# 微信小程序\_03

# 今日目标5

* 能够申明一个es6的类

class 类名{

constructor(){} 构造方法

方法(){} 普通方法

}

继承

extends

class 类 extends 类{}

* 能够了解常用的http请求类型

get post put delete

* 能够发起一个有效的网络请求

wx.request({…})

用到了promise

wx.request 并发5个

请求上线时候一定要是 https

* 能够设置和移除本地缓存

同步

大家一定要完成我们的写的缓存类

* 能够显示和隐藏消息提示框
* laravel api接口

php artisan make:resource 接口资源 单个

php artisan make:resource –c 集合 集合

# 一、申明es6的js类

* class关键字，和构造方法(constructor)
* class里面直接加方法

class User{

// 构造方法 非必须，如果有成员属性则一定要写,因为它要进行成员属性初始化

constructor(name,pass){

// 成员属性

this.name = name;

this.pass = pass;

}

// 成员方法

showName(){

alert(this.name);

}

showPass(){

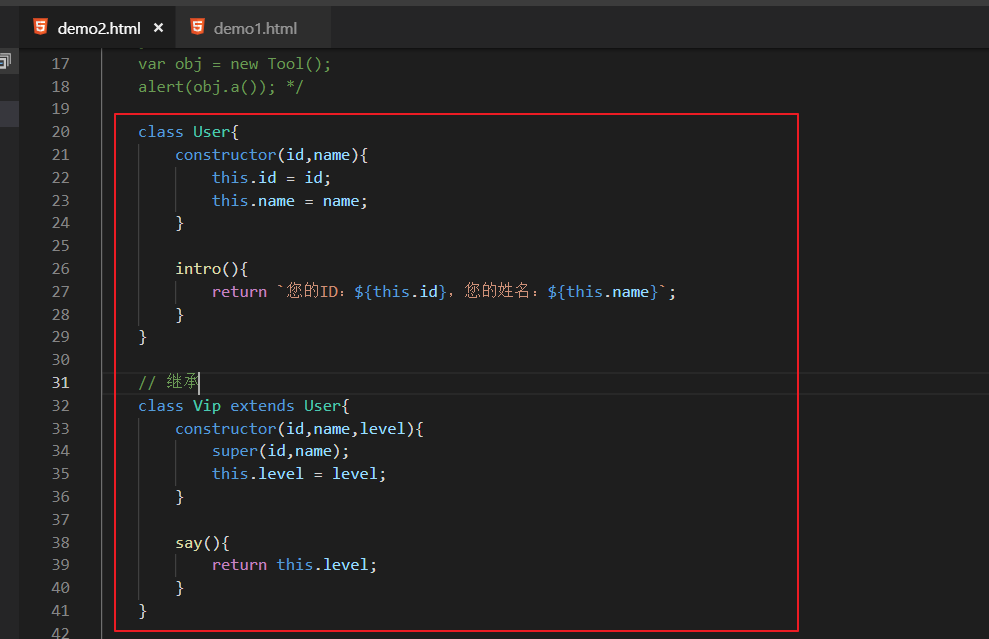
alert(this.pass);

}

}

var user = new User('admin','admin888');

user.showName();



# 二、常用小程序Api

## 2.1、发起网络请求

wx.request(OBJECT)

