# ThinkPHP框架\_03

# 今日目标5

* 能够熟悉使用数据查询方法

find

select

* 能够使用TP5框架实现分页功能

paginate

* 能够使用TP5框架的验证码组件

composer 安装 require 包名

模板中

<img src={:captcha\_src} />

验证

2种

验证器

独立验证

* 能够使用TP5框架的验证器组件

独立

$validate = Validate::make(rule,message);

$validate->check($\_post);

验证器

php think make:validate 模块/验证器名称

* 能够在TP5框架中使用cookie和session
* cookie()
* session()
* 中间件

php think make:middleware 中间件的名称

注册3种

控制器 pathinfo

路由

配置文件

# 一、模型操作

## 1.6、查询数据

# 查询单条记录

$user = User::where('name', 'thinkphp')->find();

# 查询多条记录

$list = User::where('status', 1)->limit(3)->order('id', 'asc')->select();

# 获取某个字段或者某个列的值

// 获取某个用户的积分

User::where('id',10)->value('score');

// 获取某个列的所有值

User::where('status',1)->column('name');

// 条件分组 where () or ()

$ret = Articles::where(function (Query $query){

#print\_r(func\_get\_args());

$query->where('id','>',200);

})->whereOr(function (Query $query){

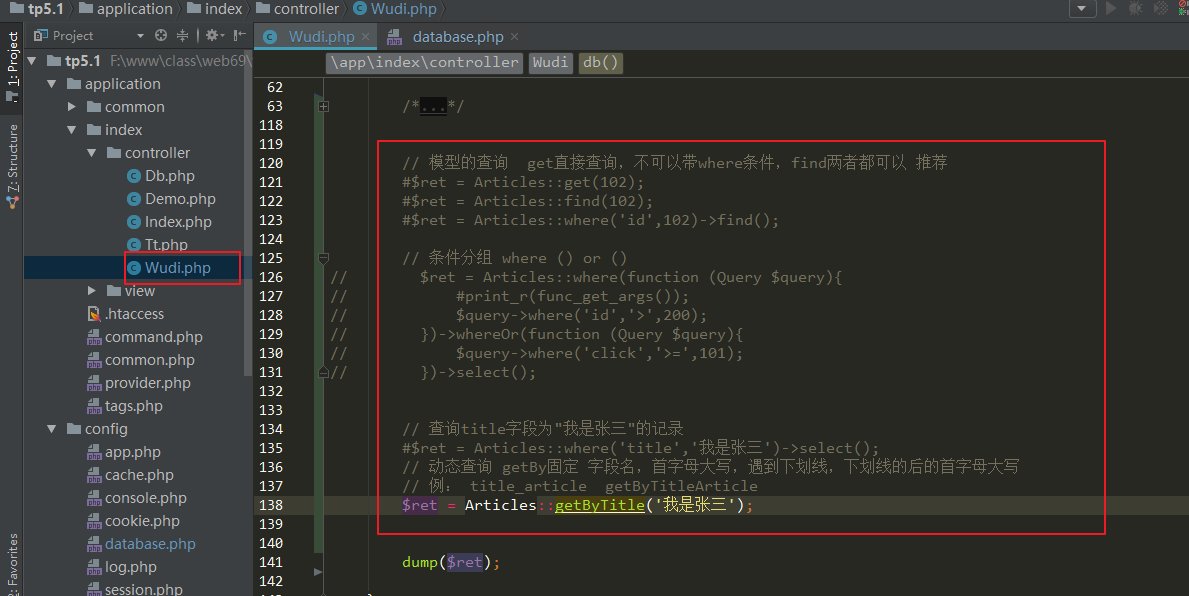
$query->where('click','>=',101);

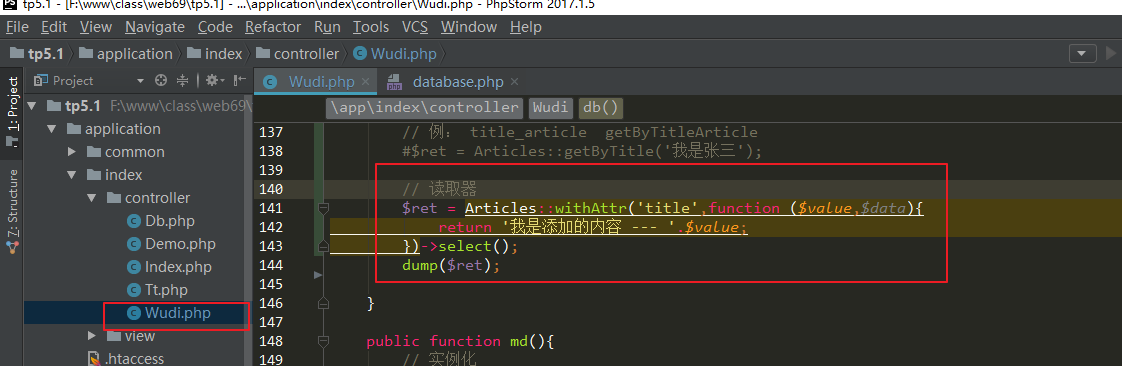
})->select();

# 动态查询

// 根据name字段查询用户

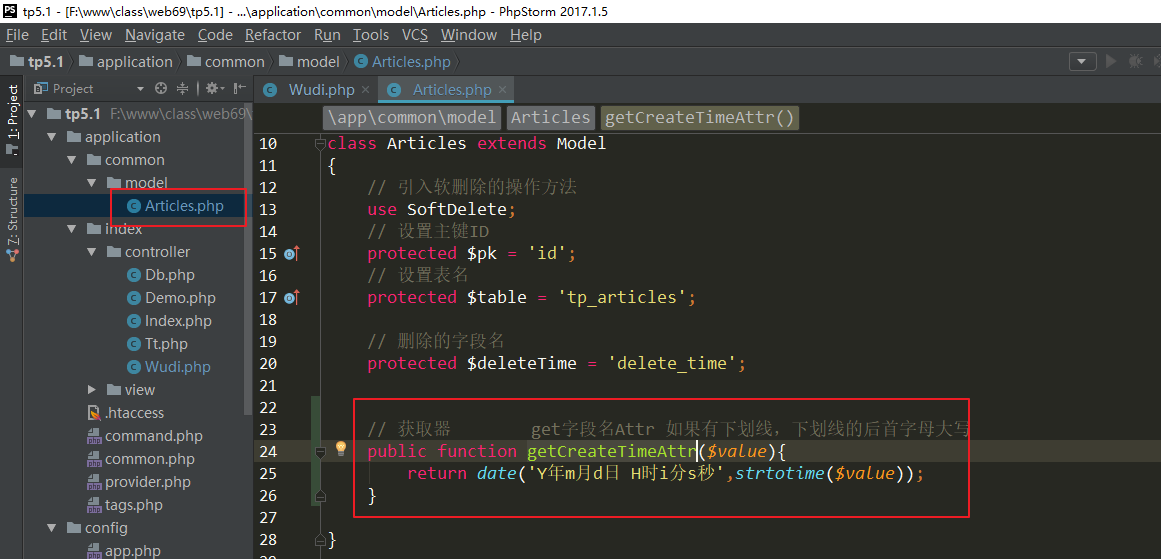
$user = User::getByName('thinkphp');

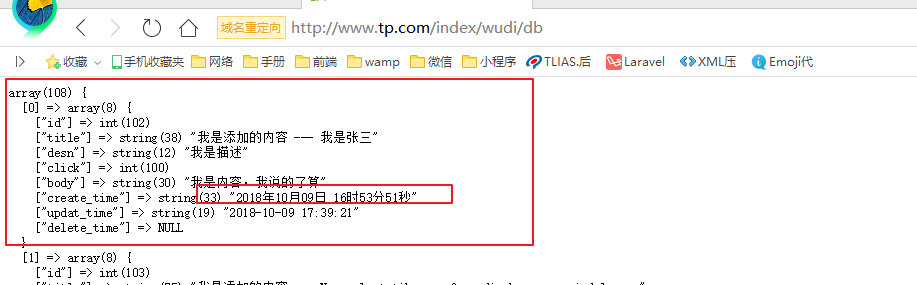




## 1.7、获取器

获取器的作用是对模型实例的（原始）数据做出自动处理。





# 二、数据分页

一般在列表页数据展示时，如果数据太多，需要进行分页展示。

ThinkPHP5.1内置了分页实现，要给数据添加分页输出功能变得非常简单，可以直接在Db类查询的时候调用paginate方法

$list = db('user')->paginate(10);

return view('index@index/user',compact('$list'));

$list = User::paginate(10)

return view('index@index/user',compact('$list'));

