

## 零基础入门篇：微信小程序开发博客项目《中》

1、添加一些图片素材（网上找的），复制到项目的根目录下面，和 **pages** 目录同级



2、查看[微信](#)文档介绍



目录结构

配置

逻辑层

注册程序

注册页面

模块化

API

视图层

WXML

数据绑定

条件渲染

列表渲染

模板

事件

# Page

Page() 函数用来注册一个页面。接受一个 object 参数，其指定页面的初始数据、生命周期函数、事件

object 参数说明:

里面默认注册好

属性	类型	描述
data	Object	页面的初始数据
onLoad	Function	生命周期函数--监听页面加载
onReady	Function	生命周期函数--监听页面初次渲染完成
onShow	Function	生命周期函数--监听页面显示
onHide	Function	生命周期函数--监听页面隐藏
onUnload	Function	生命周期函数--监听页面卸载
onPullDownRefresh	Function	页面相关事件处理函数--监听用户下拉动作
其他	App	开发者可以添加任意的函数或数据到 object

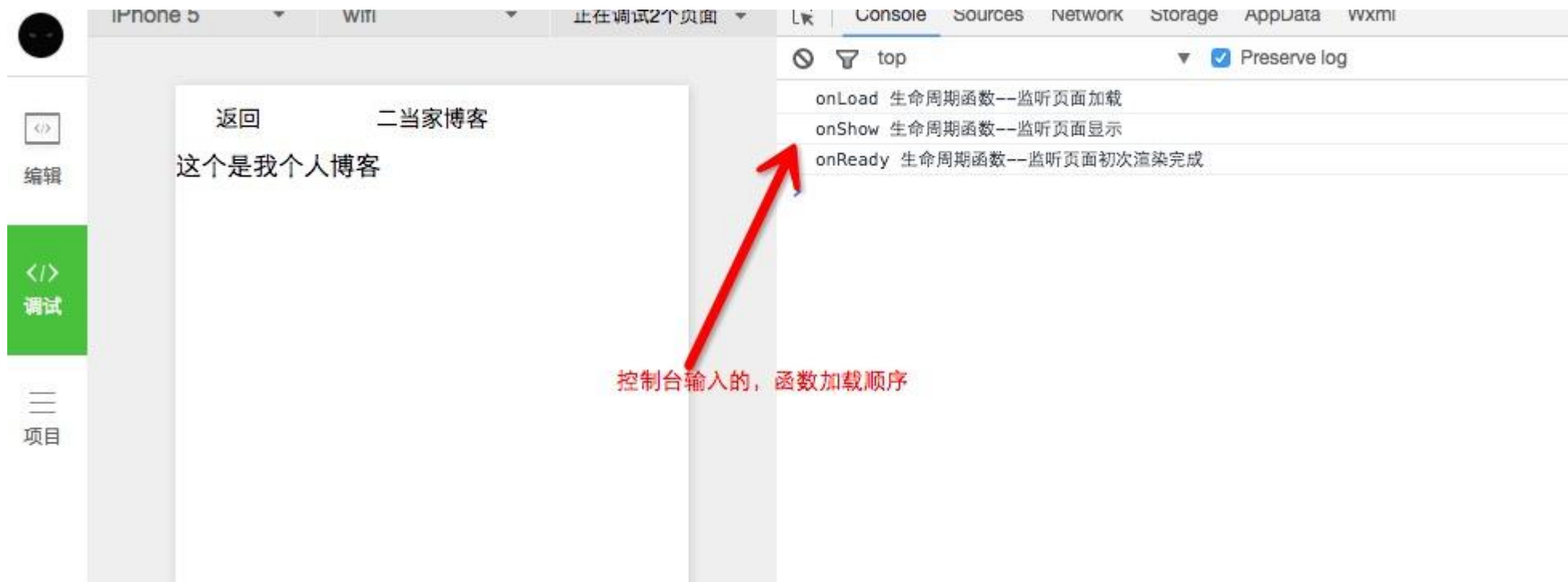
3、打开 blog 文件夹，编辑 blog.js，page 函数里面添加对象，data 对象表示数据，即和后台交互的数据都在这里存储和展示，onLoad,onReady 等函数是微信在页面加载时默认存在的函数，我们可以重写，实现我们自己的逻辑



