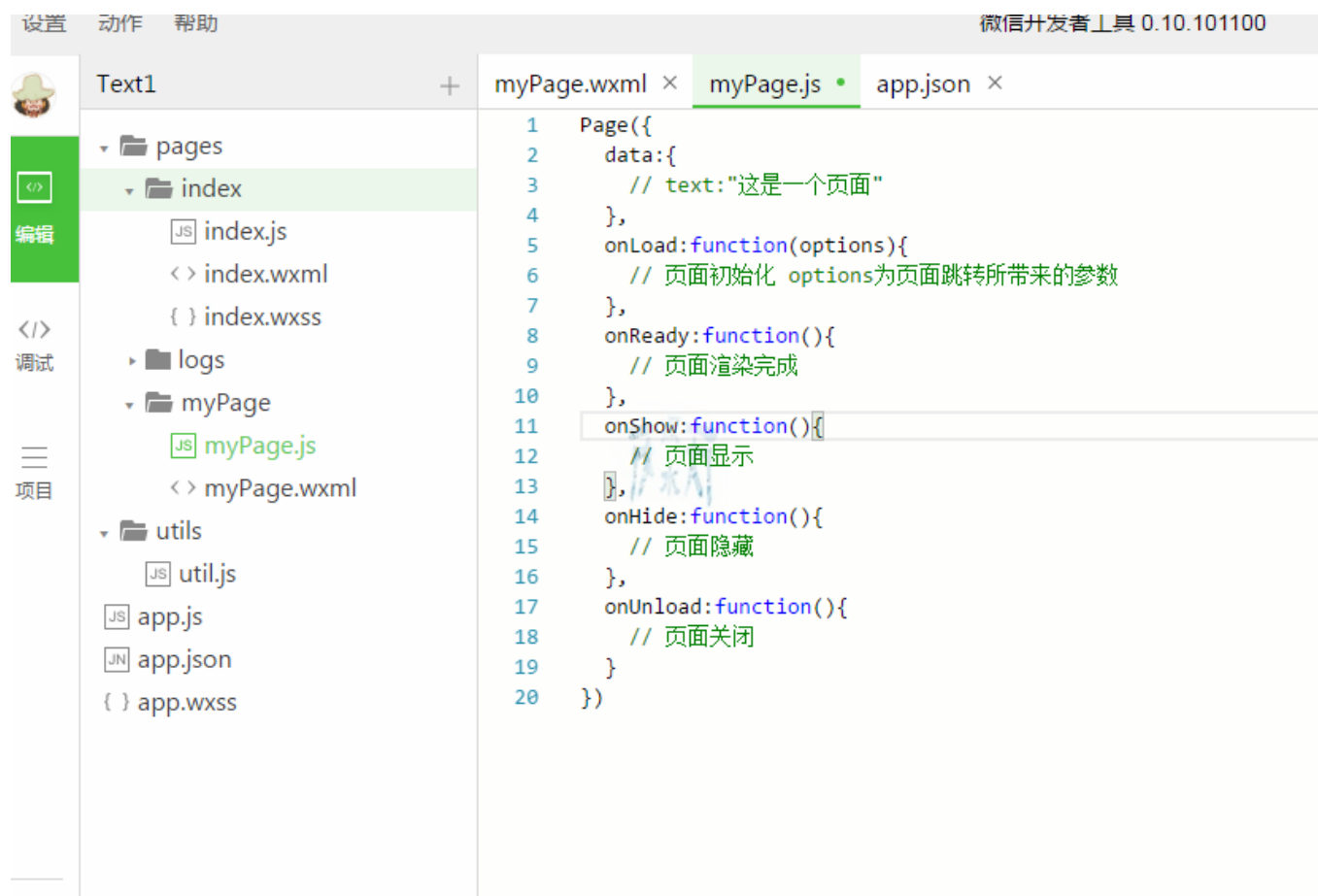
好了，然后我们新建一个自己的页面，首先建一个文件夹(myPage)，在文件夹下创建对应的布局文件(myPage.wxml)和 js 文件(myPage.js)，最后记得在 app.json 中配置页面信息，在 app.json 中需要注意了，在 pages 的数组里，第一个代表着启动时第一个加载的页面



