# 好客租房\_03

# 今日目标5

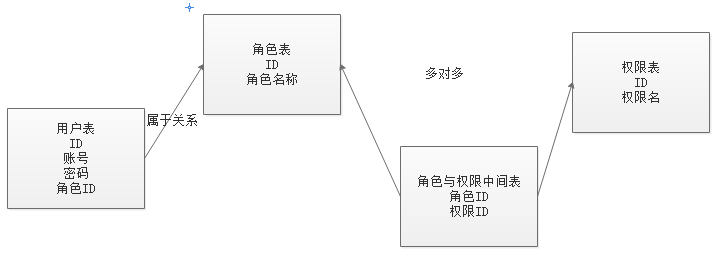
* 理解角色与权限表的设计
* 能够实现角色表的增删改查操作
* 能够实现给角色分配权限操作
* 能够使用中间件检查用户权限
* 能够实现后台菜单的动态读取显示

# 一、RBAC

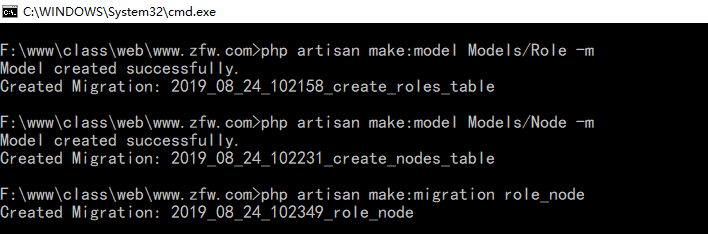
RBAC: role base access control 基于角色的用户访问权限控制权限，就是权限分配给角色，角色又分配给用户。

