# 微信小程序\_03

# 今日目标6

* 自定义一个组件
* 能够发起一个有效的网络请求 wx.request
* 能够实现列表页面带参数跳转 ?id=1&aaa=bbb
* 能够设置和移除本地缓存
* 能够了解wx.login登录获取openid
* 能够发起POST请求添加数据

# 一、自定义组件

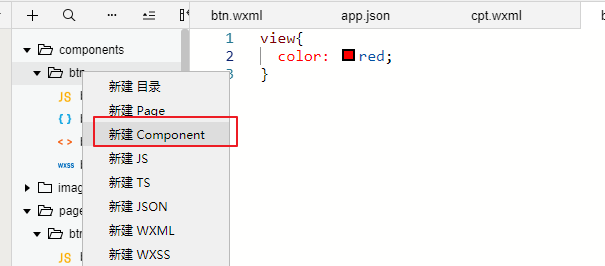
<https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/>

开发者可以将页面内的功能模块抽象成自定义组件，以便在不同的页面中重复使用；也可以将复杂的页面拆分成多个低耦合的模块，有助于代码维护。自定义组件在使用时与基础组件非常相似。

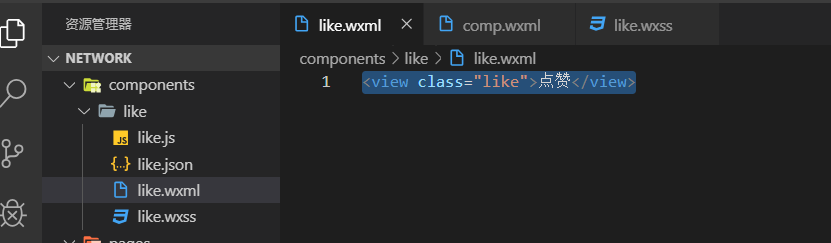
说白了就是自定义一些标签，然后这些标签能帮助我们实现一些功能且可以重复的使用。

## 3.1、创建自定义组件

打开官方开发工具，自定义目录后右击菜单，选择如下









## 3.2、使用自定义组件

使用已注册的自定义组件前，首先要在页面的 **json** 文件中进行引用声明。此时需要提供每个自定义组件的标签名和对应的自定义组件文件路径：

{

"usingComponents": {

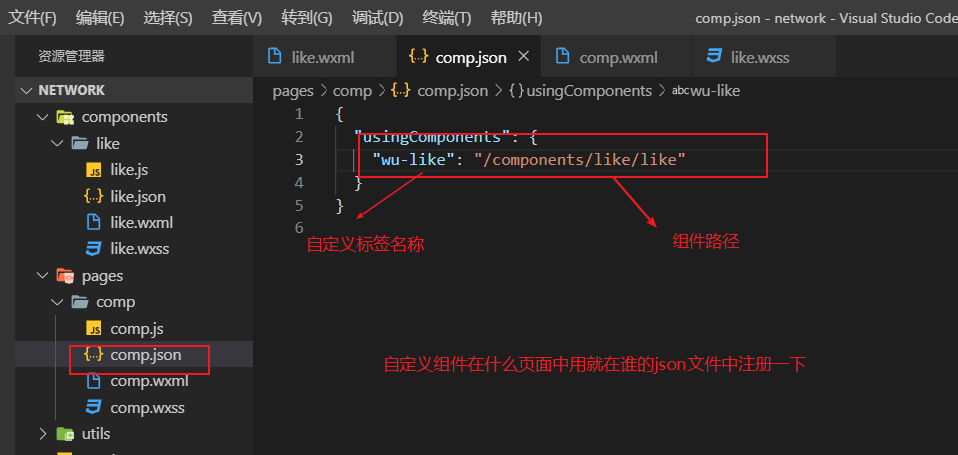
// 在父模板中调用的标签名 组件的路径

"component-tag-name": "path/to/the/custom/component"