The screenshot displays the WeChat DevTools IDE interface. On the left, the 'Project' (项目) sidebar shows the file structure of a mini-program named '小程序博客' (Mini-program Blog). The 'pages' directory is expanded, showing the 'blog' folder. Inside 'blog', the files 'blog.js' (JavaScript), 'blog.json' (JSON), 'blog.wxml' (WXML), and 'blog.wxss' (WXSS) are listed. The 'index' folder is also visible. The main editor area shows the 'blog.js' file, which contains the following code:

```
1 //获取应用实例
2 var app = getApp()
3
4 Page({
5   data: {
6     //data里面是对象数据，展示层从从这里取数据
7   },
8   onLoad: function () {
9     console.log('onLoad 生命周期函数--监听页面加载')
10    var that = this
11    //调用应用实例的方法获取全局数据
12
13  },
14  onReady: function(){
15    console.log('onReady 生命周期函数--监听页面初次渲染完成')
16  },
17  onShow: function(){
18    console.log('onShow 生命周期函数--监听页面显示')
19  },
20  onHide: function(){
21    console.log('onHide 生命周期函数--监听页面隐藏')
22  },
23  onUnload: function(){
24    console.log('onUnload 生命周期函数--监听页面卸载')
25  }
26 })
27
```

4、点击编译查看对应的效果，及函数加载顺序, **onLoad**、**onShow** 、**onReady** 的加载顺序



5、网络请求

打开微信文档，里面有对应的网络请求 api，`wx.request()`，接收一个对象，包含我们请求的各种参数（交互的域名正式环境有限制，而且需要为 **https**）

网络

发起请求

上传、下载

wx.uploadFile

wx.downloadFile

WebSocket

wx.connectSocket

wx.onSocketOpen

wx.onSocketError

wx.sendSocketMessage

wx.onSocketMessage

wx.closeSocket

wx.onSocketClose

媒体

图片

wx.chooseImage

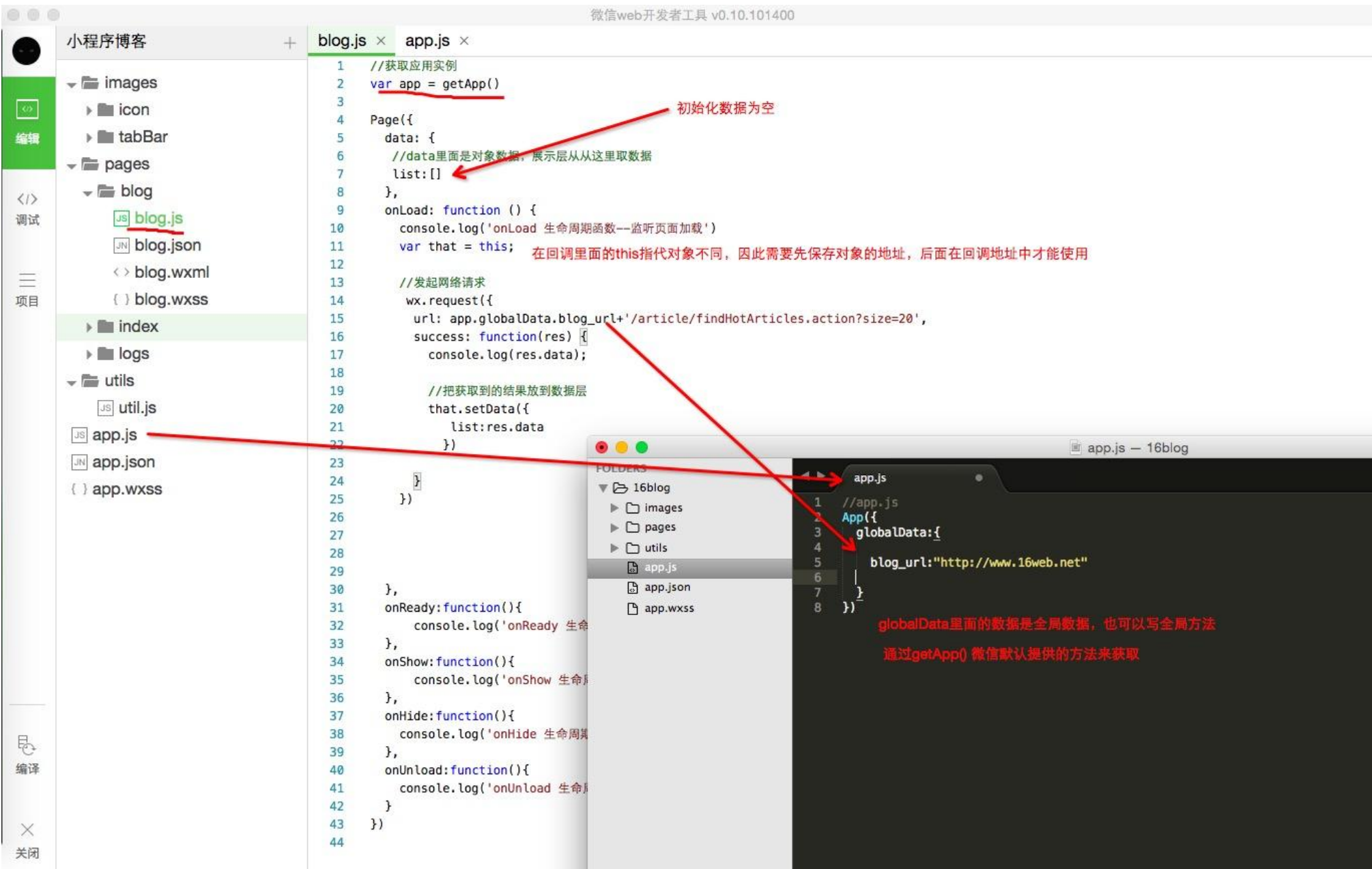
## wx.request(OBJECT)