即一个用户对应一个角色，一个角色对应多个权限，一个用户对应用户组，一个用户组对应多个权限。

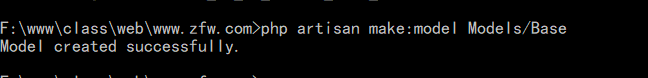
## 1.1、角色与权限表的设计

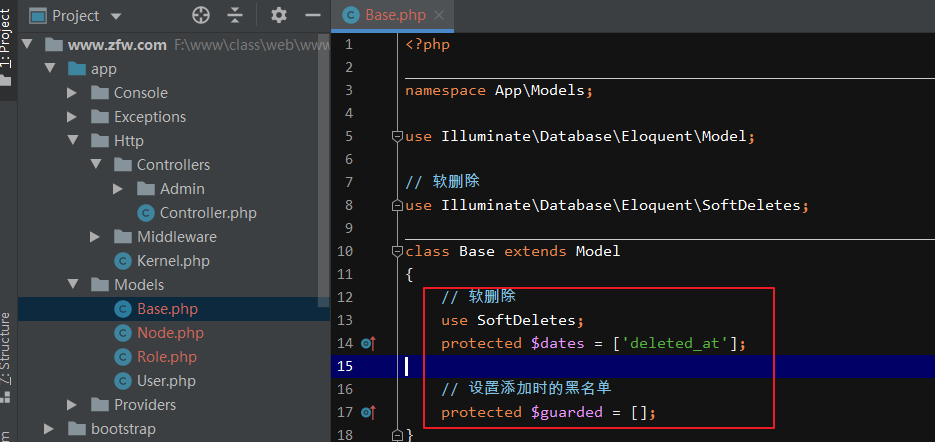


创建迁移文件和模型

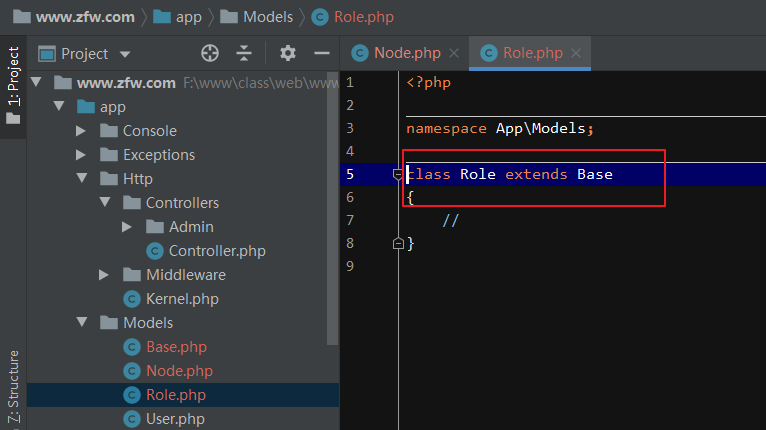


创建基础模型

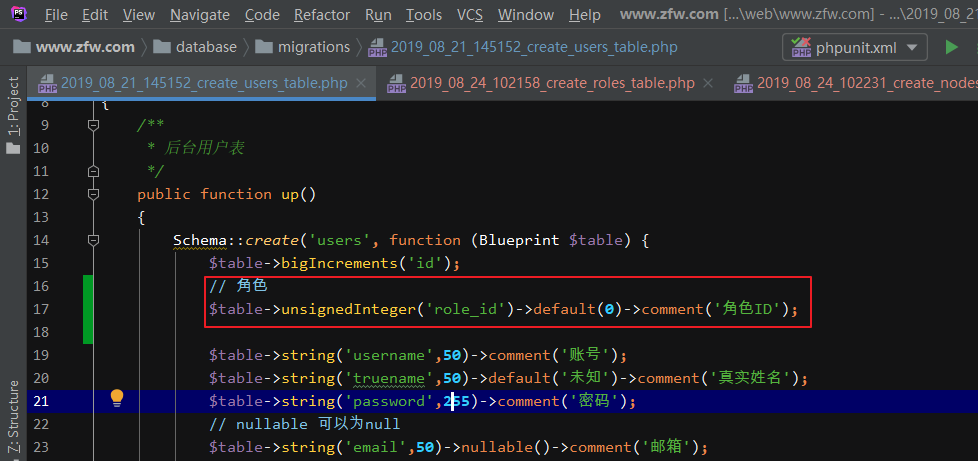




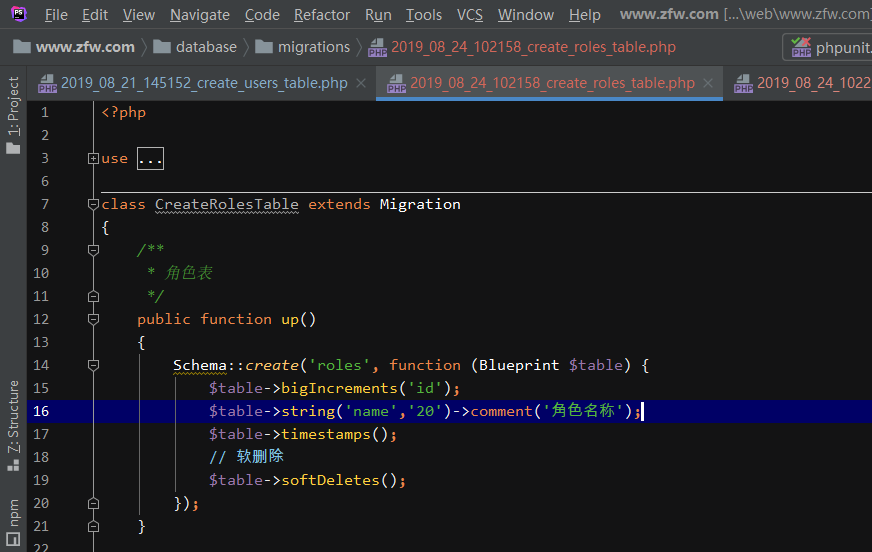
修改角色与权限模型的继承



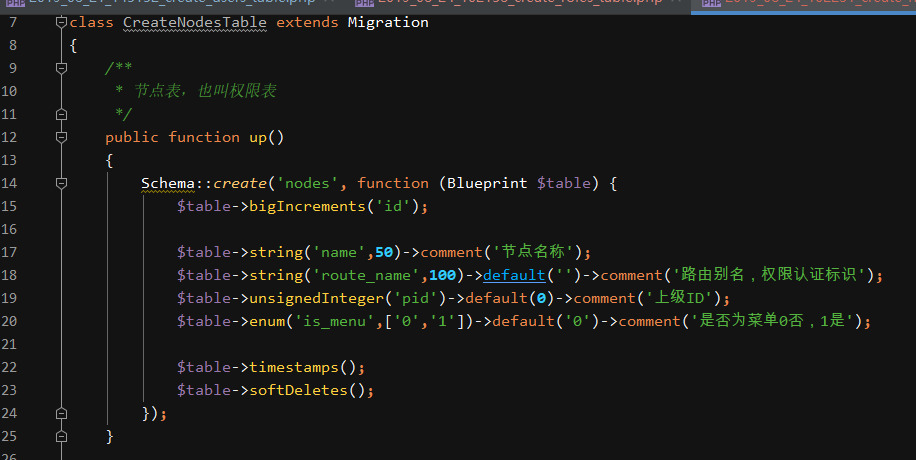
修改用户迁移文件添加角色ID



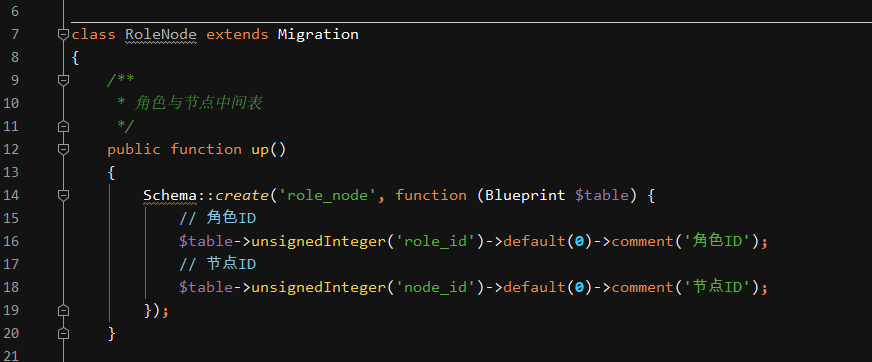
角色表



节点表(权限表)

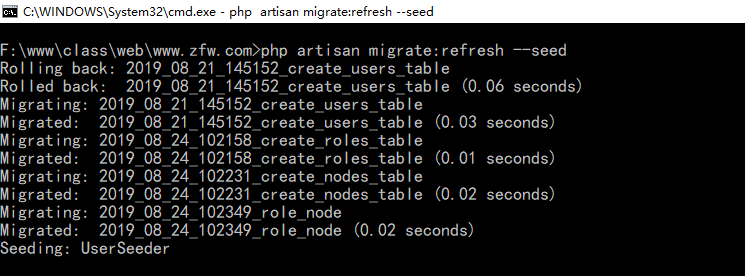


角色与节点中间表



执行迁移文件

php artisan migrate:refresh --seed



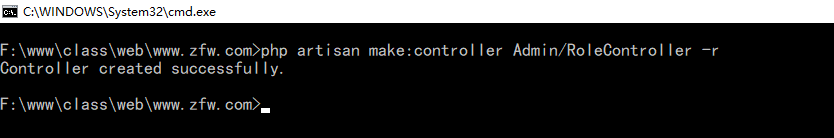
## 1.2、角色管理

### 1.2.1、角色列表

创建一个资源控制器和资源路由

# 创建资源控制器

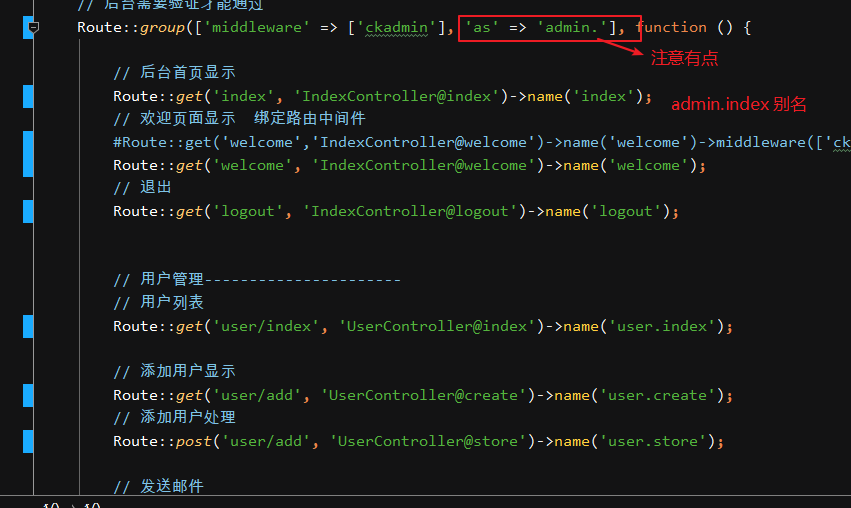
php artisan make:controller Admin/RoleController -r/--resource