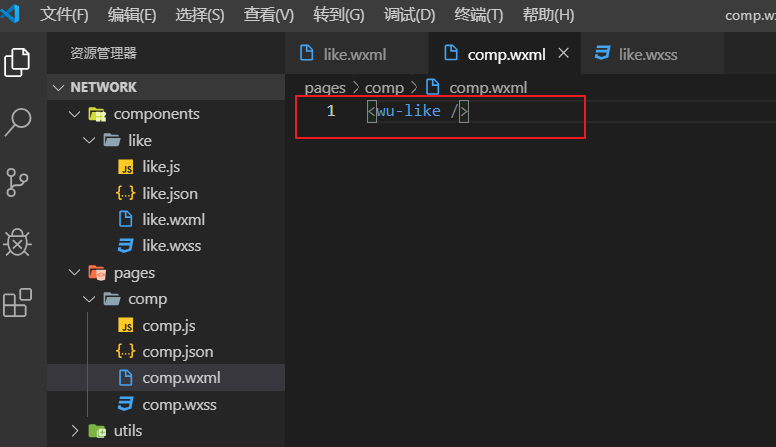
}

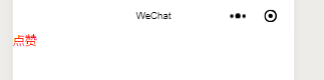
}

调用



使用



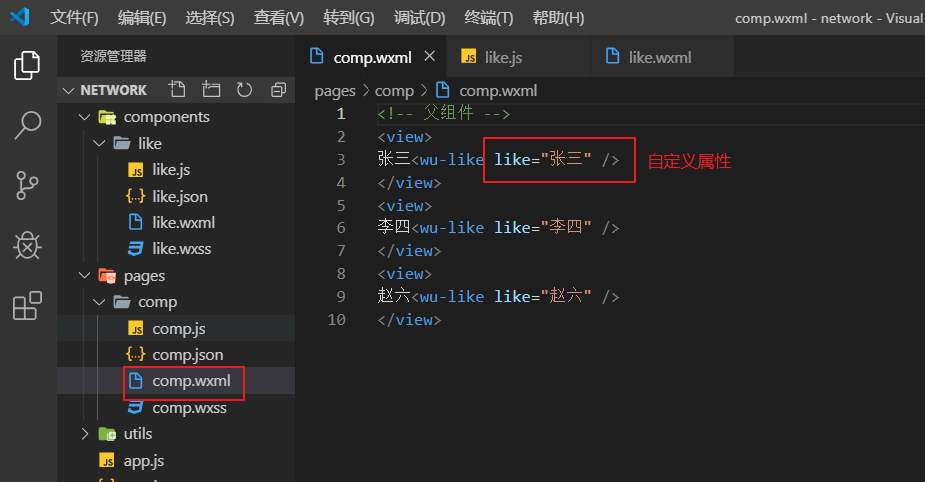


### 3.2.1、父子组件通信

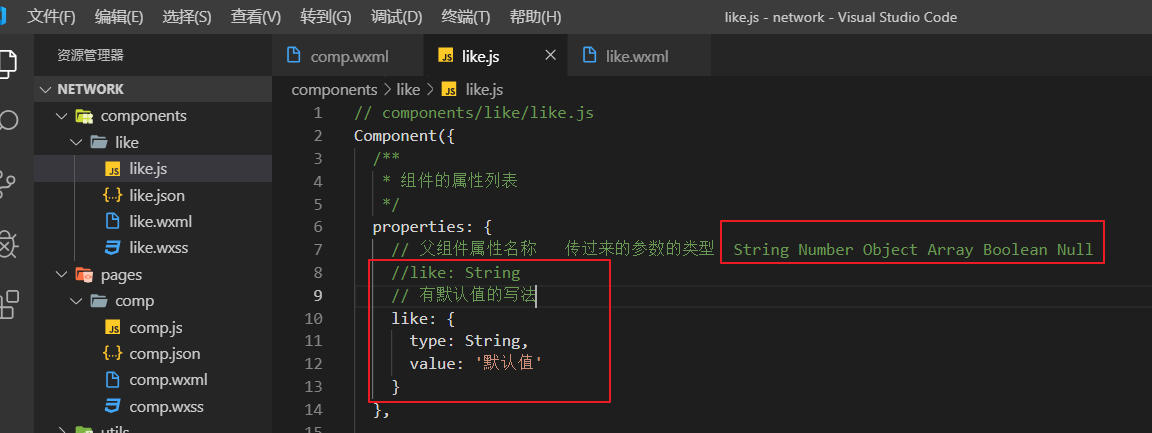
父组件向子组件通信

小程序中，父组件是通过 属性的方法和子组件通信 properties

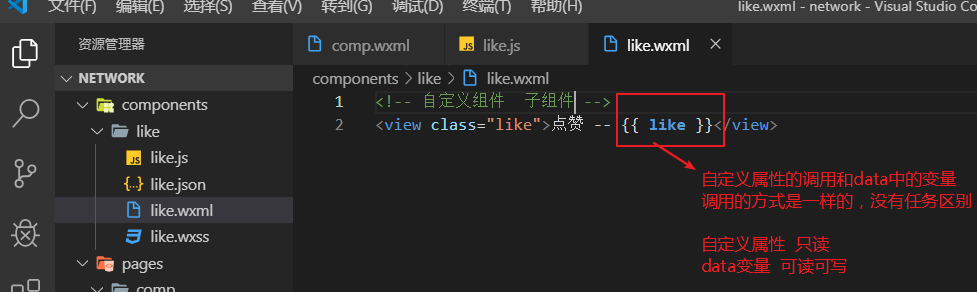
父组件中自定义一个属性



子组件中的js中申明，父组件中的属性名称和类型



在自定义组件中wxml中使用接收到的信息



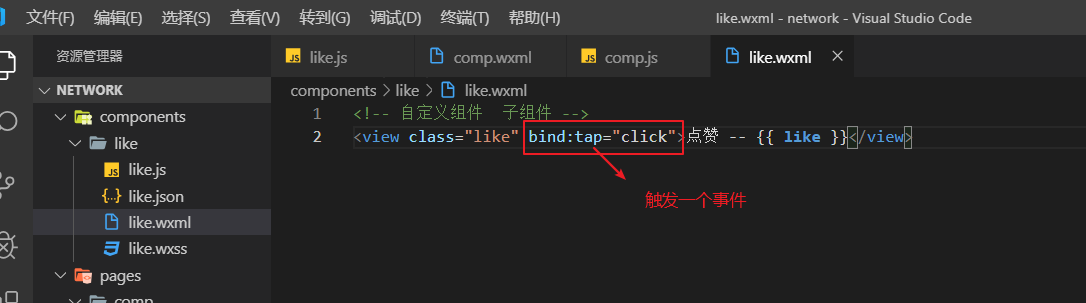
效果