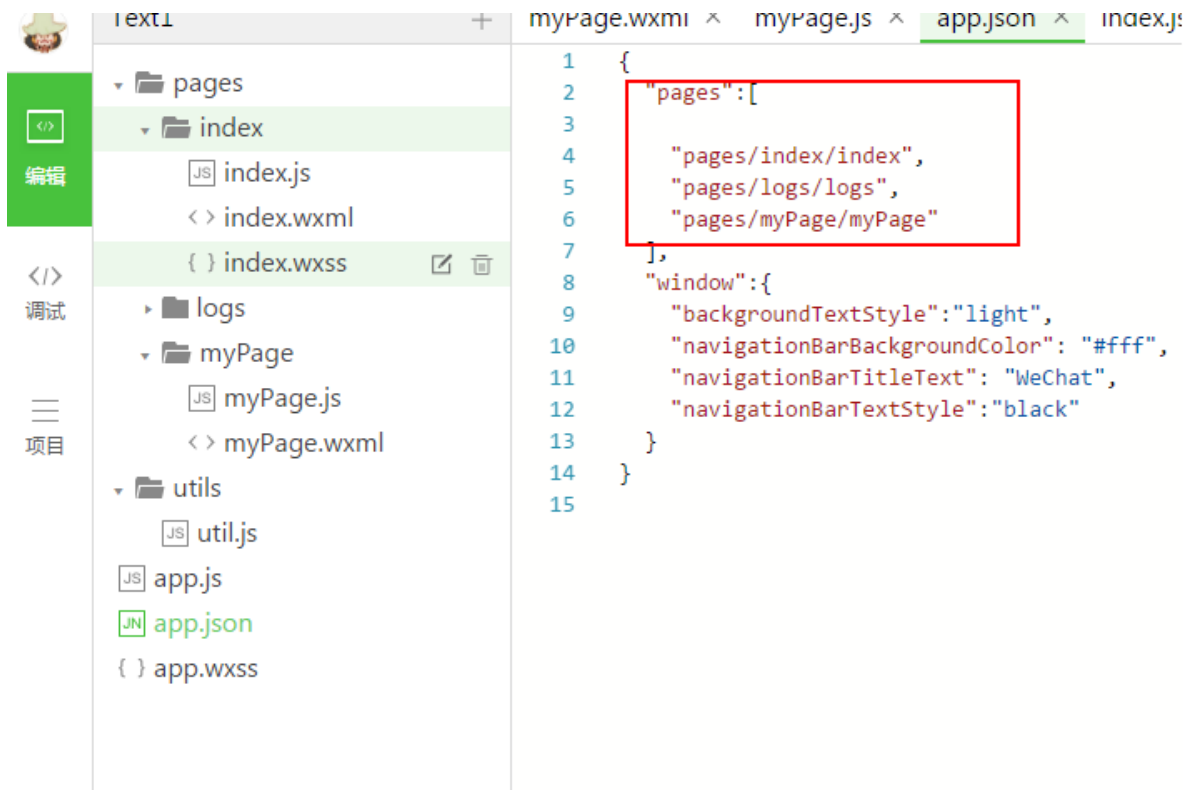
2. 然后进入调试界面，点击重启按钮，看下效果，会发现在控制台中弹出了报错的信息，如下：



3.看报错的信息，很清楚的知道在 myPage.js 文件中没有找到 Page 方法，因此我们在 myPage.js 文件中调用 Page 方法，然后调试，重启，就会发现不报错了！



4.好了，在这一点中也说到，在 app.json 中 pages 的数组里，第一个代表着启动时第一个加载的页面，那好我们把顺序调转一下，就可以验证我们的结论了



建议看官方的教程：<https://mp.weixin.qq.com/debug/wxadoc/dev/>

二、组件的使用和数据的绑定

1. 要了解微信小程序组件的使用，最好的方法就是查看微信小程序的 API 文档：<https://mp.weixin.qq.com/debug/wxadoc/dev/component/>