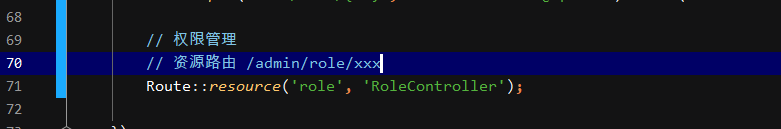


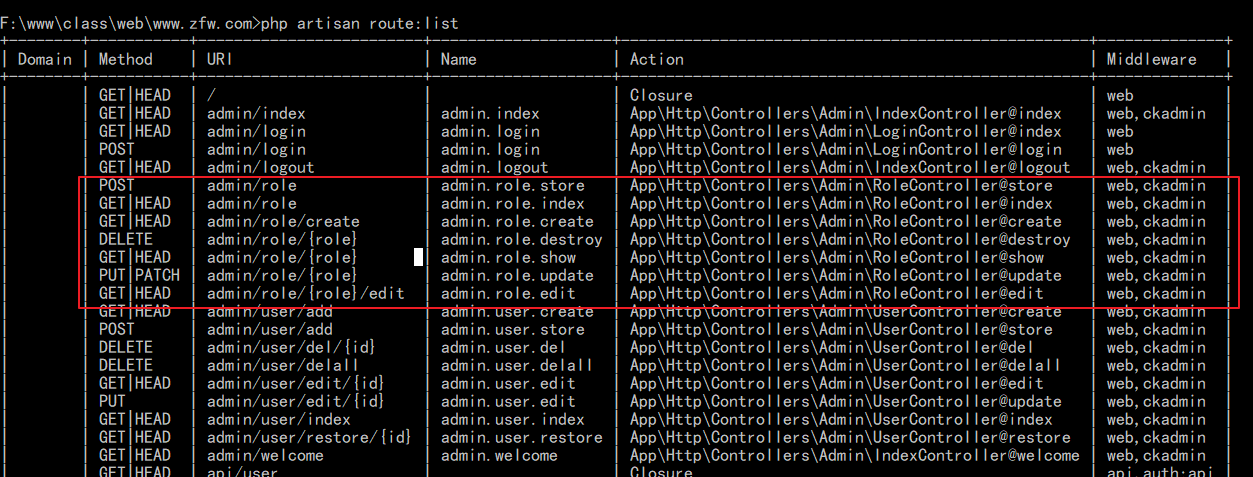
定义资源路由

Route::resource('规则前缀', '控制器名称不要方法'); 定义路由前缀

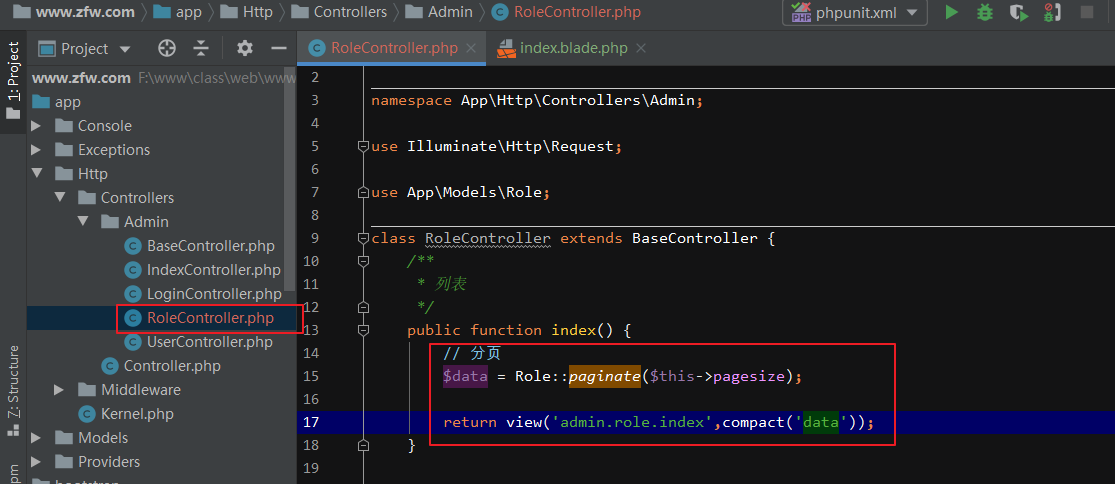
在路由分组中定义统一路由前缀



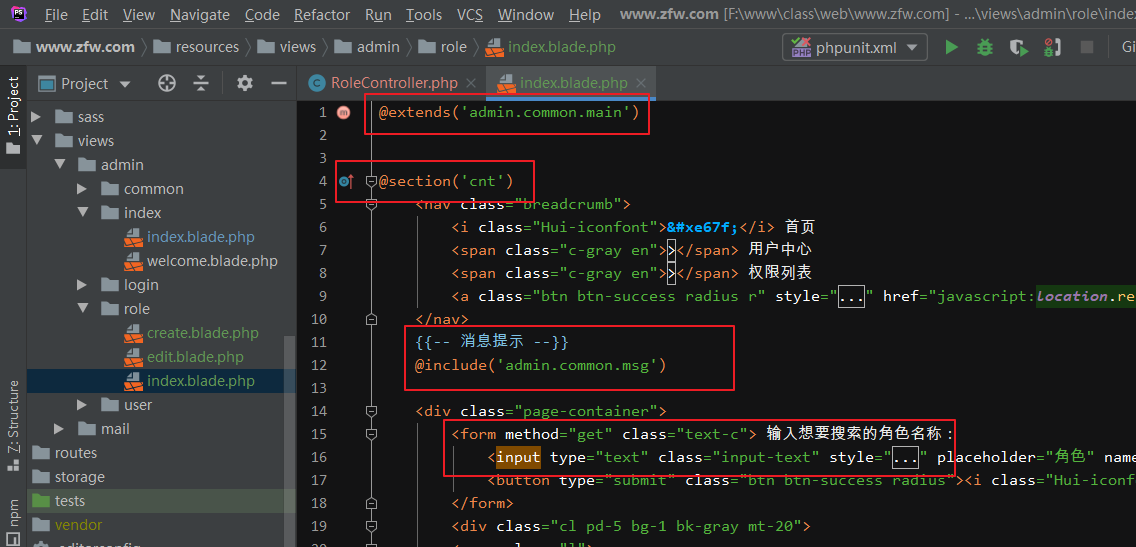




控制取数据



模板



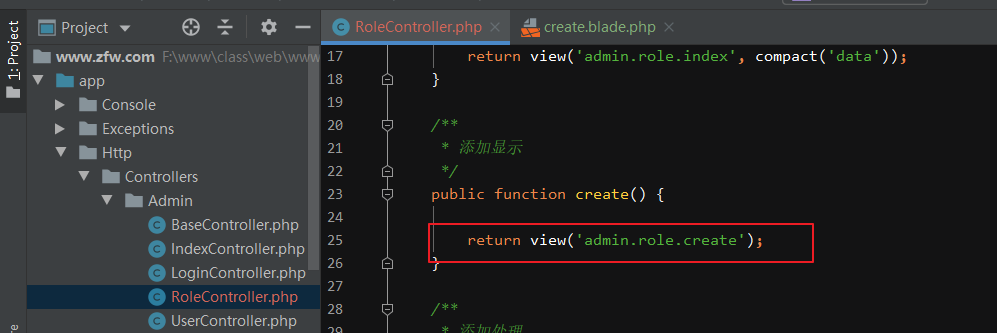


效果

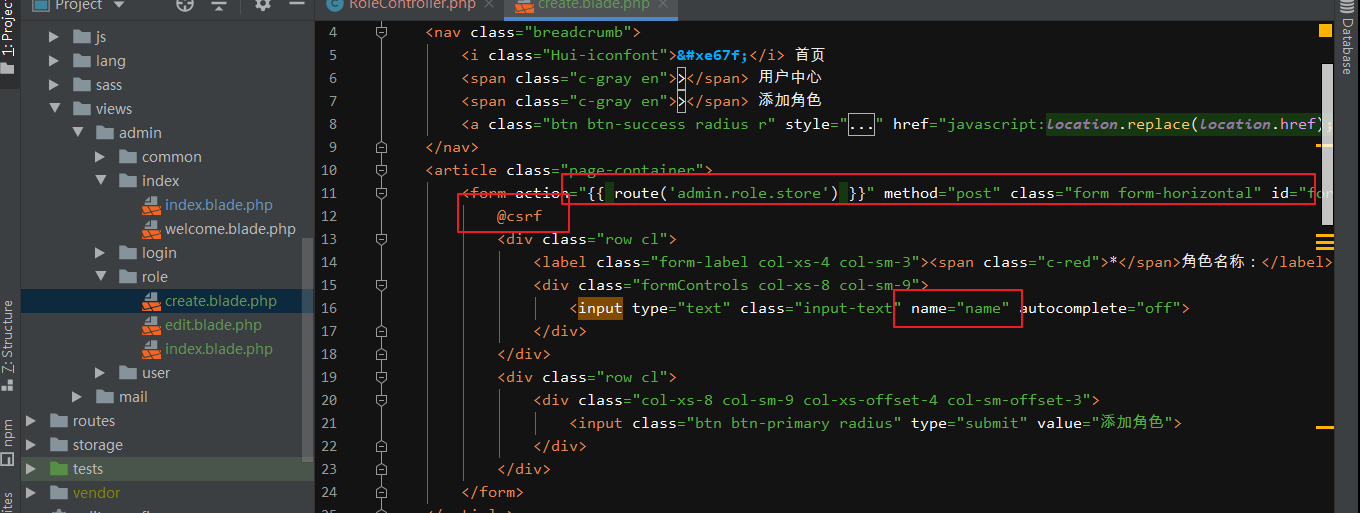