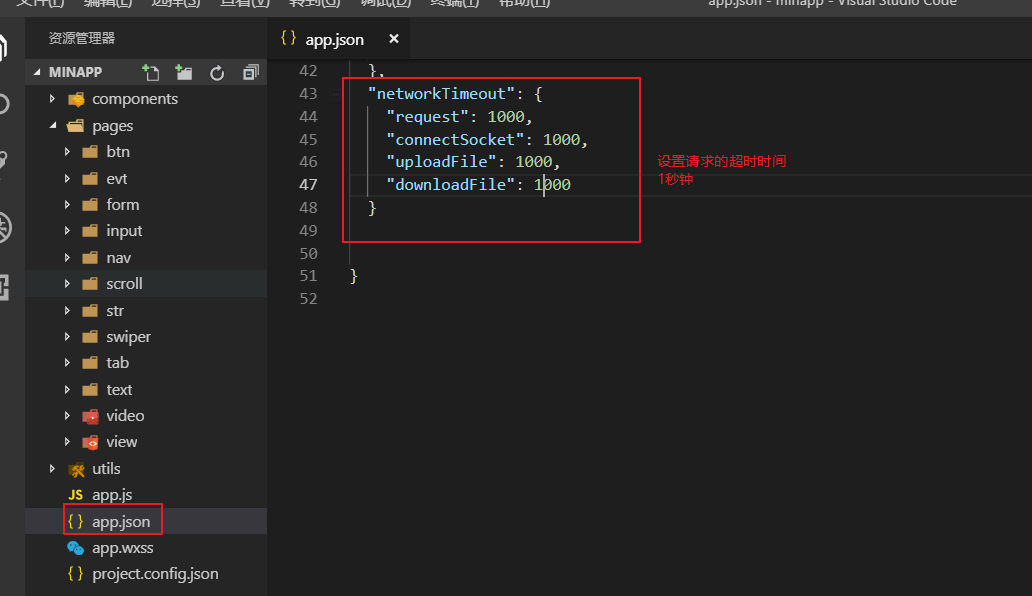
发起 HTTPS 网络请求

测试支持的并数请求数5个

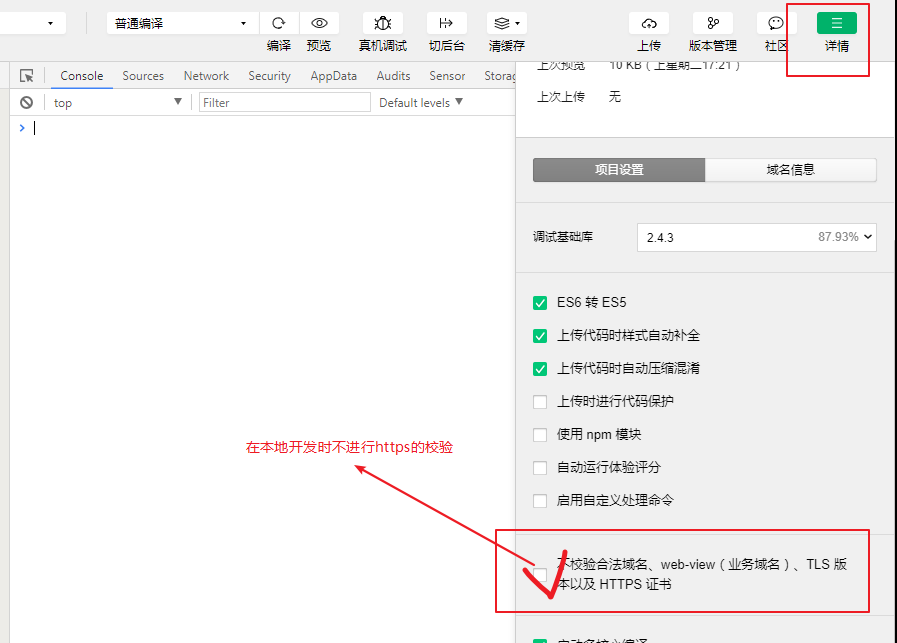
HTTPS 证书必须有效 而且是TLS1.2及以上版本

最好要设置一下 app.json中request请求的超时时间

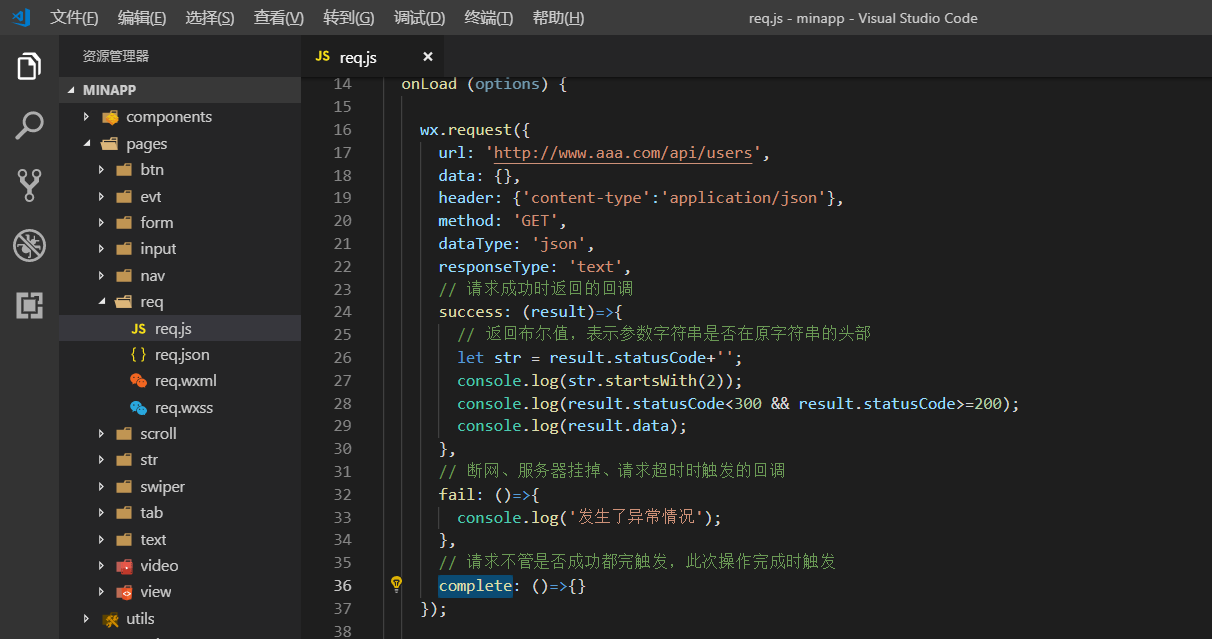
### 2.1.1、app.json文件中设置超时时间

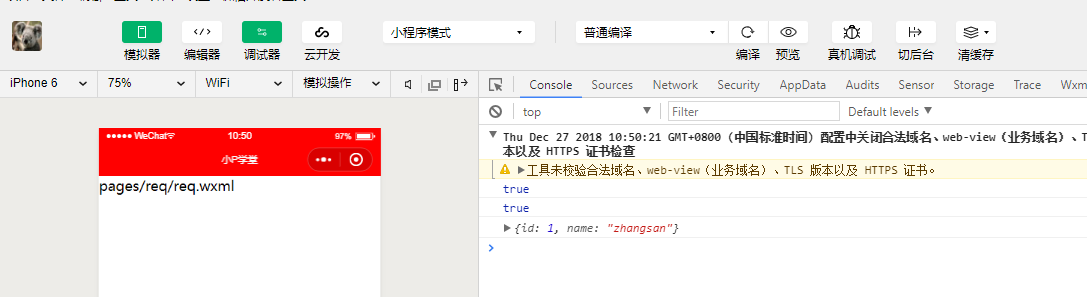


### 2.1.2、本地开发开启允许http请求

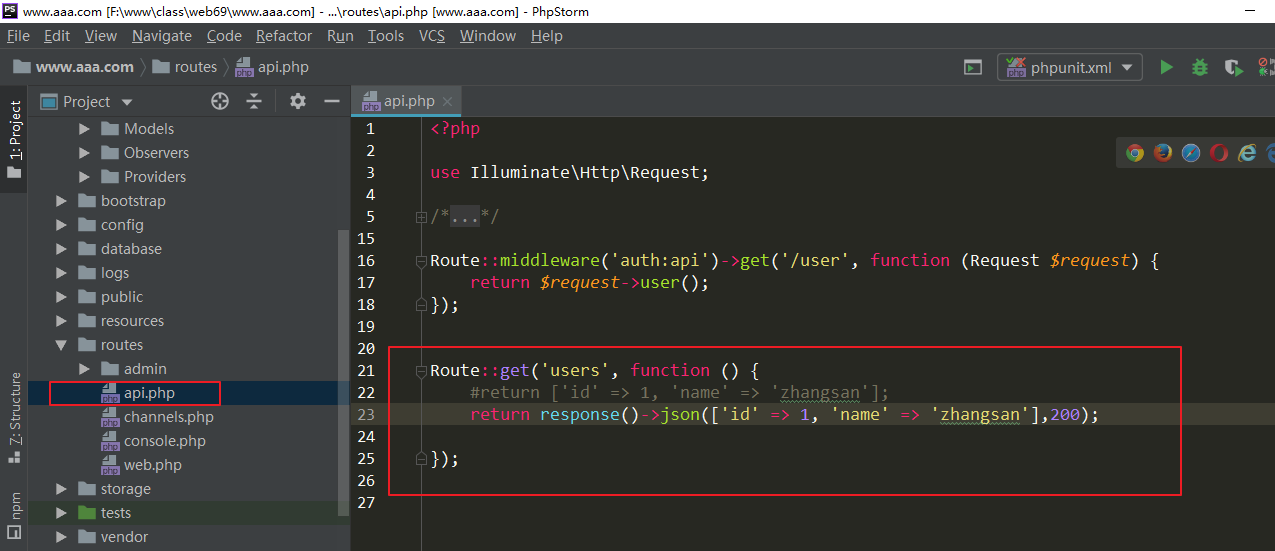


### 2.1.3、编写代码



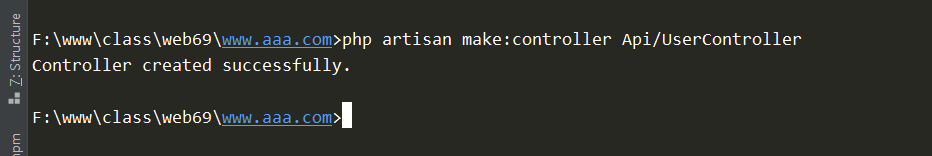


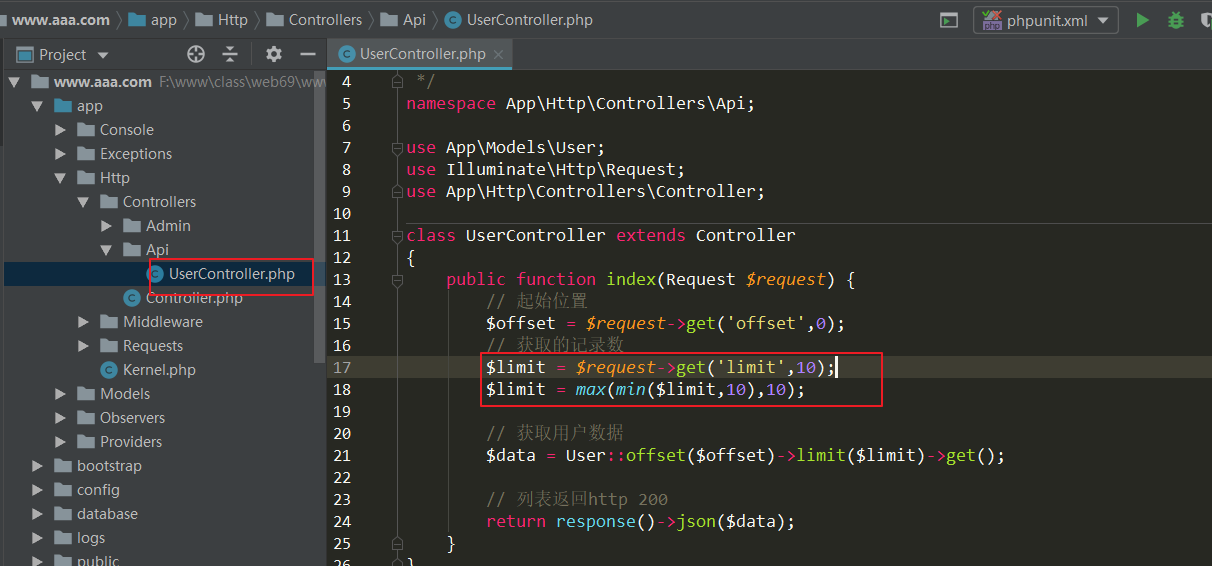
服务器端