视图容器

view

scroll-view

swiper

基础内容

icon

text

progress

表单组件

button

checkbox

form

text

文本。

支持转义符“\”。

除了文本节点以外的其他节点都无法长按选中。

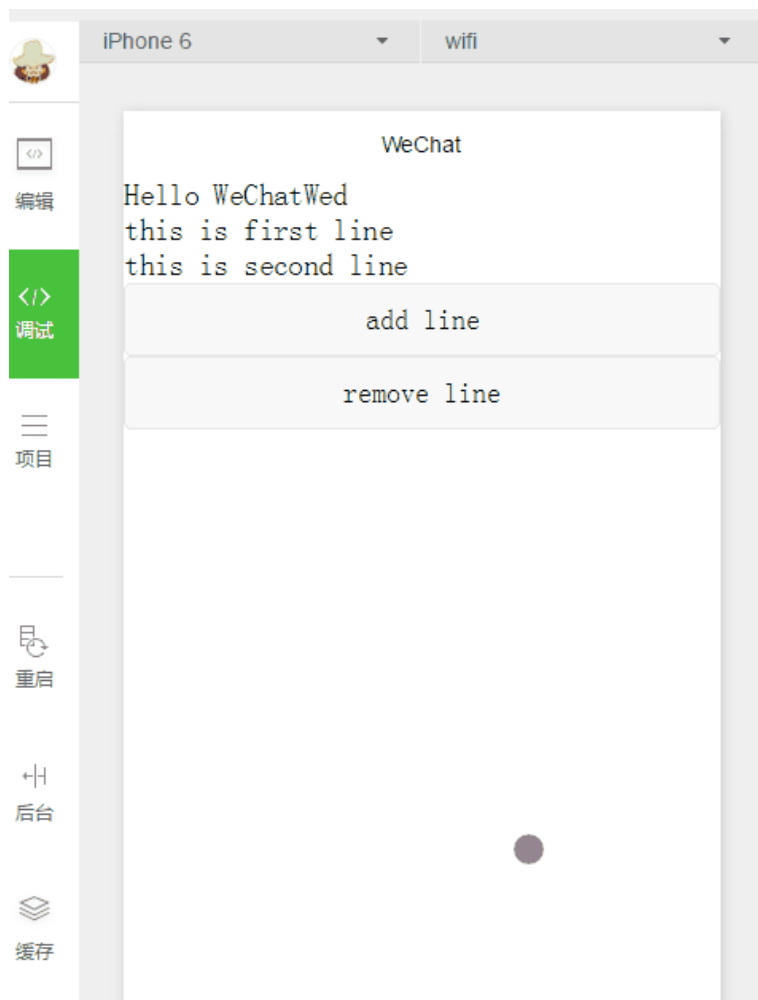
示例：

```
<view class="btn-area">
  <view class="body-view">
    <text>{{text}}</text>
    <button bindtap="add">add line</button>
    <button bindtap="remove">remove line</button>
  </view>
</view>
```

