# 好客租房\_09

# 今日目标5

* 能够了解laravel接口路由定义

routes/api.php中定义

此文件有默认的前缀 api

接口符合 restful 主流

* 能够配置好passport接口验证(jwt)

接口验证 jwt

安装

迁移

执行 secret 命令 install

在providers boot添加的过期时间

模型中use 方法trait

auth.php文件中配置 guard节点 dirver=passport

在auth登录成功后，进行生成token

* 能够接口返回json数据

response()->json(数组，http状态码);

restful规范 返回时要有http状态码

* 能够实现小程序登录wx.login

在小程序先wx.login 得到code 有效期5分钟

在小程序中wx.request 发起网络请求 给自己的服务器

自己的服务器中 get请求小程序的服务器，并且传入 appid secret code

在自己的服务器中得到了openid

在自己的服务器中存入到租客表中，租客表中openid设置唯一索引 使用try catch 捕获异常

在小程序端缓存操作 缓存了1小时

。。。

登录成功后，进行的相关业务。

* 能够实现小程序用户授权

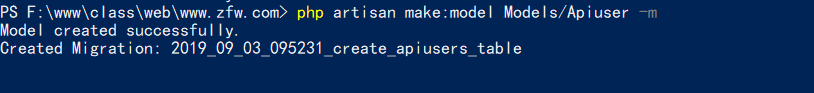
# 一、接口账号管理

## 1.1、设计账号表

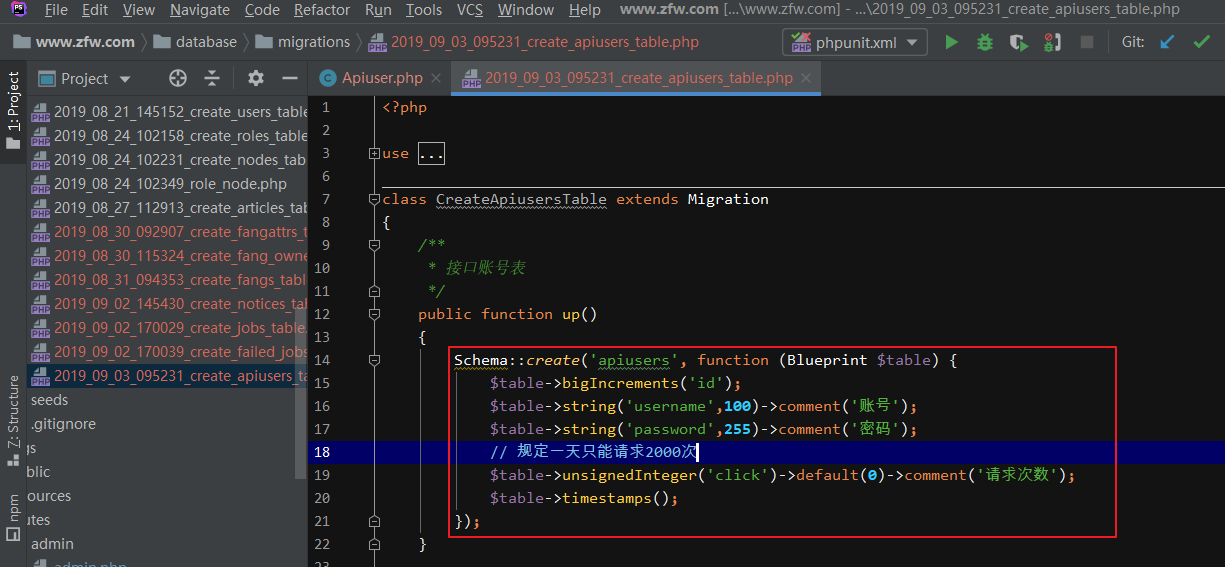
接口账号表，字段设计

id,账号，密码，是否删除，类型，ip白名单 ，有效期，金额