### 2.1.4、用户列表

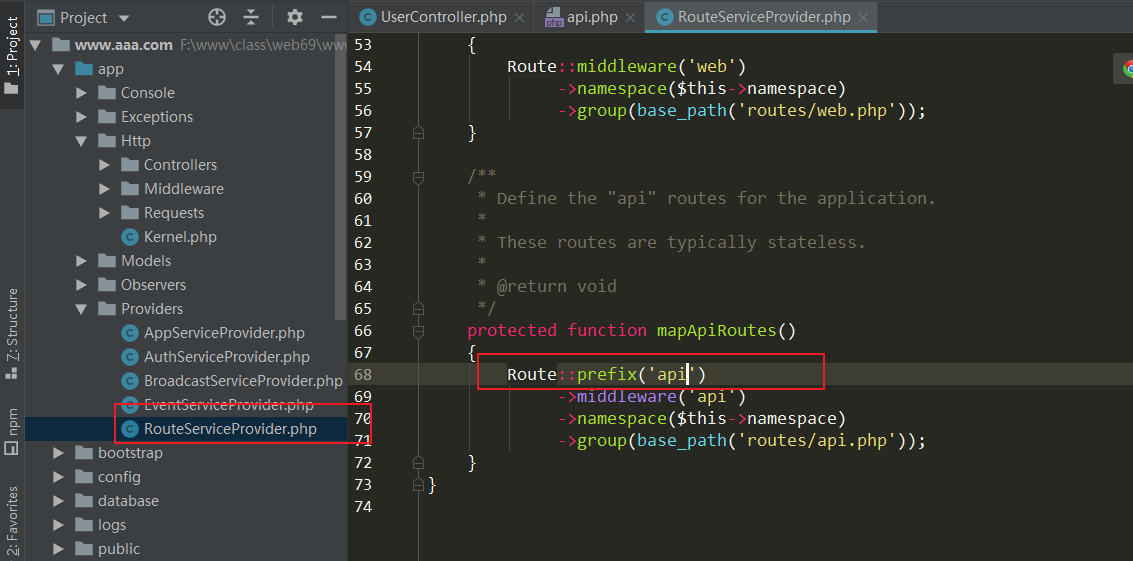
#### 2.1.4.1、服务器端代码的编写





写路由

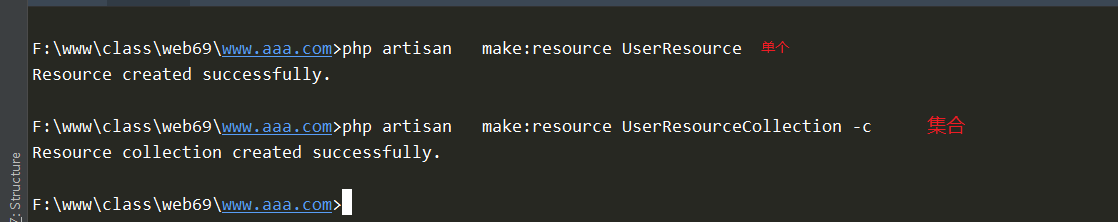
可以更换接口路由的前缀名称

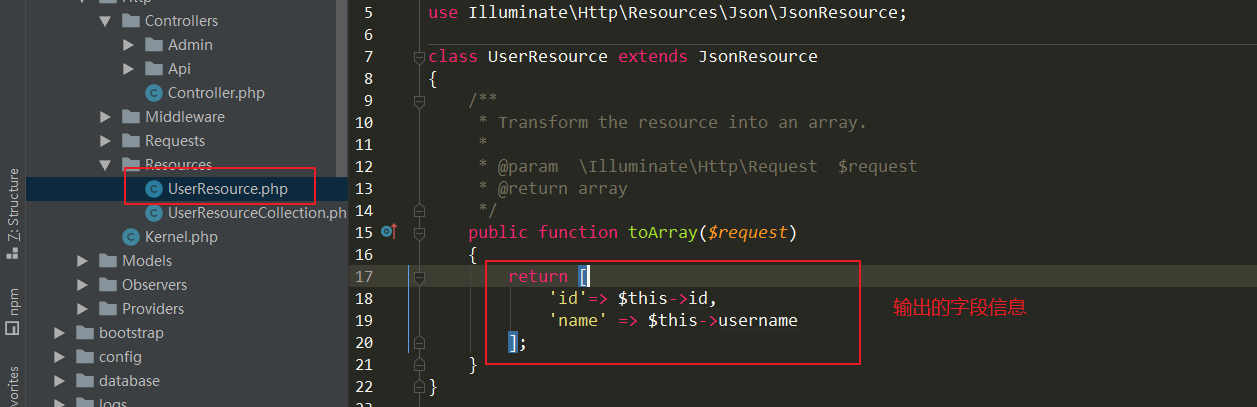




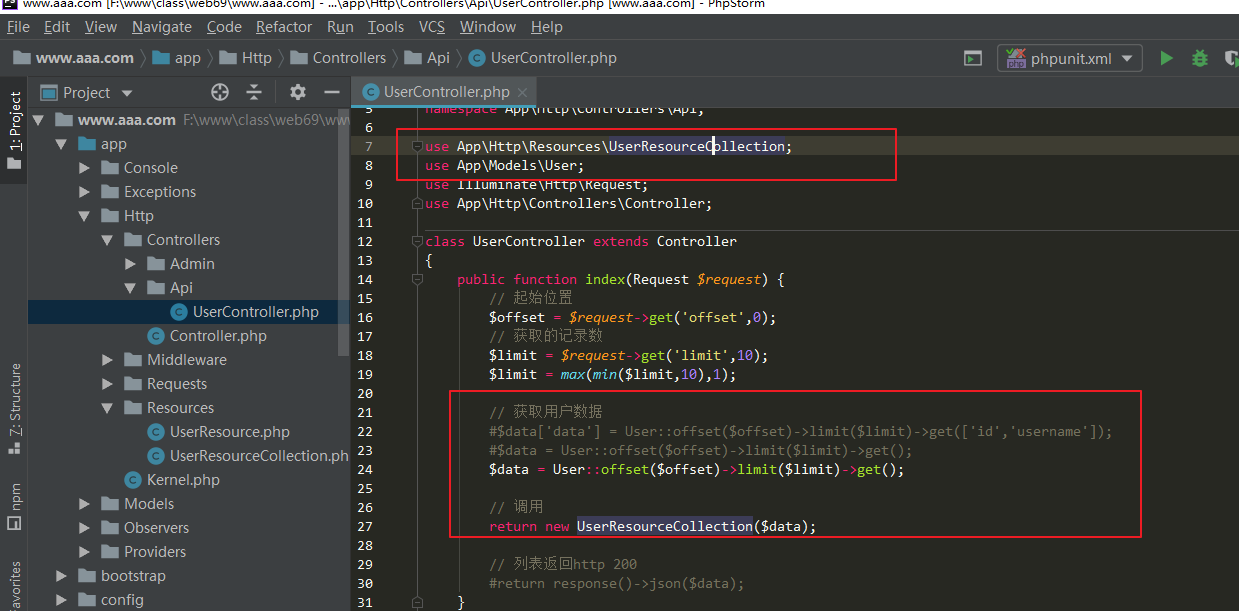


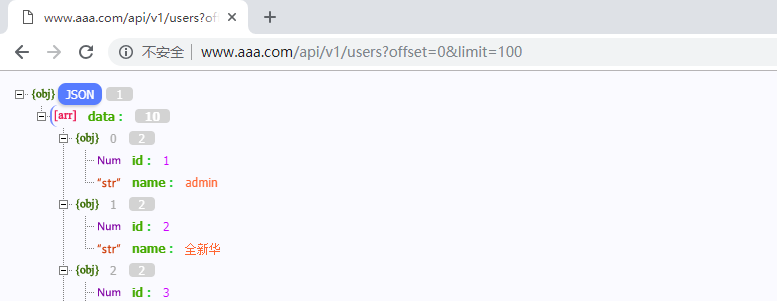
#### 2.1.4.2、make:resource生成接口输出





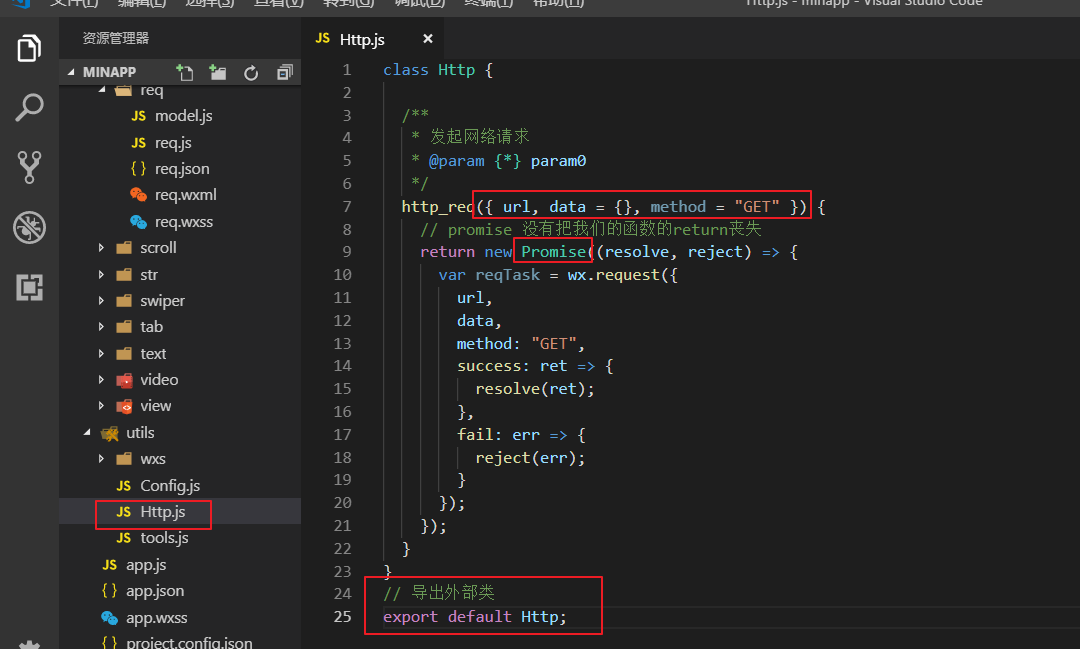
控制器



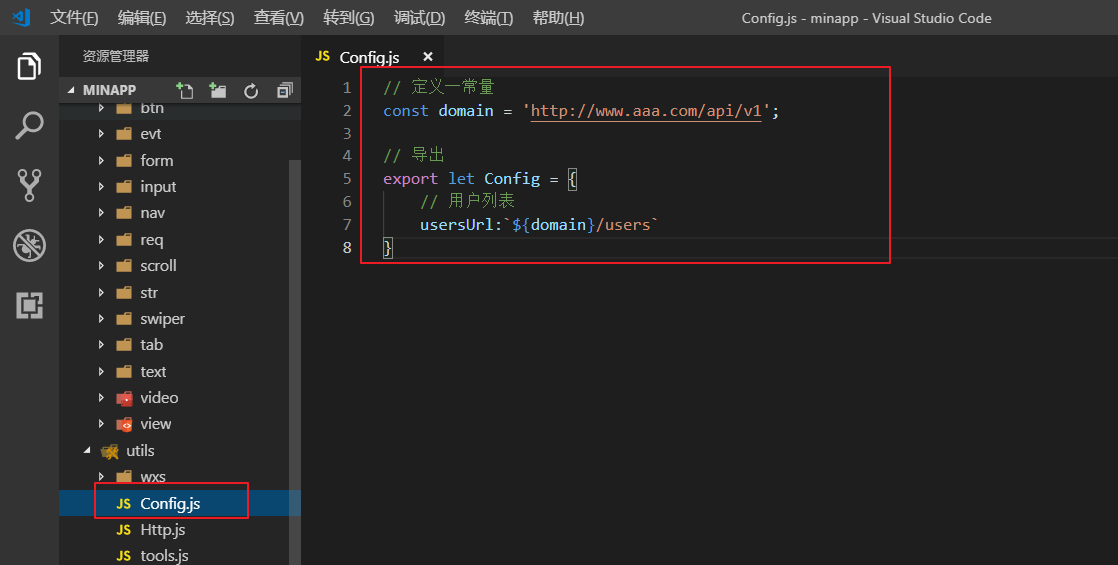


#### 2.1.4.3、加载更多小程序写法

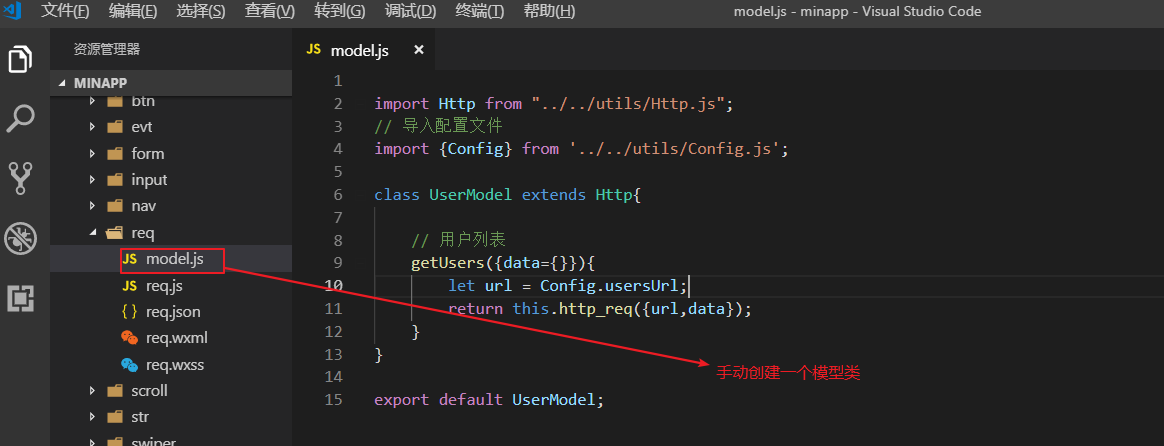
首先定义一个http请求的类



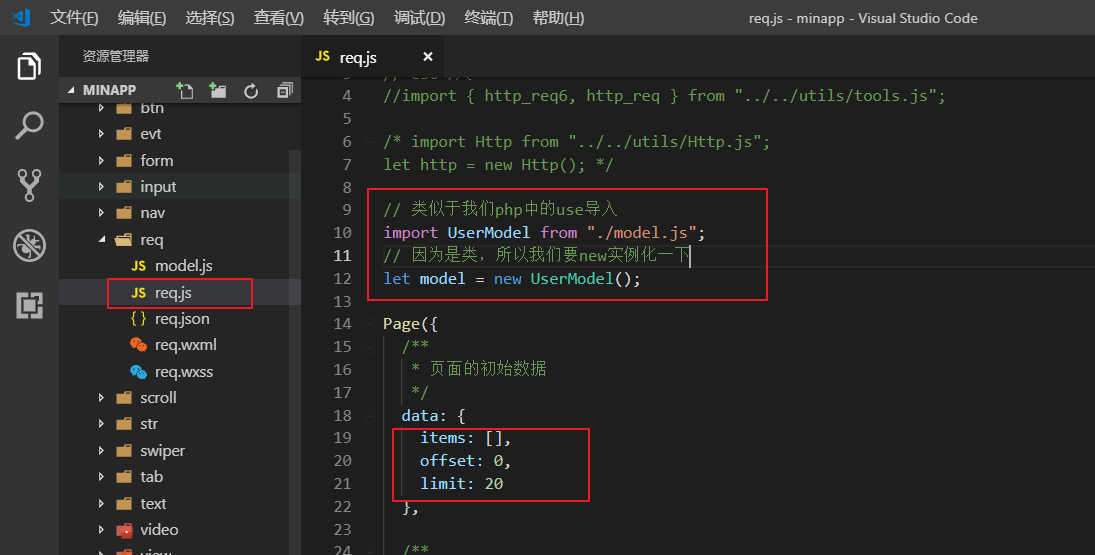