### 1.2.2、添加角色

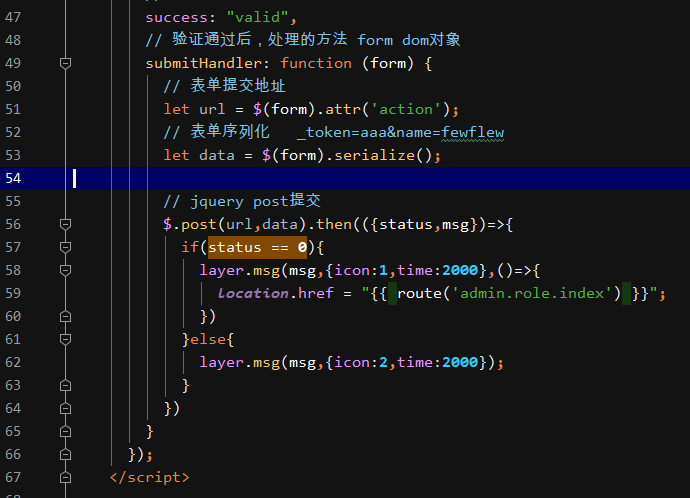
显示添加界面



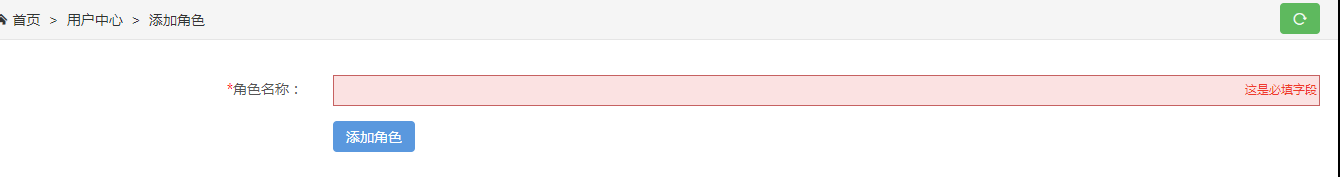
模板 ajax提交







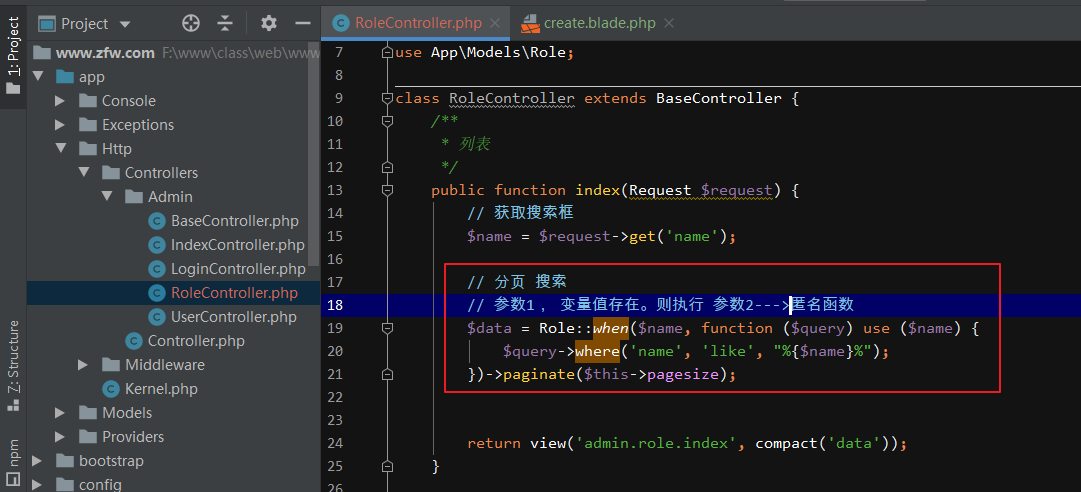
效果





### 1.2.3、搜索角色

使用模型查询中的when方法

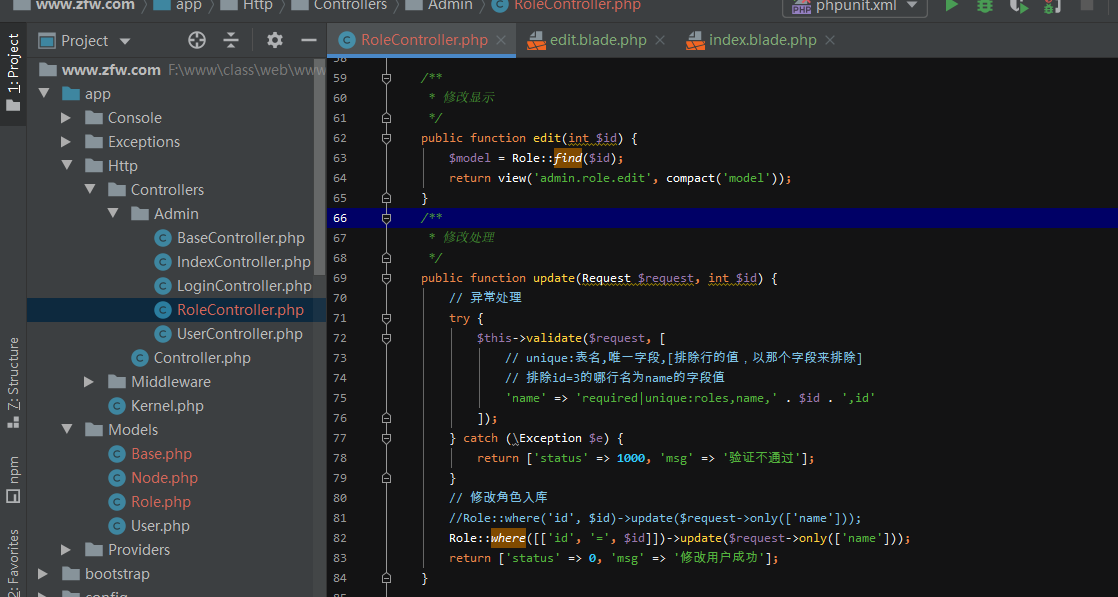


效果

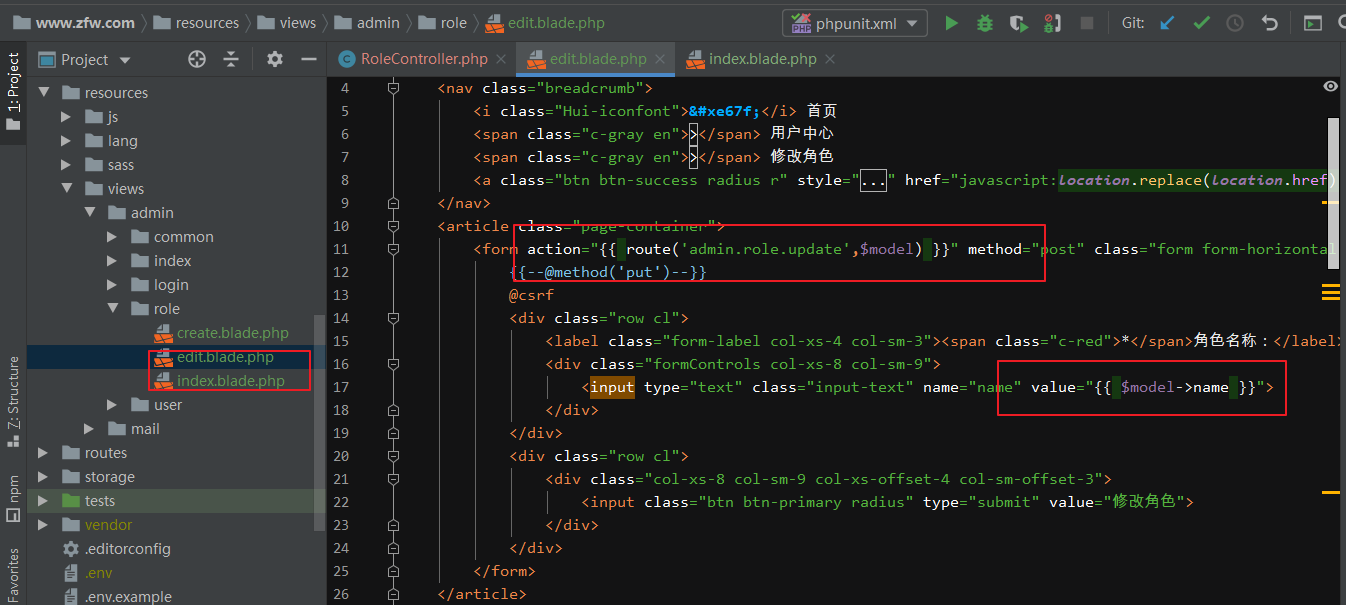