模板文件中分页输出代码如下

<div>

<ul>

{volist name='list' id='user'}

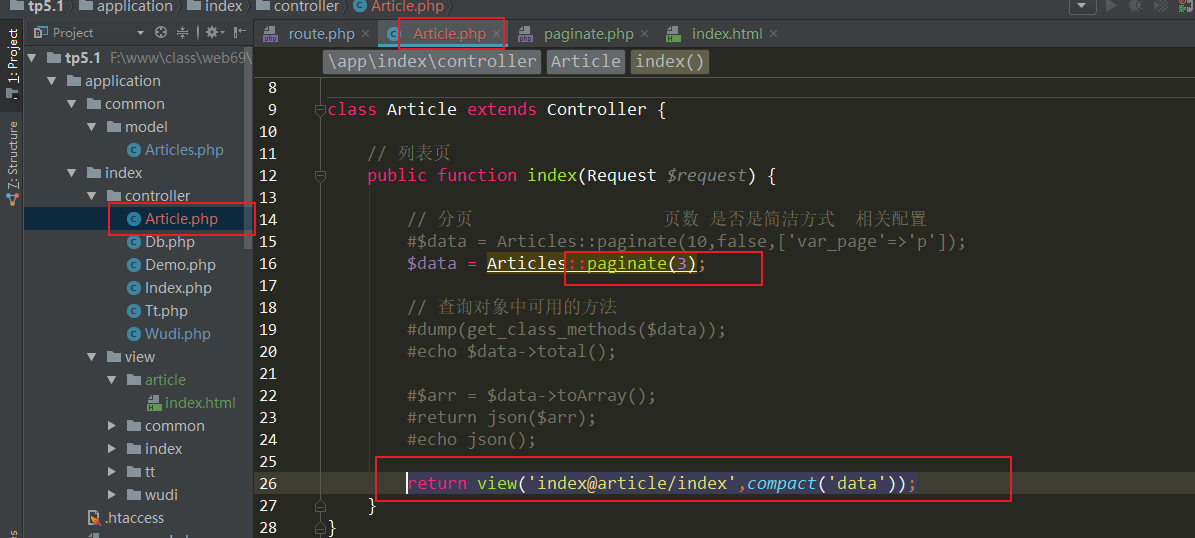
<li> {$user.nickname}</li>

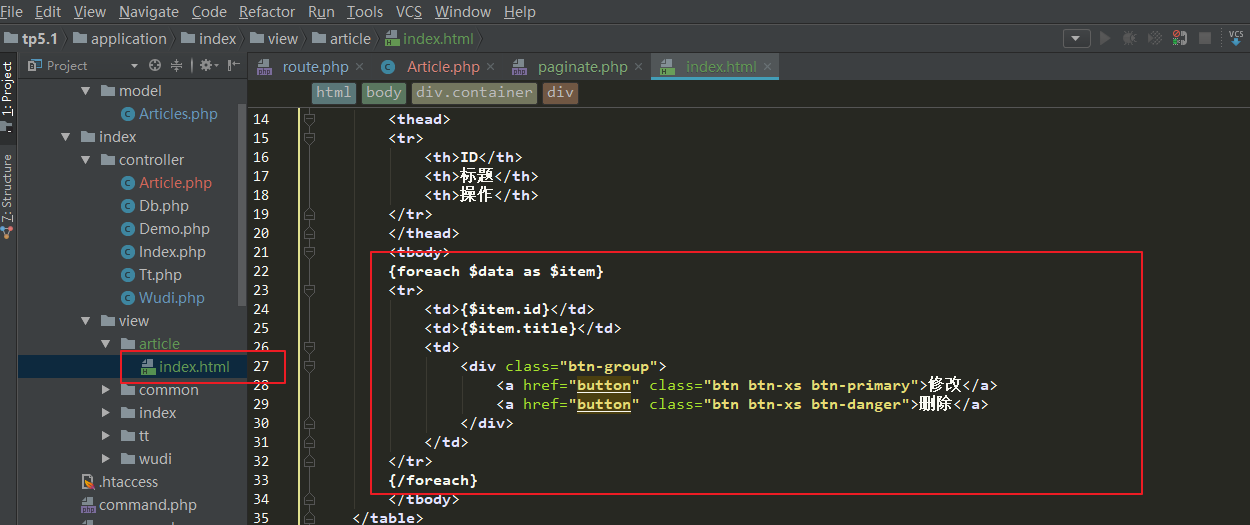
{/volist}

</ul>

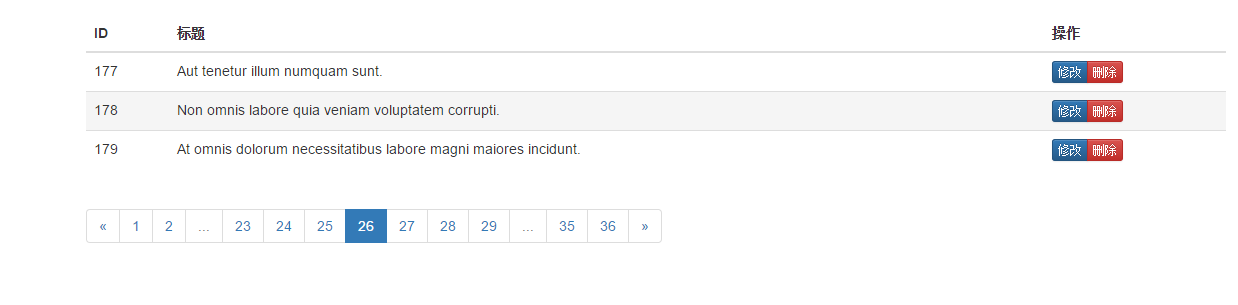
</div>

**{$list|raw}**





效果



# 三、表单验证

表单验证是为了防止访问者跳过客户端验证(js验证)而造成的系统安全问题，一但非法用户绕过客户端验证而服务器端没有加以验证，这样就是很不安全了，所以项目必须要进行服务器端表单验证。

ThinkPHP5.1推荐使用验证器进行数据验证（也支持使用\think\Validate类进行独立验证）

内置规则：

<https://www.kancloud.cn/manual/thinkphp5_1/354107>

## 3.1、独立验证

<https://www.kancloud.cn/manual/thinkphp5_1/354107>

$validate = Validate::make([

'name' => 'require|max:25',

'email' => 'email'

],[

'name.require' => '名称必须填写',

'name.max' => '名称最多只能是25字',

'email.email' => '邮箱不合法'

]);

$data = [

'name' => 'thinkphp',

'email' => 'thinkphp@qq.com'

];

if (!$validate->check($data)) {

dump($validate->getError());

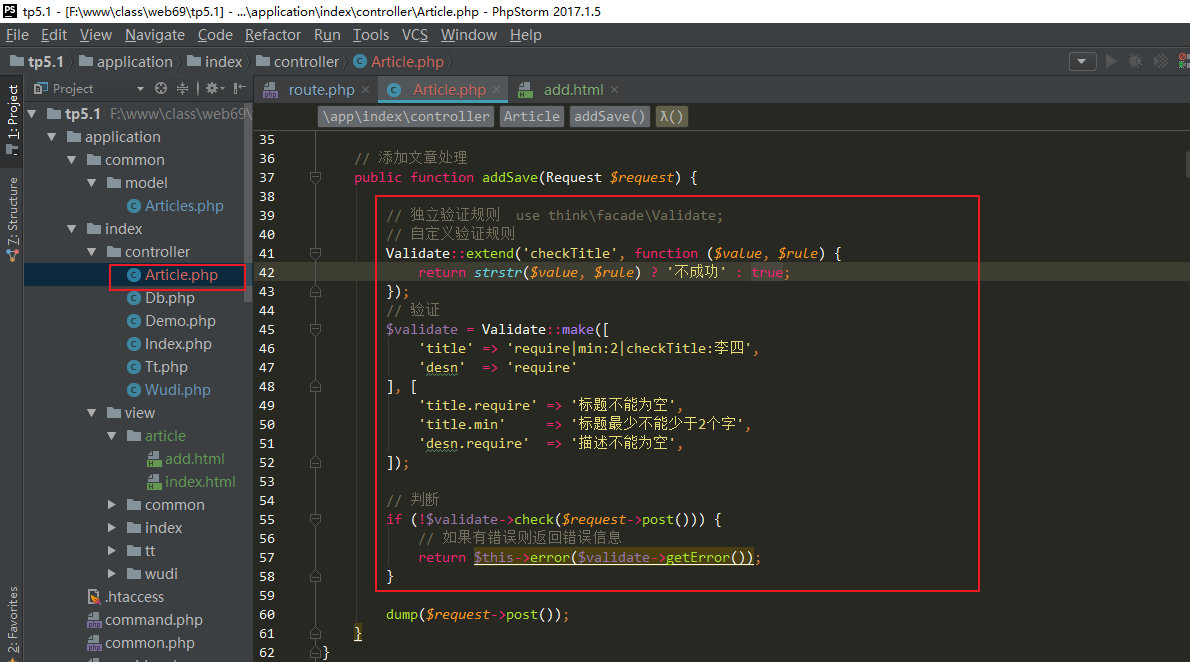
}

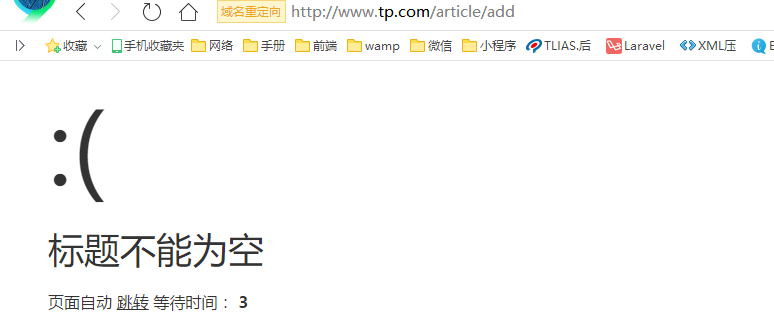
# 自定义验证规则

Validate::extend('checkName', function ($value, $rule) {

return $rule == $value ? true : '名称错误';