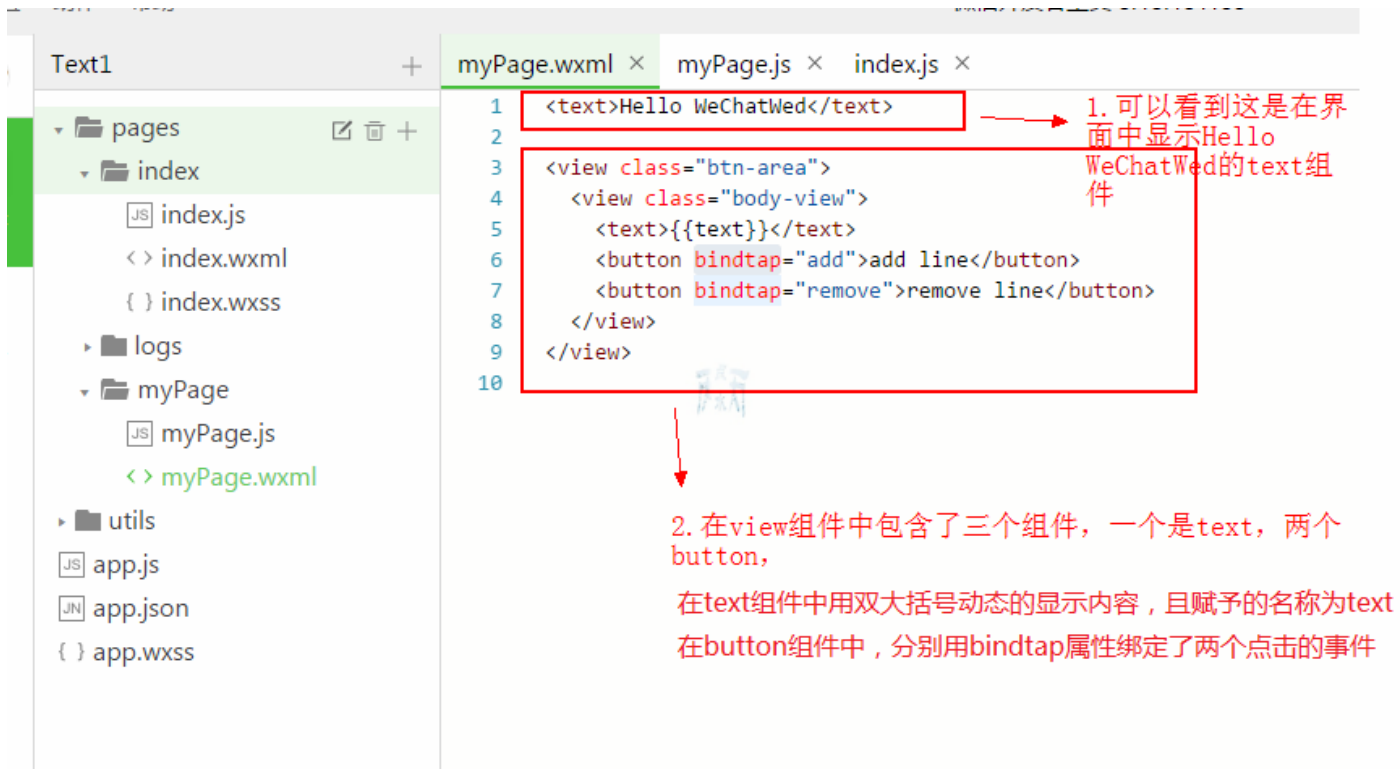
```
var initData = 'this is first line\nthis is second line'
var extraLine = [];
Page({
  data: {
    text: initData
```

可以看到 API 文档详细的介绍了各种组件的使用，要熟悉组件的使用，最好的方式就是多练了！

2.看着文档，把官方文档的例子直接贴进来，然后我们也创建一个 text 组件，看看效果先吧：



结合代码进行分析：



js 代码：

设置 动作 帮助

微信开发者工具 0.10.101100

Text1

+

myPage.wxml × myPage.js × index.js ×

pages

index

index.js

index.wxml

index.wxss

logs

myPage

myPage.js

myPage.wxml

utils

app.js

app.json

app.wxss

</>

编辑

</>

调试

三

项目

编译

×

关闭

1.定义两个常量

var initData = 'this is first line\nthis is second line'

var extraLine = [];

3.实现两个button组件的点击事件的方法

add: function(e) {

extraLine.push('other line')

this.setData({

text: initData + '\n' + extraLine.join('\n')

})

},

remove: function(e) {

if (extraLine.length > 0) {

extraLine.pop()

this.setData({

text: initData + '\n' + extraLine.join('\n')

})

}

},

add: 往数组中添加数据, 然后text组件显示

remove: 判断extraLine数组的长度是否大于0, 大于0就移除数组中的最后一个元素, 然后显示

2.给动态赋值的text组件设置值

data:{

// text:"这是一个页面"

text: initData

},

onLoad:function(options){

// 页面初始化 options为页面跳转所带来的参数

},

onReady:function(){

// 页面渲染完成

},

onShow:function(){

// 页面显示

}

三、渲染标签

微信小程序给我提供了两种渲染标签：

1.判断标签 wx:if 和 wx:else



Text1 + myPage.wxml × myPage.js × index.js ×

pages

- index
 - index.js
 - index.wxml
 - index.wxss
- logs
- myPage
 - myPage.js
 - myPage.wxml
- utils
- app.js
- app.json
- app.wxss