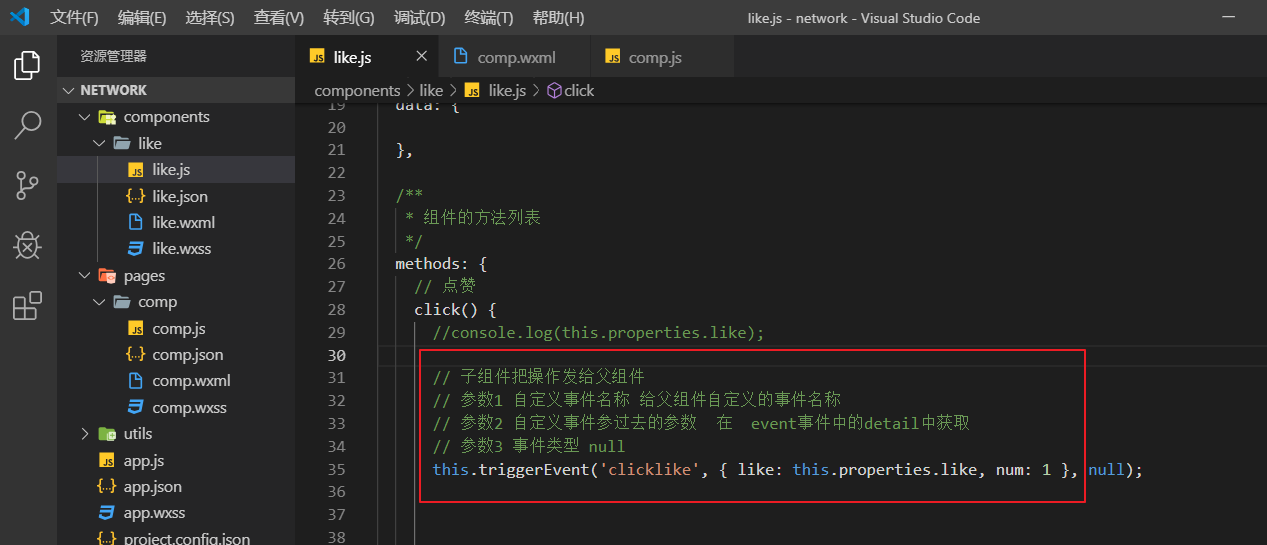
* 子组件向父组件通信

小程序使用 自定义事件来实现

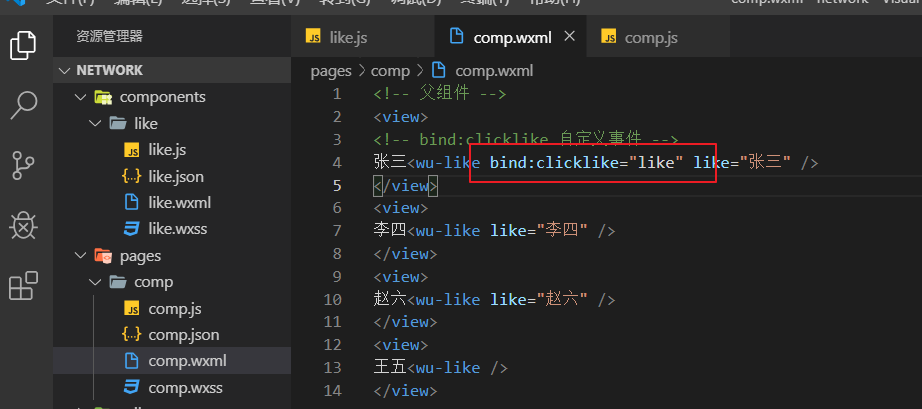
组件的wxml中



在组件js中自定义一个事件



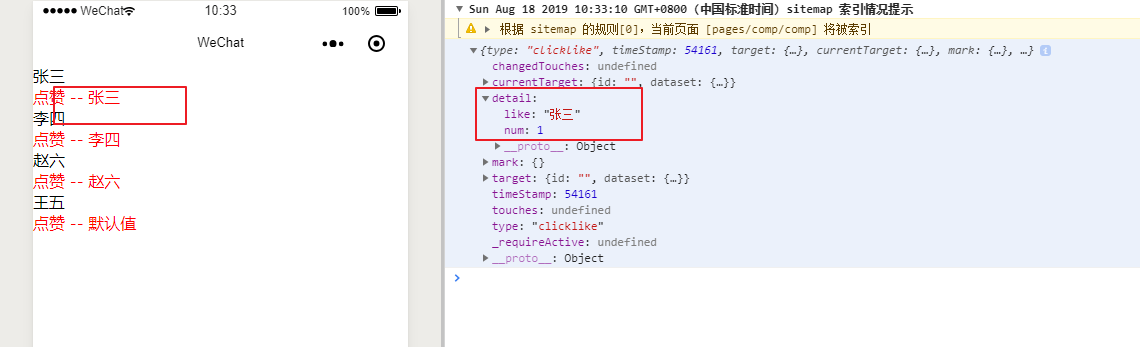
在父组件中使用自定义事件



在父组件中实现事件方法



效果



# 二、网络请求

<https://developers.weixin.qq.com/miniprogram/dev/api/network/request/wx.request.html>

wx.request(OBJECT)

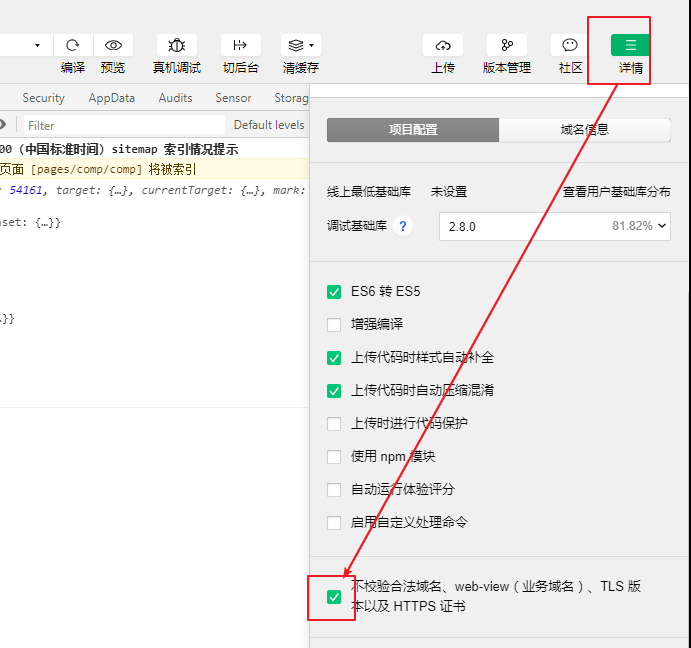
发起 HTTPS 网络请求,上线时一定要有https服务器，但在本机开发时可以没有https服务器