wx.request 发起的是https请求。一个微信小程序，同时只能有5个网络请求连接。 本地开发无https限制

### OBJECT参数说明：

参数名	类型	必填	说明
url	String	是	开发者服务器接口地址 正式环境需要在后台配置安全域名
data	Object、String	否	请求的参数
header	Object	否	设置请求的 header，header 中不能设置 Referer
method	String	否	默认为 GET，有效值：OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT
success	Function	否	收到开发者服务成功返回的回调函数，res = {data: '开发者服务器返回的内容'}
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

6、编辑 blog.js，在 onLoad 函数里面发起网络请求，接口地址是我测试用的地址，会返回对应的数据列表，注意的地方是 this 指向问题，因为是在回调函数里面获取数据的，然后需要放到 data 对象里面，所以需要提前保存 this 对象的地址



现在进入页面就可看到控制台打印接收到的数据了（数据请求也可以这样发送）



```
1. //发起网络请求
2.   wx.request({
3.     url: app.globalData.blog_url+'/article/findHotArticles.action',
4.     data:{
5.       size:20
6.     },
7.     success: function(res) {
8.       console.log(res.data);
9.
10.      //把获取到的结果放到数据层
11.      that.setData({
12.        list:res.data
13.      })
14.
15.    }
16.  })
```