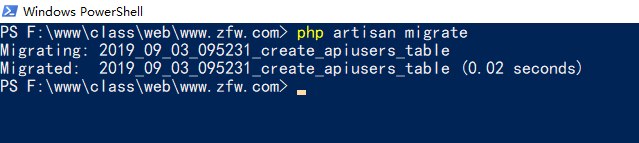
创建模型并同时创建迁移文件

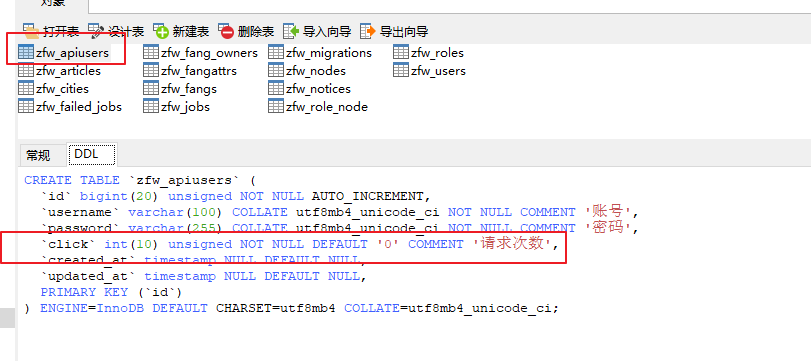


编写迁称文件，创建数据表



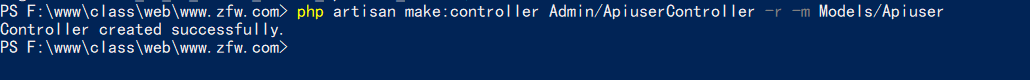
执行迁移文件



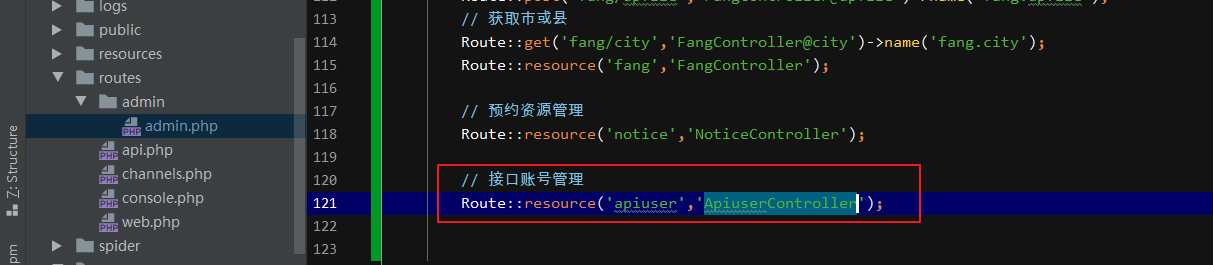


## 1.2、接口账号列表

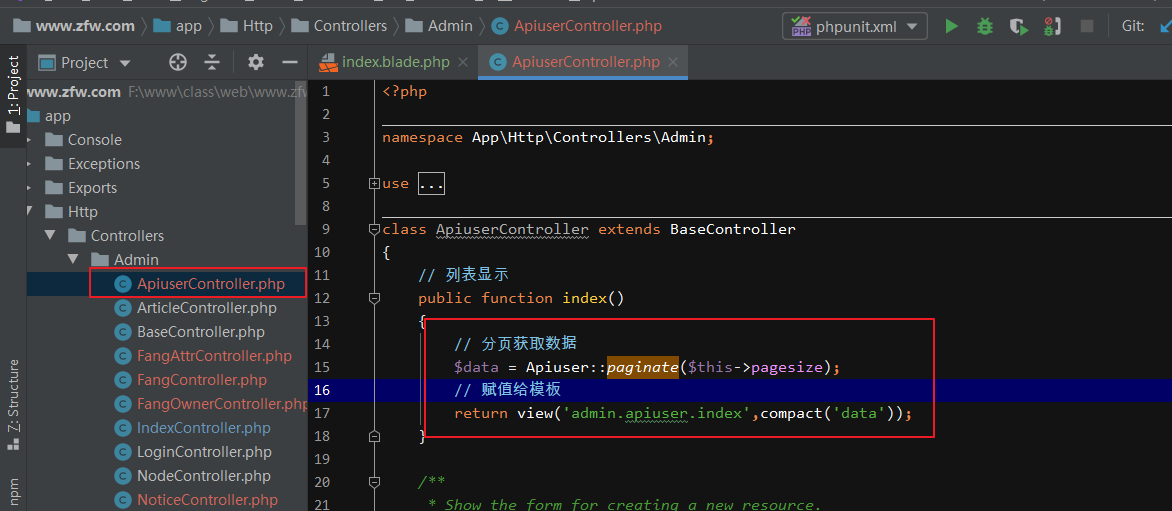
创建资源控制器



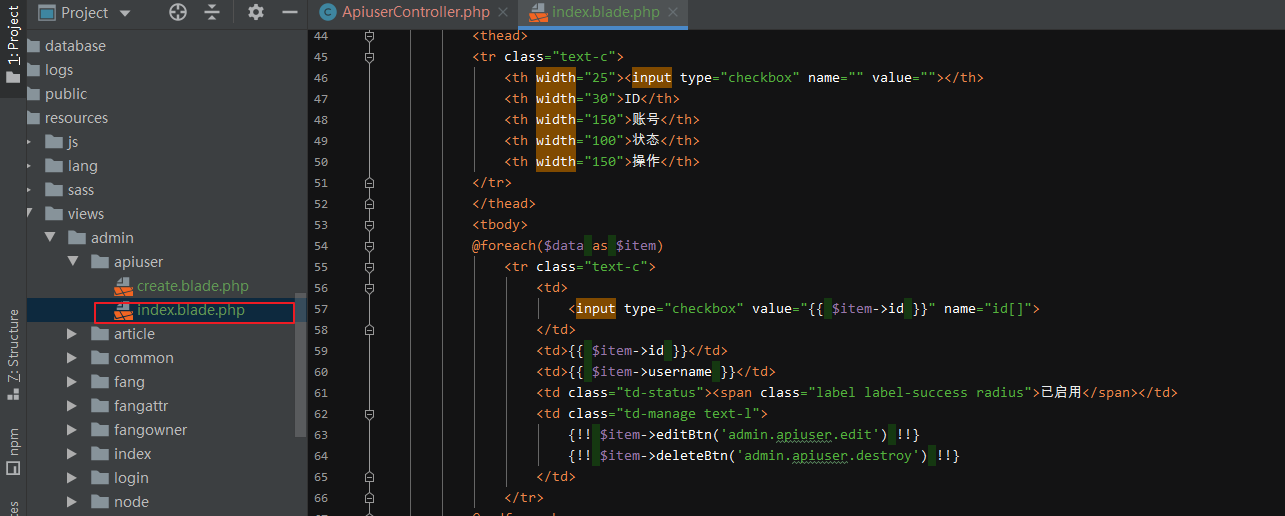
定义资源路由



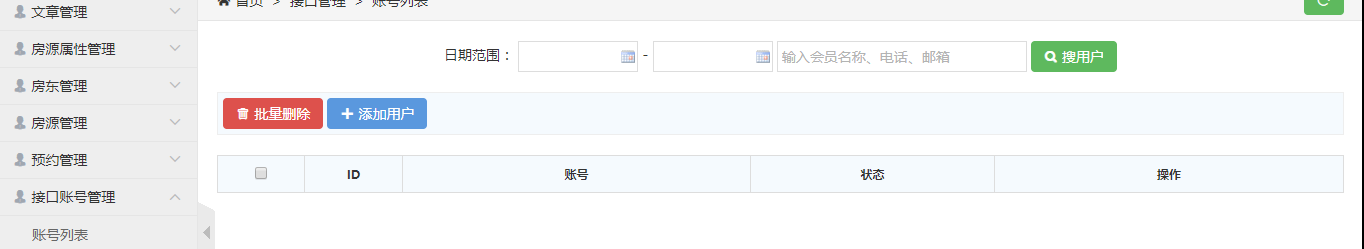
在控制器中指定模板路径



模板

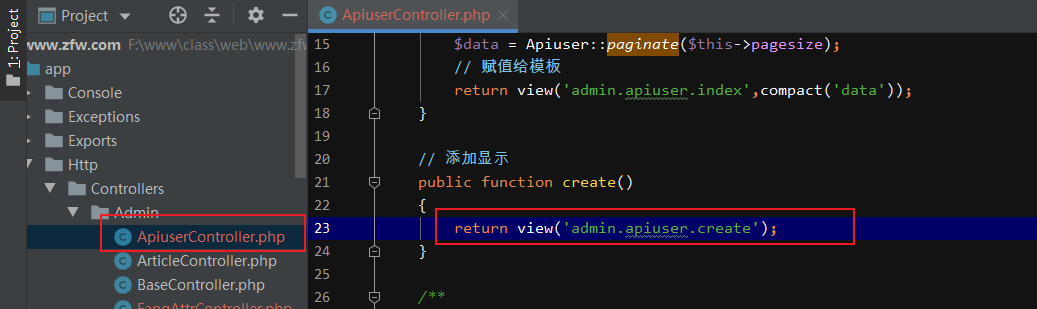


效果

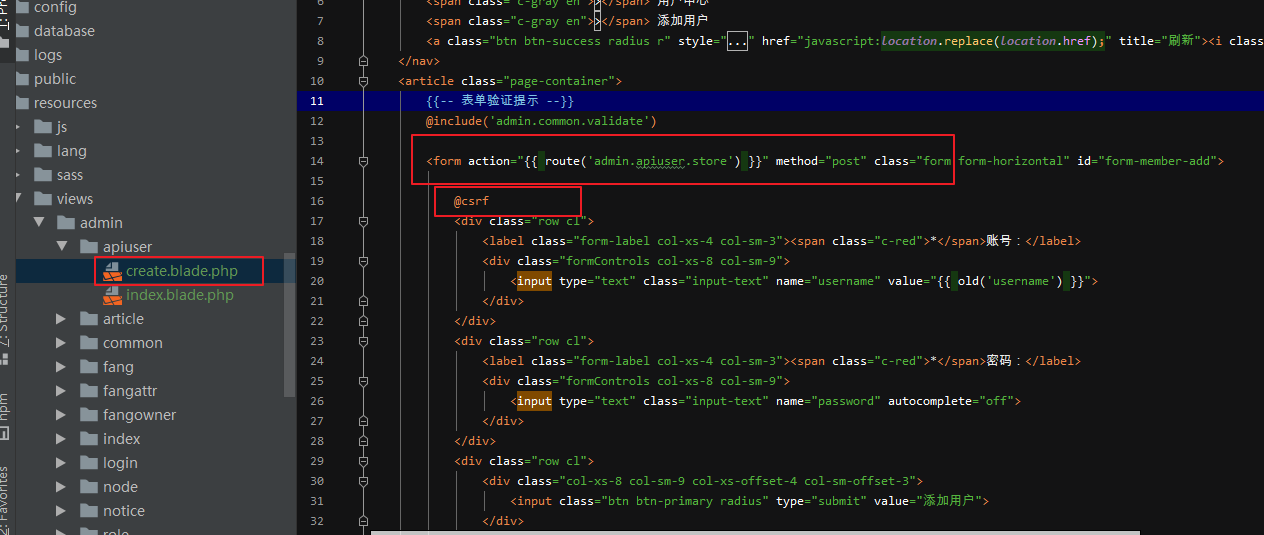


## 1.3、添加账号

控制器中显示添加模板

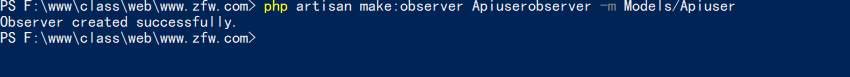


添加账号模板

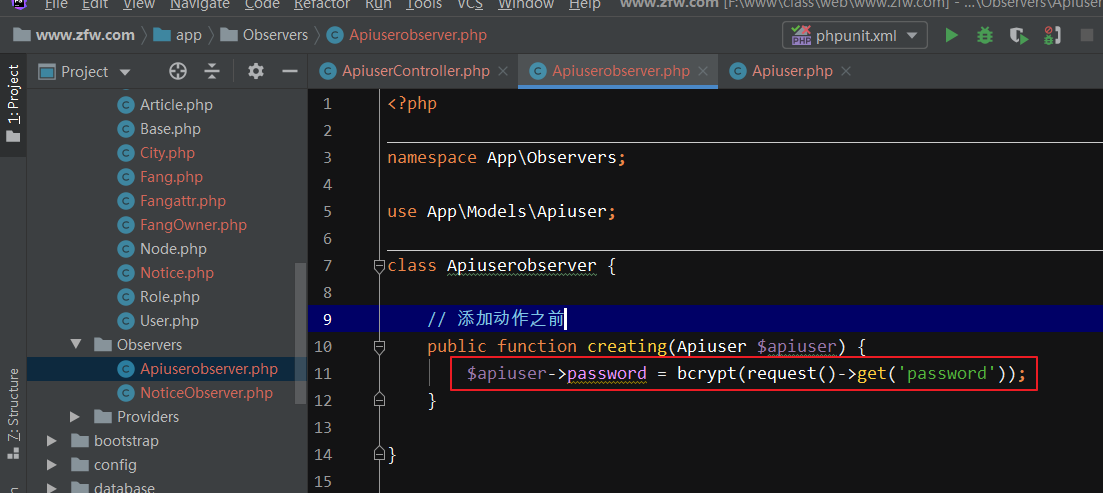


定义模型观察类

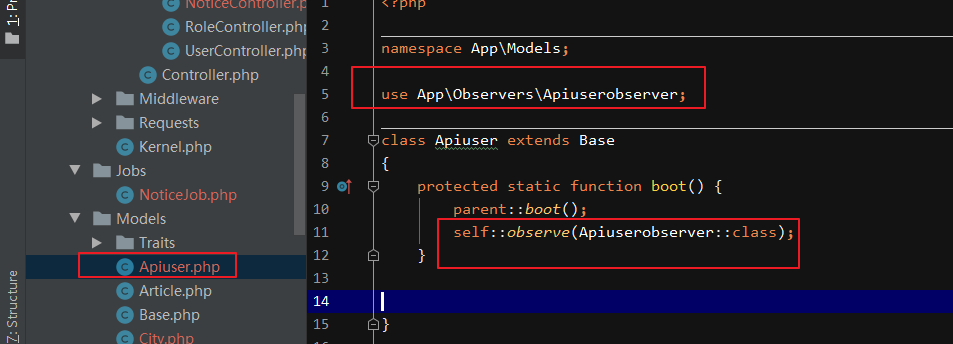
php artisan make:observer Apiuserobserver -m Models/Apiuser