定义一个全局的配置文件

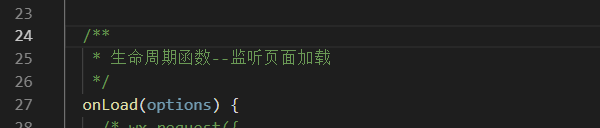


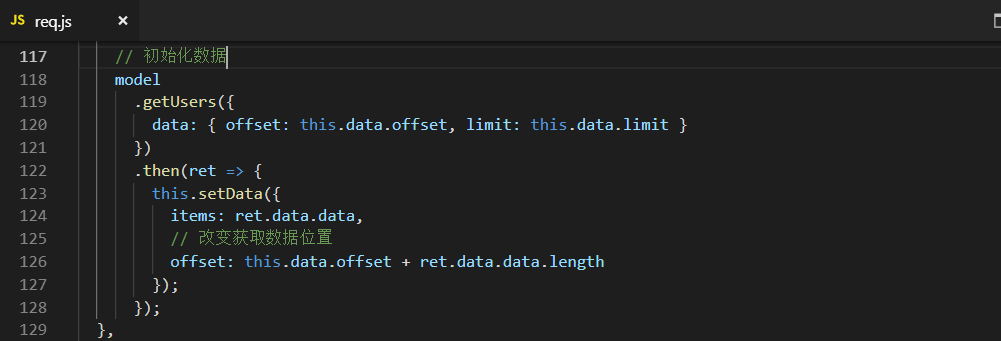
在页面中创建一个model.js文件



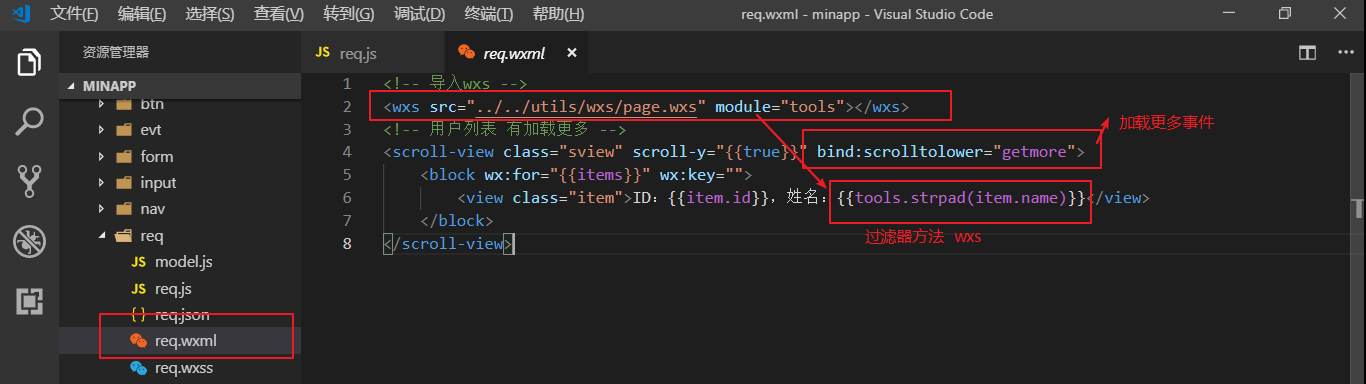
在page.js文件中 控制文件中



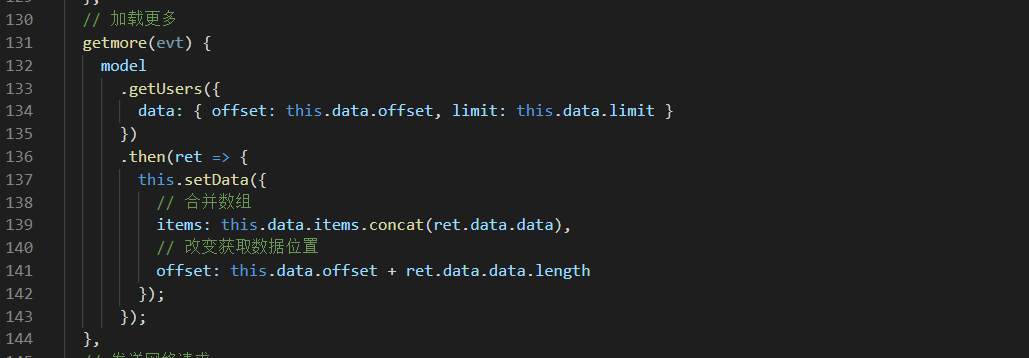




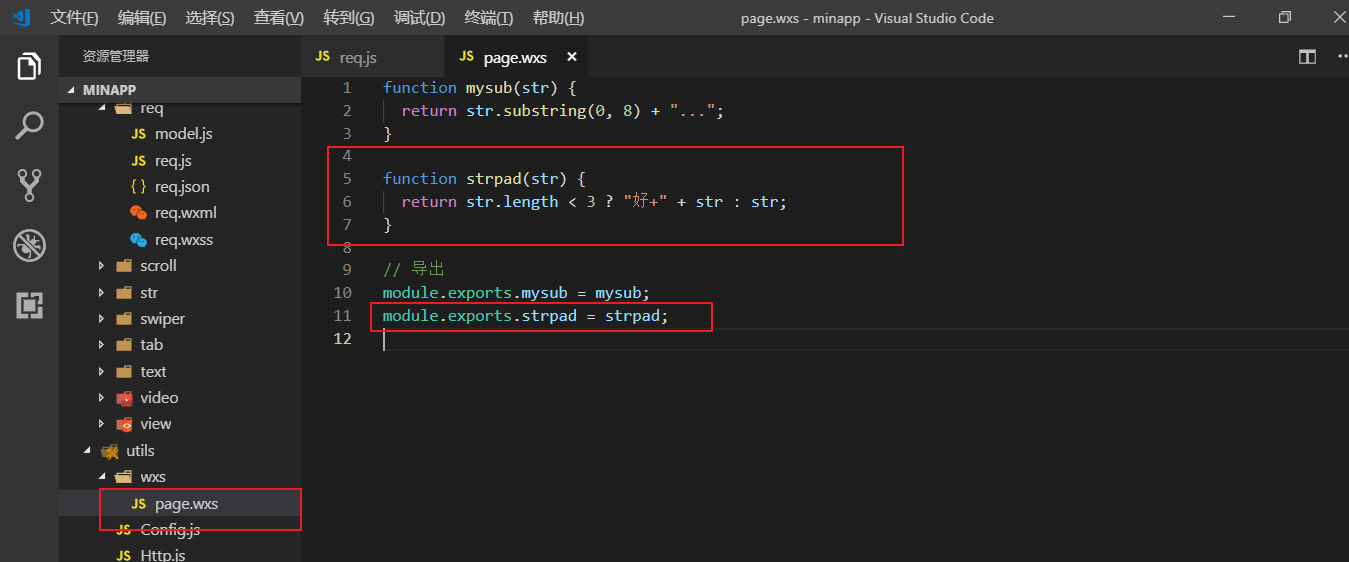
加载更多



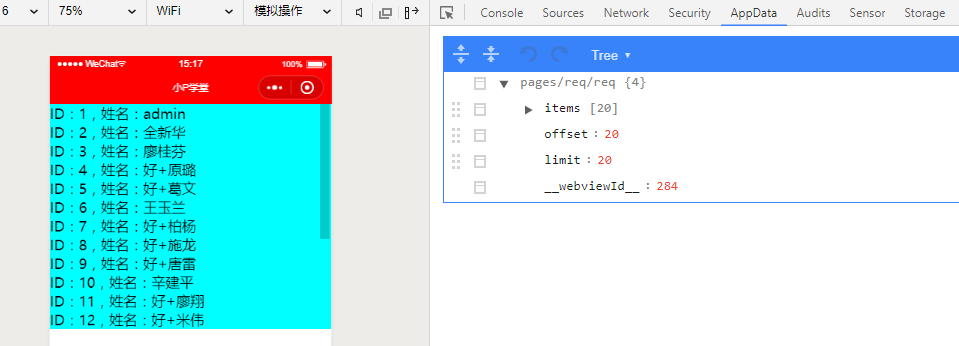
req.js文件中添加对应的方法



过滤器方法的编写 wxs



效果



## 2.2、数据缓存

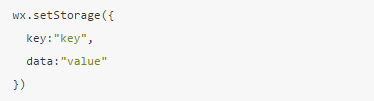
每个微信小程序都可以有自己的本地缓存，即对本地缓存进行设置、获取和清理。同一个微信用户，同一个小程序 storage 上限为 10MB。

