

微信小程序购物商城系统开发系列《一》目录结构

作者灵动生活，的实战开发系列，一步步带领你完成开发任务：

相信大家都已经蠢蠢欲试建立一个自己的小程序，去完成一个独立的商城网站。

先别着急我们一步步来，先尝试下写一个自己的小 demo.

这一篇文章我们主要的是介绍一下小程序的一些目录结构，以及一些语法，为我们后面的微信小程序商城系统做铺垫。

首先我们来了解下小程序的目录结构

```

└─ pages
  └─ index
    └─ index.js
    └─ index.xml
    └─ index.wxss
└─ logs
└─ utils
└─ app.js
└─ app.json
└─ app.wxss
```

Pages 我们新建的一些页面将保存在这个文件夹下面，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。

Utils 我们写的一些公共的工具 js 可以放里面。

app.js 是小程序的脚本代码。我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。调用框架提供的丰富的 API，如本例的同步存储及同步读取本地数据。

```

//app.js
App({
  onLaunch: function () {
    //调用API从本地缓存中获取数据
    var logs = wx.getStorageSync('logs') || []
    logs.unshift(Date.now())
    wx.setStorageSync('logs', logs)
  },
  getUserInfo:function(cb){
    var that = this
    if(this.globalData.userInfo){
      typeof cb == "function" && cb(this.globalData.userInfo)
    }else{
      //调用登录接口
      wx.login({
        success: function () {
          wx.getUserInfo({
            success: function (res) {
              that.globalData.userInfo = res.userInfo
              typeof cb == "function" && cb(that.globalData.userInfo)
            }
          })
        }
      })
    }
  },
  globalData:{
    userInfo:null
  }
})

```

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。



```
{
```

```
  "pages":[
```

```
    "pages/index/index",
```

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

```
  ],
```

```
  "window":{
```

```
    "backgroundTextStyle":"light",
```


```
    "navigationBarBackgroundColor": "#fff",
```

```
"navigationBarTitleText": "WeChat",

"navigationBarTextStyle":"black"

}

}
```



注：app.json 中是对整个小程序的全局设置，当然每个页面也有自己的 json 文件。每个页面下的 json 文件只能对 window 进行设置，所以页面 json 文件下只需写入 window 的配置即可

```
{

"backgroundTextStyle":"light",

"navigationBarBackgroundColor": "#fff",
```

```
"navigationBarTitleText": "WeChat",
```

```
"navigationBarTextStyle":"black"
```

```
}
```



app.wxss 是整个小程序的公共样式表。我们可以在页面组件的 class 属性上直接使用 app.wxss 中声明的样式规则。

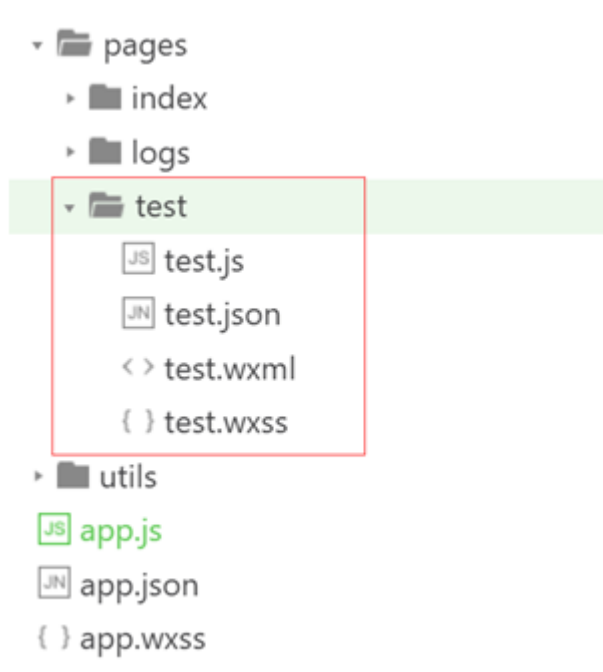
每个页面中也可以定义自己的 wxss。Wxss 也就是微信所提供的 css 样式表，里面的写法几乎和 css 差不多。不过他支持的选择器相对比较少。在写的时候我们以前的一些项目的 css 基本可以拿过来用，不过除了下面这些选择器外，你使用了其他的选择器，可能使整个页面报错！