});





## 3.2、验证器

thinkphp5.1之后推荐使用的方式,验证器，就是一个独立的文件，此文件就干一件事件，验证。

# 创建验证器

php think make:validate 模块名/验证器名(首字母大写)

# 验证器

namespace app\index\validate;

use think\Validate;

class User extends Validate

{

protected $rule = [

'name' => 'require|max:25',

'age' => 'number|between:1,120',

'email' => 'email',

];

protected $message = [

'name.require' => '名称必须',

'name.max' => '名称最多不能超过25个字符',

'age.number' => '年龄必须是数字',

'age.between' => '年龄只能在1-120之间',

'email' => '邮箱格式错误',

];

}

# 控制器

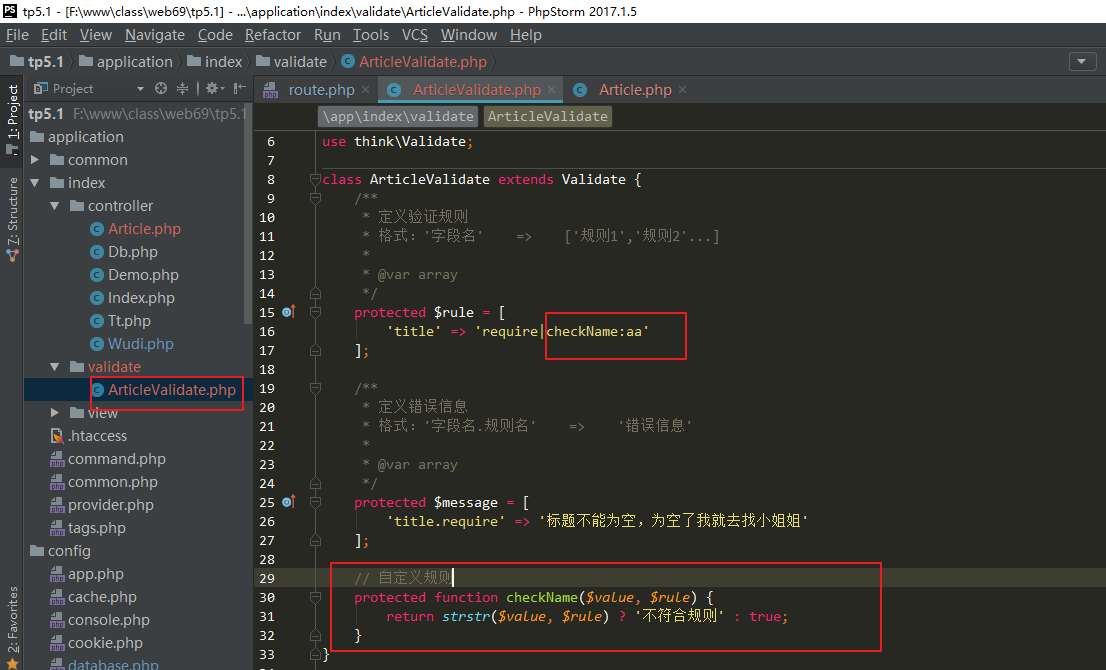
$ret = $this->validate($request->post(),User::class);

if (true !== $ret){

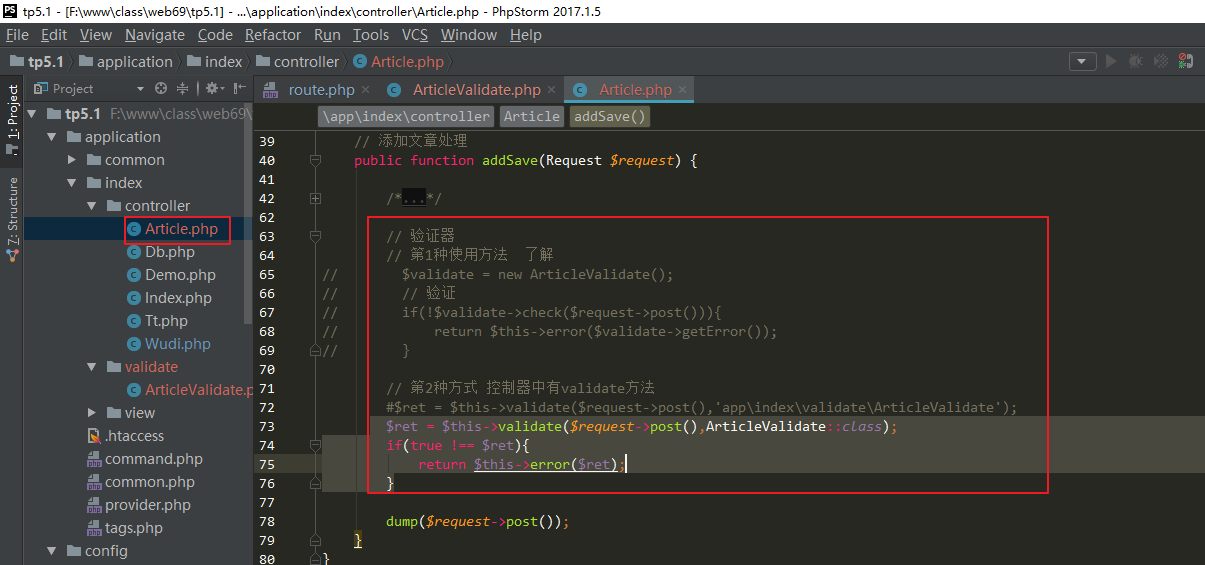
dump($ret);