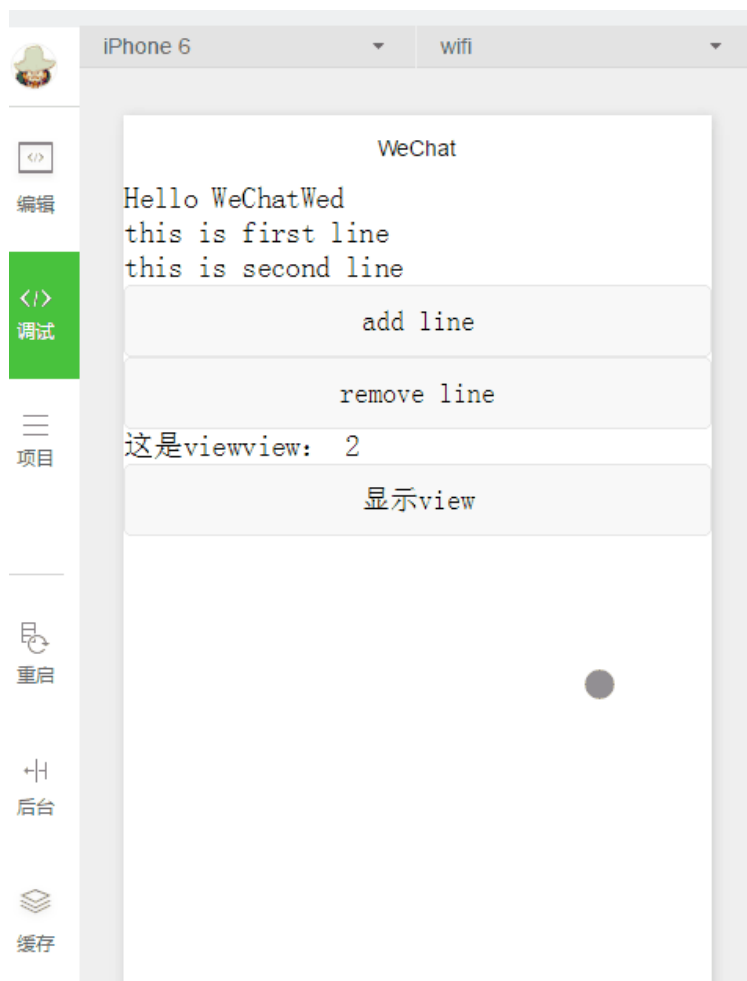
```
4 Page( {
5
6   add: function( e ) {
7     extraLine.push( 'other line' )
8     this.setData( {
9       text: initData + '\n' + extraLine.join( '\n' )
10    })
11  },
12  remove: function( e ) {
13    if( extraLine.length > 0 ) {
14      extraLine.pop()
15      this.setData( {
16        text: initData + '\n' + extraLine.join( '\n' )
17      })
18    }
19  },
20
21  show: function( e ) {
22    var show = this.data.isShow
23    this.setData( {
24      textShow: "这是viewview: ",
25      isShow: !show
26    })
27  },
28 },
29
30
31 data: {
32   // text:"这是一个页面"
33   text: initData,
34   isShow: true
35 },
```