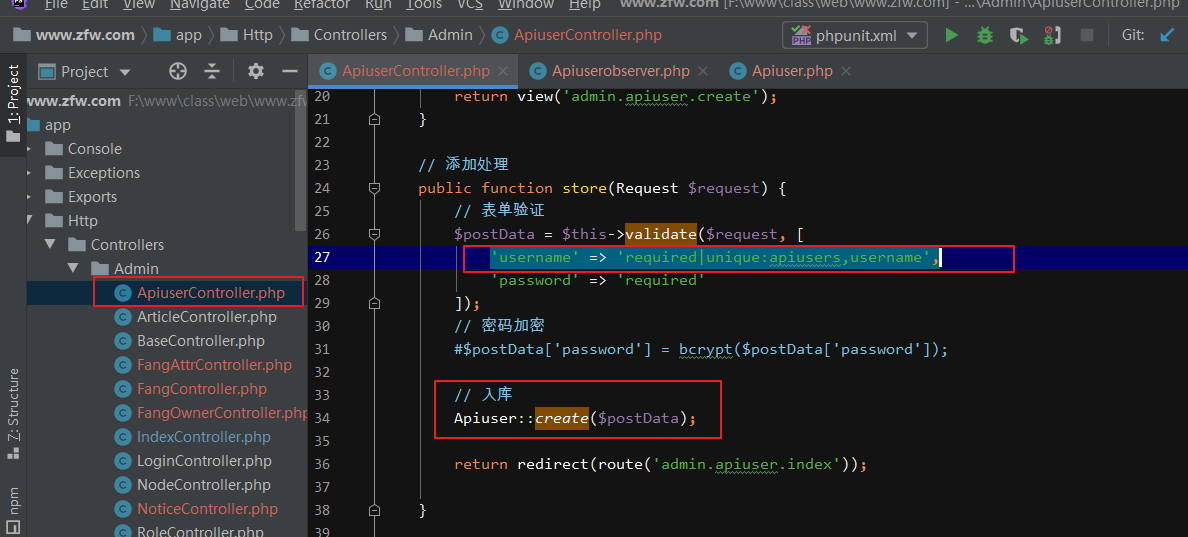
在添加数据之前，让密码字段加密

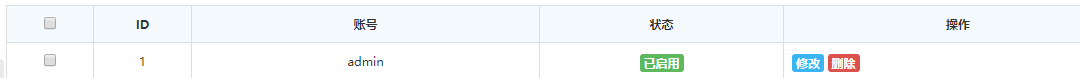


在模型类中添加监听



在控制器实现create添加

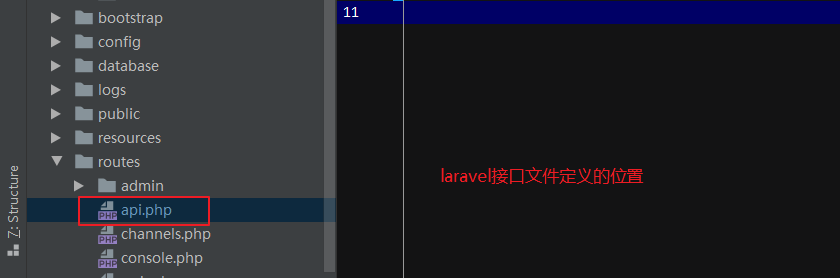




## 1.4、模拟登录

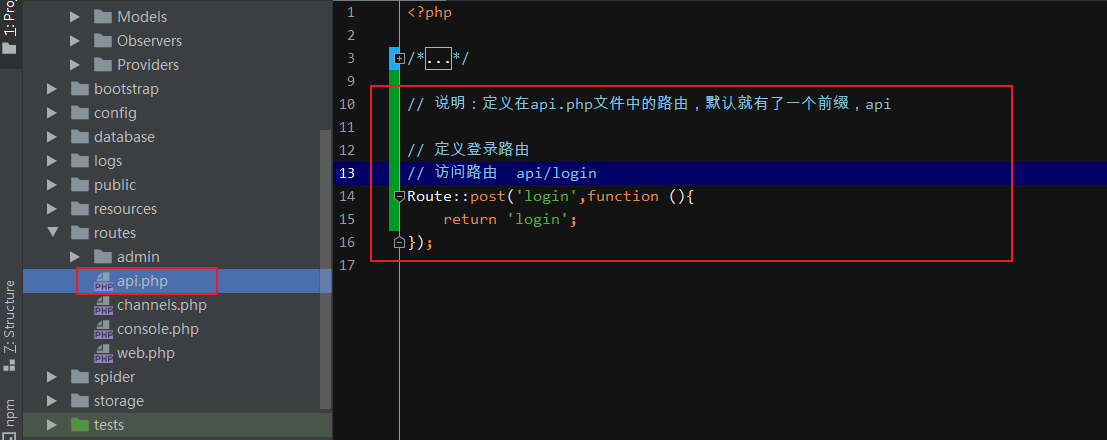
laravel如何来定义接口路由,接口路由定义在什么文件中？

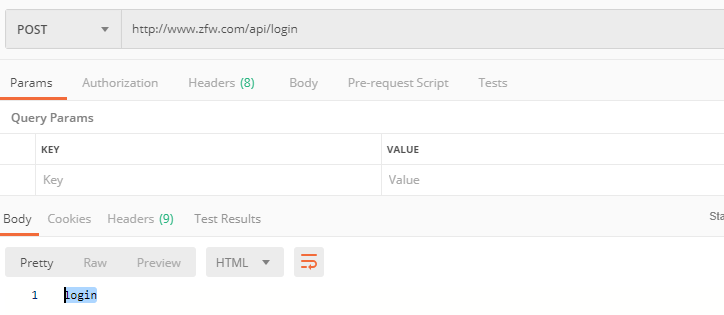
答：laravel接口路由定义在routes/api.php文件中，默认引导的位置