7、ok，现在就是展示数据的时候了,打开微信微信官方文档，主要使用的是 **view** 标签，详情参考文档

视图容器

view

scroll-view

swiper

基础内容

icon

text

progress

表单组件

button

checkbox

form

input

label

picker

radio

scroll-view

可滚动视图区域。

用可滚动的view标签,上面有多种熟悉可以使用到,如下拉刷新,和上拉刷新

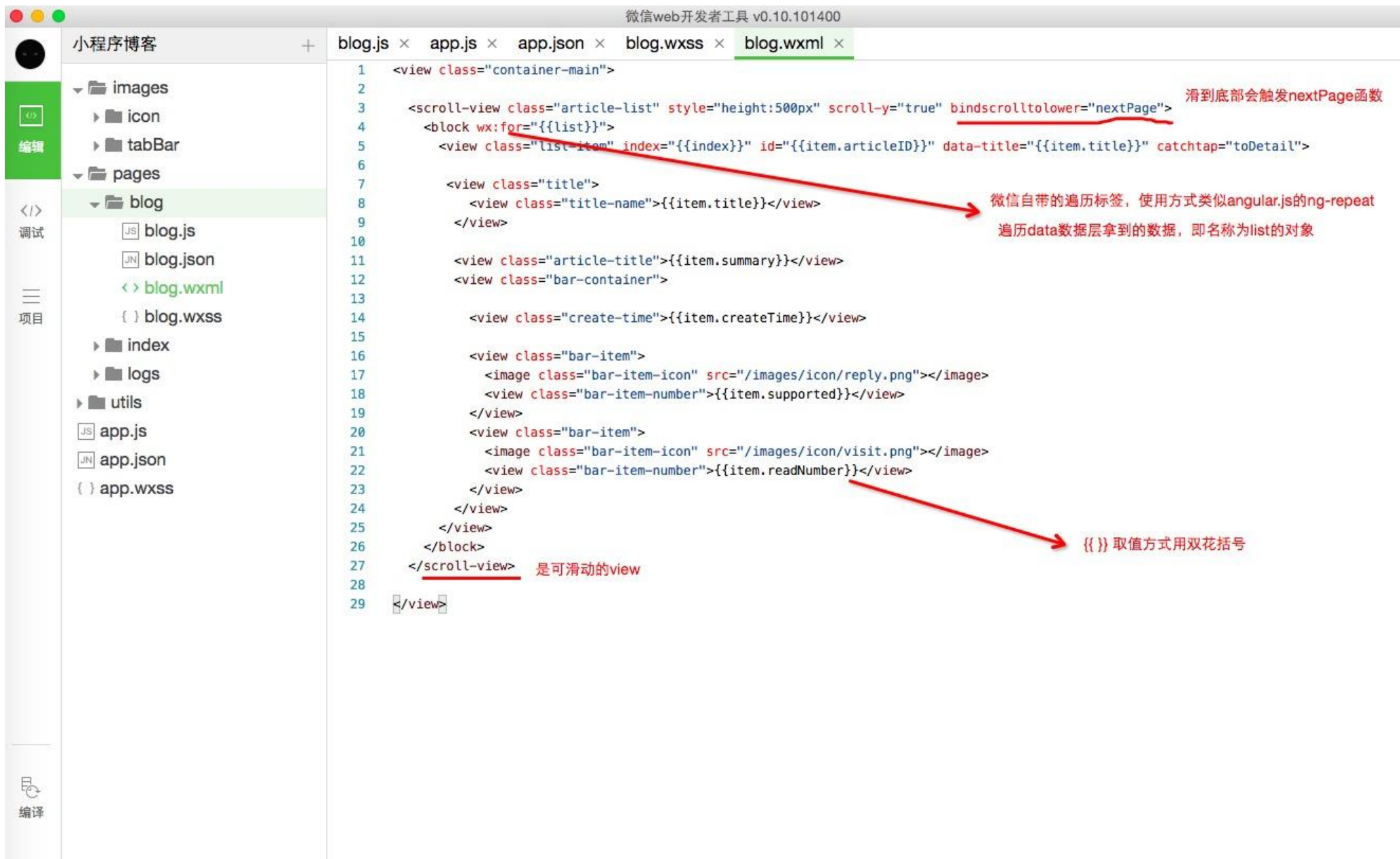
属性名	类型	默认值	说明
scroll-x	Boolean	false	允许横向滚动
scroll-y	Boolean	false	允许纵向滚动
upper-threshold	Number	50	距顶部/左边多远时 (单位px), 触发 scrolltoupper 事件
lower-threshold	Number	50	距底部/右边多远时 (单位px), 触发 scrolltolower 事件
scroll-top	Number		设置竖向滚动条位置
scroll-left	Number		设置横向滚动条位置
scroll-into-view	String		值应为某子元素id, 则滚动到该元素, 元素顶部对齐滚动区域顶部
bindscrolltoupper	EventHandle		滚动到顶部/左边, 会触发 scrolltoupper 事件
bindscrolltolower	EventHandle		滚动到底部/右边, 会触发 scrolltolower 事件

上拉刷新

下拉刷新

8、展示层代码, 使用<block wx:for="{{list}}"> 遍历文章对象, {{}} 取的是 data 数据层里面的数据





绑定的下拉函数（不做处理）

小程序博客

+

blog.js ×

app.js ×

app.json ×

blog.wxss ×

blog.wxml ×

images

icon

tabBar

pages

blog

blog.js

blog.json

blog.wxml

blog.wxss

index

logs

utils

app.js

app.json

app.wxss

&lt;/&gt;