点击事件

1.初始化值

虽然之前没有接触过 js，可是代码也挺容易理解的，逻辑思维跟 Android 还是有点像，不过想学好，还是要学下 js 的，所以最近我也会学下 js

最后我们看下效果图吧，可以看到当位 true 的时候就显示 1 false 时显示 2

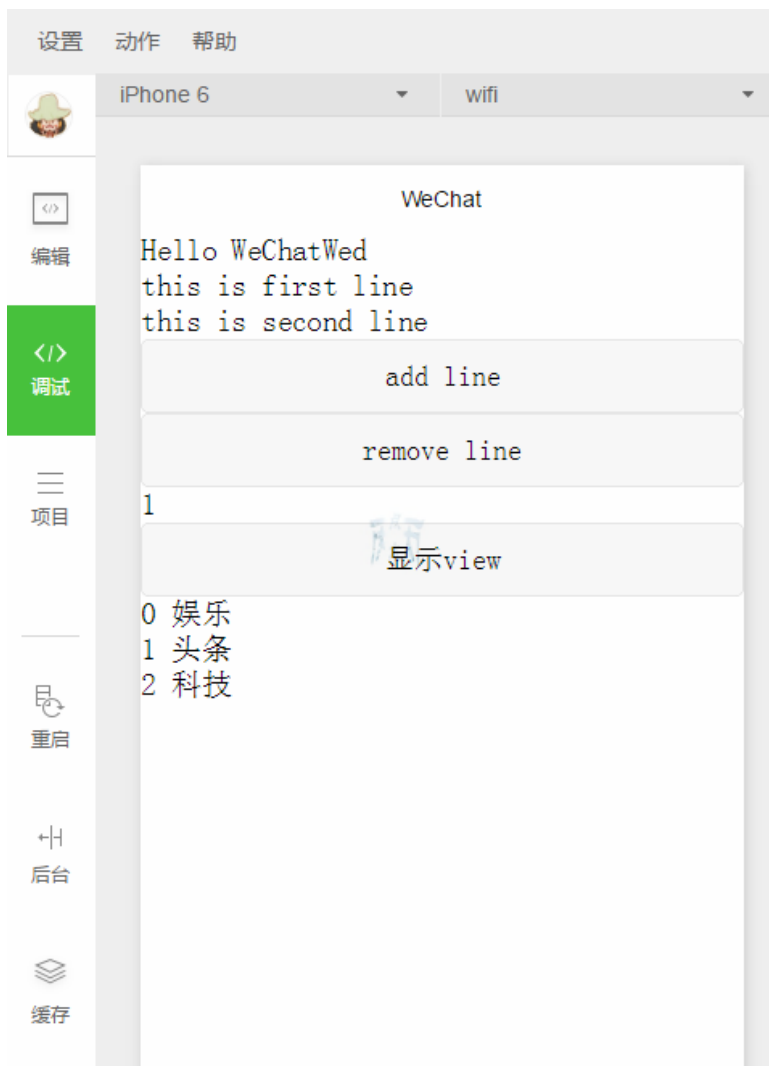


2.循环标签 (wx:for)