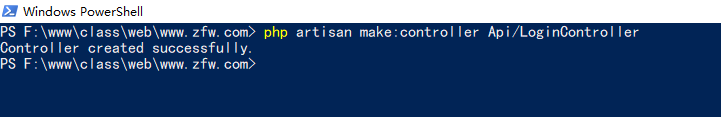
定义接口用户登录路由

注：定义在api.php文件中的路由，默认都有一个api前缀

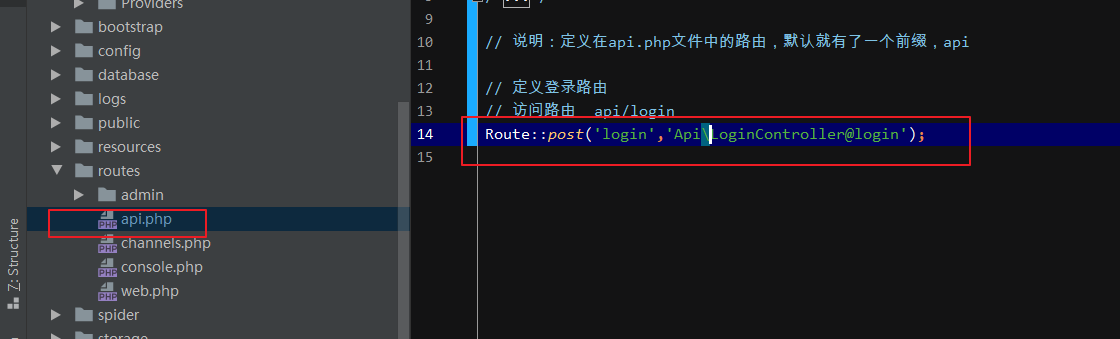




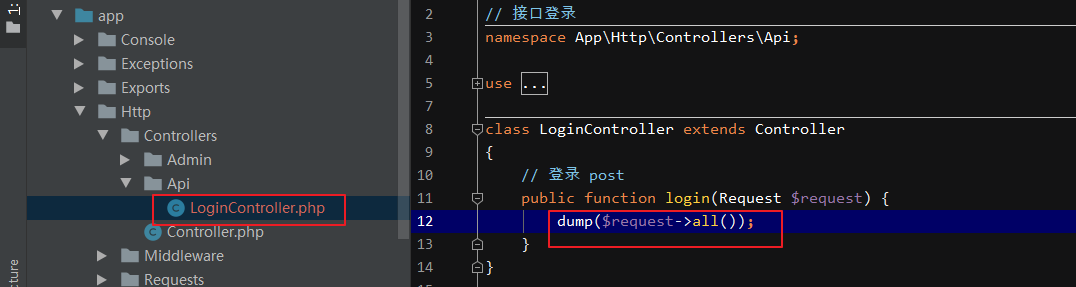
创建接口登录控制器



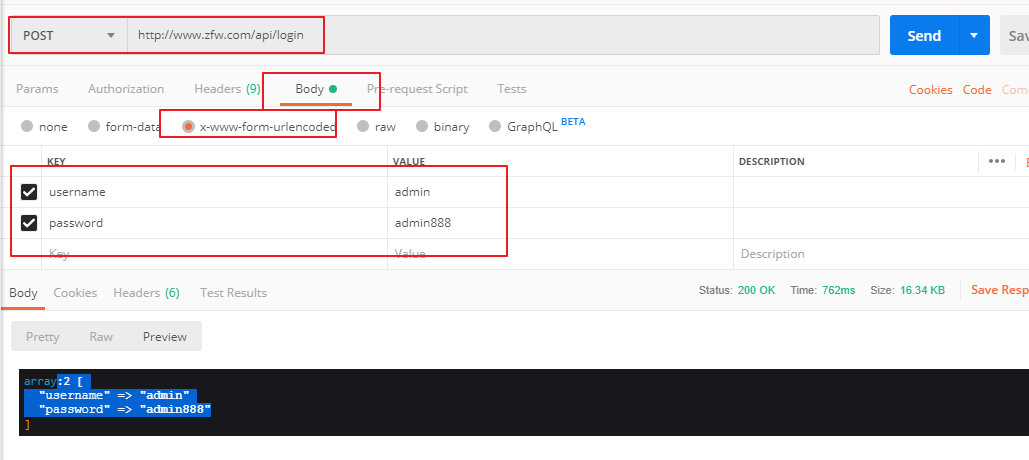
定义好登录路由



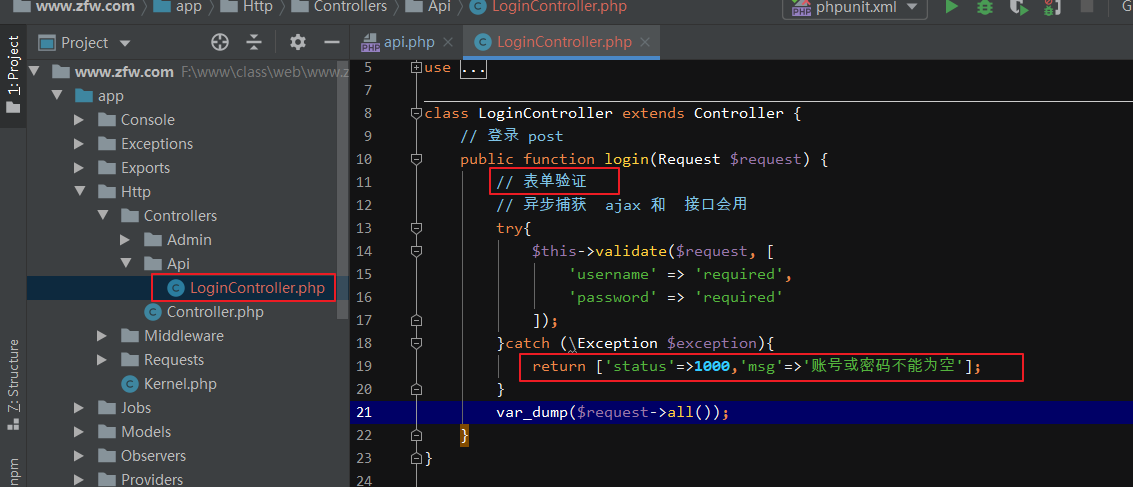
控制器中获取请求的数据



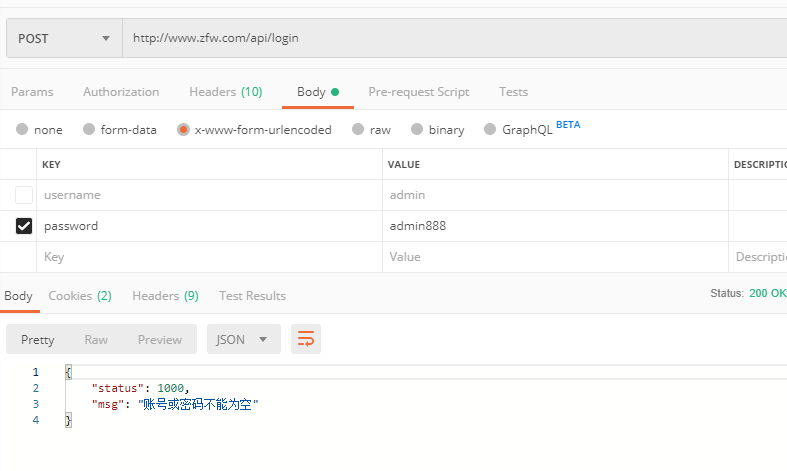
postman测试



在表单提交接受页面进行表单验证



注：使用异常的捕获 try catch

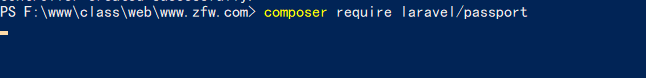


### 1.4.1、配置passport

用passport实现接口的登录和jwt生成(生成token)

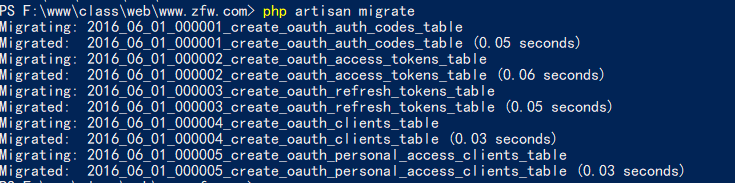
**安装passport插件**

composer require laravel/passport



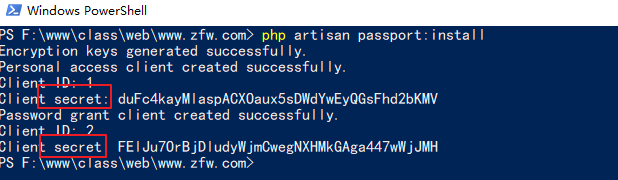
**执迁迁移文件，生成数据表，保存token数据**

php artisan migrate



生成客户端授权码

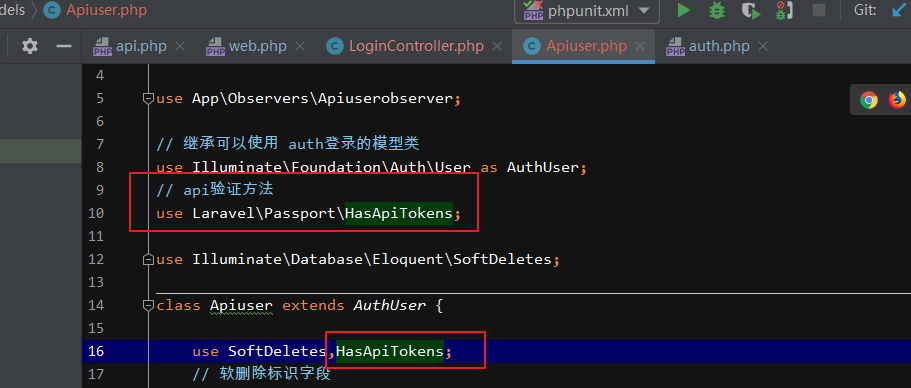
php artisan passport:install



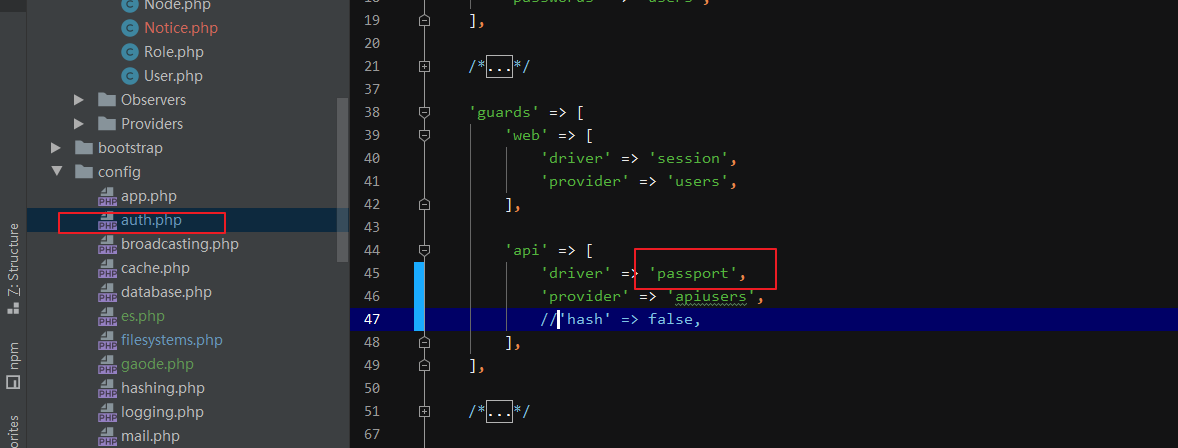
注：secret保存起来

修改接口账号模型

use Laravel\Passport\HasApiTokens;



修改config/auth.php文件中的api配置



**令牌的有效期**

在app/Proivders/AuthServiceProvide.php文件中的boot方法添加有效期

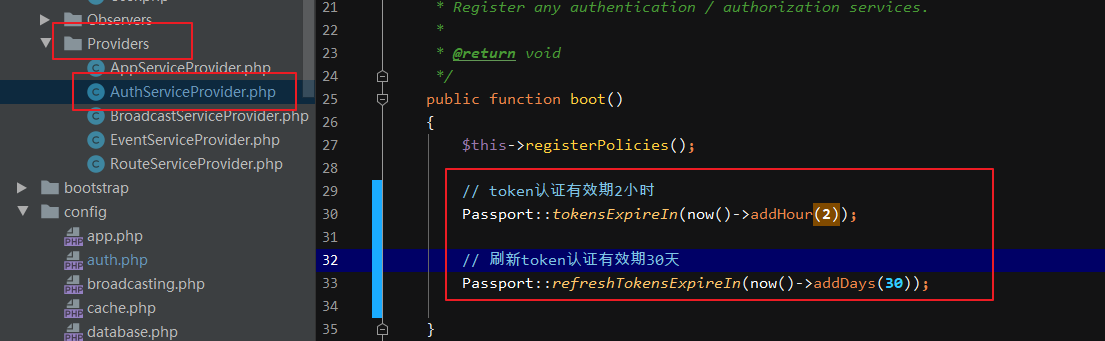
use Laravel\Passport\Passport;

// token认证有效期2小时

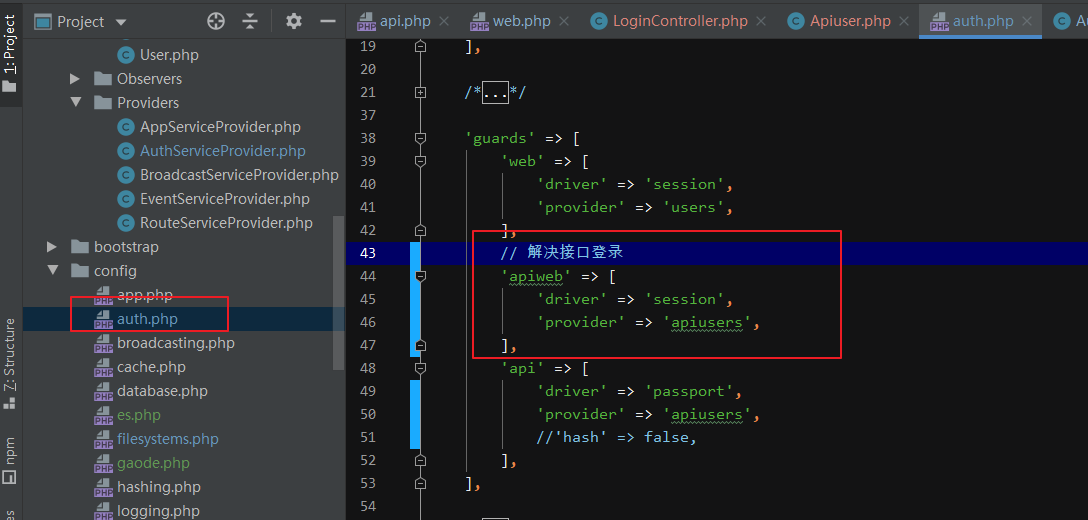
Passport::tokensExpireIn(now()->addHour(2));