### 1.2.3、修改角色

修改控制器



模板



效果



### 1.2.4、删除角色

## 1.3、权限(节点)管理

### 1.3.1、节点列表

节点列表不分页

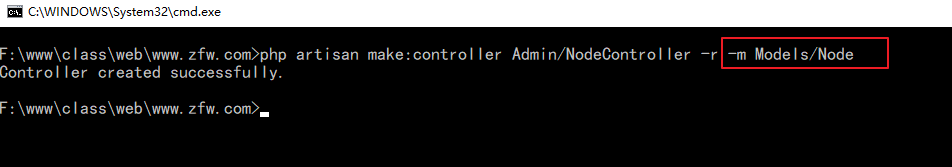
定义资源路由



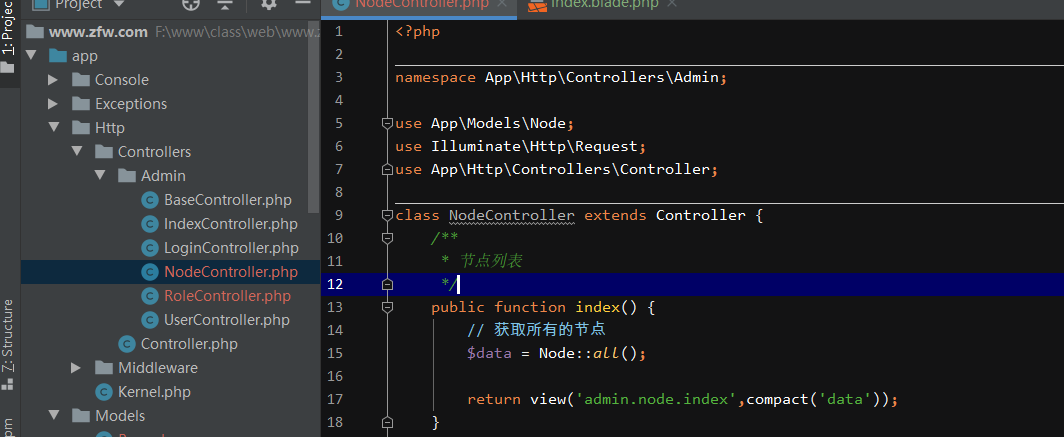
创建资源控制器，并关联模型

php artisan make:controller Admin/NodeController -r -m Models/Node

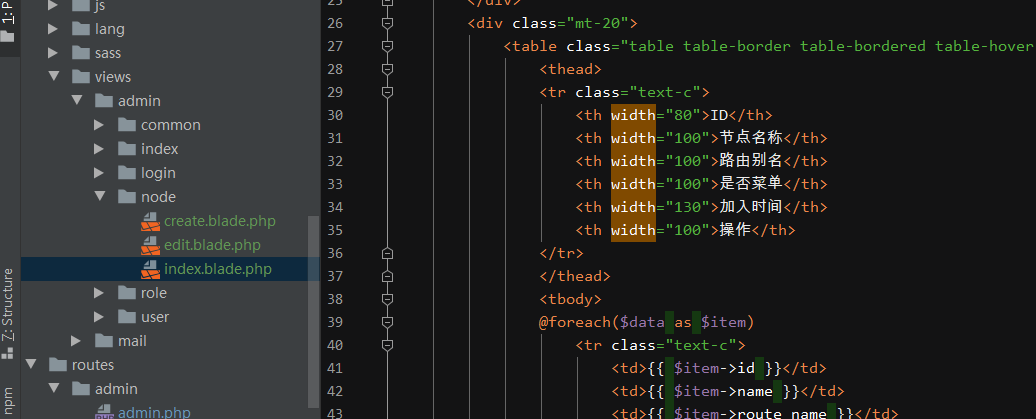
-m 模型



控制器



模板