测试支持的并数请求数5个

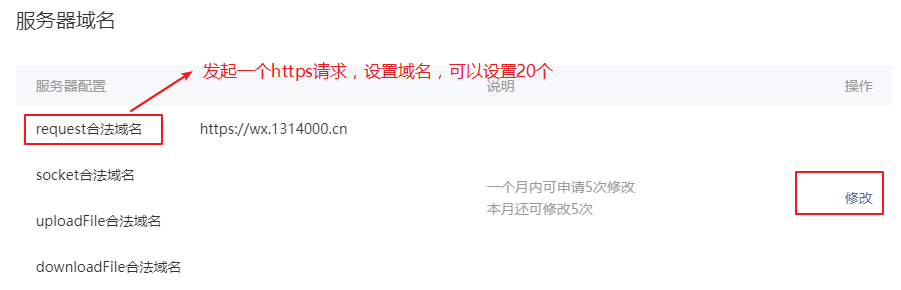
HTTPS 证书必须有效 而且是TLS1.2及以上版本

最好要设置一下 app.json中request请求的超时时间

如果是测试环境开发时，可以在小程序官方开发工具中勾选不检查https证书



正式环境，一定需要一个https服务器，同时还需要在小程序官方后台设置有效的域名



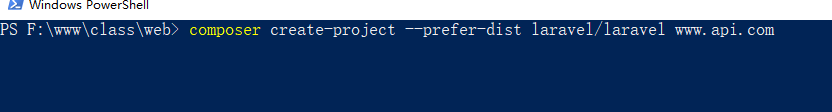


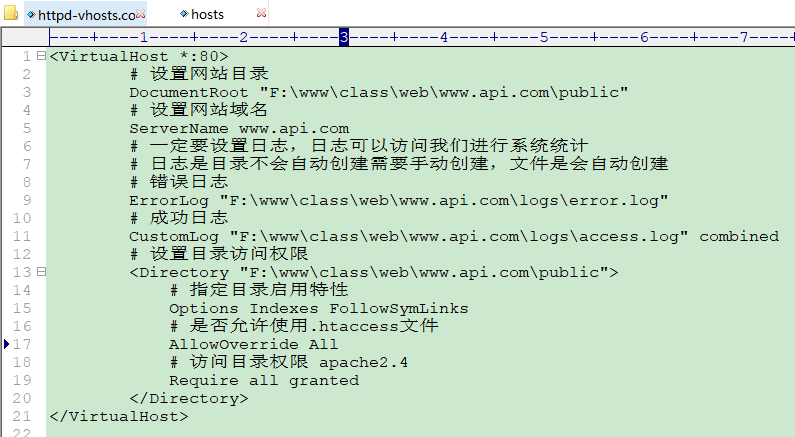
## 2.1、编写后台接口

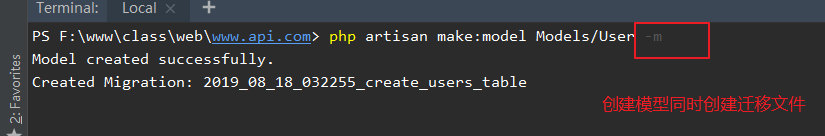
### 2.1.1、用户表设计

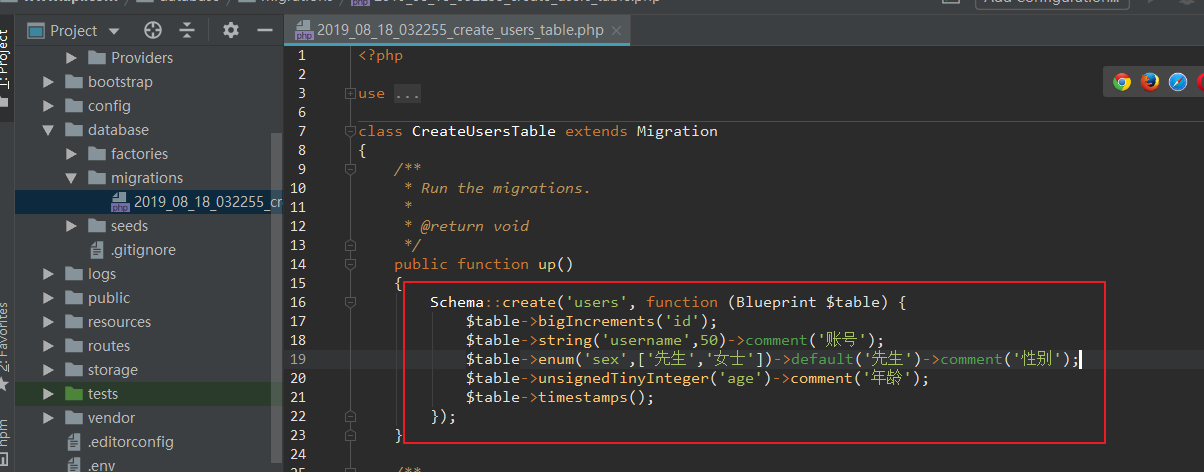
id，姓名，年龄