注意： 如果用户储存空间不足，我们会清空最近最久未使用的小程序的本地缓存。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。

### 2.2.1、设置缓存

设置缓存分为同步和异步

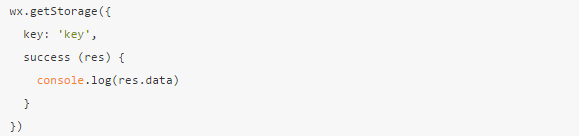
* wx.setStorage(OBJECT) 异步
* wx.setStorageSync(KEY,DATA) 同步



### 2.2.2、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



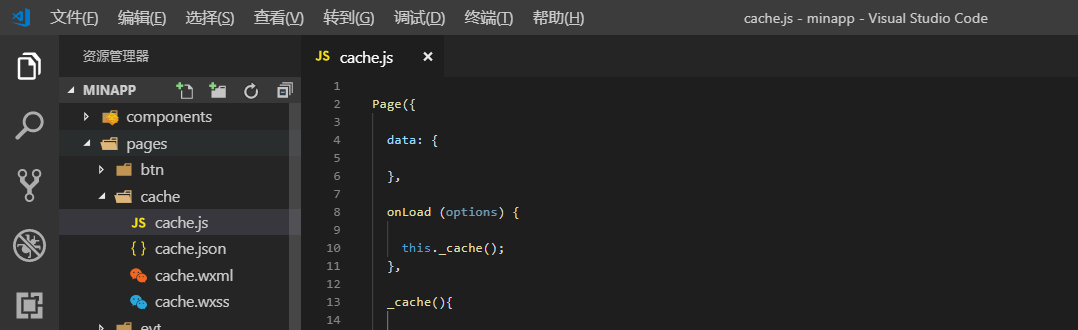
### 2.2.3、移除本地缓存

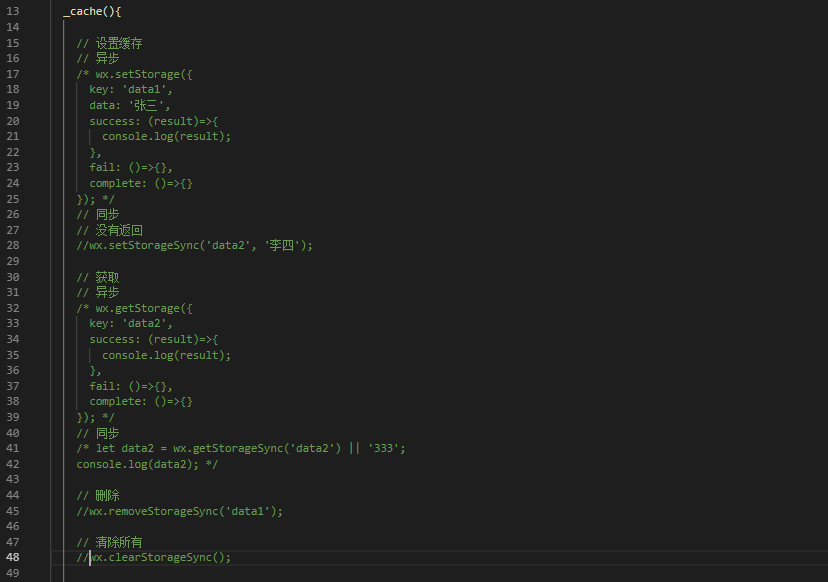
从本地缓存中异步移除指定 key, 分为同步和异步

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

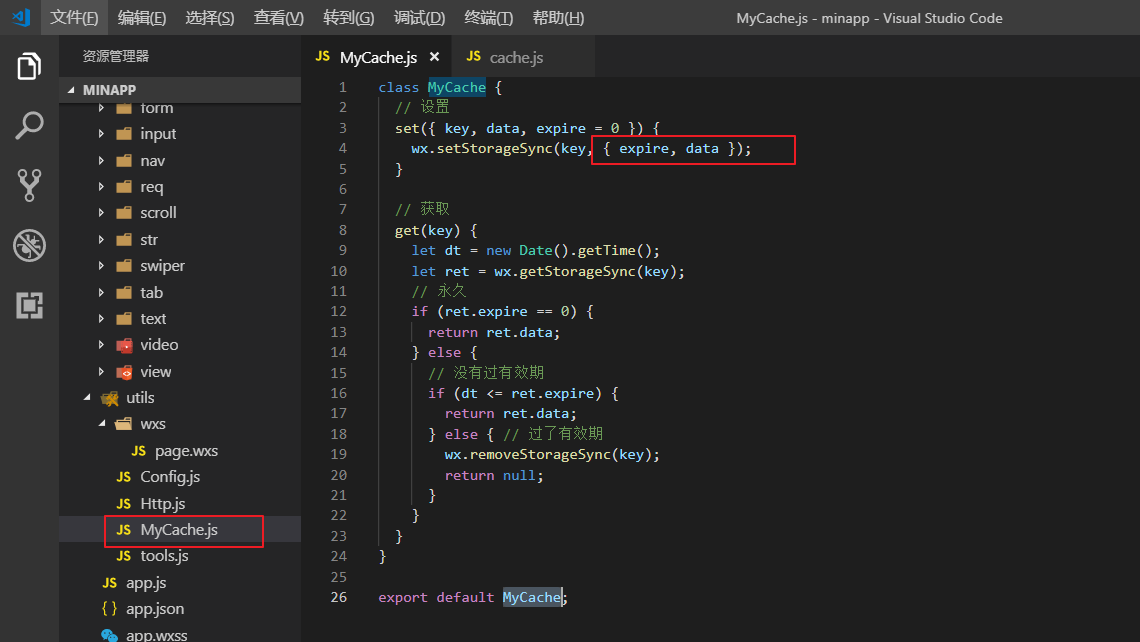
### 2.2.4、清除本地缓存

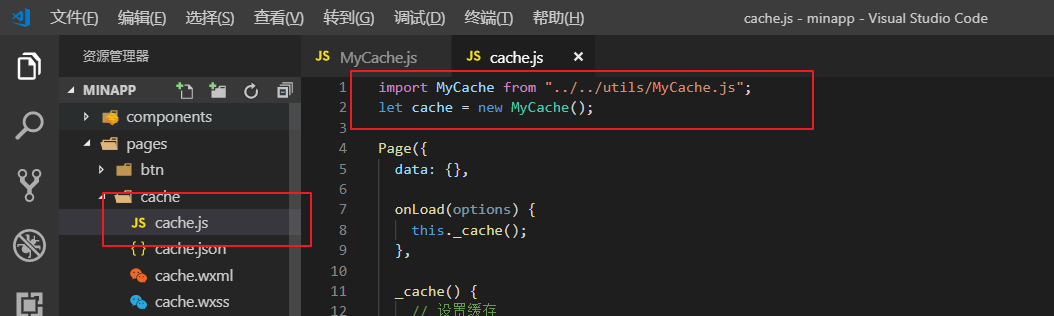
* wx.clearStorage()
* wx.clearStorageSync()

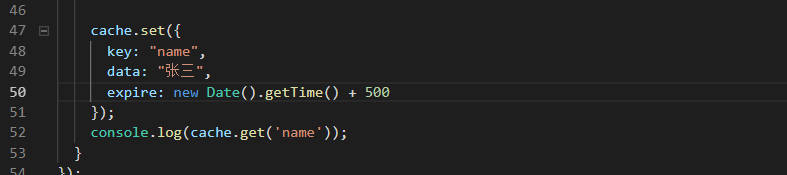


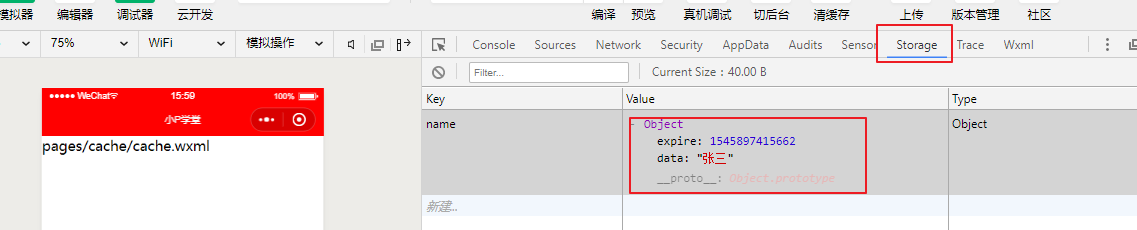


### 2.2.5、自定义缓存









## 2.3、交互反馈

### 2.3.1、显示、隐藏加载框

wx.showLoading(Object object) # 显示

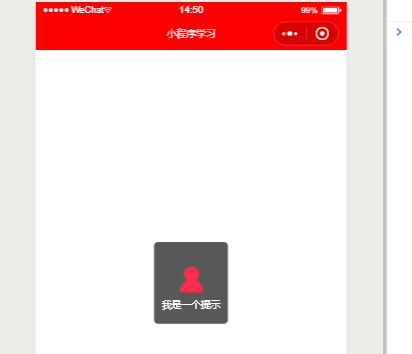
wx.hideLoading(Object object) # 隐藏



### 2.3.2、显示和隐藏消息提示框

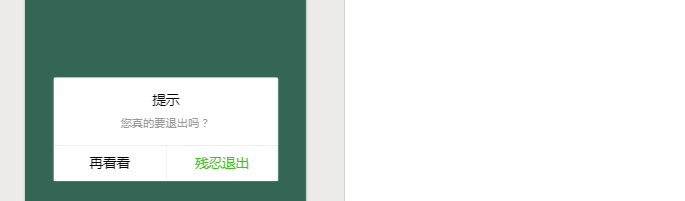
wx.showToast(OBJECT)

wx.hideToast()