目前支持的选择器有：

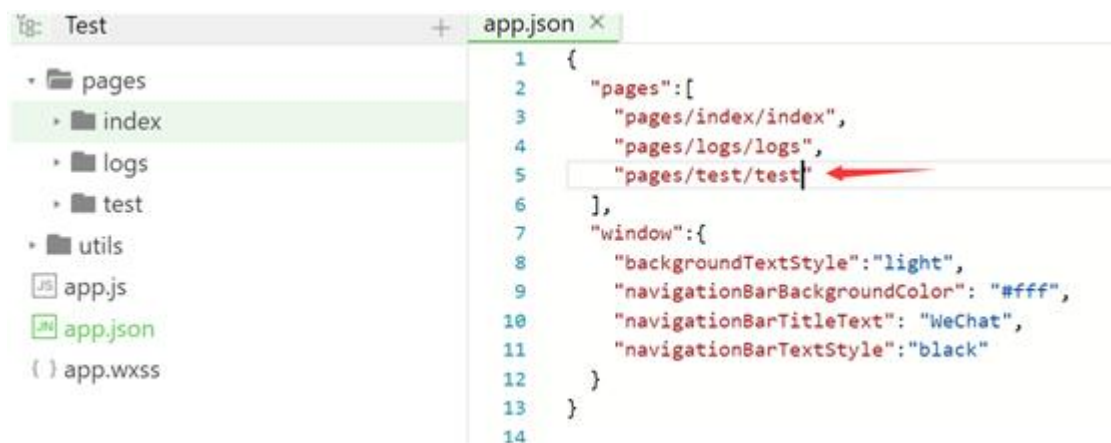
选择器	样例	样例描述
.class	.intro	选择所有拥有 class="intro" 的组件
#id	#firstname	选择拥有 id="firstname" 的组件
element	view	选择所有 view 组件
element, element	view, checkbox	选择所有文档的 view 组件和所有的 checkbox 组件
::after	view::after	在 view 组件后边插入内容
::before	view::before	在 view 组件前边插入内容

Index.whtml wxml 后缀的文件夹也就是微信小程序提供给我们的页面结构文档，就相当于我们以前的 web 页面的 html 差不多，只不过他自定义了一些自己的标签。

接下来我们就新建一个页面来尝试下



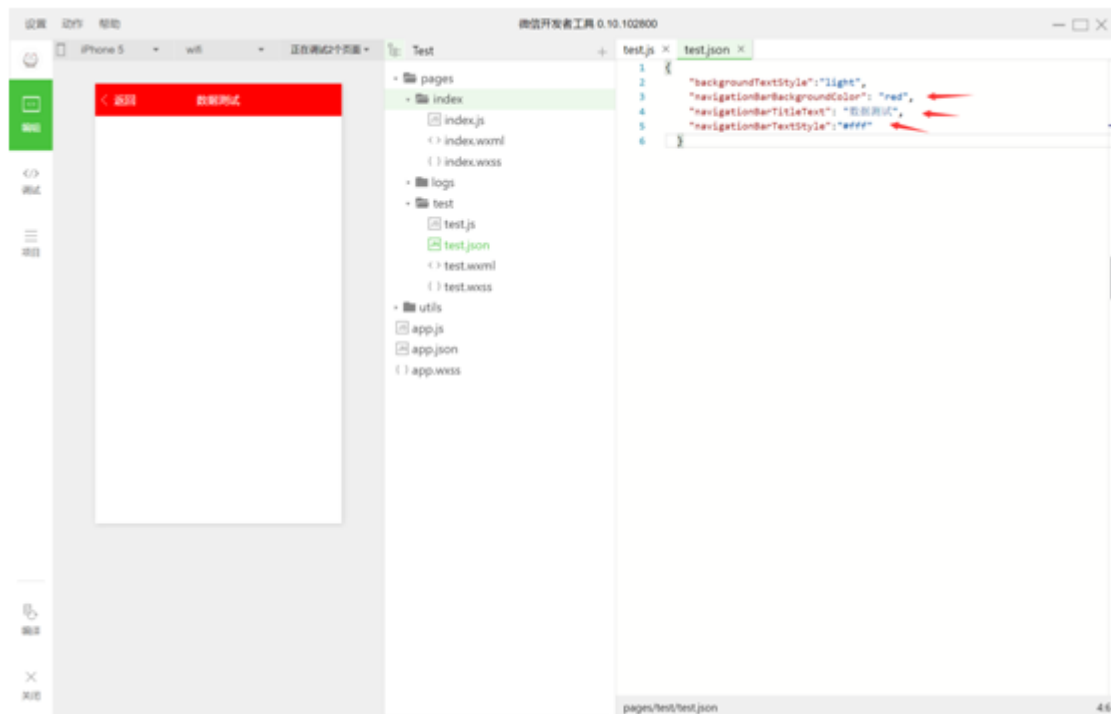
在 pages 下新建页面 test 注意新建的.js,.json,.xml,.wxss。和 page 下定义的 test 文件夹名称保持一致，微信微信小程序会自动读取这些文件，并生成小程序实例。