// 刷新token认证有效期30天

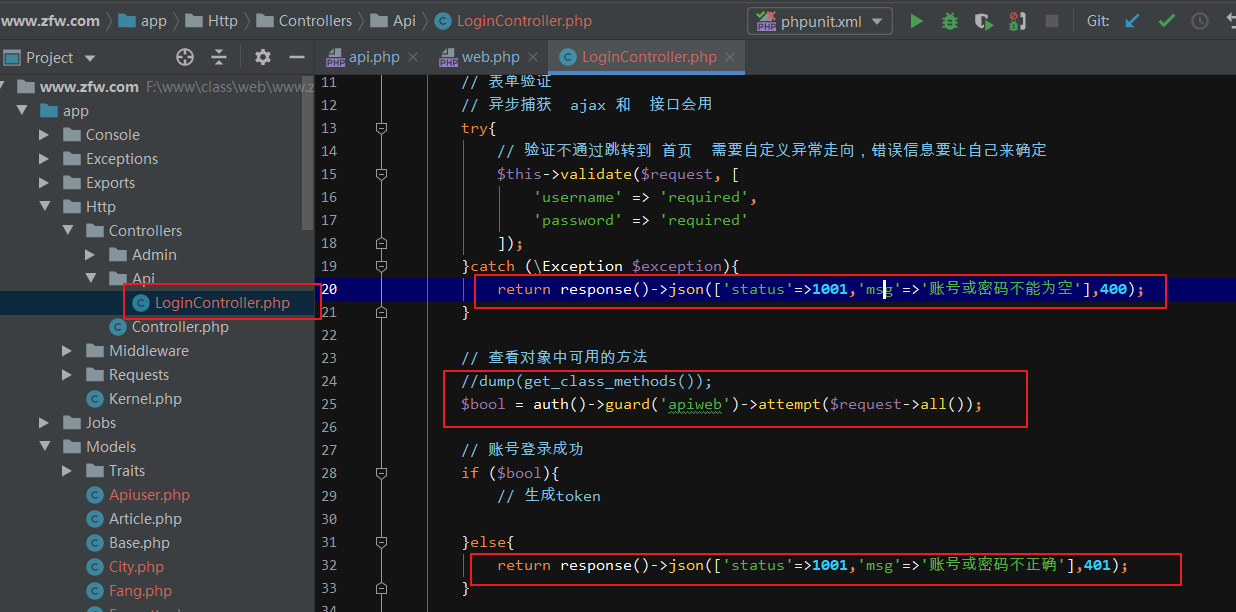
Passport::refreshTokensExpireIn(now()->addDays(30));



