

## 有渔微信小程序系统概述《三》**view** 层及小程序框架概述

### 小程序的 **view** 层

MVC 模式这里就不讲了，不懂 MVC 模式的就自己去查阅相关资料。小程序的 view 层由 WXML 与 WXSS 编写，由组件来进行展示。view 层将逻辑层的数据反应成界面显示，同时将界面发生的事件发送给逻辑层。

WXML(WeiXin Markup language)用于描述页面的结构，可以想象成 Html 文件。

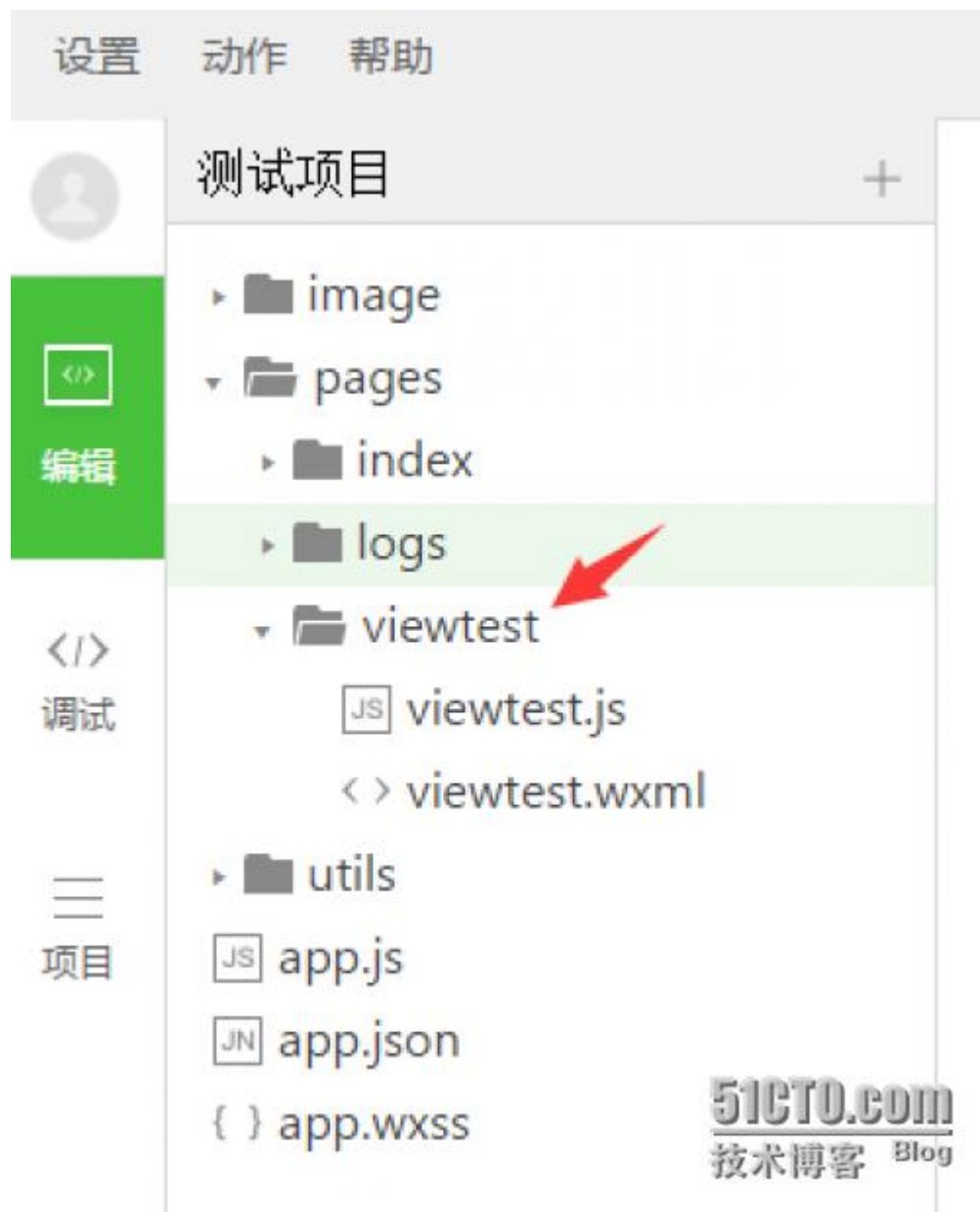
WXSS(WeiXin Style Sheet)用于描述页面的样式，可以想象成 Css 文件。

组件(Component)是视图的基本组成单元，可以想象成 Html 中的组件。

下面我们用简单的例子来看看 WXML 具有什么能力：

### 1、测试环境准备

(1) pages 目录里建立 viewtest 目录，专门用来做 view 层测试。



(2) index 里添加触发 viewtest 的相关代码

### ① index.wxml

<view>修改成：<view bindtap="bindUserTap">

### ② index.js

添加下面的代码：

```
bindUserTap: function() {  
  
  wx.navigateTo({  
  
    url: '../viewtest/viewtest'  
  
  })  
  
},
```

### ③ app.json

把 viewtest 路径加入 pages 参数里：

```
"pages": [  
  
  "pages/index/index",  
  
  "pages/logs/logs",  
  
  "pages/viewtest/viewtest"  
  
],
```