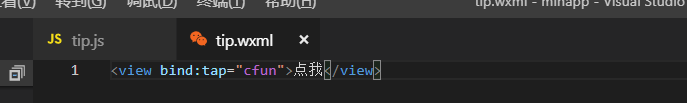


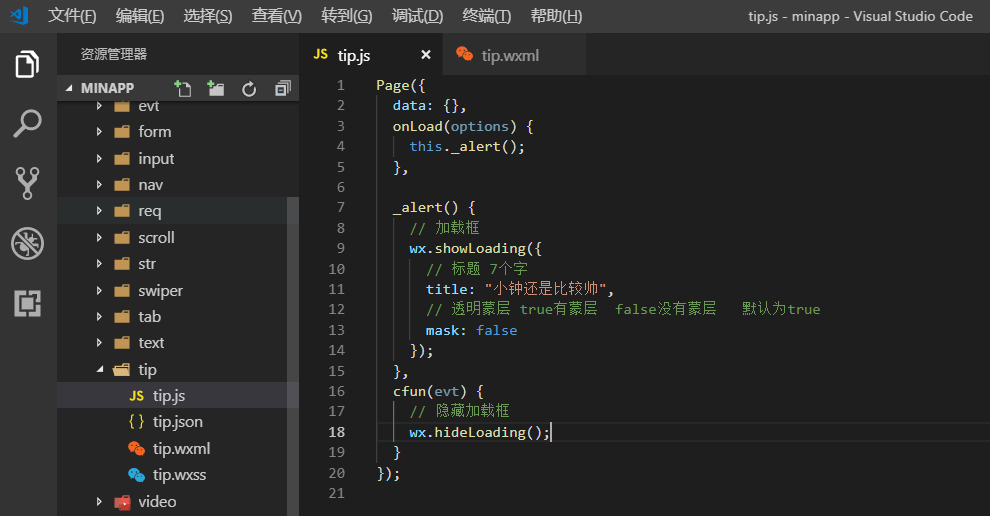
### 2.3.3、显示模态弹窗

wx.showModal(OBJECT)