解决auth在接口中登录没有attempt问题

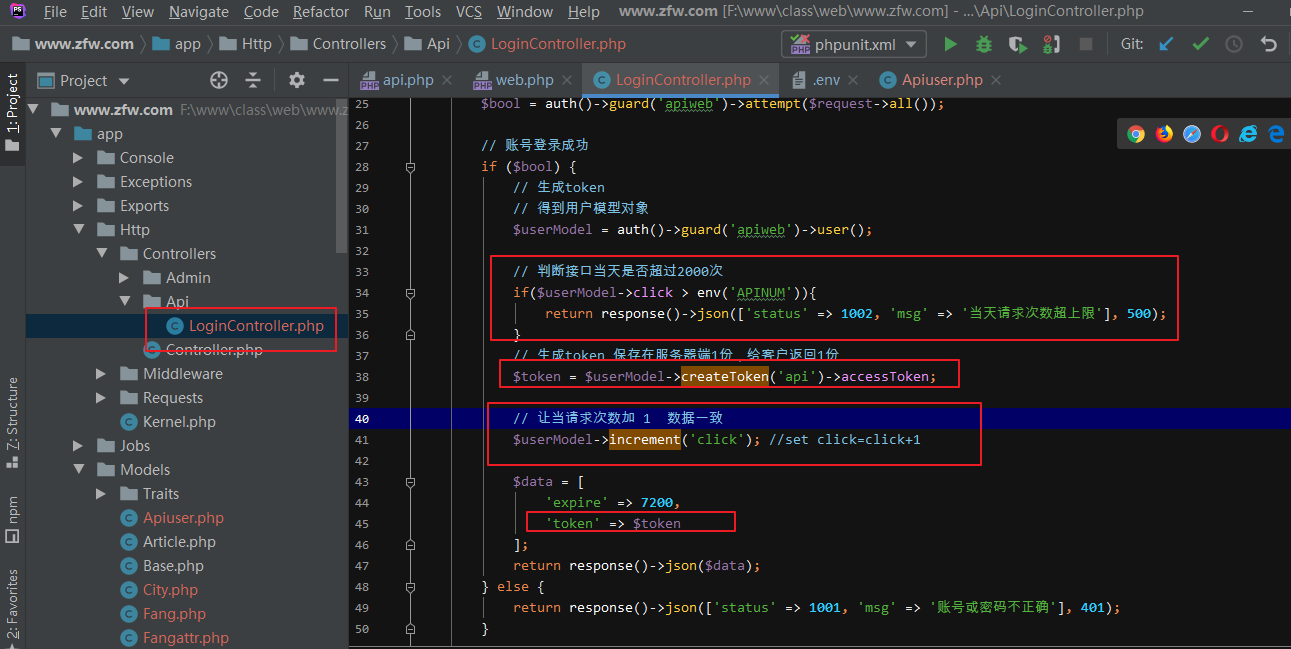


实现登录，并且注意，如果登录不成功，数据返回符合restful规范。

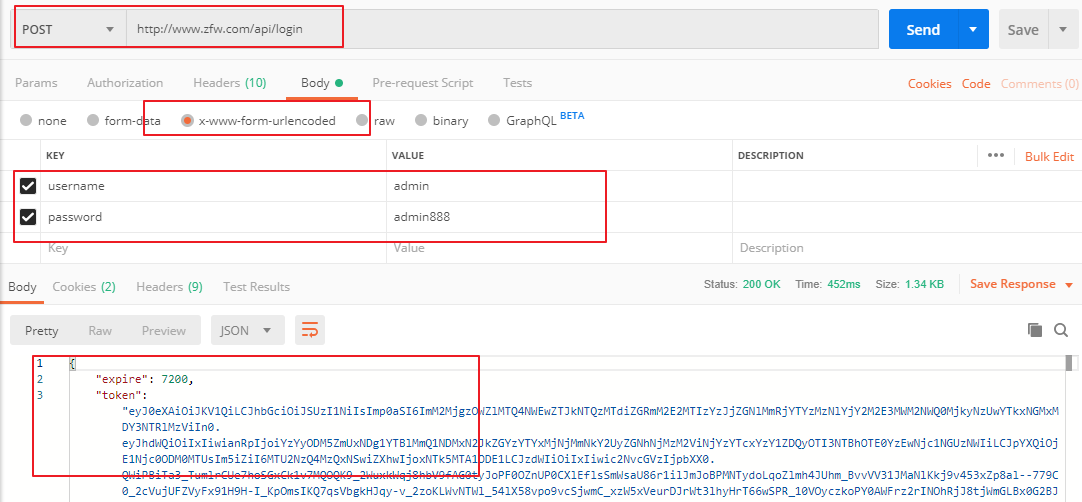


## 1.5、接口token生成

生成接口验证token在登录成功后生成。



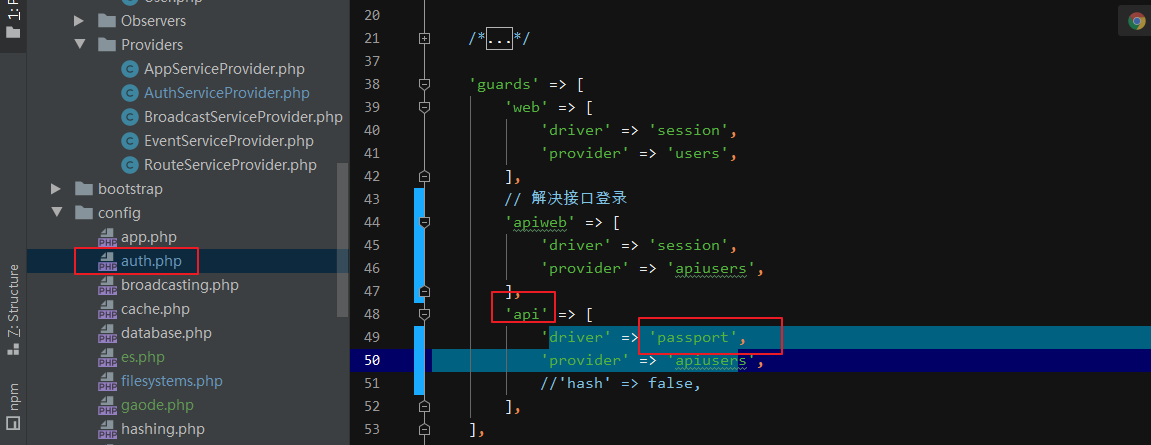
postman测试得到了token



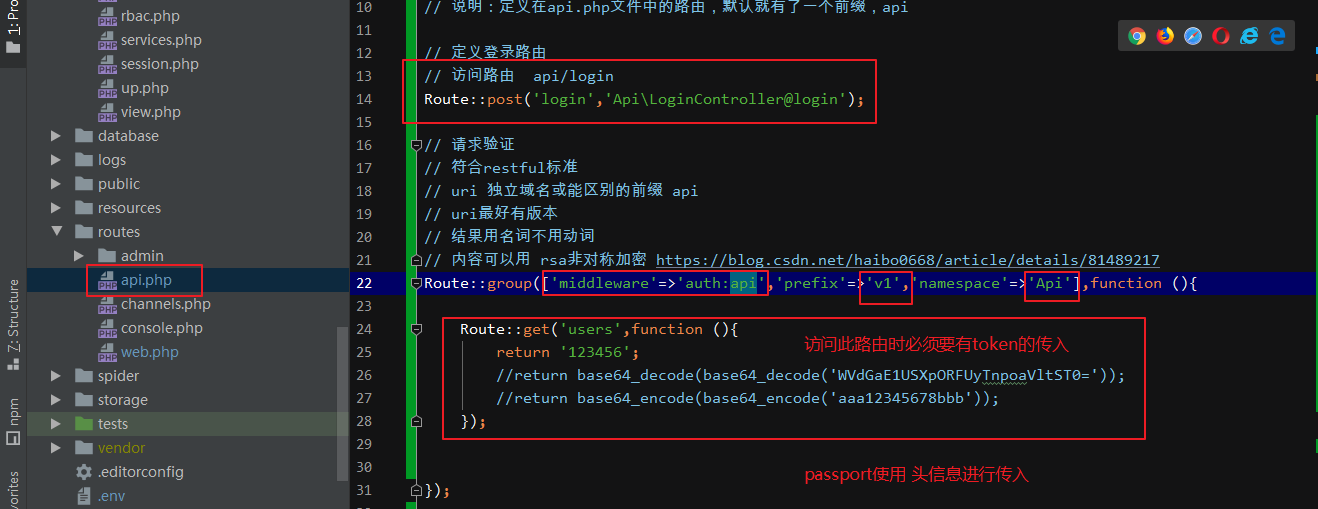
## 1.6、接口安全验证

使用了jwt进行验证

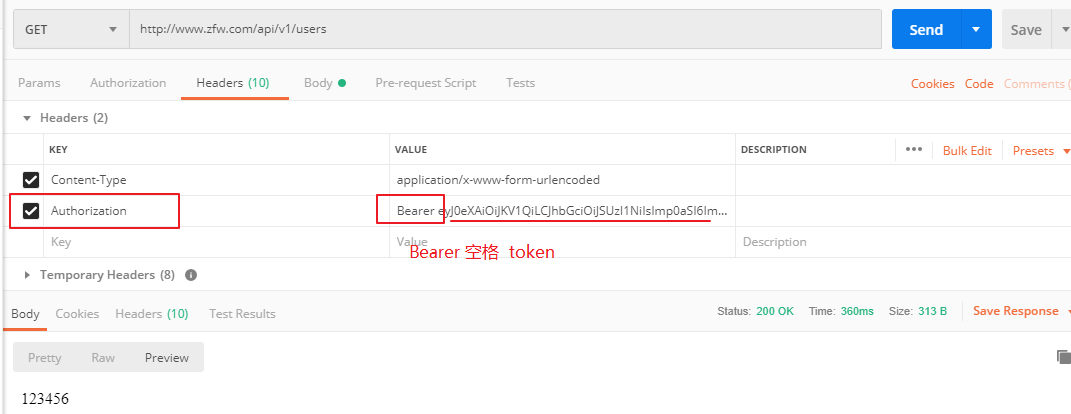
需要对接口路由绑定中间件,前提是auth.php文件中的一定要把api节点中的driver更换为passport



在定义路由中绑定jwt认证中间件

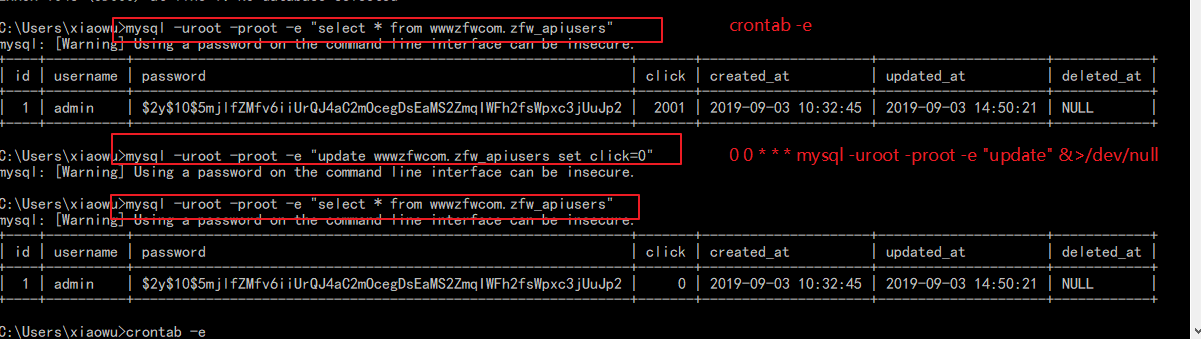


postman测试



**注：有的时候，不光光使用jwt来安全验证，还有时候还需要对内容进行加密，一般使用rsa加密。**

用户访问自动清零



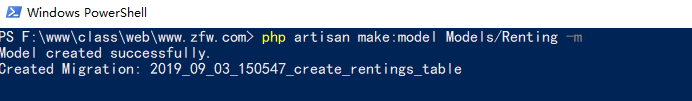
# 二、租客用户信息

## 2.1、租客表设计

通过效果图得到了设计的租客用户表



创建模型同时创建迁移文件



Schema::create('rentings', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('openid',50)->default('')->comment('小程序ID号');

$table->string('nickname',50)->default('')->comment('昵称');

$table->string('phone',15)->default('')->comment('手机');

$table->enum('sex',['男','女'])->default('男')->comment('性别');

$table->unsignedTinyInteger('age')->default(0)->comment('年龄');

$table->string('avatar',200)->default('')->comment('头像');

$table->string('card',18)->default('')->comment('身份证号');

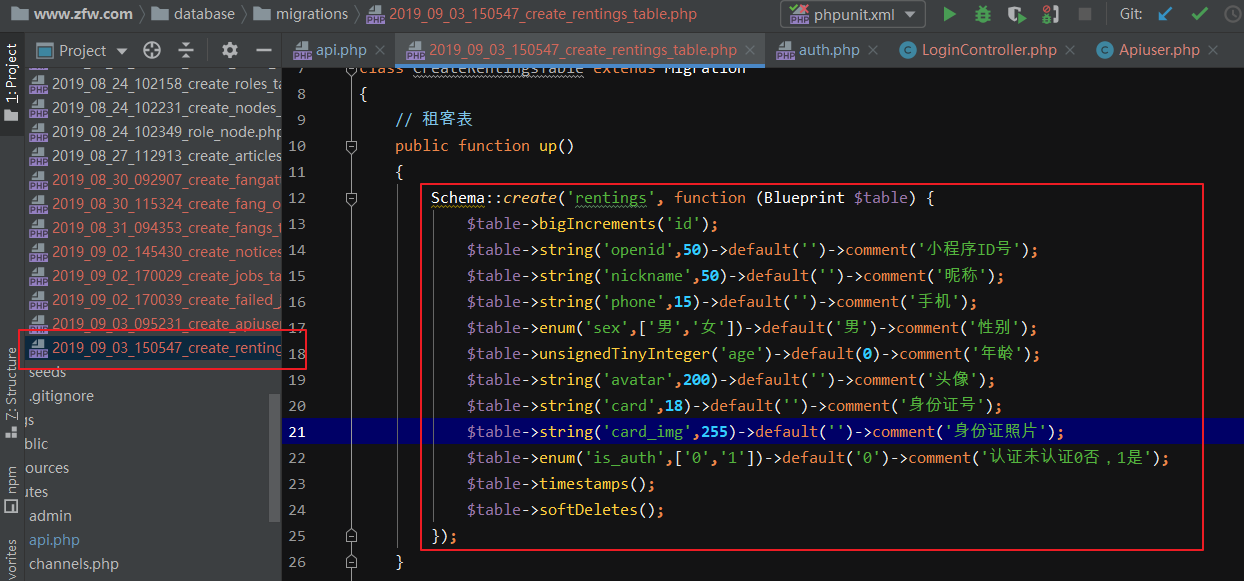
$table->string('card\_img',255)->default('')->comment('身份证照片');

$table->enum('is\_auth',['0','1'])->default('0')->comment('认证未认证0否，1是');

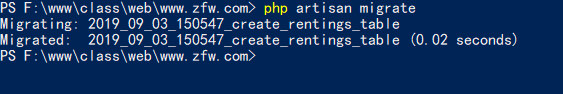
$table->timestamps();

$table->softDeletes();

});