安装laravel项目

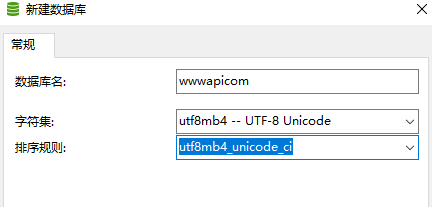




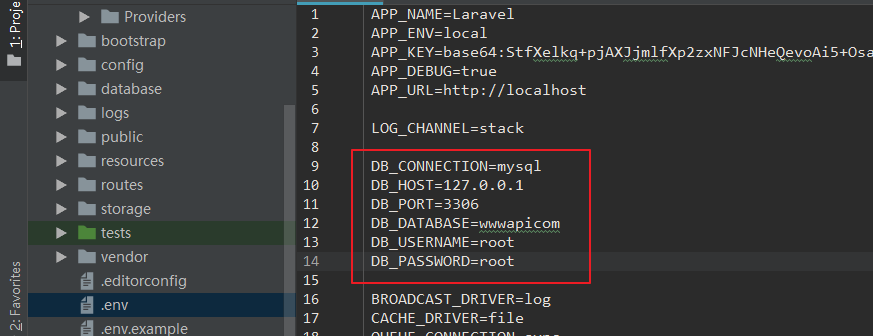




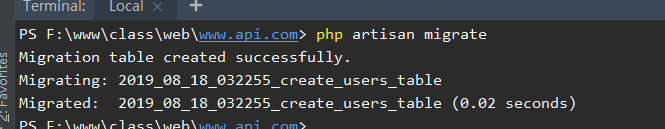
配置数据库



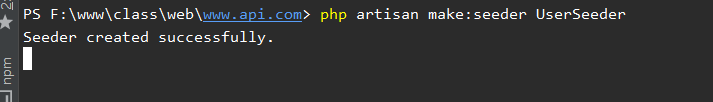
在根目录下面的.env文件设置数据库相关信息



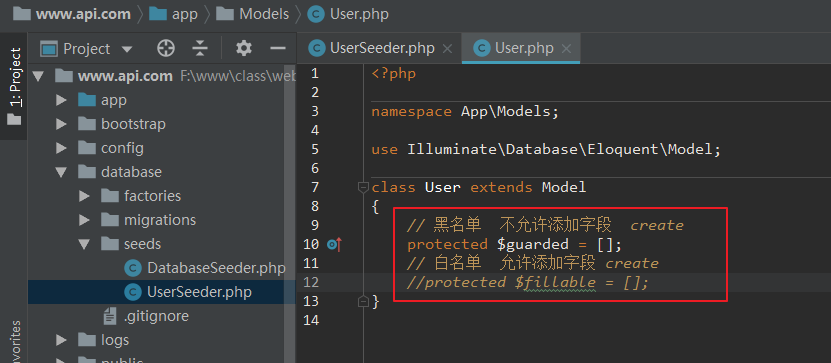
执行迁移



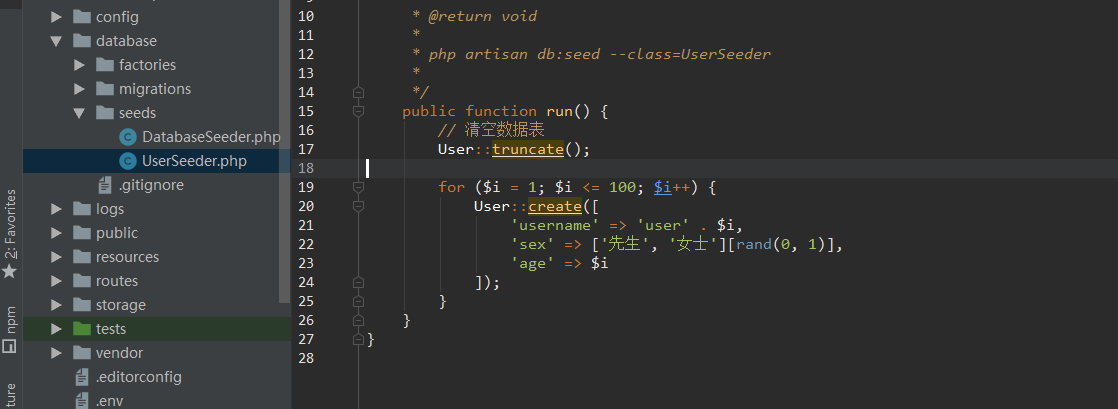
创建填充文件



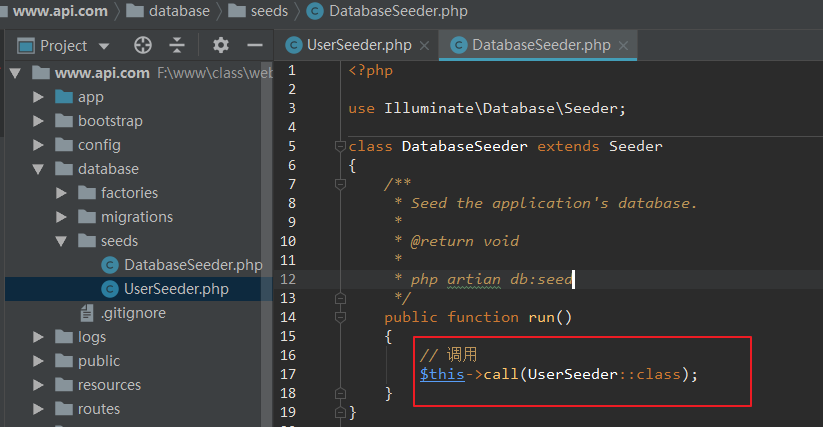
在模型中添加对应的类限制



在填充文件中定义填充数据代码



调用



执行

php artisan db:seed