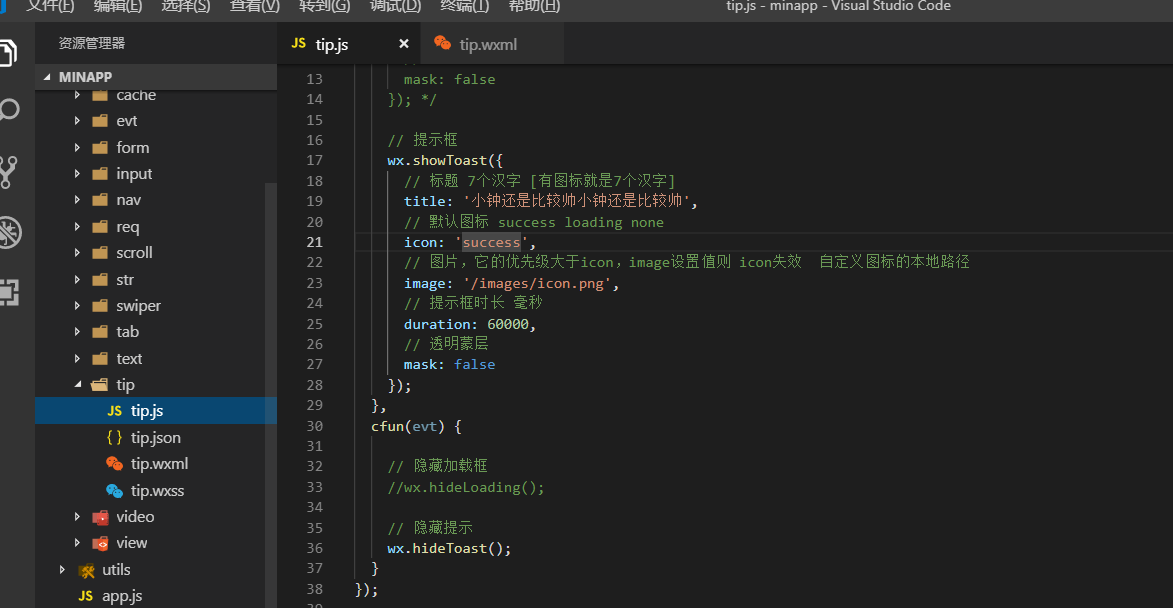


加载框





提示框



确认框

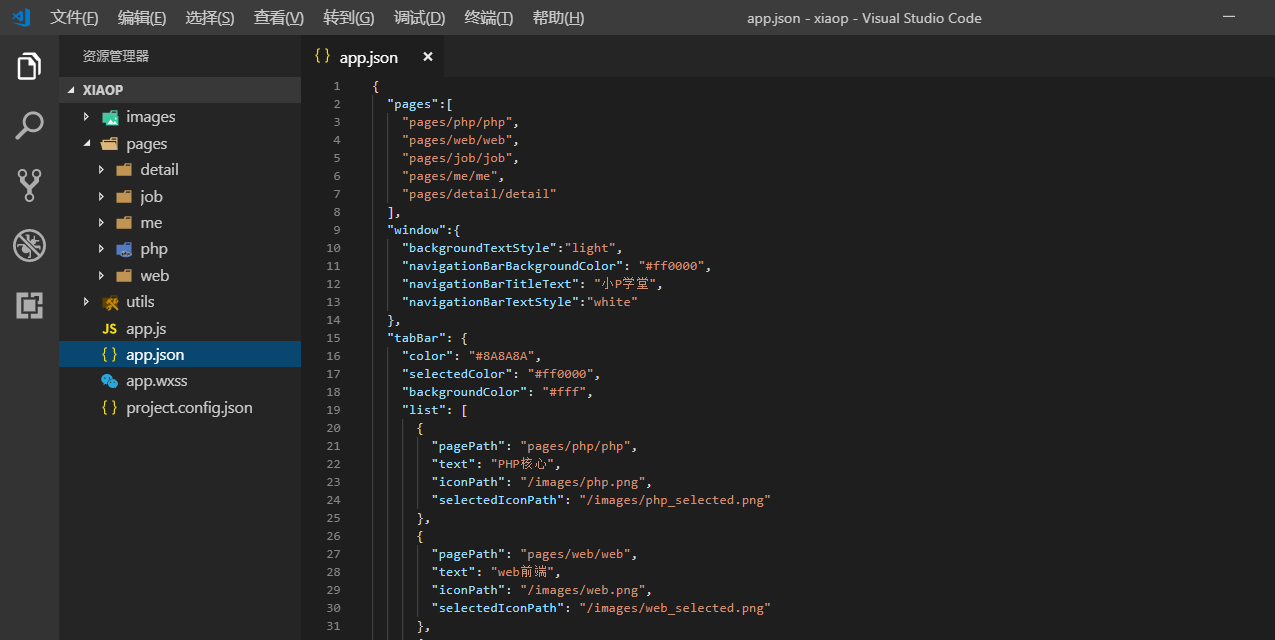




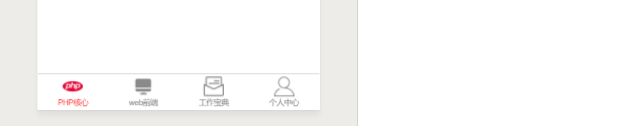
# 三、小P学堂

## 3.1、小程序项目初始化

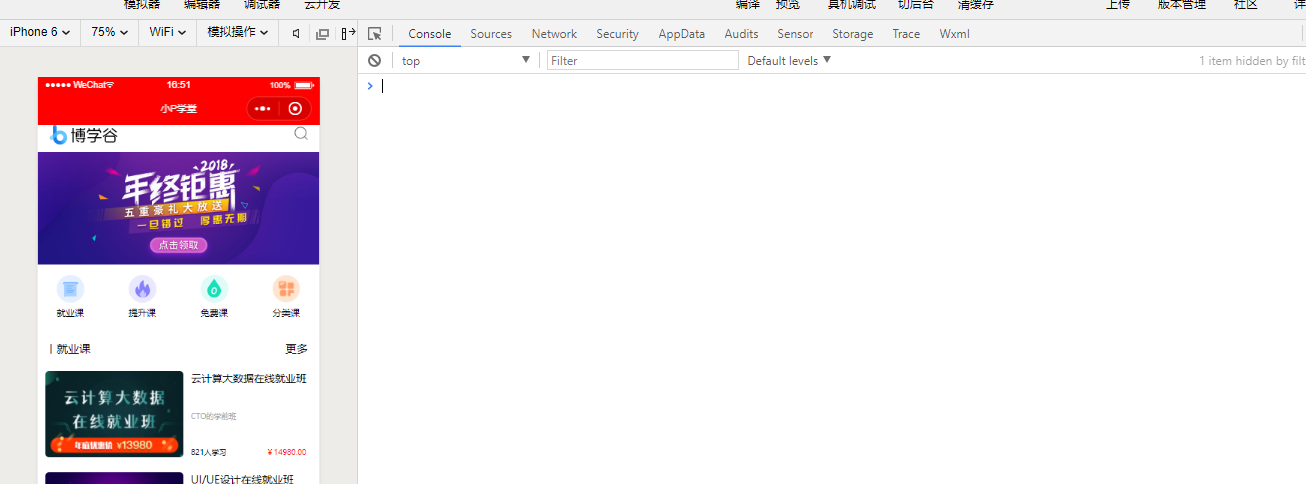
设置app.json文件



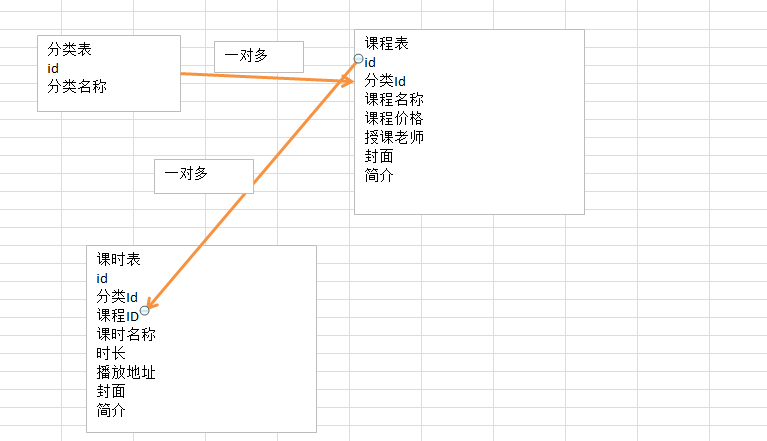




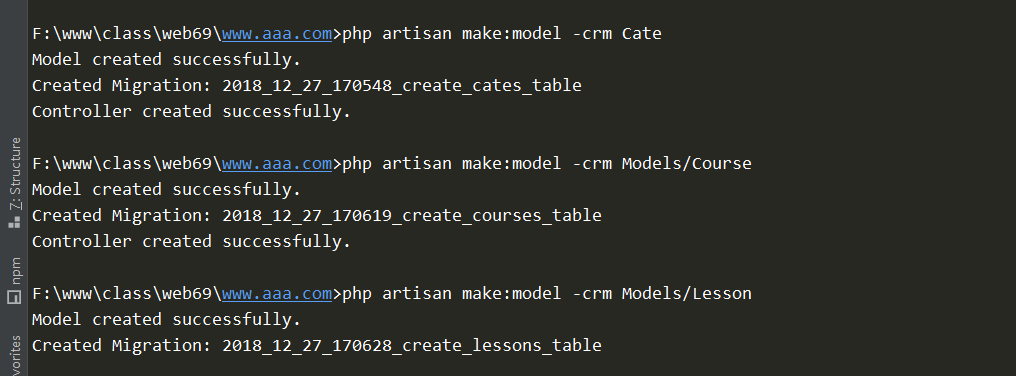
## 3.2、PHP列表静态页



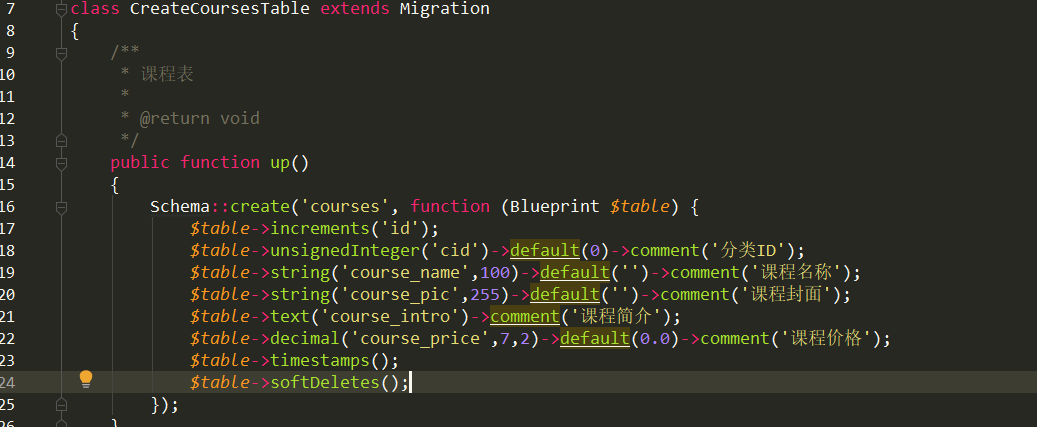
## 3.3、数据表设计

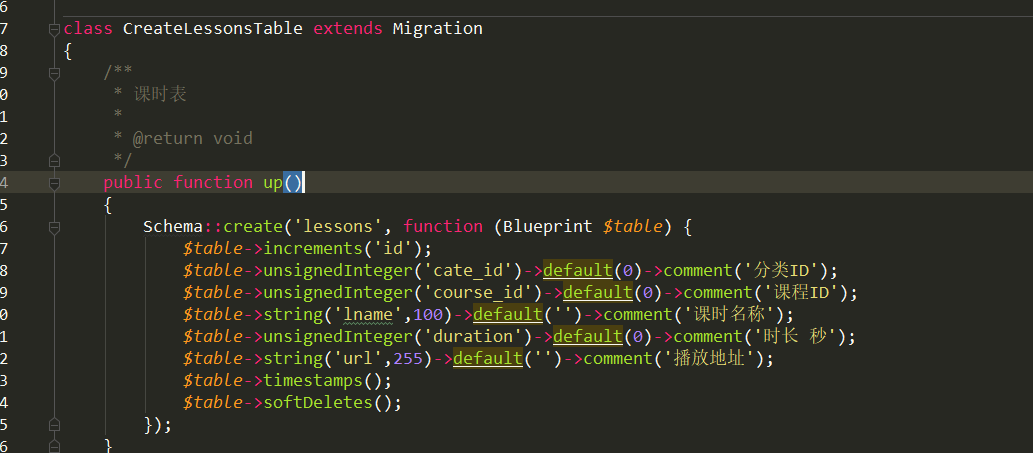


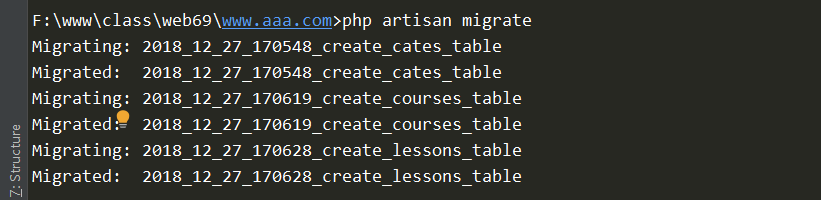
创建控制器 模型和迁移文件

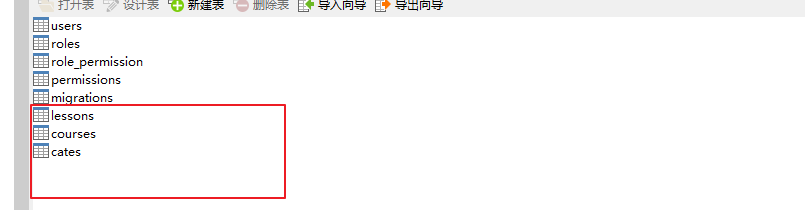