执行迁移文件

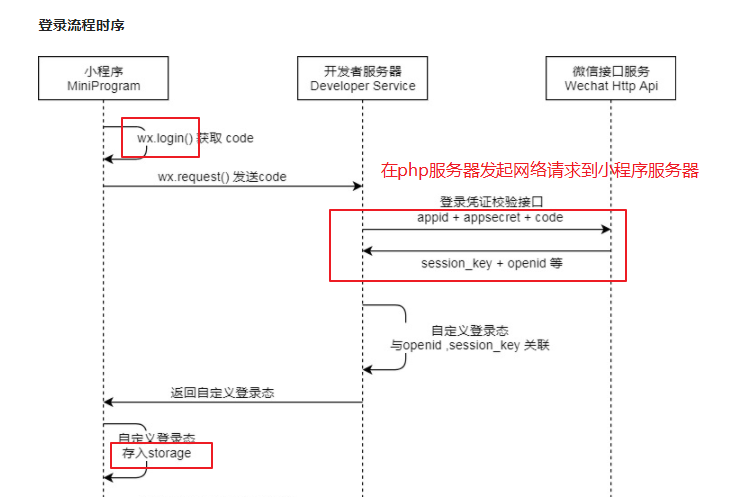


## 2.2、用户登录(添加租客)

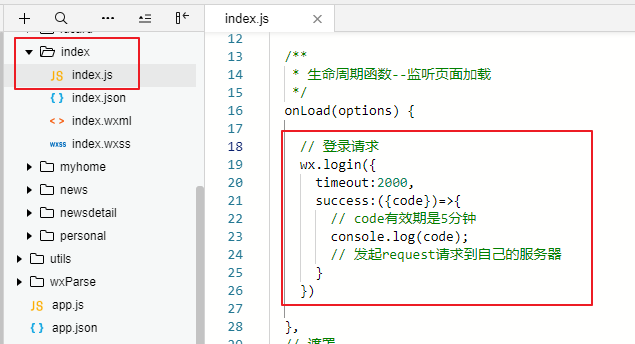
明确一定，小程序是如何实现得到openid，用户的小程序登录呢？

答：看官方示例文档：

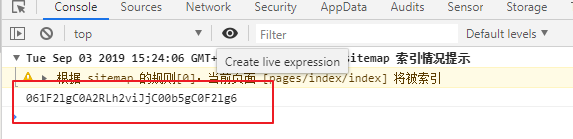
<https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/login.html>



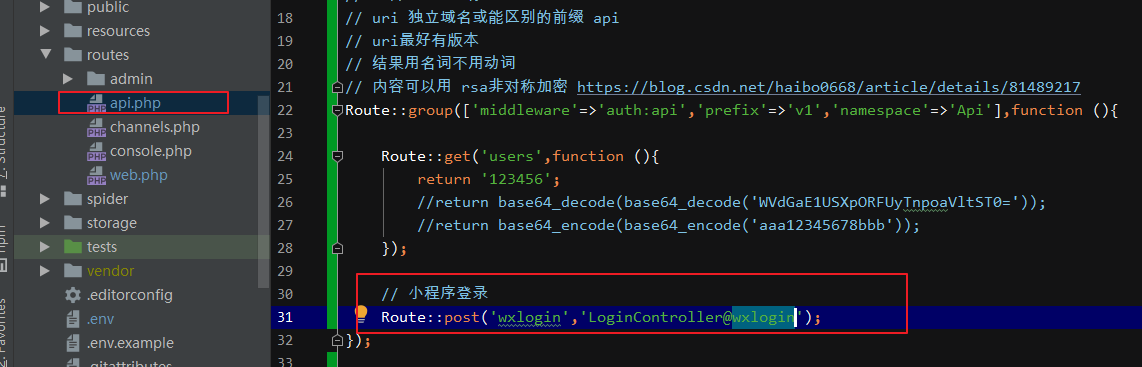
在小程序中随便找了一个页面，进行登录



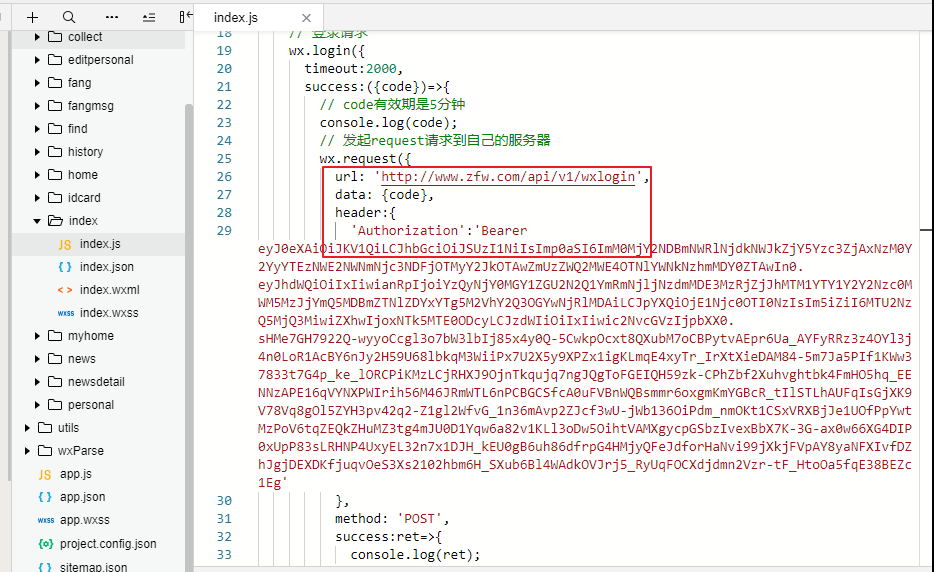
**得到code值，注：code有效期为5分钟**



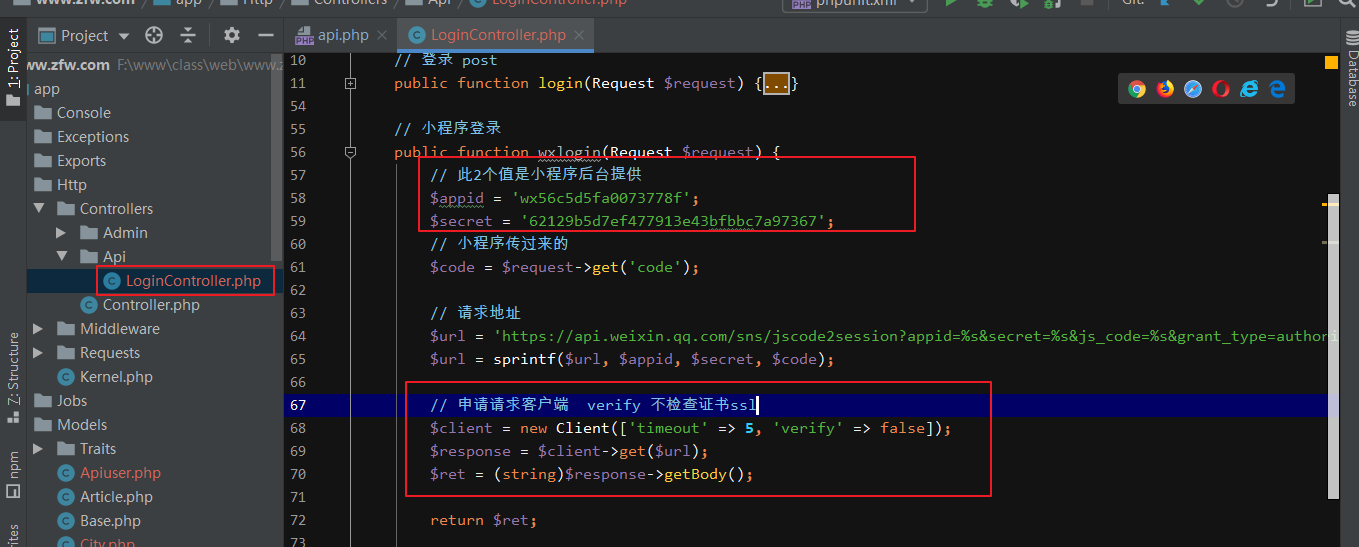
在自己接口服务器中定义登录处理的路由



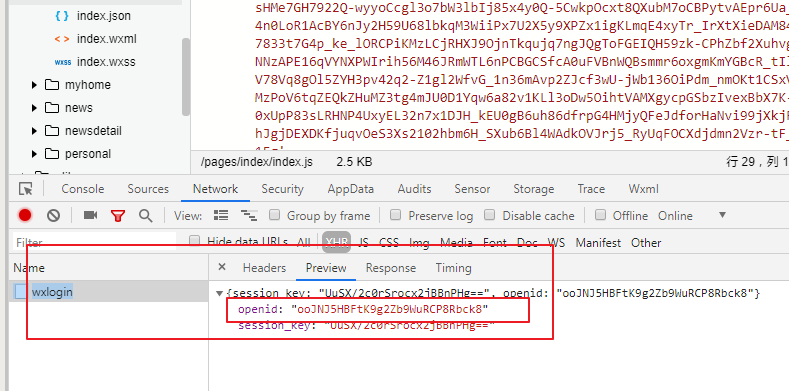
在登录控制器中创建小程序登录代码



在后台接受参数， 对小程序服务器发起GET请求得到openid

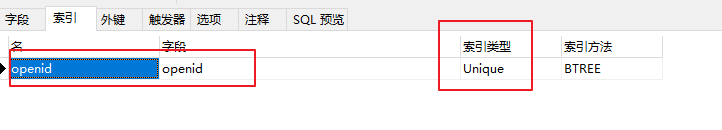


在小程序登录请求中就可以得到了openid

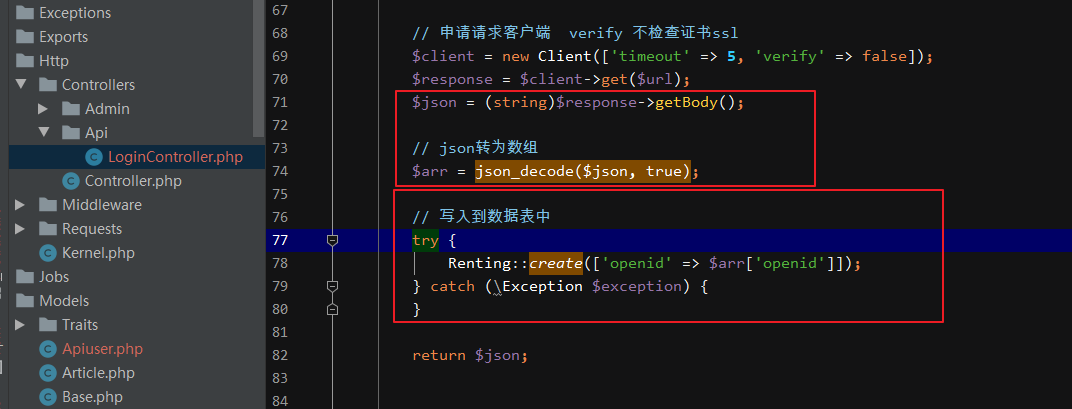


在服务器端，登录成功后，返回得到的小程序中，进行入库操作，使用了一点小技巧，数据库索引，唯一索引(优点：查询速度快，数据不能重复)