}

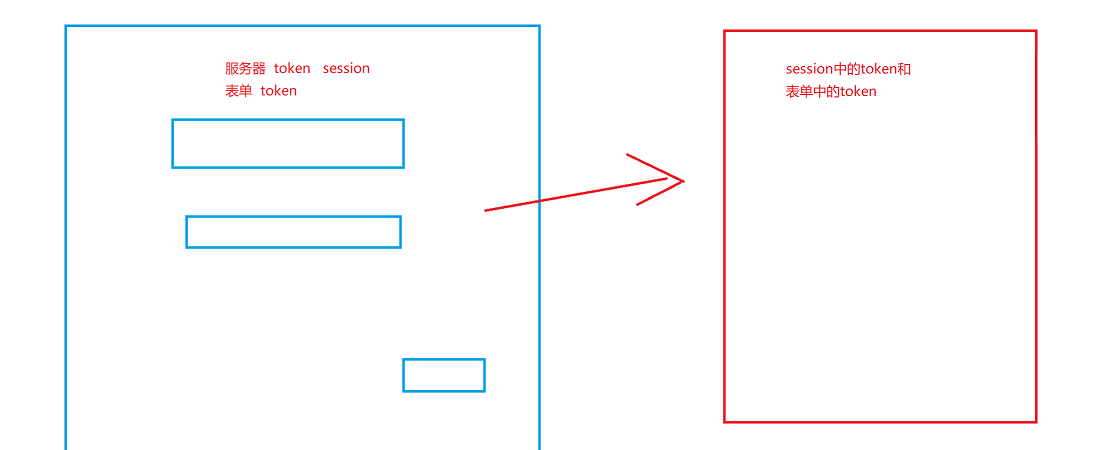
验证器

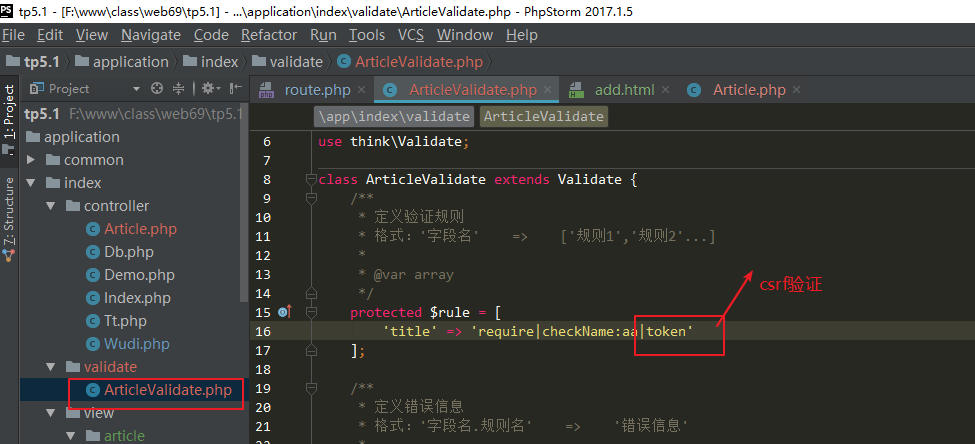


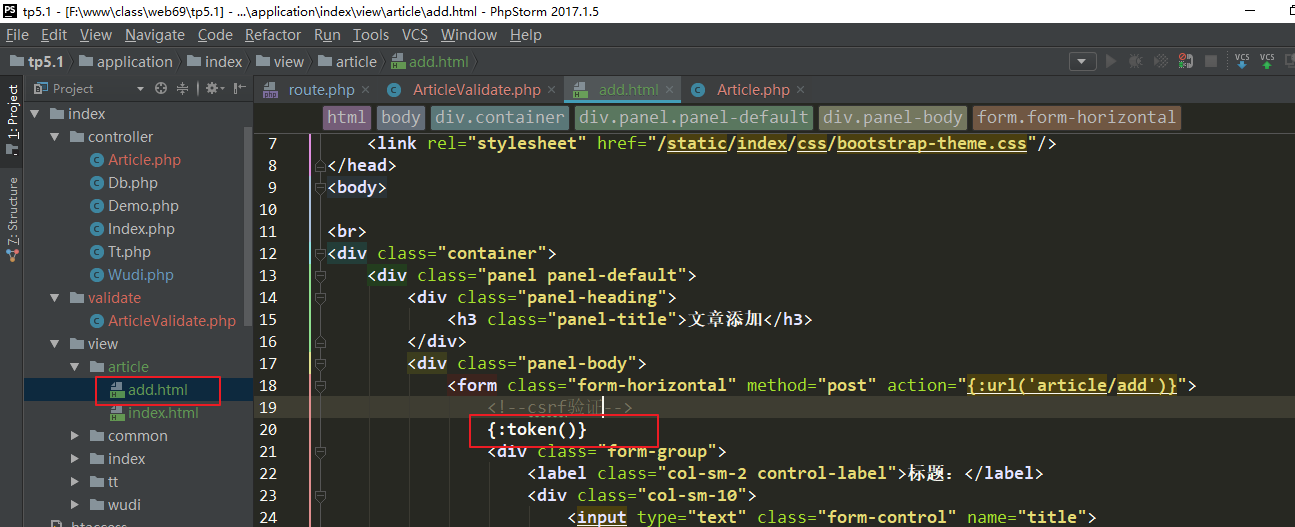
控制器

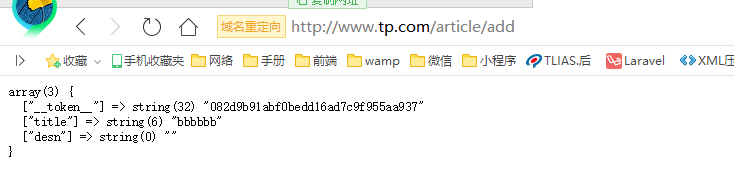


## 3.3、csrf验证









# 四、验证码

<https://www.kancloud.cn/manual/thinkphp5_1/354122>

tp5框架中默认没有自带验证码功能，需要我们自己去引入安装

# 官方提供的tp官方验证码类

composer require topthink/think-captcha

# 模板中引用

<div>{:captcha\_img()}</div>

或者

<div><img src="{:captcha\_src()}" alt="captcha" /></div>

# 验证

$this->validate($data,[

'code'=>'require|captcha'

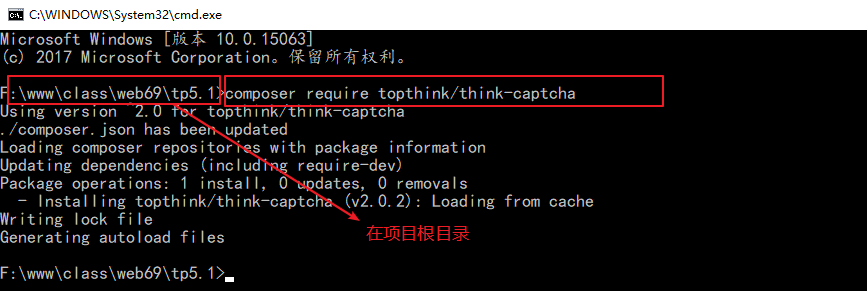
]);

或者

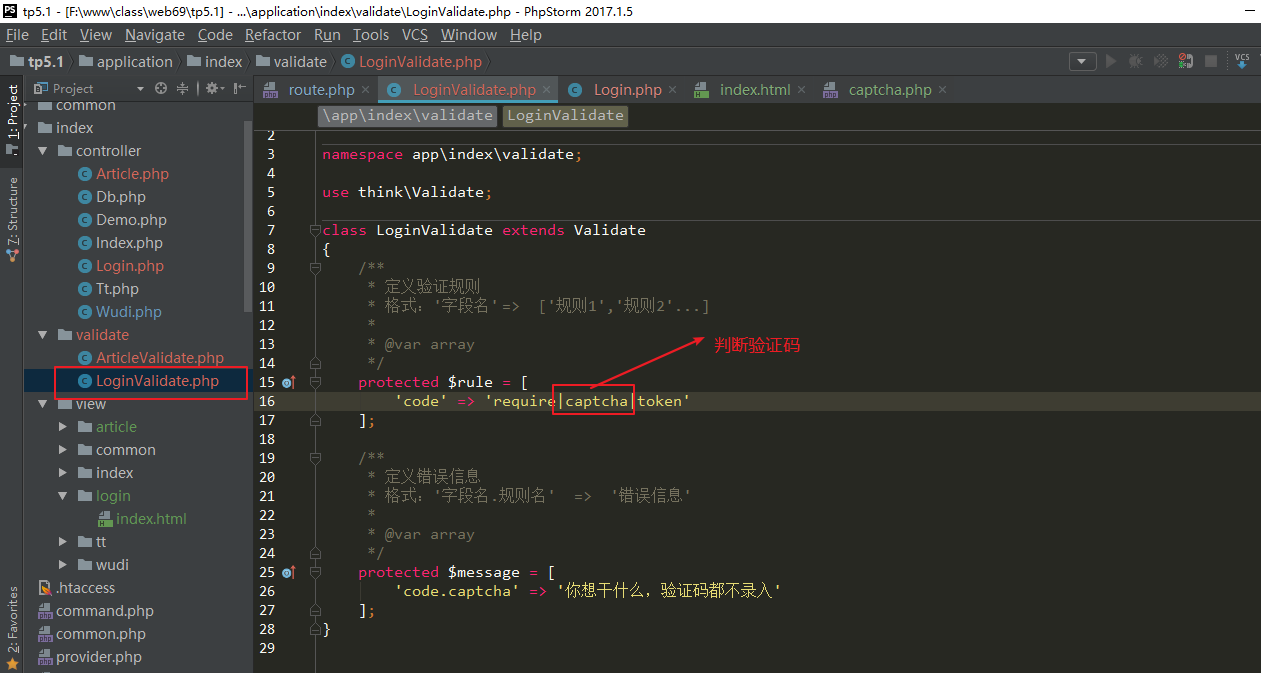
if(!captcha\_check($captcha)){

// 验证失败

};