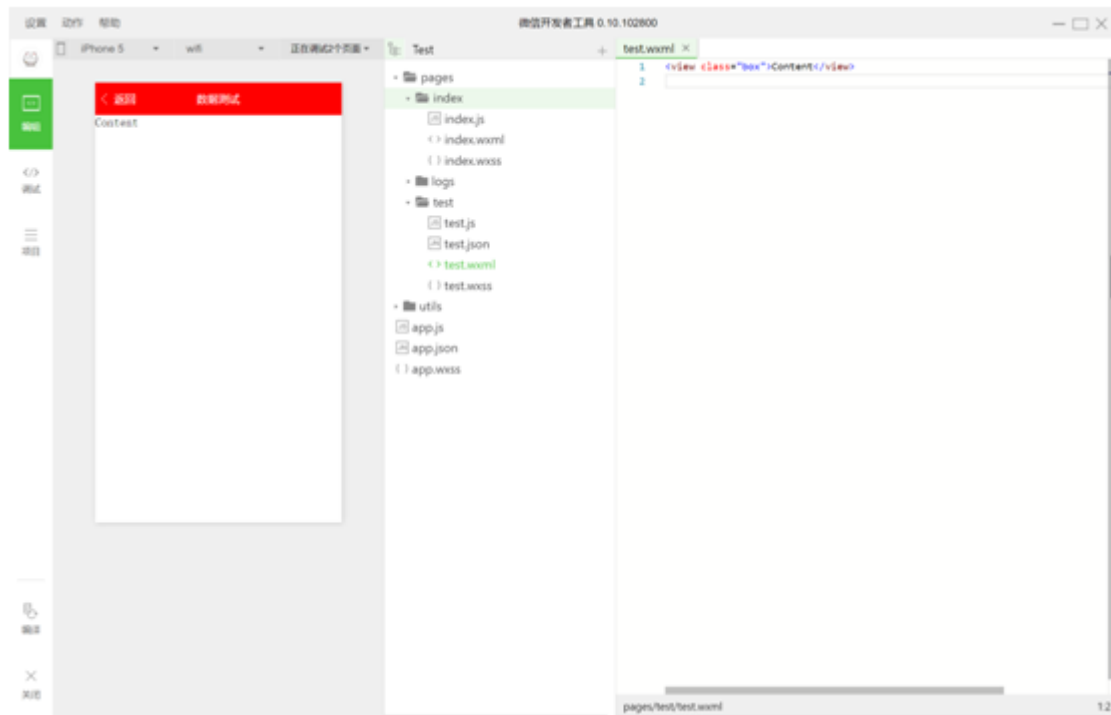
首先我们要在 app.json 下配置好我们刚新建的页面

注意:页面配置先后的顺序，在实际操作中发现，第一个配置的为首页。

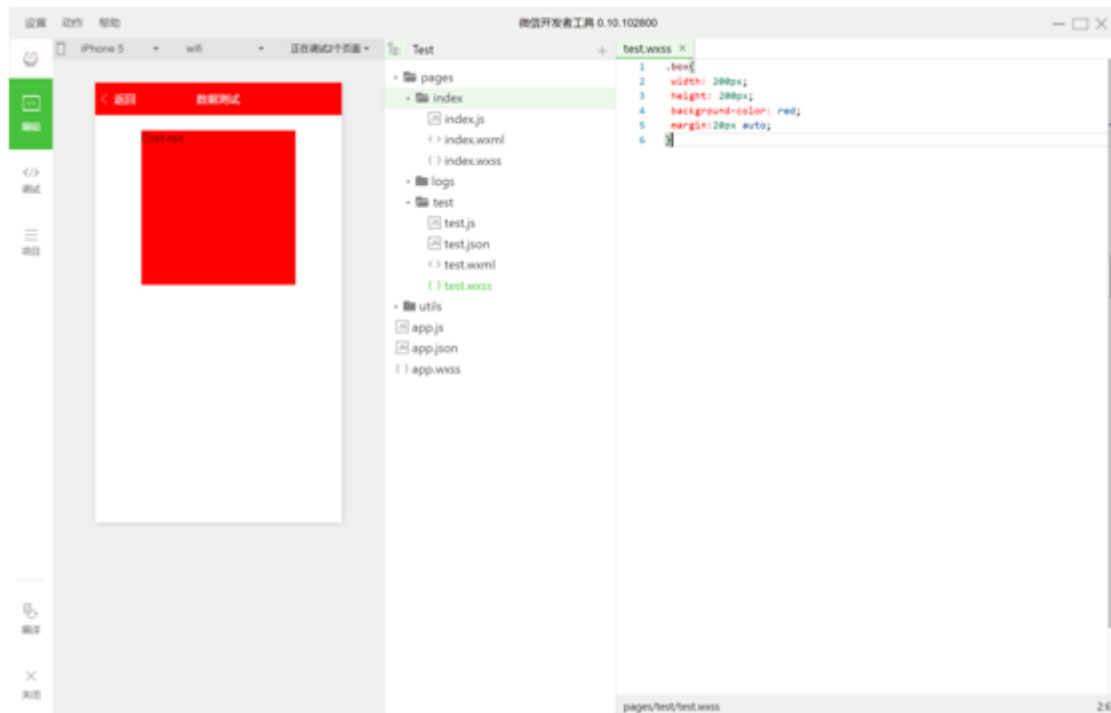
配置好后我们就可以编写我们的页面代码。



大家可以看到，我们在 app.json 页面可以直接配置当前页面的一些 window 属性



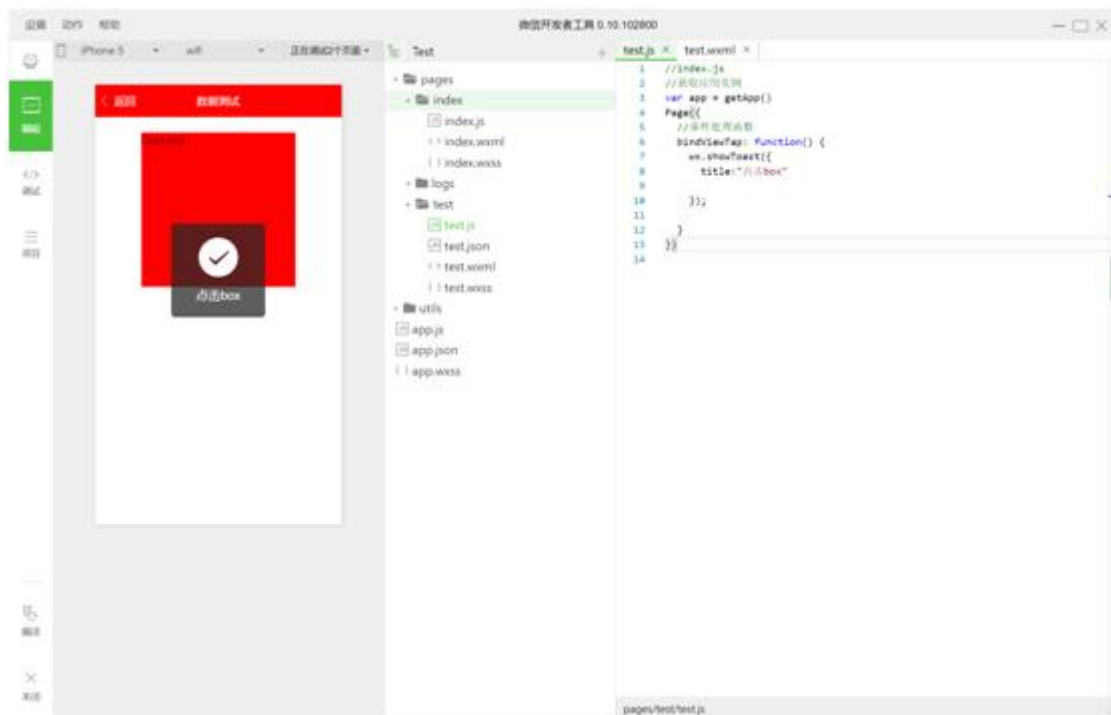
Wxml 中我们可以编写一些我们页面展示的结构，也就是我们以前写 html 一样。这中间我们使用到了 `<view>` 标签，他就相当于我们以前用的 div 差不多。



Wxss 大家看到是不是和我们以前写的 css 差不多。这边是个我们之前定义的 .box 加一个样式。

test.js 是页面的脚本文件，在这个文件中我们可以监听并处理页面的生命周期函数、获取小程序实例，声明并处理数据，响应页面交互事件等。

```
<view class="box" bindtap="bindViewTap">Content</view>
```



这边我们使用 bintap 给 view 绑一个点击事件然后再弹出一个提示框。

注意：这边 js 的写法和我们以前的写法完全一样，只不过这里面不能使用 window 对象和 document 对

象，同样的也不能使用 jquery ,zepto 等框架，因为这些框架中会使用到 window 和 document 对象。

好啦！今天只是给大家简单介绍下小程序的目录结构和简单的使用了下，下一节我们会来开始我们商城

系统的编写。