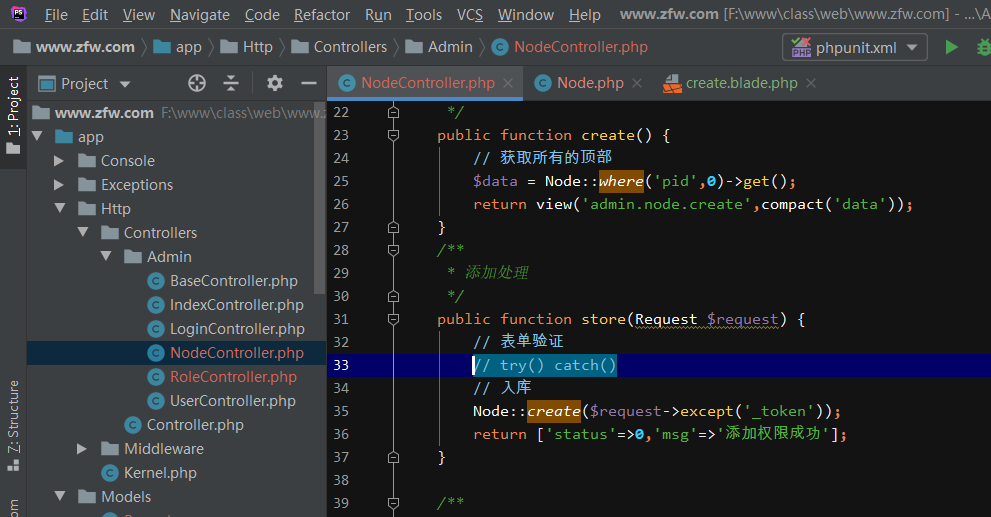
效果



### 1.3.2、添加节点

使用vue进行界面管理，实现添加功能

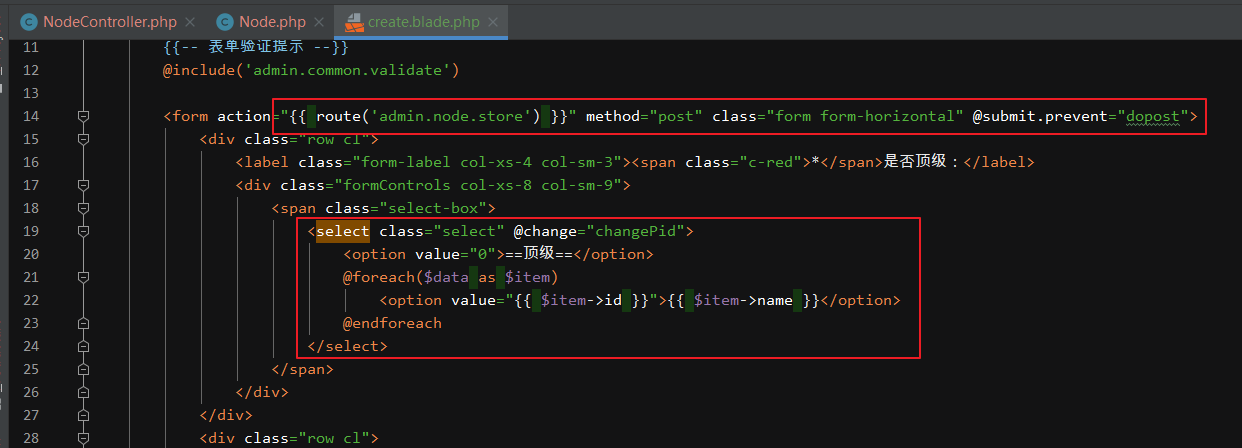
控制器

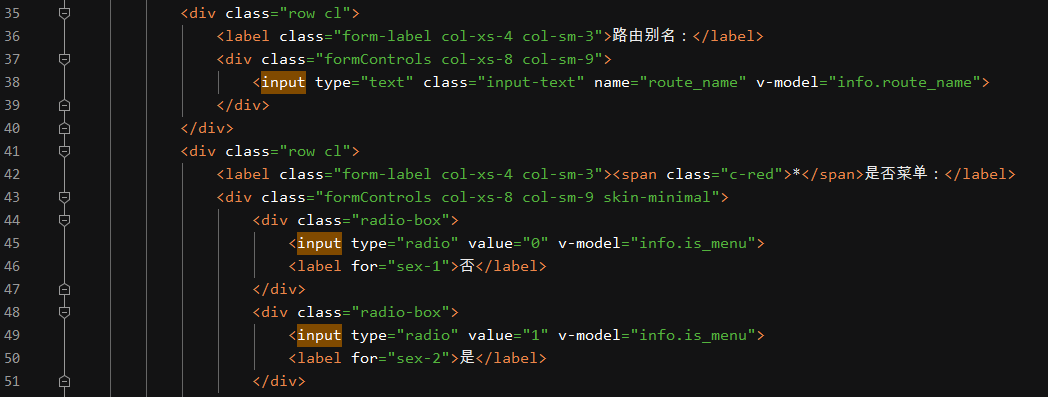


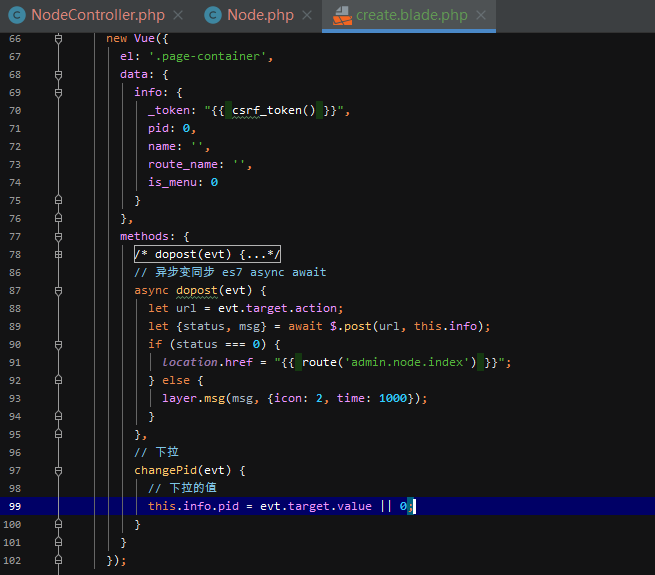
模型中定义修改器



模板

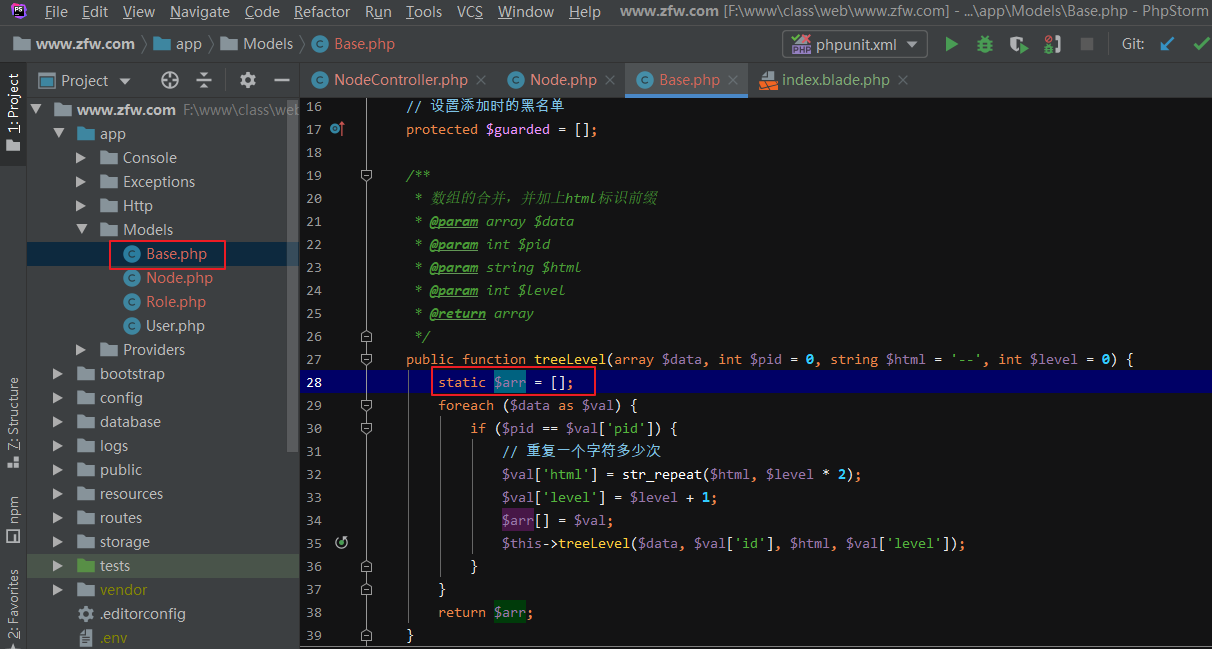




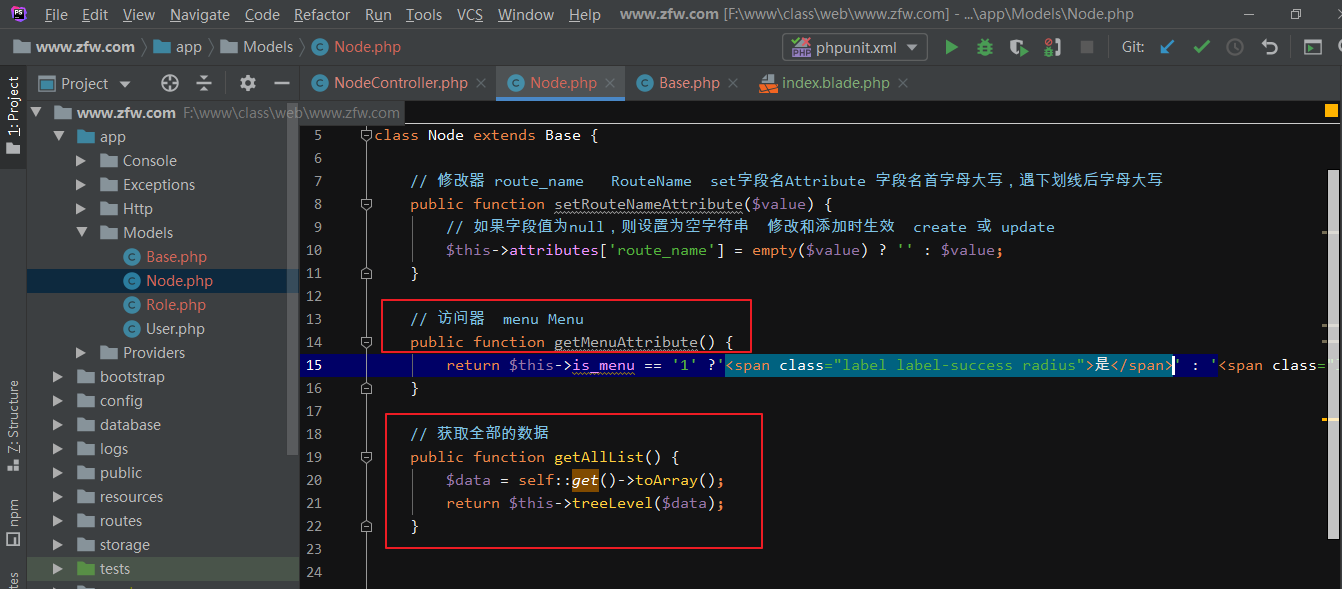