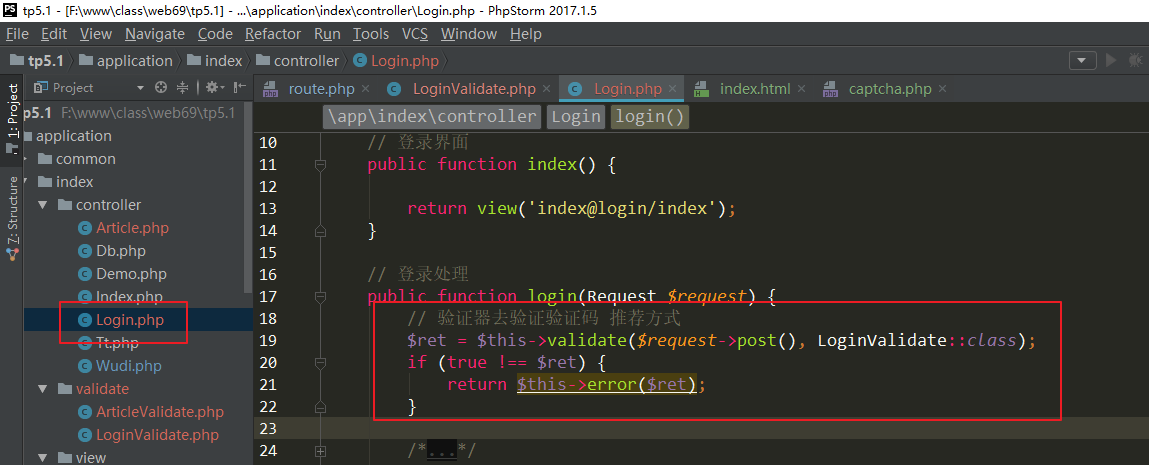


验证器

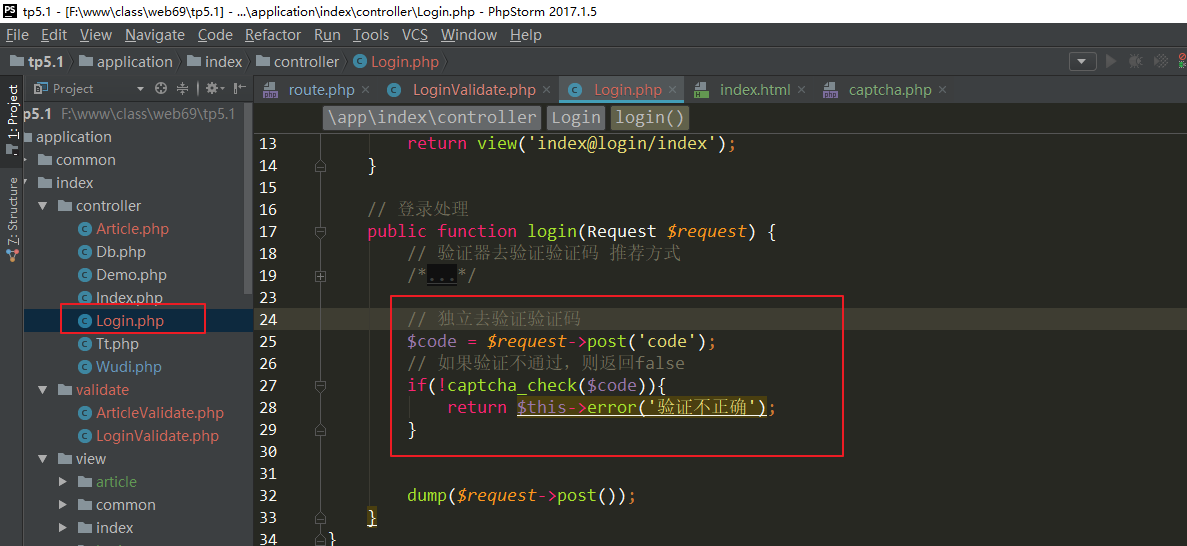


控制器

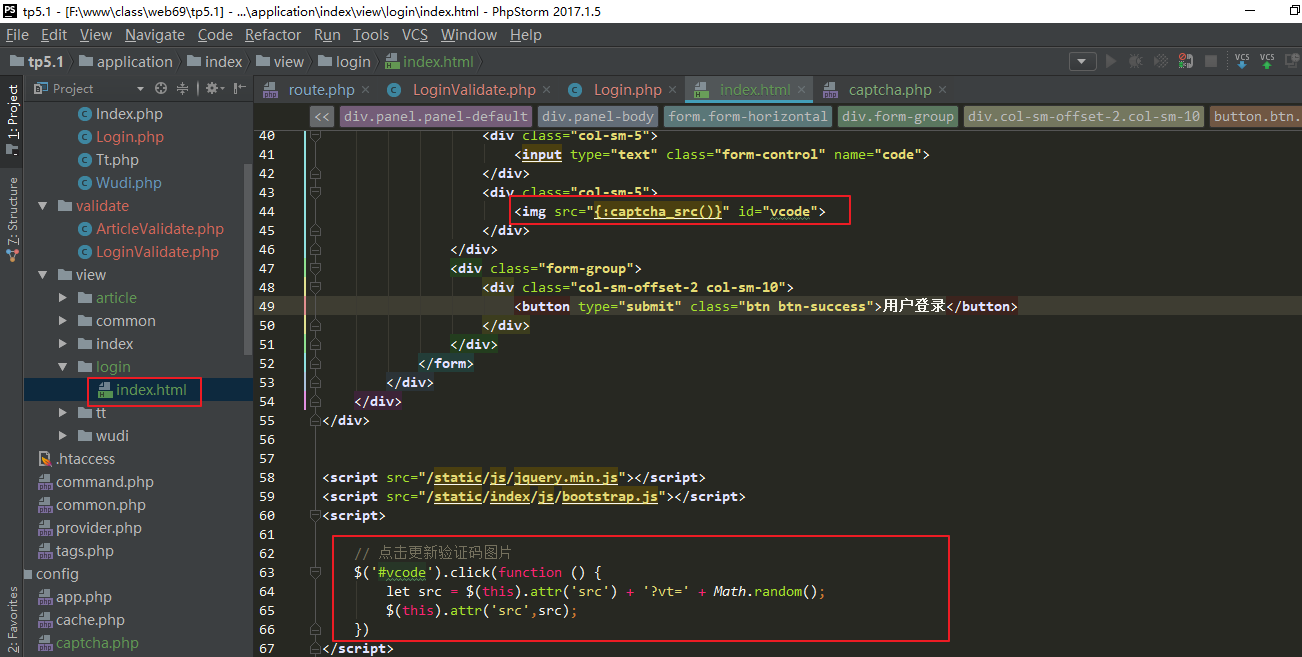
验证器去验证



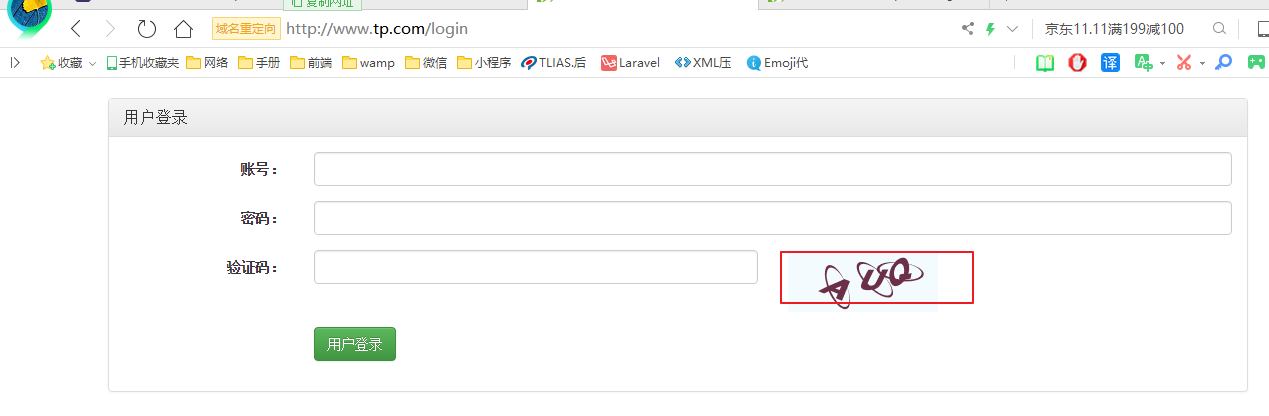
独立去验证验证码



html模板



效果



# 五、cookie和session处理

## 5.1、cookie操作

ThinkPHP采用think\facade\Cookie类提供Cookie支持。think\Cookie

配置文件位于 config/cookie.php中

# 设置

// 设置Cookie 有效期为 3600秒

Cookie::set('name','value',3600);

cookie('name', 'value', 3600);

# 判断是否存在

Cookie::has('name');

cookie('?name')

# 获取

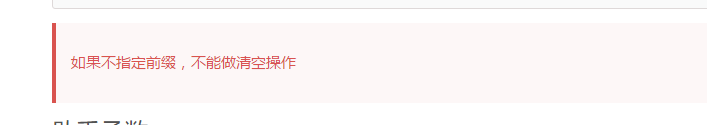
Cookie::get('name');

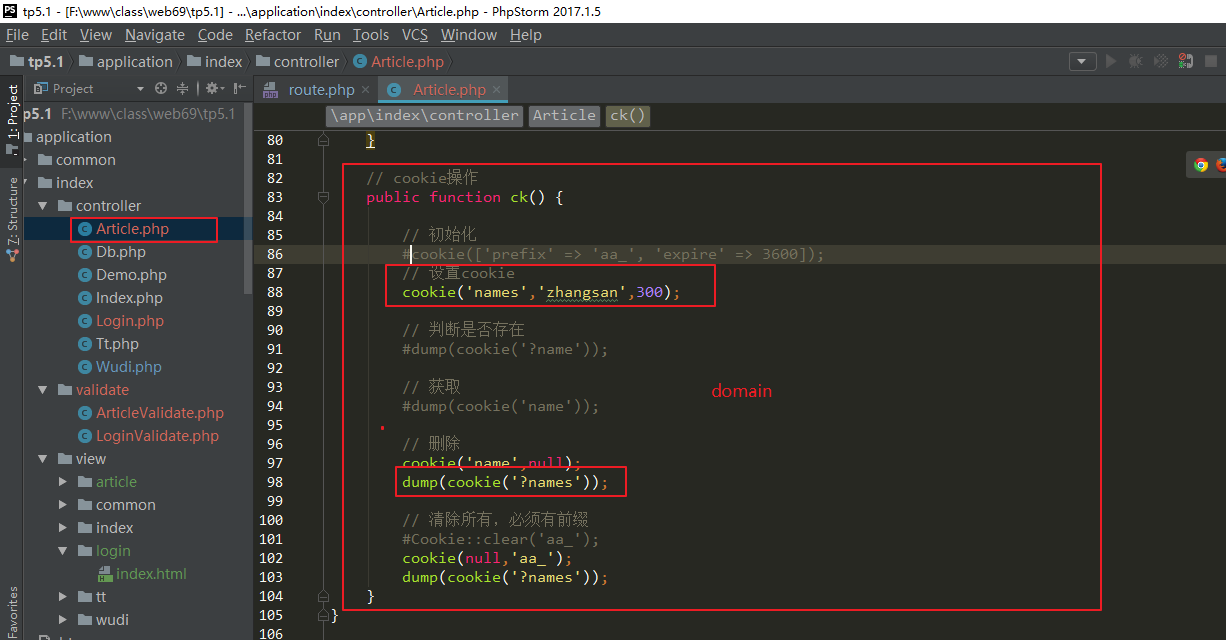
cookie('name');