### 2.1.2、用户列表

使用restful规范编写相对应的接口

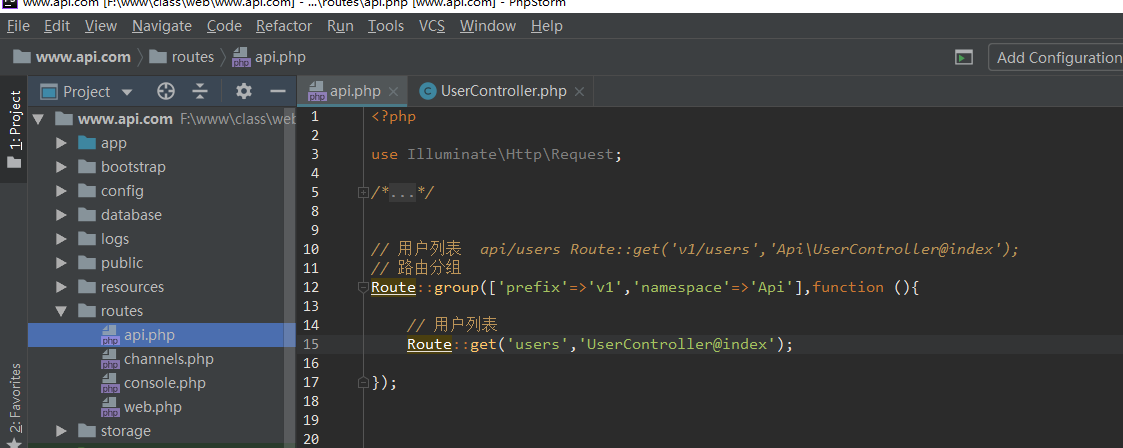
get 获取资源 获取多个资源或单个资源 200

post 添加资源 201

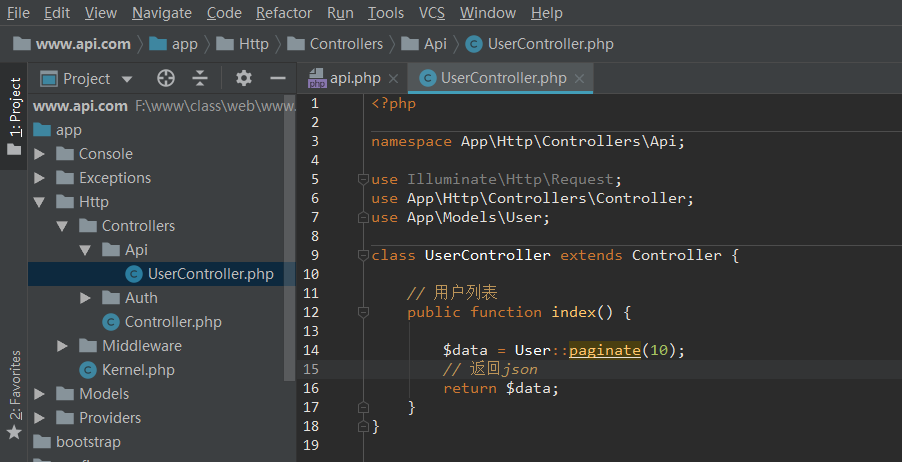
put 修改资源 201/202

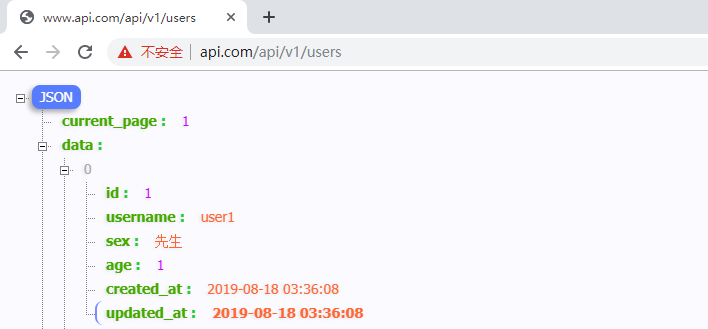
delete 删除资源 204 无响应体

定义get获取资源路由



用户列表分页代码





## 2.2、发起网络请求

wx.request({

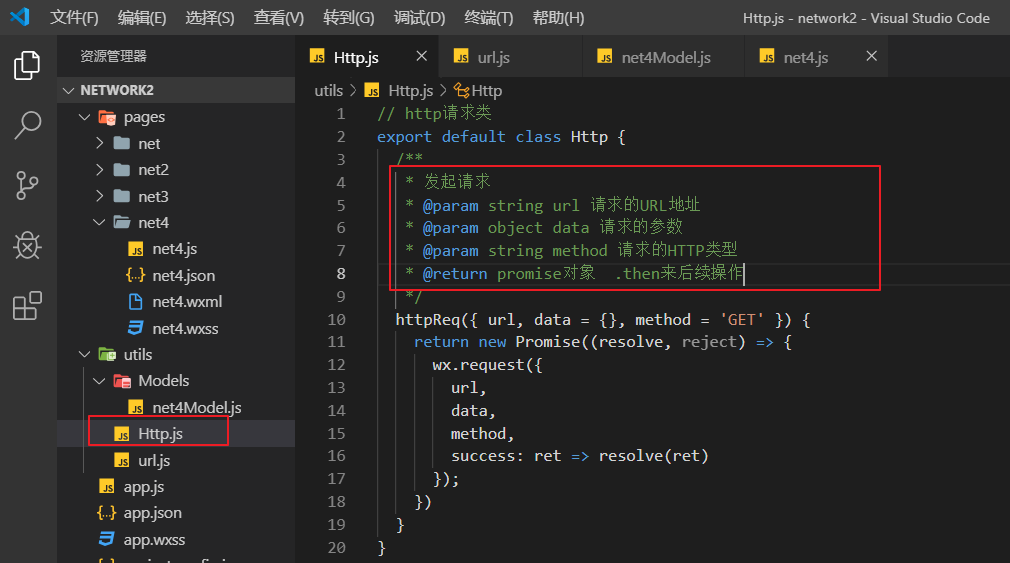
url,

method,

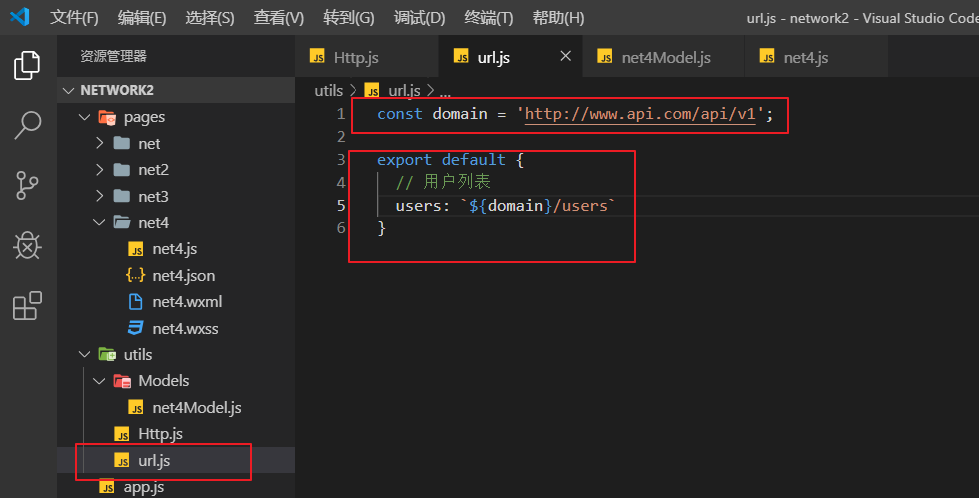
data

})