调试

三

项目

编译

编译

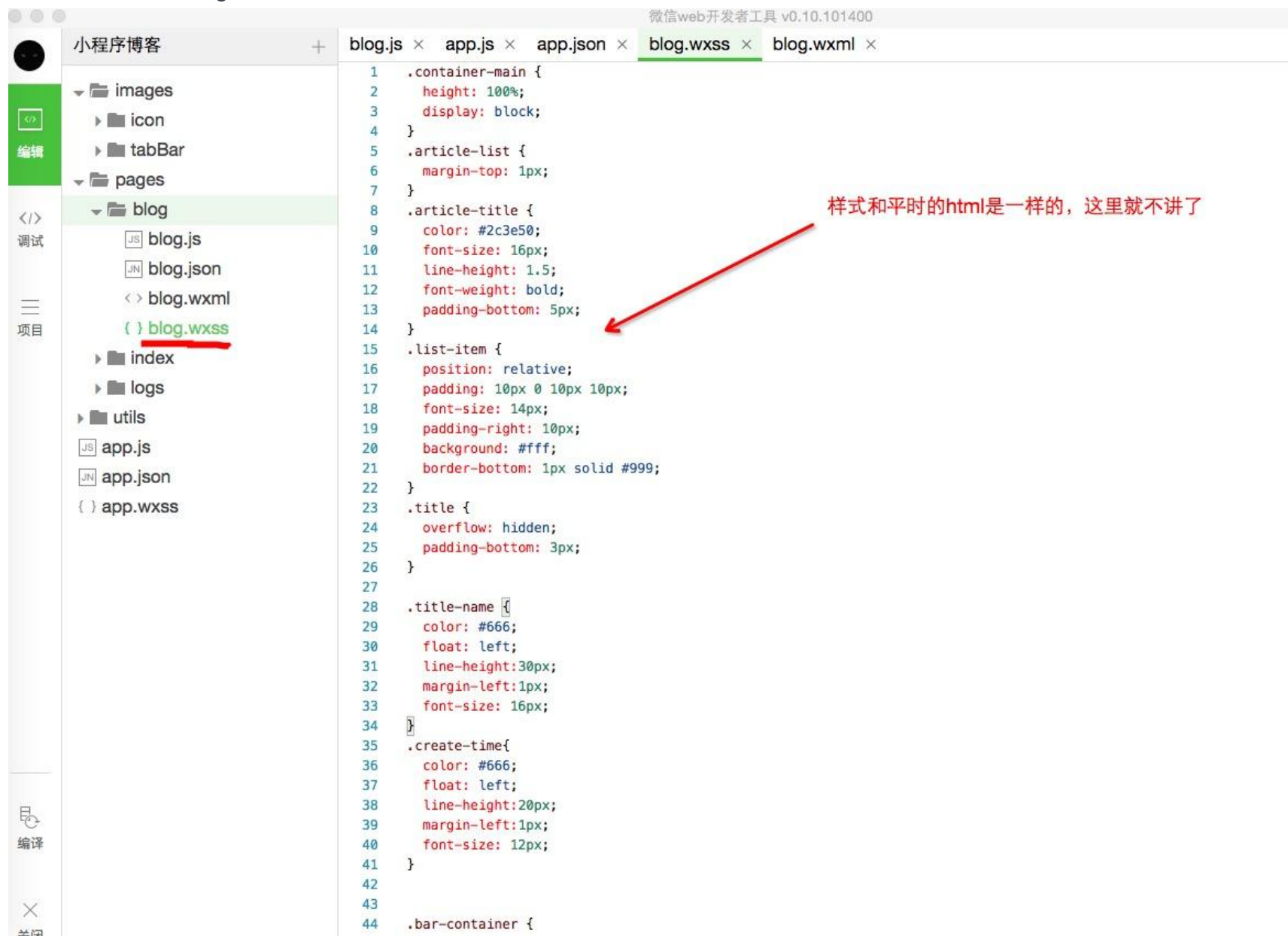
X

关闭

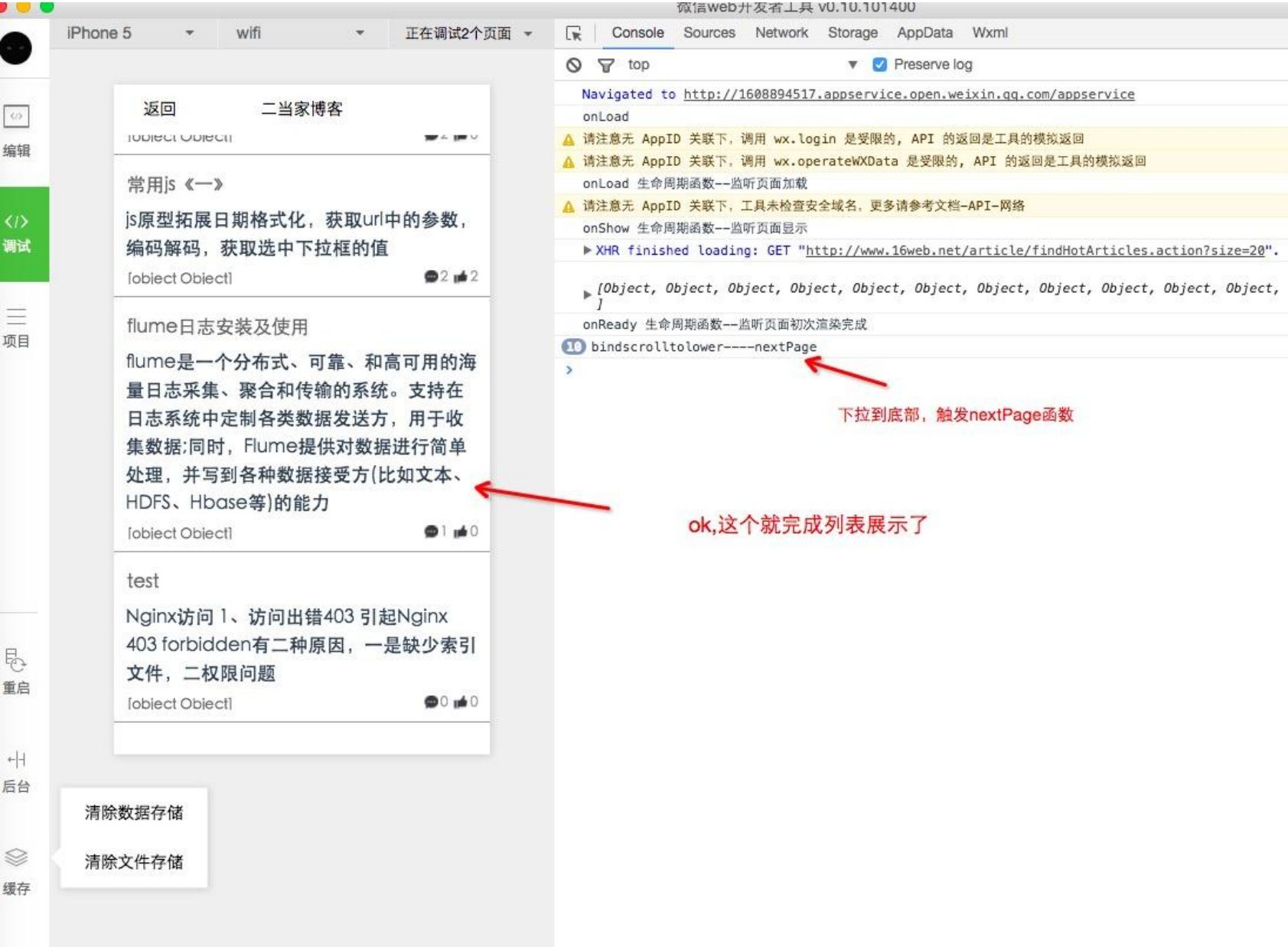
```
5  data: {
6    //data里面是对象数据，展示层从从这里取数据
7    list:[]
8  },
9  onLoad: function () {
10   console.log('onLoad 生命周期函数--监听页面加载')
11   var that = this;
12
13   //发起网络请求
14   wx.request({
15     url: app.globalData.blog_url+'article/findHotArticles.action',
16     data:{
17       size:20
18     },
19     success: function(res) {
20       console.log(res.data);
21
22       //把获取到的结果放到数据层
23       that.setData({
24         list:res.data
25       })
26
27     }
28   })
29
30   },nextPage:function(){
31     console.log("bindscrolltolower----nextPage");
32   },
33   onReady:function(){
34     console.log('onReady 生命周期函数--监听页面初次渲染完成')
35   },
36   onShow:function(){
37     console.log('onShow 生命周期函数--监听页面显示')
38   },
39   onHide:function(){
40     console.log('onHide 生命周期函数--监听页面隐藏')
41   },
42   onUnload:function(){
43     console.log('onUnload 生命周期函数--监听页面卸载')
44   }
45
46
47 })
48
```

下拉函数

10、处理样式，view 标签可以使用 css3 的样式，并且微信本身也提供一些计算尺寸的方法，也支持样式选择器，类选择器，id 选择器，直接看代码，编辑我们的 blog.wxss



11、现在就可以保存，然后点击左侧的编译，就完成了列表展示



已经可以见世人了，码字不容易，篇幅较长，请留意第三篇