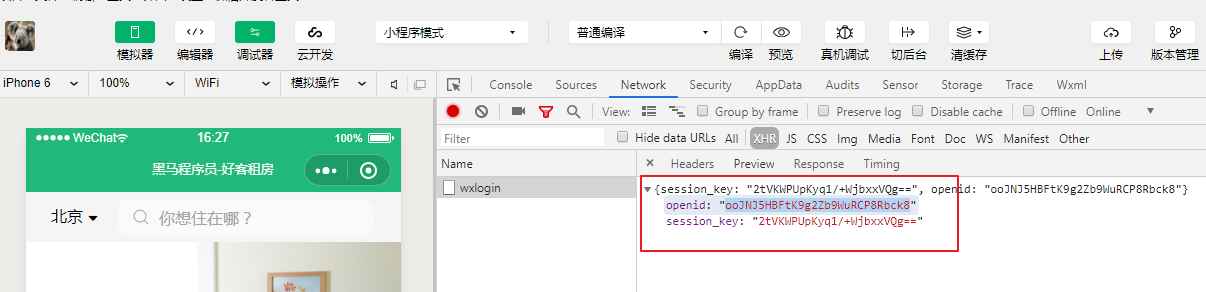
给租客表中的openid添加唯一索引



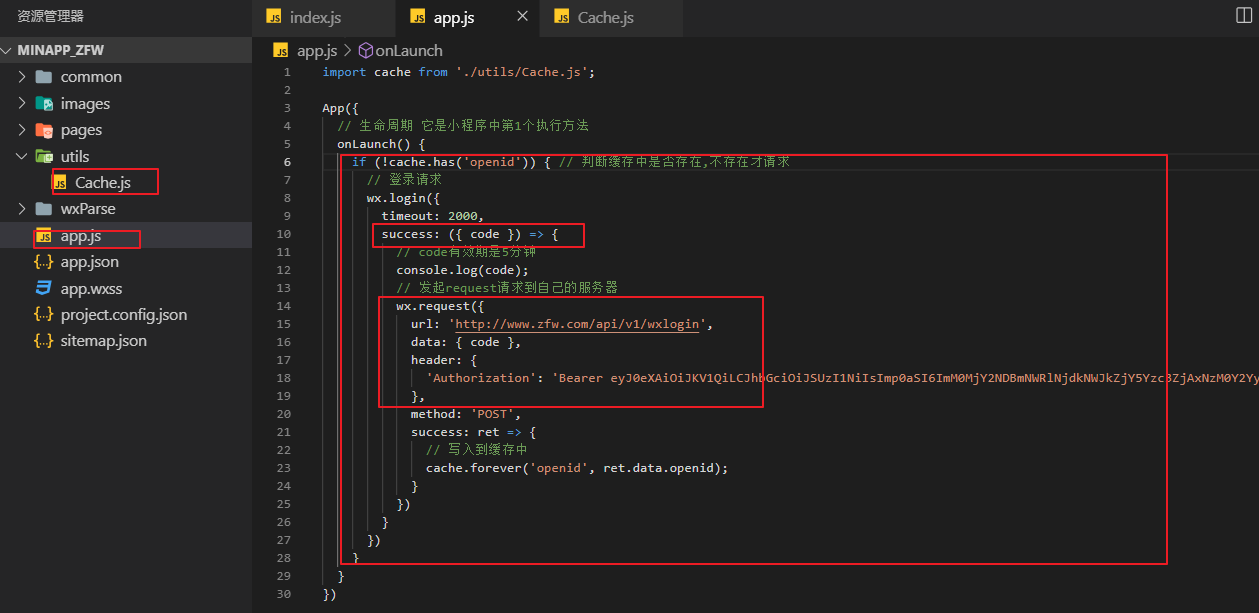
在得到openid后，进行入库操作，但是有唯一索引的存在，可以添加时会有异步的出现，所以使用try catch来捕获异步



在小程序中还是可以得到openid



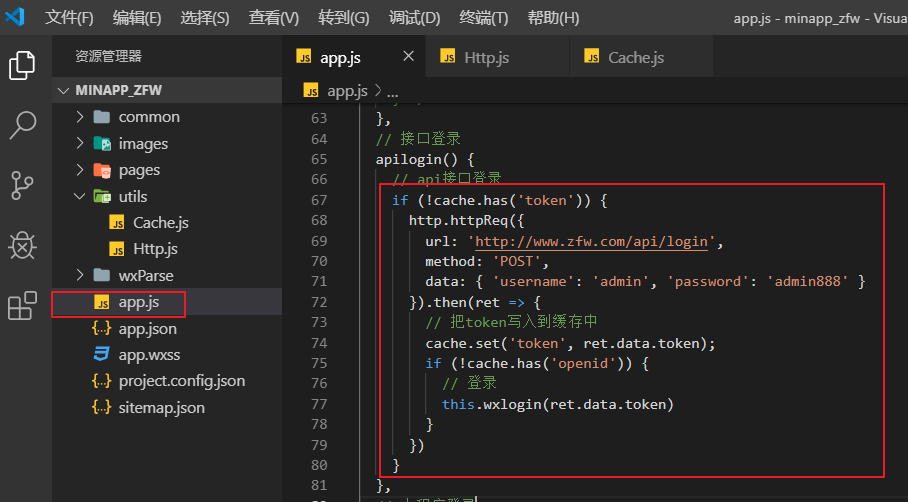
在小程序端，小程序的登录是应该在所有的页面都执行的文件中去执行，app.js文件就是入口文件。



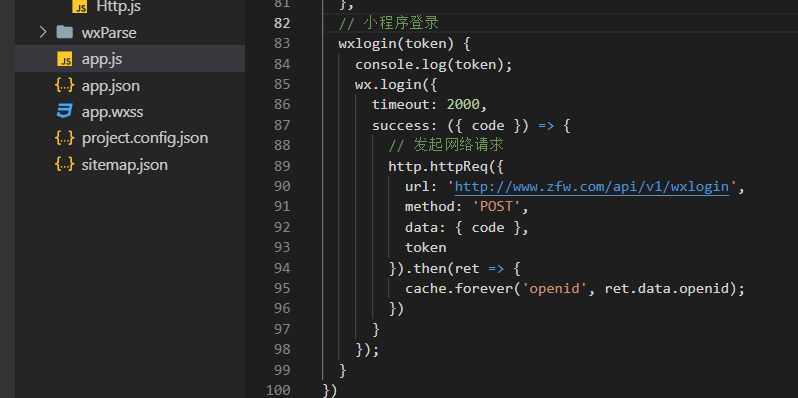
注：请求登录完成后，一定要写入到缓冲中，不要每次打开页面都进行登录网络请求。

小程序中实现api接口登陆和小程序登录

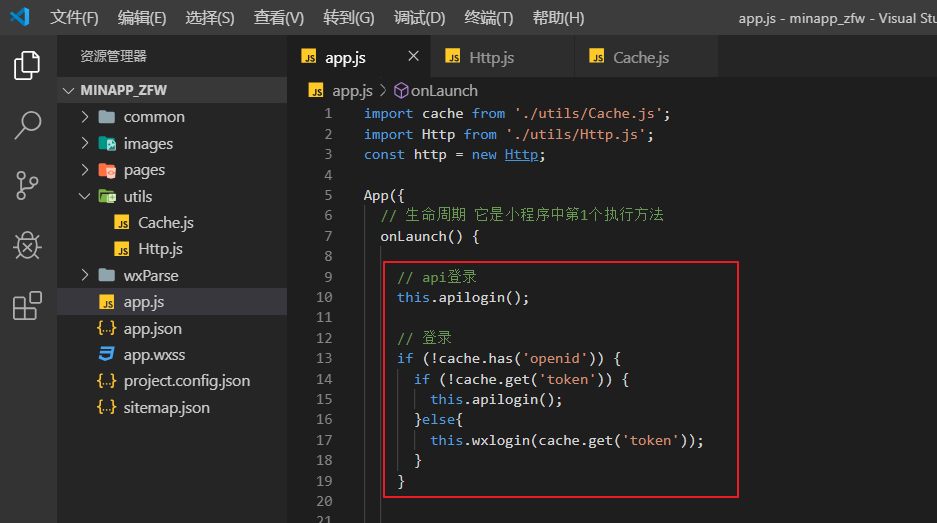
小程序的接口登陆



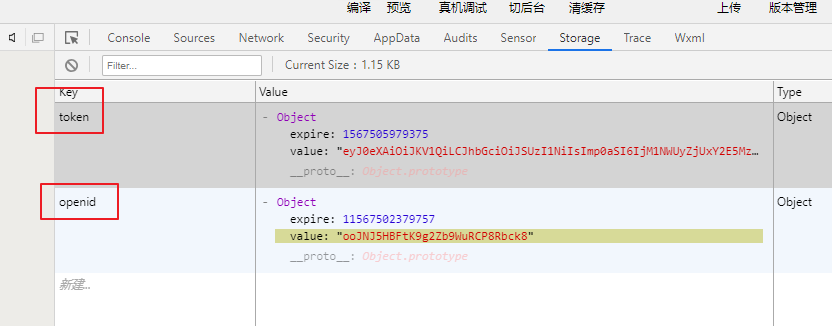
小程序登录业务



登录



小程序缓存中就得到数据



## 2.3、用户授权

## 2.4、个人资料

## 2.5、身份认证

## 2.6、看房通知