看下循环标签在布局文件中的使用吧



看下效果图：



四、样式模板的使用

从上面的代码就可以看到，一个界面可能会用到很多布局的代码，可是界面与界面之间可能会有很多重复的样式，然而这些样式的代码，每次都

要重新写就很麻烦了，而且也浪费资源，微信小程序像其他开发一样，也有样式模板的使用方式

1.使用 include 的方式包含模板样式

例如：`<include src="../../templates/myText" />`

具体的使用看图片：

设置 动作 帮助

微信开发者工具

Text1

+

myText.wxml ×

myPage.wxml ×

pages

index

logs

myPage

myPage.js

myPage.wxml

templates

myText.wxml

utils

app.js

app.json

app.wxss

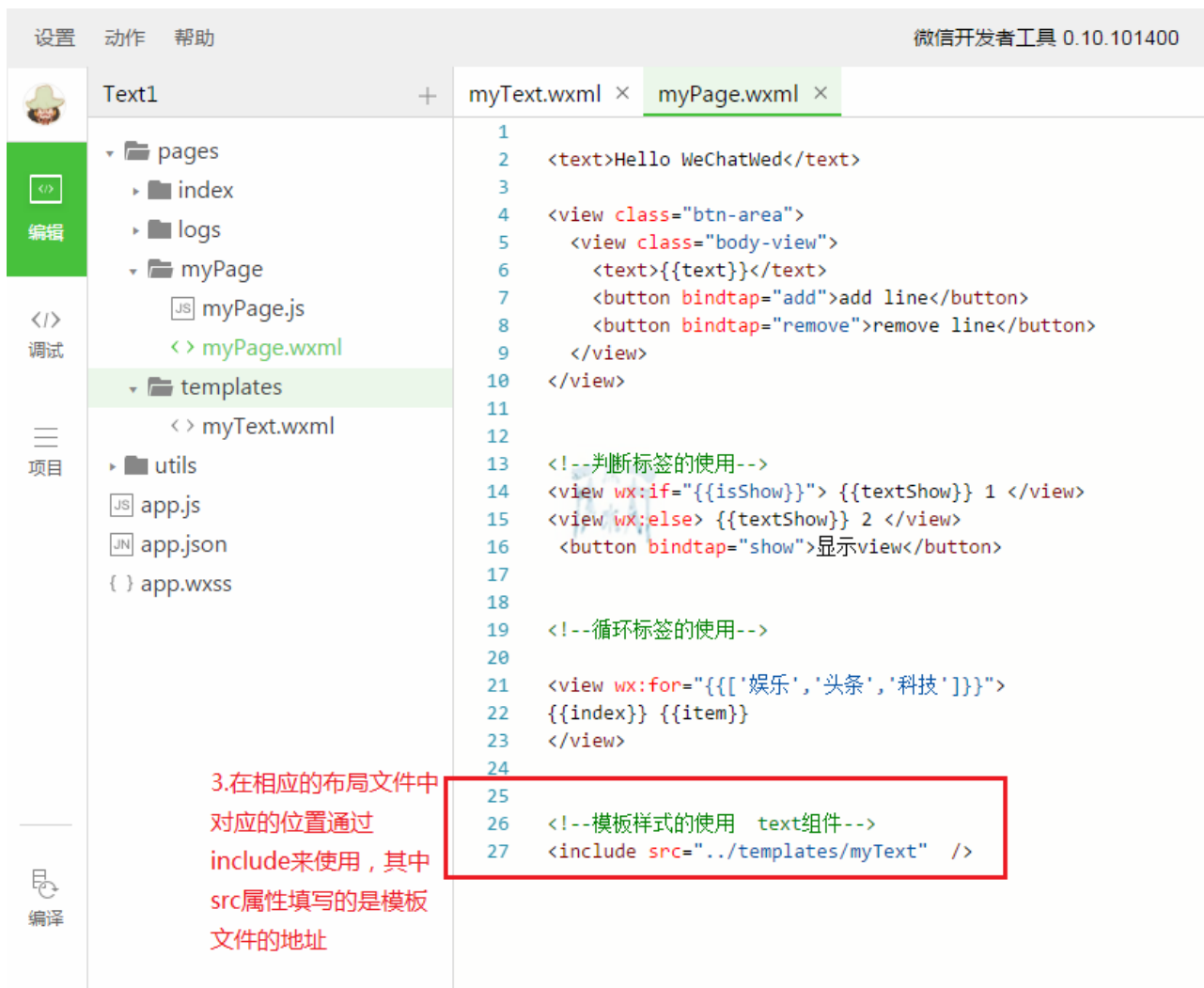
1. 创建了templates的文件夹，用来存放模板，创建了一个myText的样式