### 1.3.3、节点列表层级显示

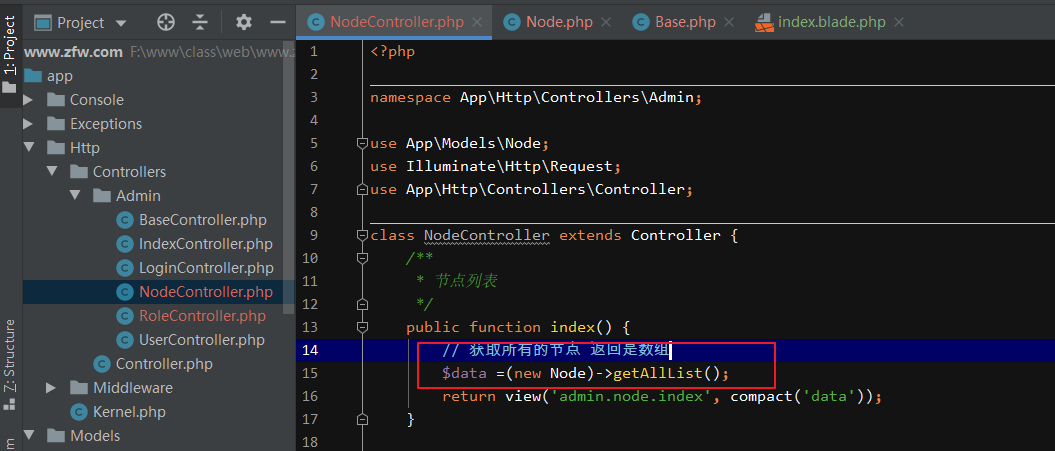
定义一个递归层级显示方法



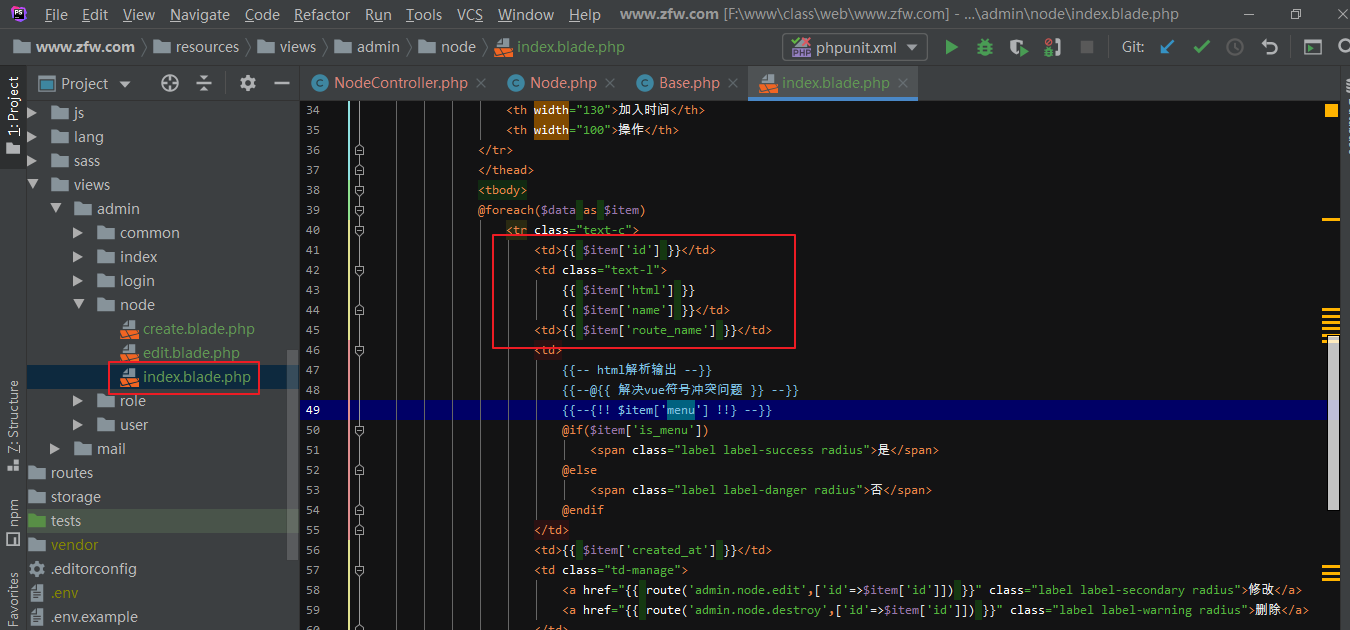
节点模型中获取数据



控制器中调用



模板



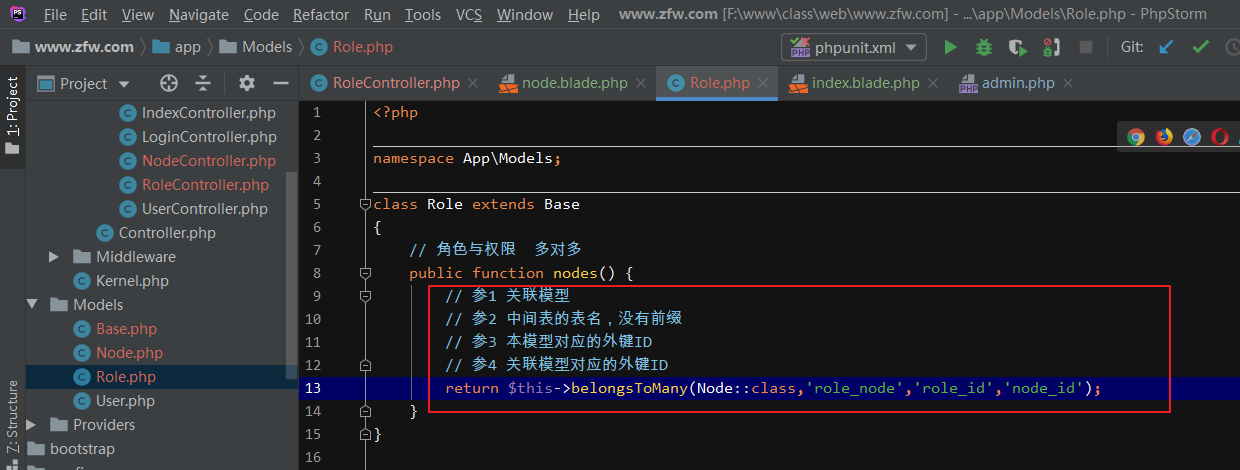
显示



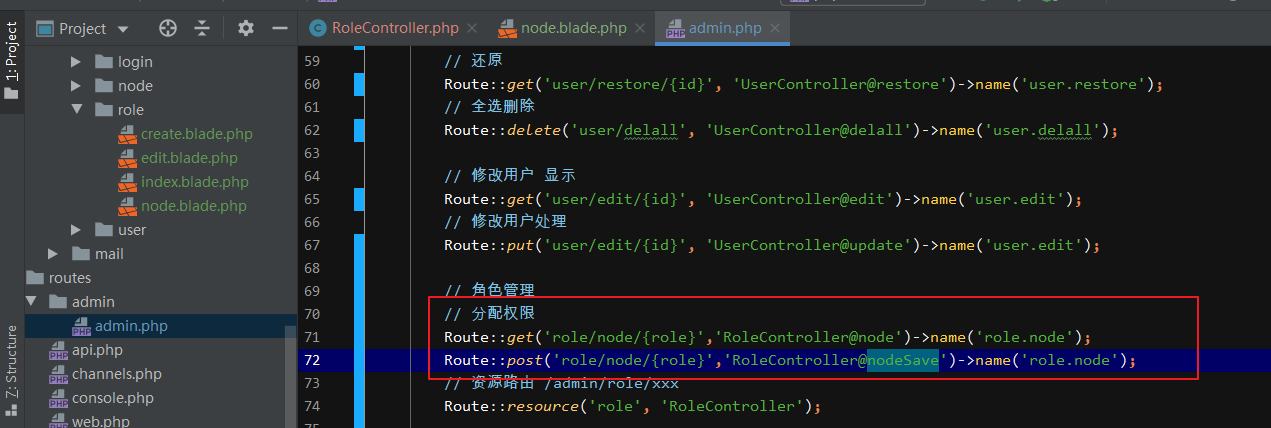
## 1.4、后台权限控制

### 1.4.1、给角色分配权限

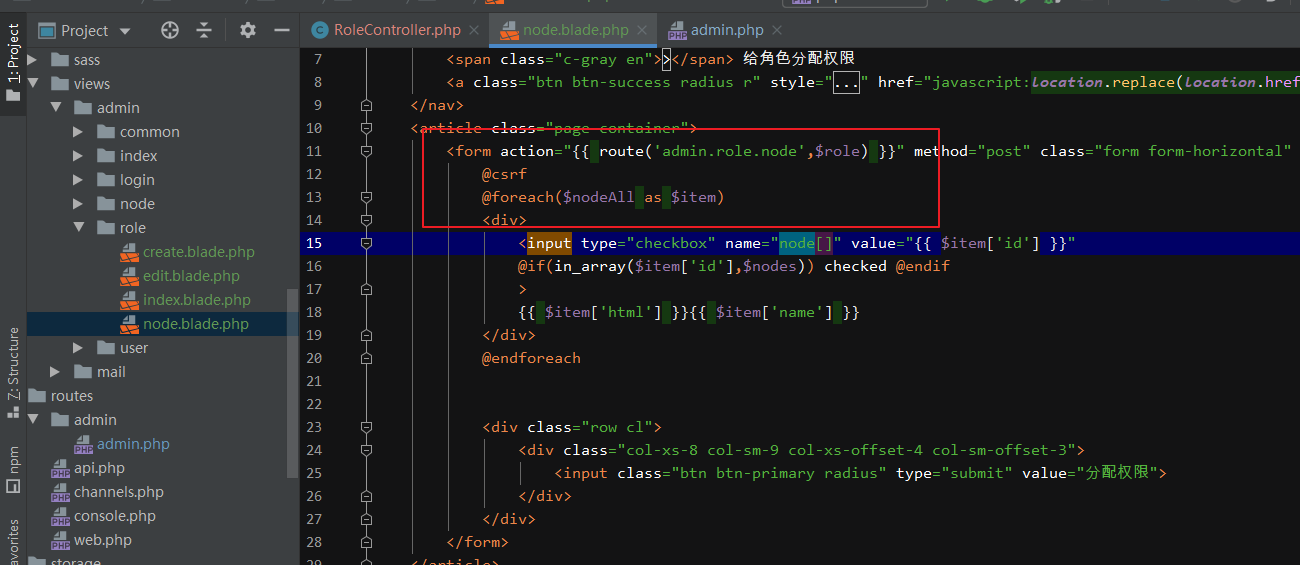
角色与权限之间的关系： 多对多关系 belongsTomany