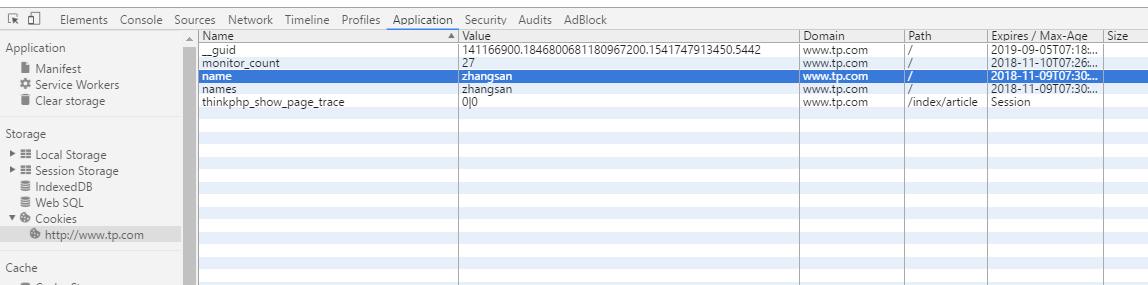
# 删除

Cookie::delete('name');

cookie('name', null);







## 5.2、session操作

可以直接使用think\facade\Session类操作Session

配置文件位于config/session.php文件中

# 设置

Session::set('name','thinkphp');

session('name', 'thinkphp');

# 闪存

Session::flash('name','value');

# 判断是否存在

Session::has('name');

session('?name');

# 取值

Session::get('name');

session('name');

# 删除

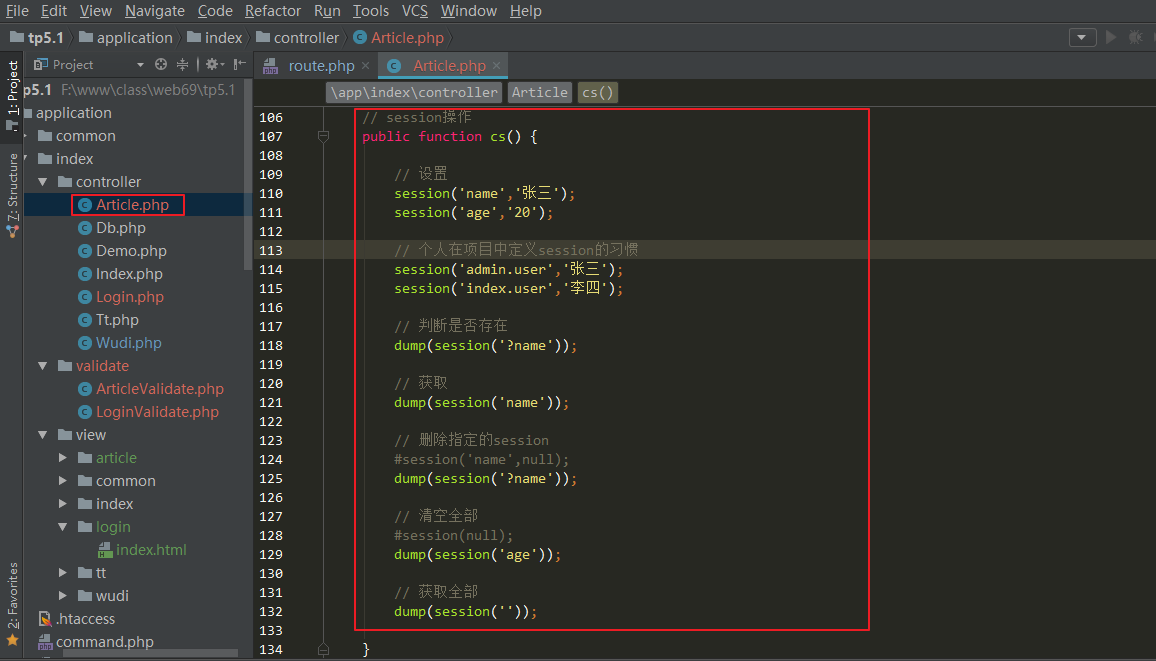
Session::delete('name');

session('name', null);

# 清空

Session::clear();

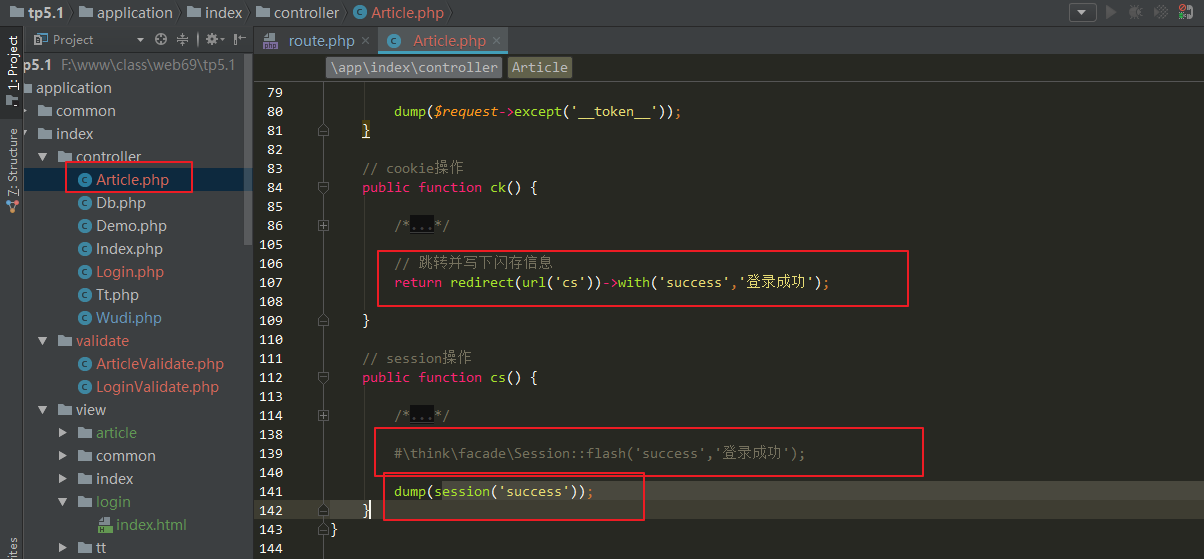
session(null);



## 5.3、闪存

# 定义好，在下一次HTTP请求中获取到，第2没有了

Session::flash('name','value');