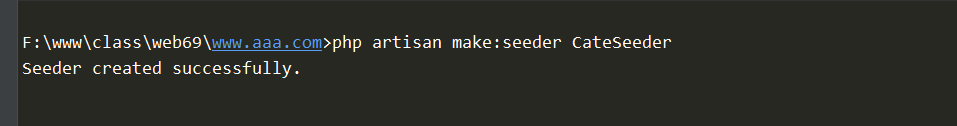


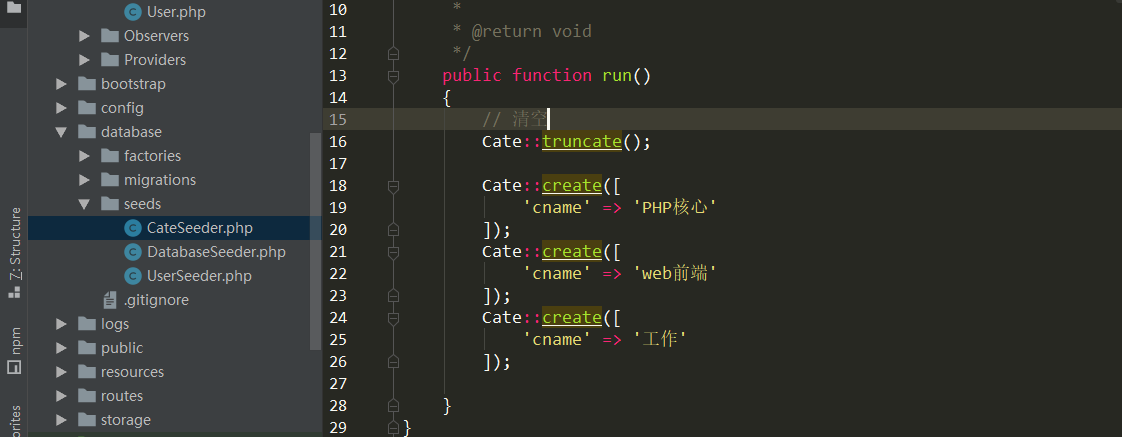


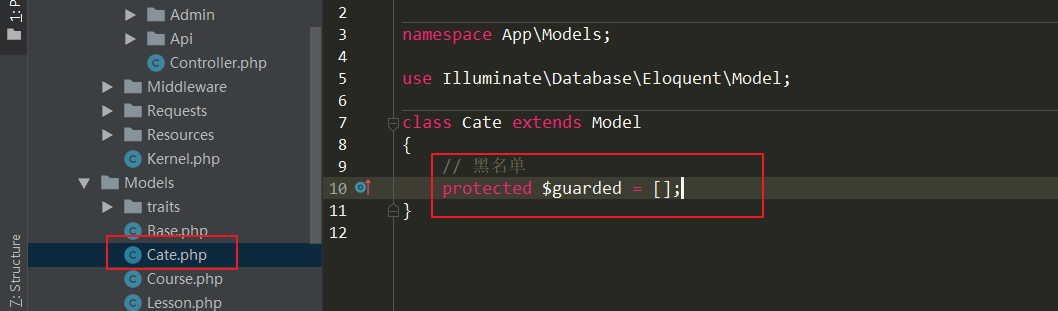


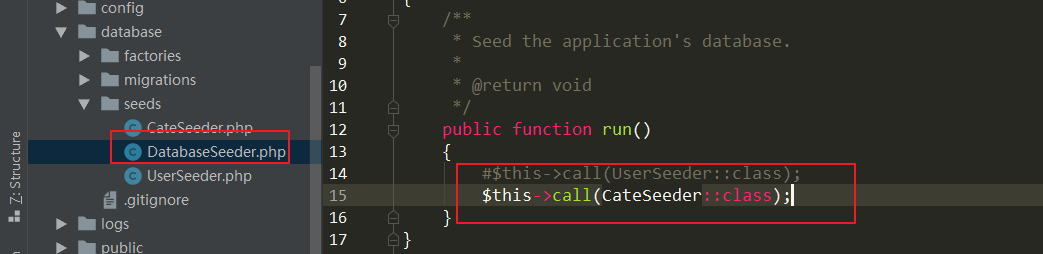


## 3.4、模拟分类数据

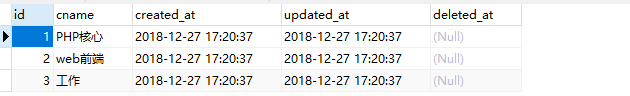










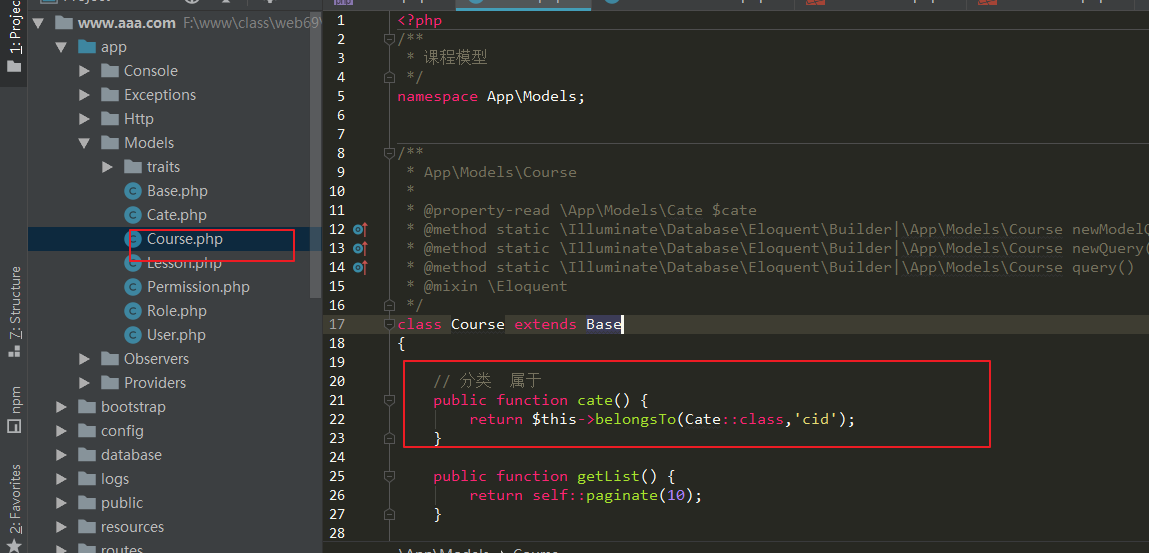


## 3.5、课程管理

定义一个资源路由



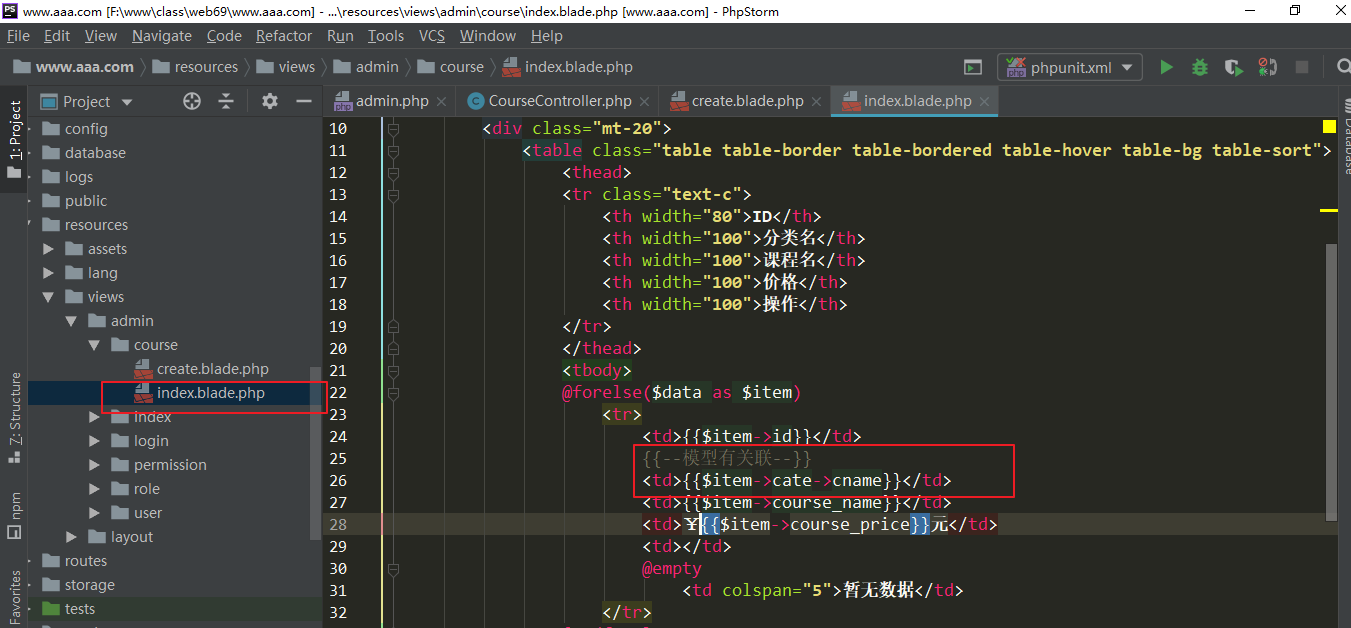
让课程的模型和分类的模型关联起来



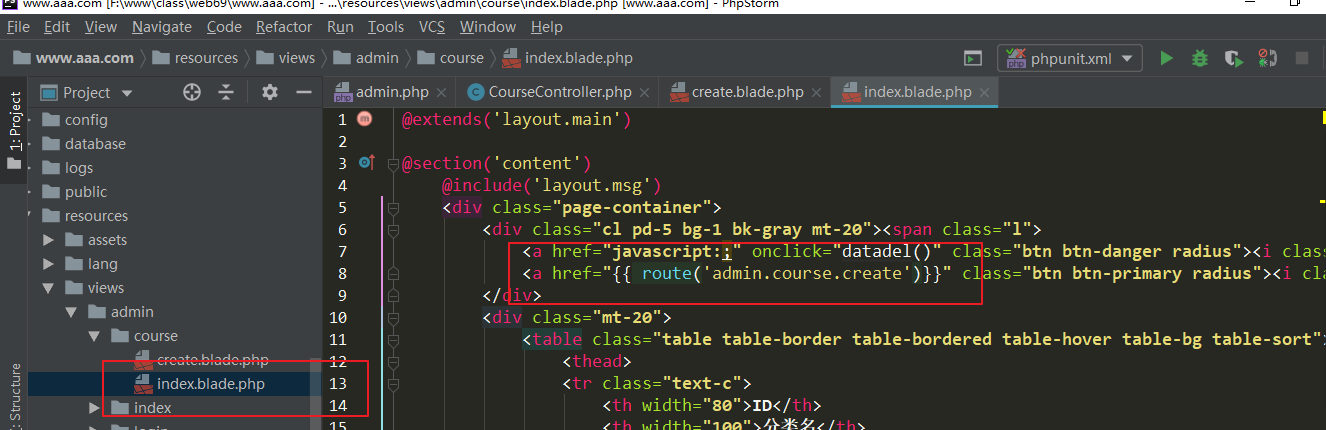
在控制器调用



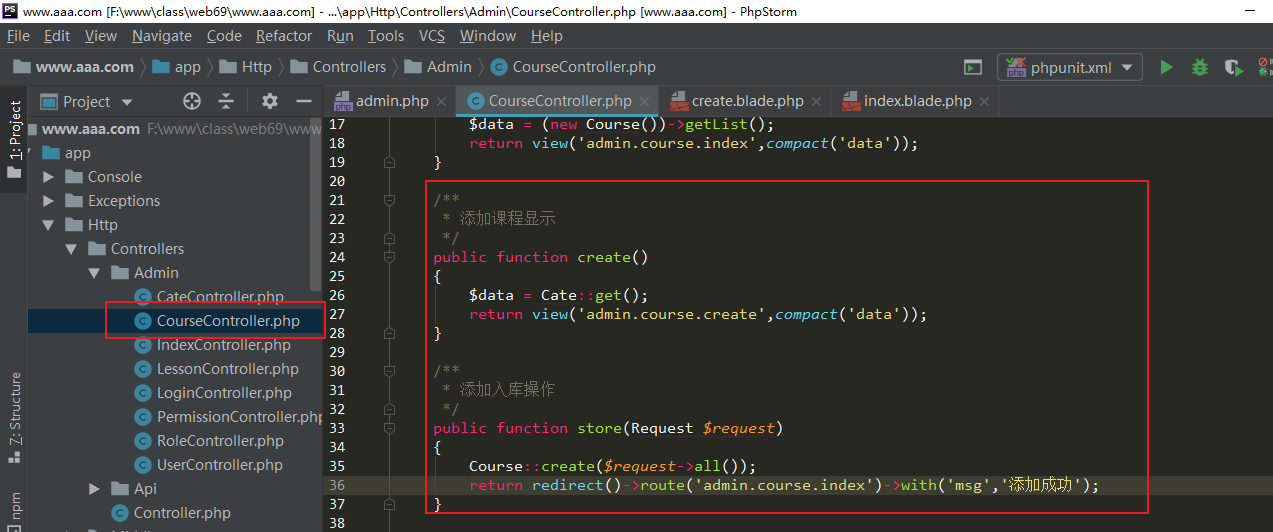
列表模板



给添加课程按钮加链接



添加控制器



添加模板