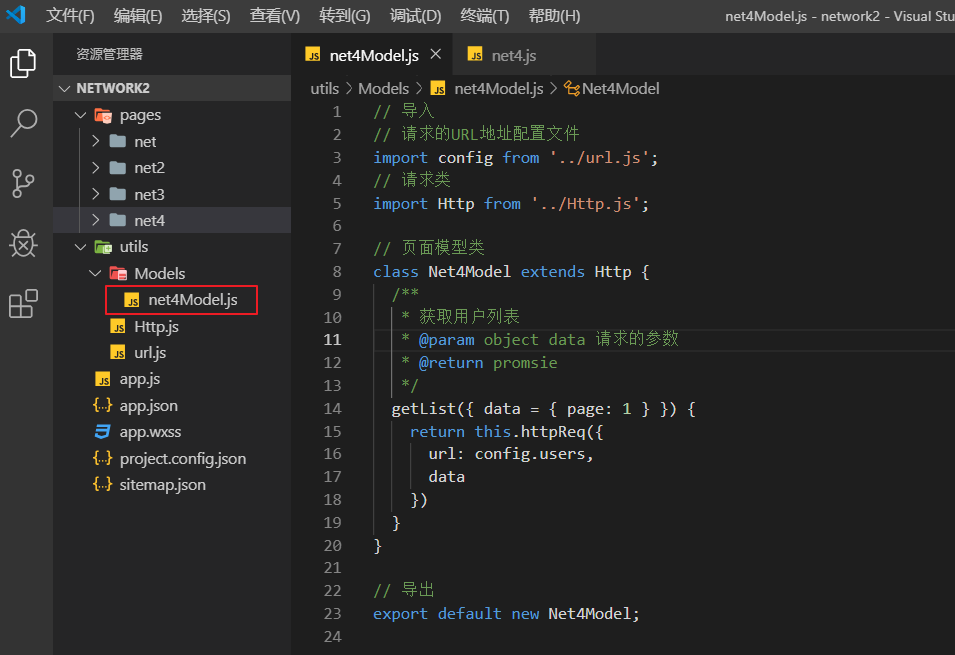
为了解决回调问题，使用 es6中的promise来封装处理方法，为了更的后期维护，又使用了es6模块化来封装请求



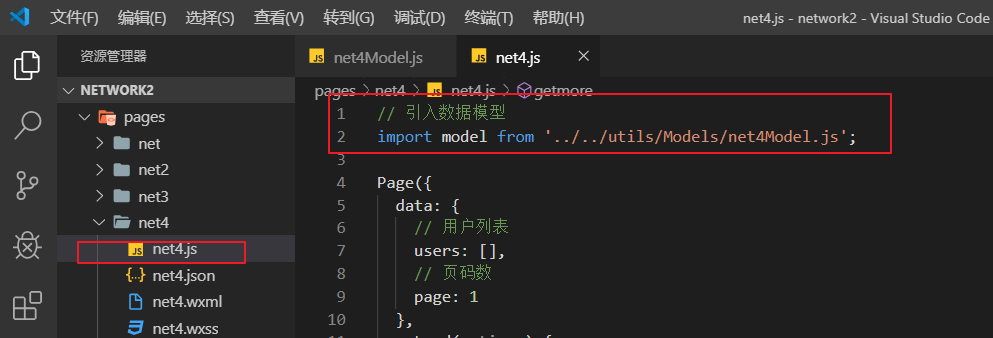
定义一个请求的URL地址的配置文件

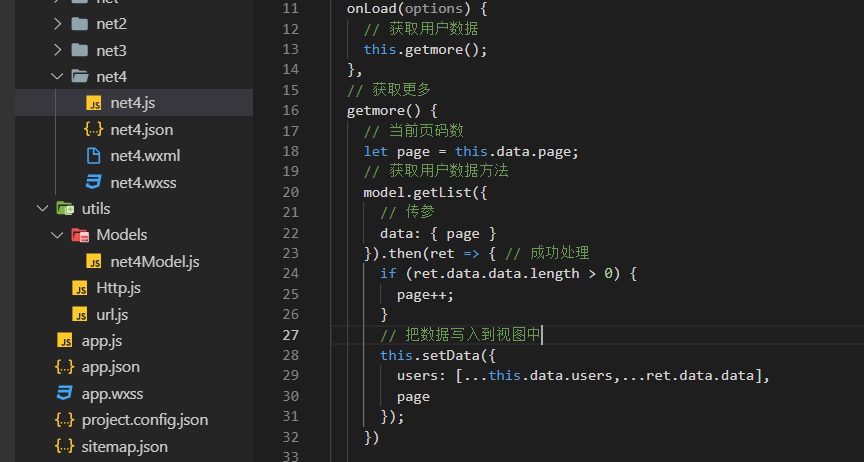


定义一个页面模型类，来获取数据

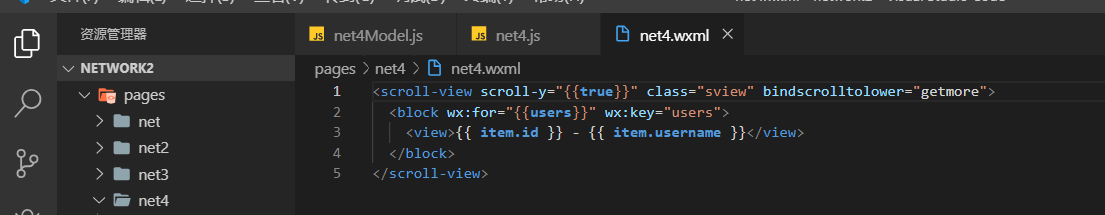


在想要展示的页面中引入对应的模型





wxml展示数据



效果



# 三、数据缓存

每个微信小程序都可以有自己的本地缓存，即对本地缓存进行设置、获取和清理。同一个微信用户，同一个小程序 storage 上限为 10MB。

注意： 如果用户储存空间不足，会清空最近最久未使用的小程序的本地缓存(LRU)。我们不建议将关键信息全部缓存起来，以防储存空间不足或用户换设备的情况。

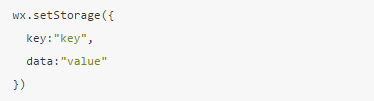
小程序的缓存是没有有效期的

## 3.1、设置缓存

<https://developers.weixin.qq.com/miniprogram/dev/api/storage/wx.setStorageSync.html>

设置缓存分为同步和异步

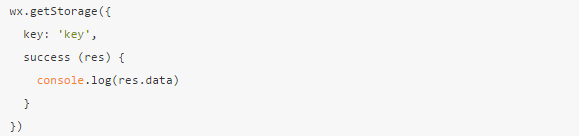
* wx.setStorage(OBJECT) 异步
* wx.setStorageSync(KEY,DATA) 同步