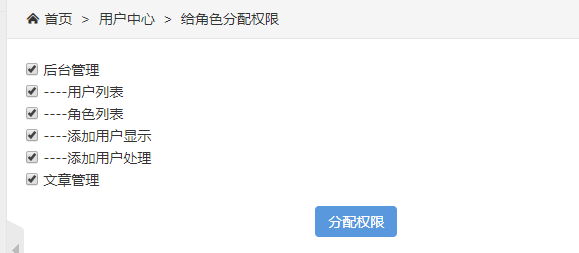
定义路由



模板



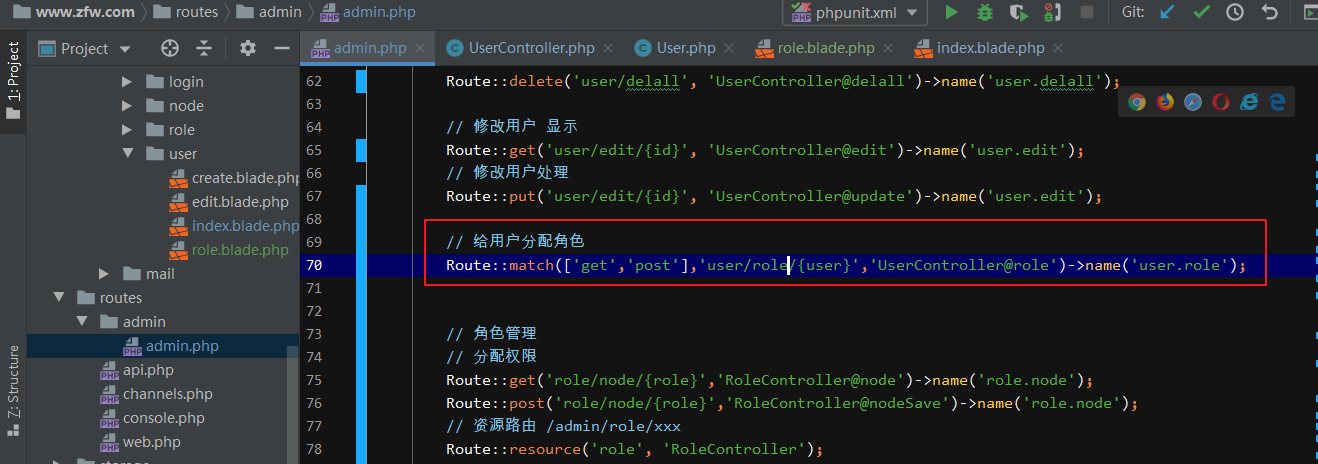
效果



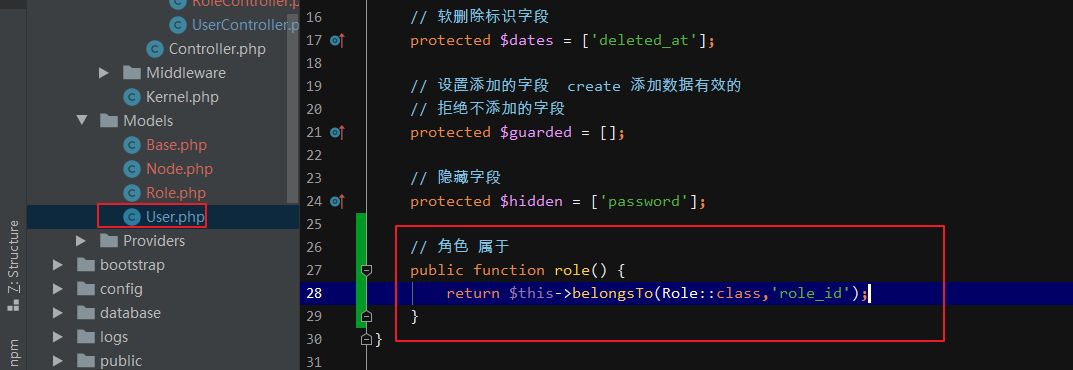
### 1.4.2、给用户分配角色

用户和角色之间是关系： 属于关系 beLongsTo

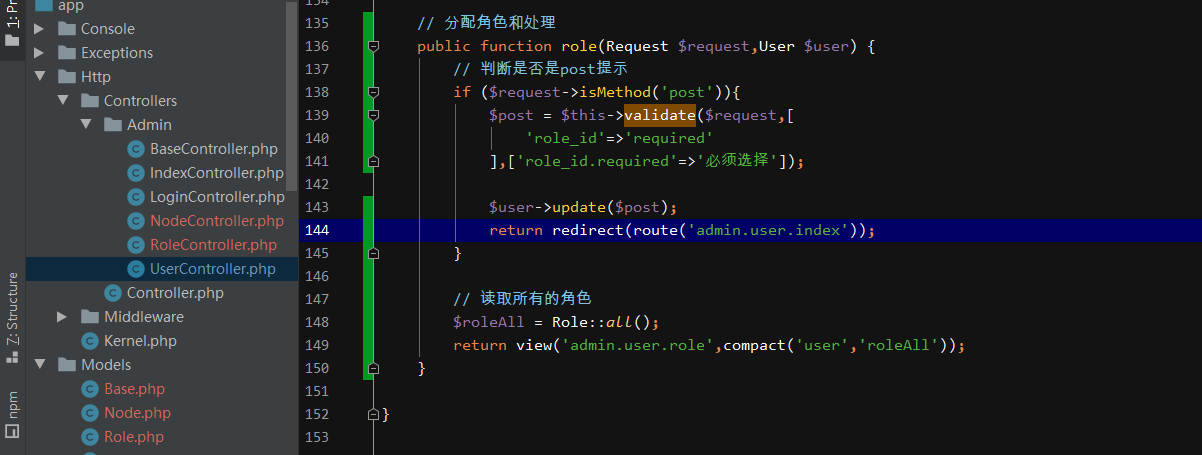
定义路由



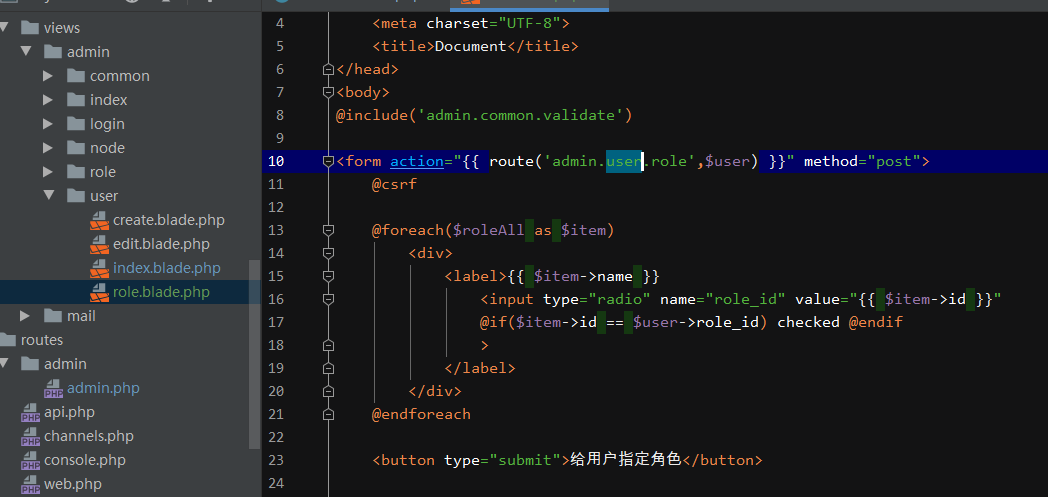
模型



控制器



模板



效果



## 1.5、后台菜单显示