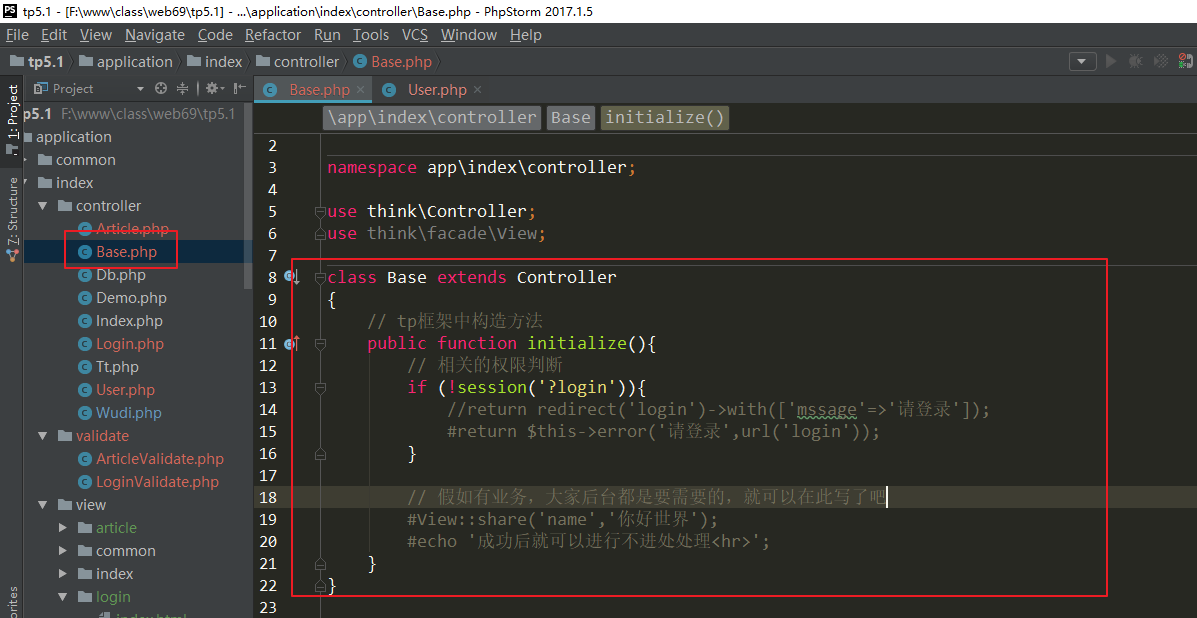


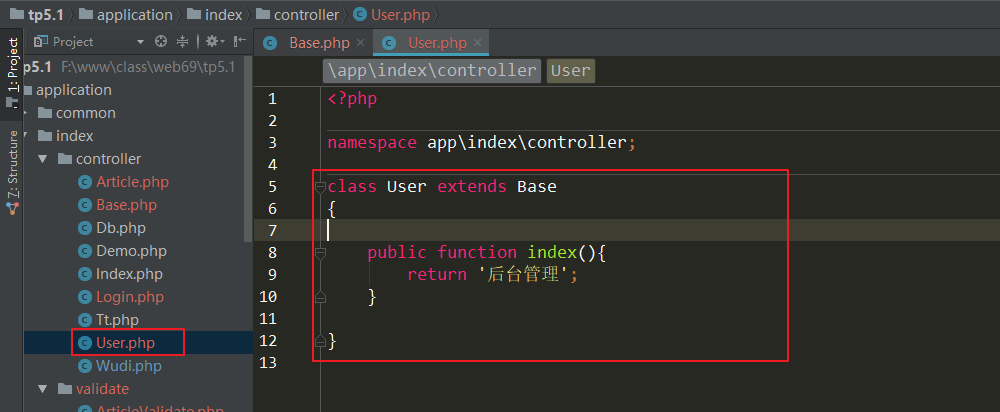
# 六、中间件

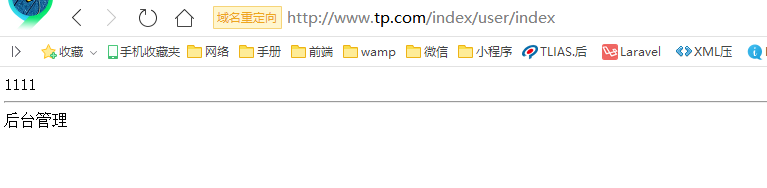
中间件主要用于拦截或过滤应用的HTTP请求，并进行必要的业务处理。

## 6.0、传统式用户过滤

通过继承基类来达到权限的判断。





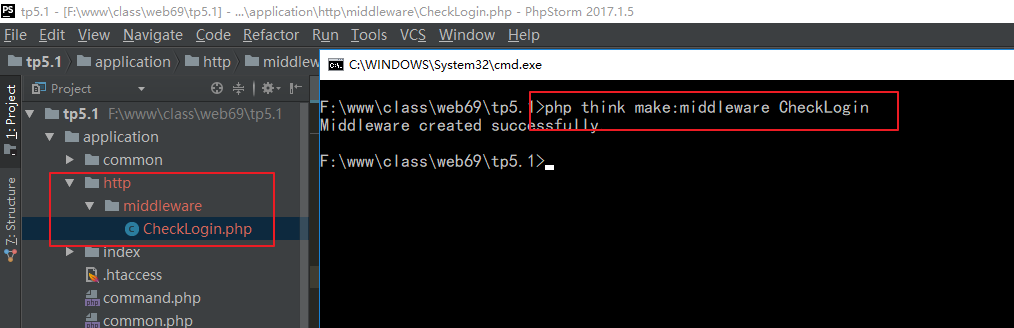


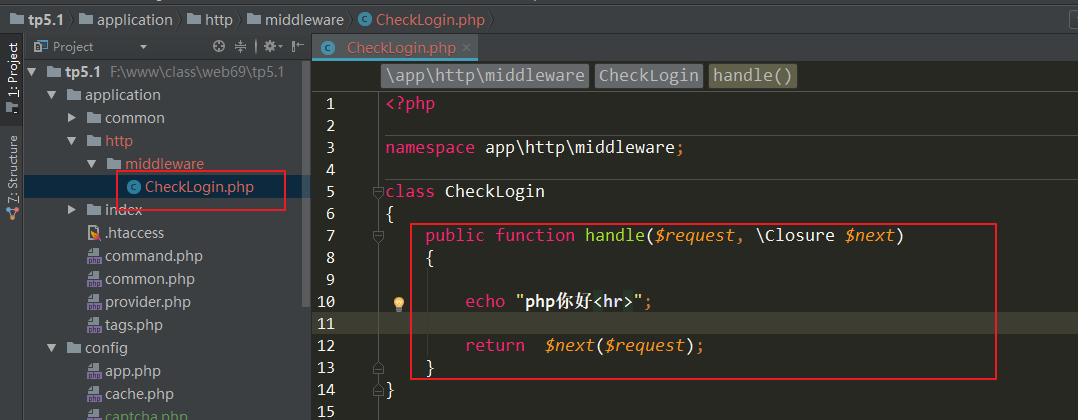
## 6.1、定义中间件

<https://www.kancloud.cn/manual/thinkphp5_1/564279>

php think make:middleware 中间件的名称

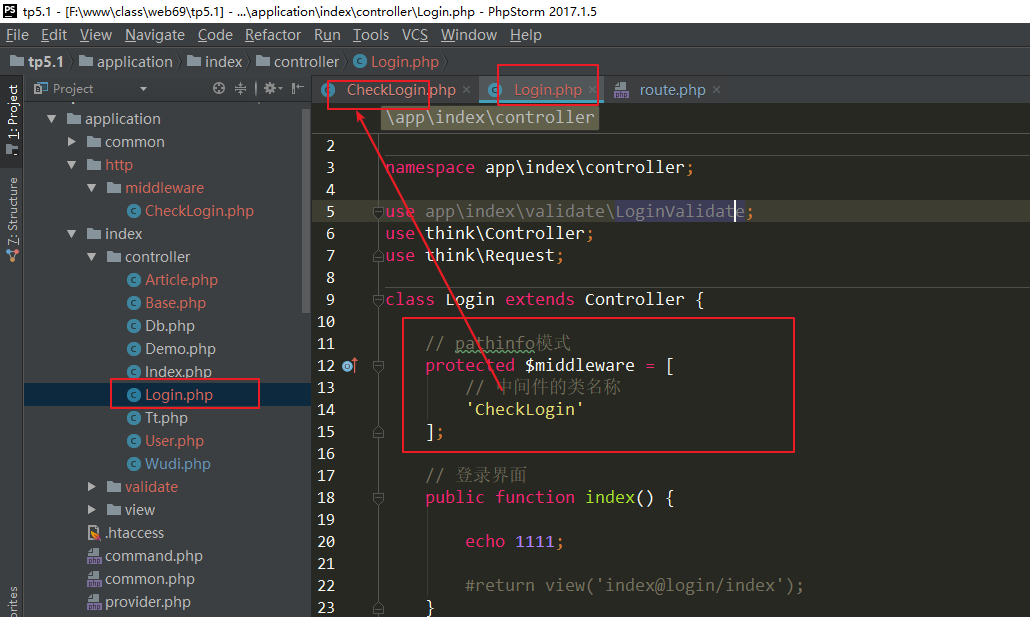
这个指令会 application/http/middleware目录下面生成一个中间件文件。