## 3.2、获取本地缓存数据

获取缓存分为同步和异步

* wx.getStorage(OBJECT) 异步
* wx.getStorageSync(KEY) 同步



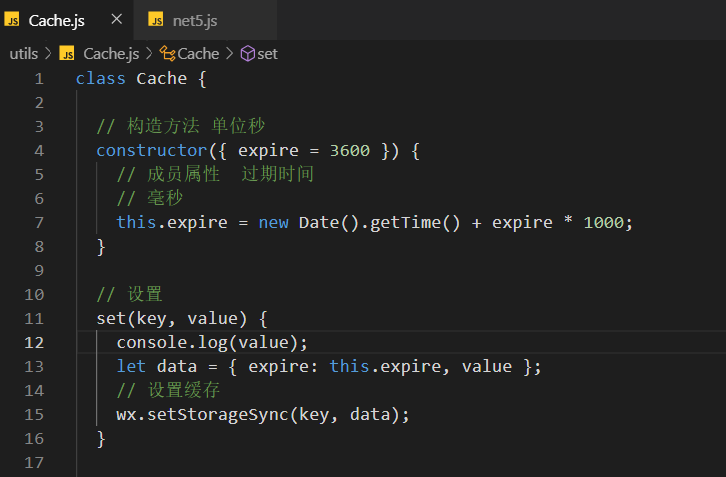
## 3.3、移除本地缓存

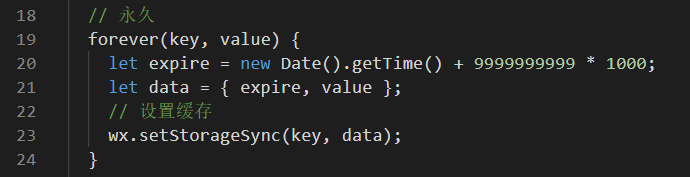
从本地缓存中异步移除指定 key, 分为同步和异步

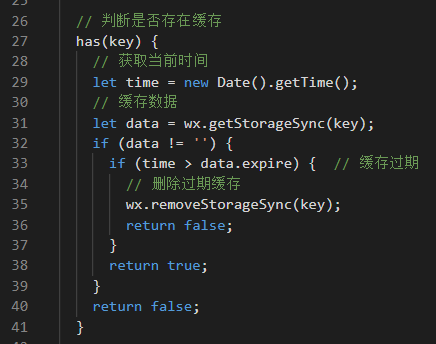
* wx.removeStorage(OBJECT) 异步
* wx.removeStorageSync(KEY) 同步

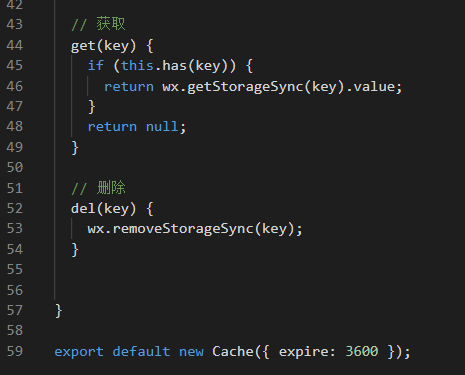
## 3.4、清除本地缓存 --- 不要用

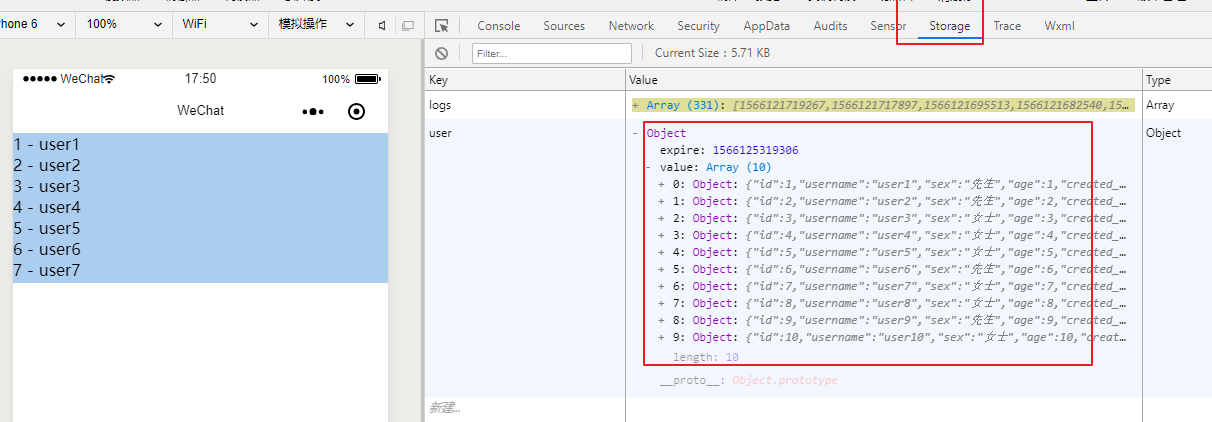
* wx.clearStorage()
* wx.clearStorageSync()





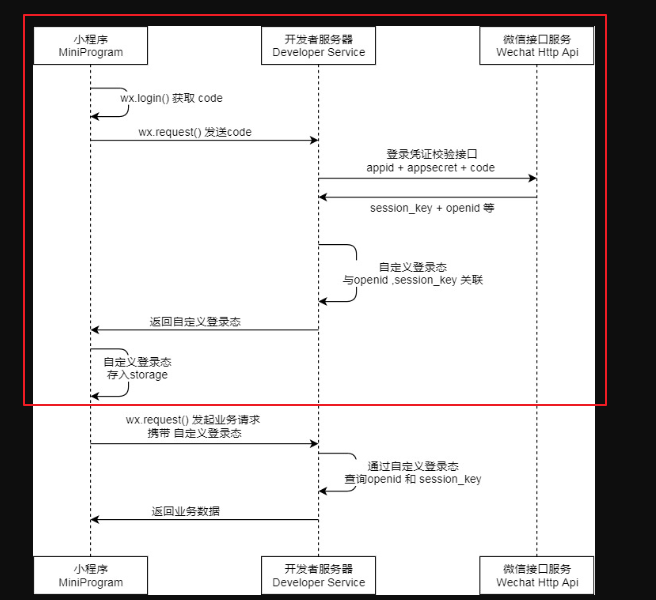






# 四、用户登录

<https://developers.weixin.qq.com/miniprogram/dev/api/wx.login.html>



## 4.1、用户登录

## 4.2、openid入库

## 4.3、用户授权