1 <text>

2 text组件模板的使用

3 </text>

2. 填写text组件的样式



使用这种方式引用模板，实际上就是等同于把 templates 中的 myText 的代码全部复制到 include 这里

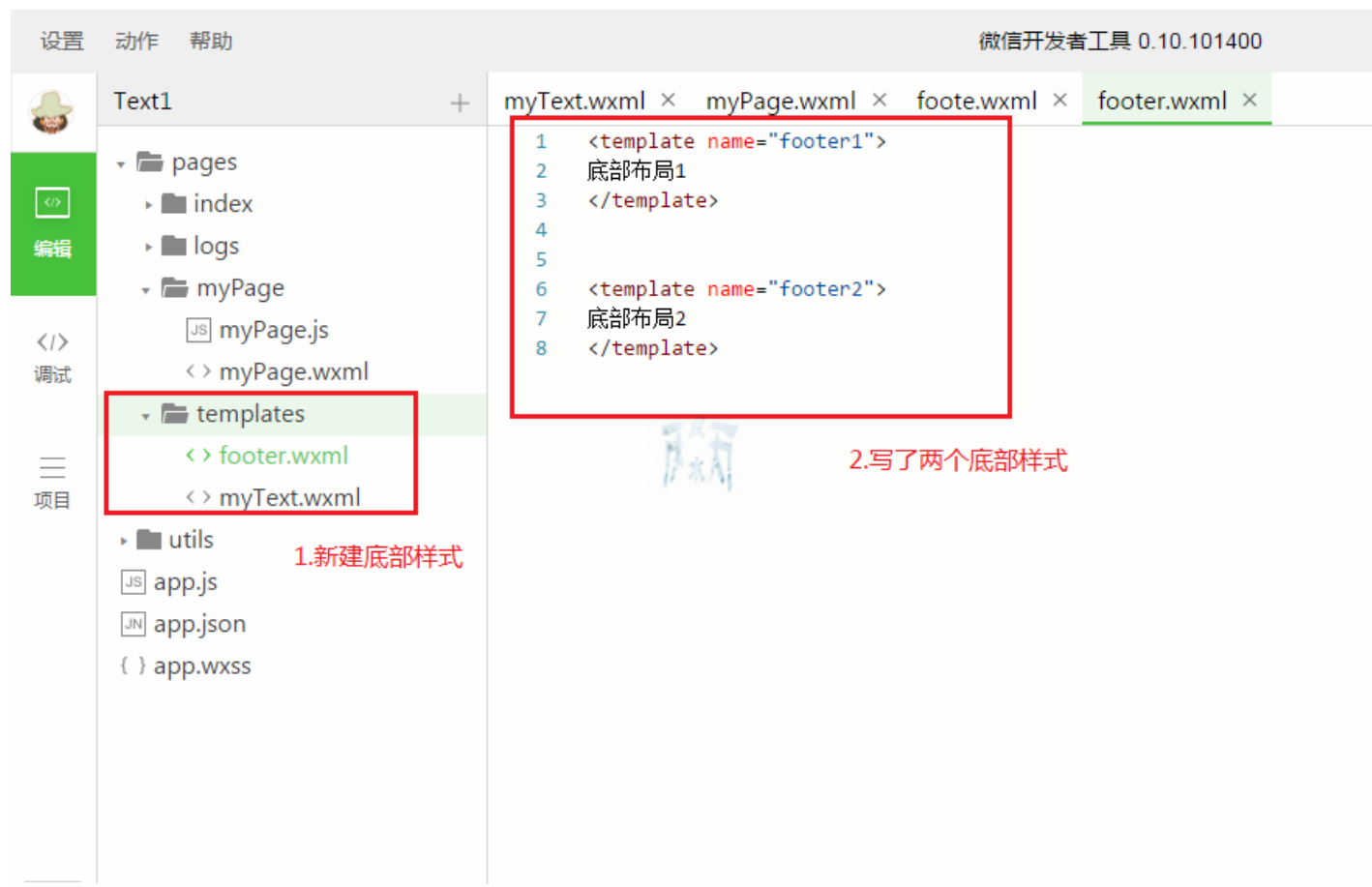
2.使用 import 的方式

例如：

<!--模板样式的使用 底部样式-->

<import src="../../templates/footer" />

<template is="footer1" />



设置 动作 帮助 微信开发者工具 0.10.101400

Text1 + myText.wxml × myPage.wxml × foote.wxml × footer.wxml ×

编辑

调试

项目

pages

index

logs

myPage

myPage.js

myPage.wxml

templates

footer.wxml

myText.wxml

utils

app.js

app.json

app.wxss

这里不能直接使用include的方式进行调用模板，而且因为底部样式有两个，我们也要用is属性来声明调用那个

11

12

13 <!--判断标签的使用-->

14 <view wx:if="{{isShow}}"> {{textShow}} 1 </view>

15 <view wx:else> {{textShow}} 2 </view>

16 <button bindtap="show">显示view</button>

17

18

19 <!--循环标签的使用-->

20

21 <view wx:for="{{['娱乐','头条','科技']}}">

22 {{index}} {{item}}

23 </view>

24

25

26 <!--模板样式的使用 text组件-->

27 <include src="../../templates/myText" />

28

29 <!--模板样式的使用 底部样式-->

30 <import src="../../templates/footer" />

31 <template is="footer1" />

32


33 <!--模板样式的使用 底部样式-->

34 <import src="../../templates/footer" />

35 <template is="footer2" />


最后看下效果图：

设置 动作 帮助




iPhone 6


wifi




编辑




调试




项目



重启



后台



缓存

WeChat

Hello WeChatWed
this is first line
this is second line

add line

remove line

1

显示view

0 娱乐
1 头条
2 科技

text组件模板的使用
底部布局1 底部布局2