## 2、例子

### (1) 数据绑定

```
<!--viewtest.wxml-->
```

```
<view> {{message}} </view>
```

```
// viewtest.js
```

```
Page({  
  data: {  
    message: 'Hello MINA!'  
  }  
})
```

### (2) 列表渲染

```
<!--viewtest.wxml-->
```

```
<view wx:for="{{array}}"> {{item}} </view>
```

```
// viewtest.js
```

```
Page({  
  data: {  
    array: [1, 2, 3, 4, 5]  
  }  
})
```

(3) 条件渲染

```
<!--viewtest.wxml-->
```

```
<view wx:if="{{view == 'WEBVIEW'}}"> WEBVIEW </view>
```

```
<view wx:elif="{{view == 'APP'}}"> APP </view>
```

```
<view wx:else="{{view == 'MINA'}}"> MINA </view>
```

```
// viewtest.js
```

```
Page({
```

```
data: {  
  view: 'MINA'  
}  
})
```

(4) 模板

```
<!--viewtest.wxml-->
```

```
<template name="staffName">
```

```
<view>
```

```
  FirstName: {{firstName}}, LastName: {{lastName}}
```

```
</view>
```

```
</template>
```

```
<template is="staffName" data="{{...staffA}}"></template>
```

```
<template is="staffName" data="{{...staffB}}"></template>
```

```
<template is="staffName" data="{{...staffC}}"></template>
```

```
// viewtest.js
```

```
Page({  
  data: {  
    staffA: {firstName: 'Hulk', lastName: 'Hu'},  
    staffB: {firstName: 'Shang', lastName: 'You'},  
    staffC: {firstName: 'Gideon', lastName: 'Lin'}  
  }  
})
```

(5) 事件

```
<!--viewtest.wxml-->
```

```
<view bindtap="add"> {{count}} </view>
```

```
// viewtest.js
```

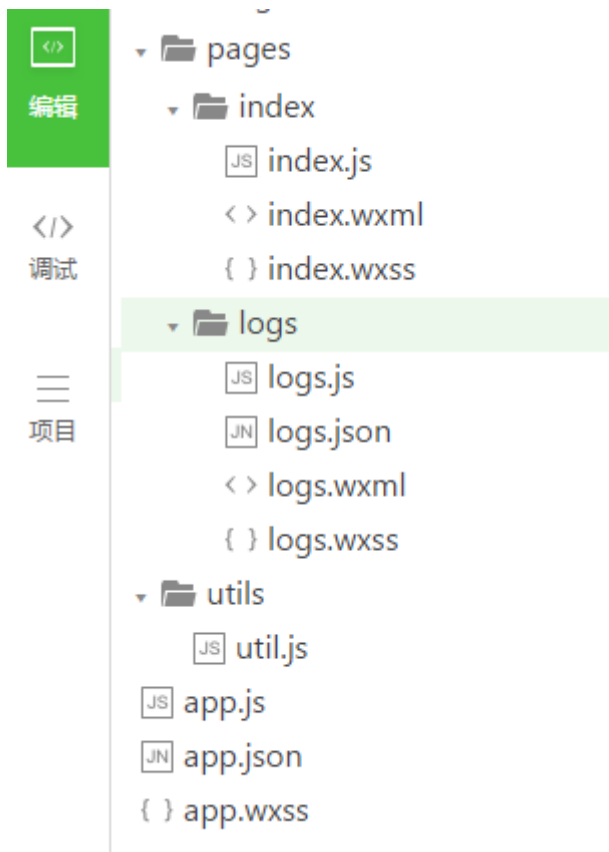
```
Page({  
  data: {  
    count: 1  
  },  
  add: function(e) {  
    this.setData({  
      count: this.data.count + 1  
    })  
  }  
})
```

## 小程序框架概述

### 1、目录结构

先来看下例子 HelloWorld 的目录结构。





这里，app 是小程序的入口，pages 是小程序的页面。比如 index 是小程序的首页，logs 是小程序的日志页。utils 是小程序的工具库。

小程序主体部分由 3 个文件组成，必须放在项目的根目录，如下：

文件	必填	作用
app.js	是	小程序逻辑

app.json	是	小程序公共设置
app.wxss	否	小程序样式

小程序的页面由 4 个文件组成，如下：

文件	必填	作用
js	是	页面逻辑
wxml	是	页面结构
json	否	页面公共设置
wxss	否	页面样式

2、配置

小程序使用 app.json 文件来对进行全局配置，设定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等。

每一个页面也可以使用.json 文件来对本页面的窗口表现进行配置。页面的配置比 app.json 全局配置简单得多 ,只能设置 window 配置项的内容 ,页面中配置项会覆盖 app.json 的 window 中相同的配置项。

### 3、逻辑层 APP Service

小程序开发框架的逻辑层是由 JavaScript 编写，也就是小程序中的 js 文件。逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈。

### 4、视图层 view

视图层由 WXML 与 WXSS 编写，由组件来进行展示，也就是小程序中的 wxml、wxss 文件。它将逻辑层的数据反应成视图，同时将视图层的事件发送给逻辑层。

其中：

WXML(WeiXin Markup language)用于描述页面的结构。

WXSS(WeiXin Style Sheet)用于描述页面的样式。

组件(Component)是视图的基本组成单元。

关于小程序配置、逻辑层、视图层的详细说明，我们在后面用专门的文章进行讲解。