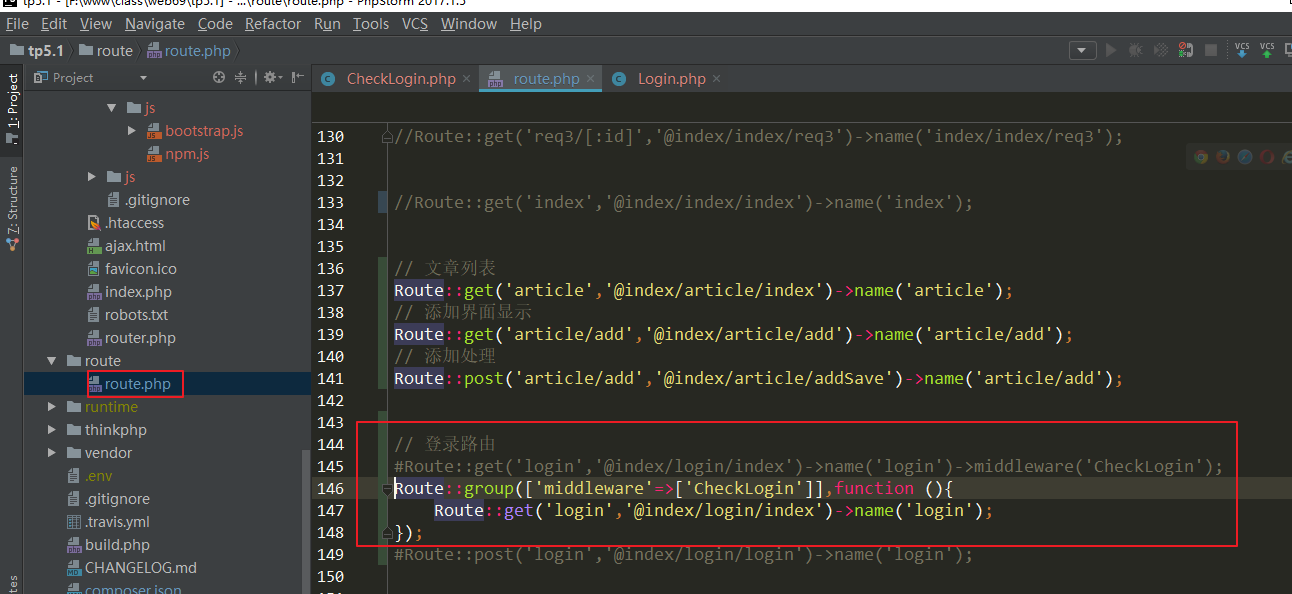


## 6.2、注册中间件

### 6.2.1、控制器注册 pathinfo

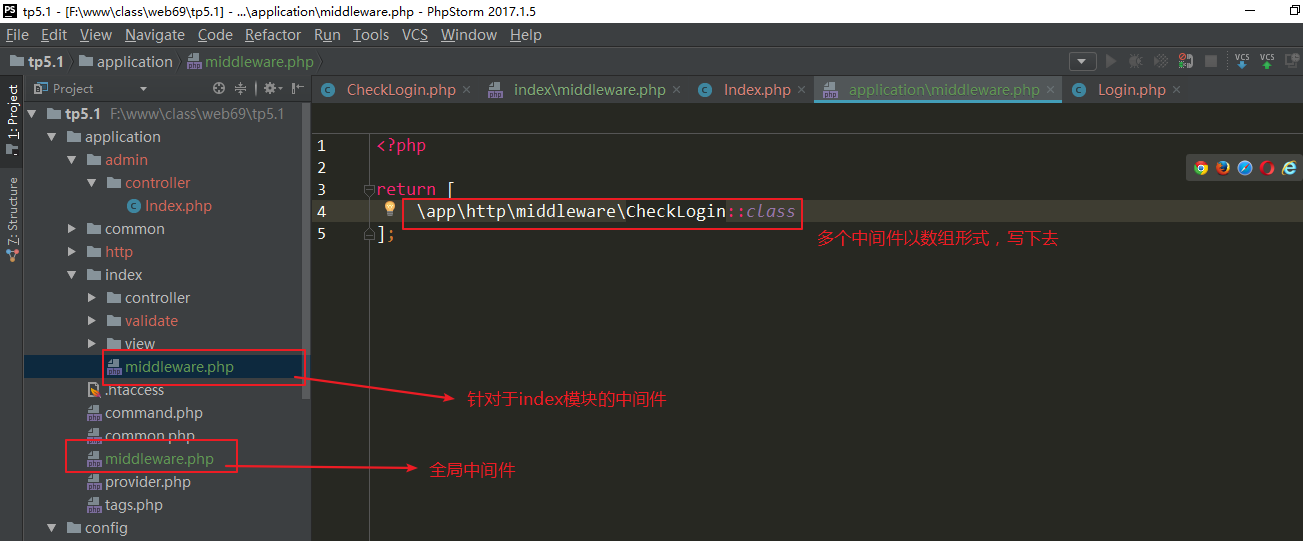


### 6.2.2、自定义路由 middleware

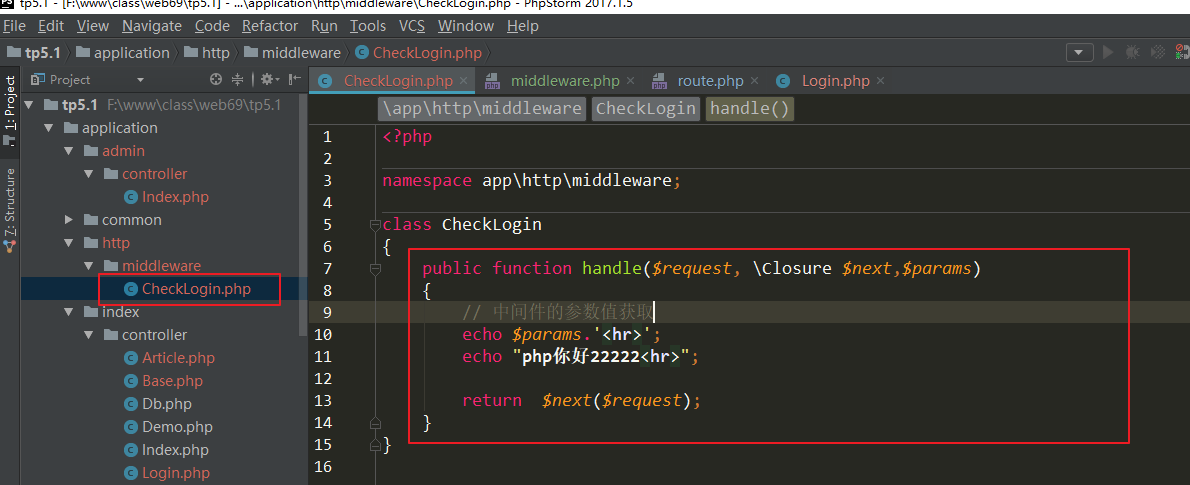


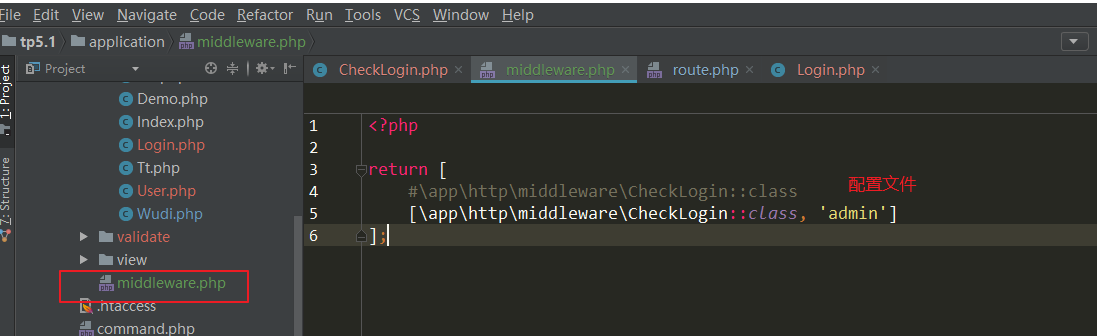
### 6.2.3、配置文件 middleware.php

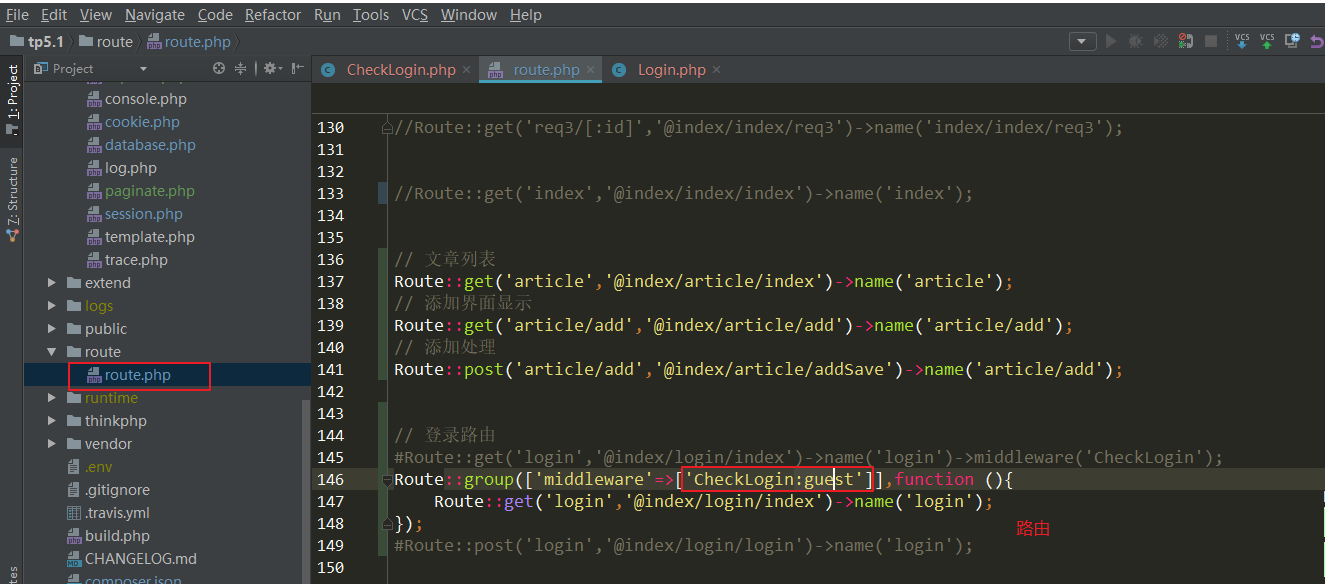
此文件可以放在模块下，就是对此模块注册此中间件，在application下面，就是全局中间件,说白了，就是所有的模块讲都有此中间件。



### 6.2.4、中间件参值









# 七、后台登录

见代码