# THINKPHP5.1

## 一、ThinkPHP5.1简介

一、需要掌握的知识

Php5，php7更好

前端知识：html，div，css，html5，css3，

Javascrpt，jquery，ajax

Linux，lnmp一件安装包

Windows：phpstudy、wampserver3

工具：sublime，atom

二、安装：不再支持下载安装，只能使用composer或git安装。

## 二、MVC相关

自动生成模块：





URL解读：



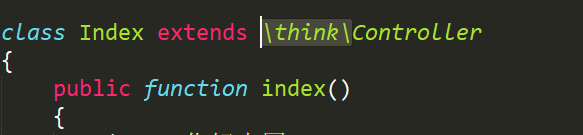
Index.php通过URL重写被隐藏了



命名空间是一个虚拟的空间，目的是为了解决类方法常量等重名的问题。

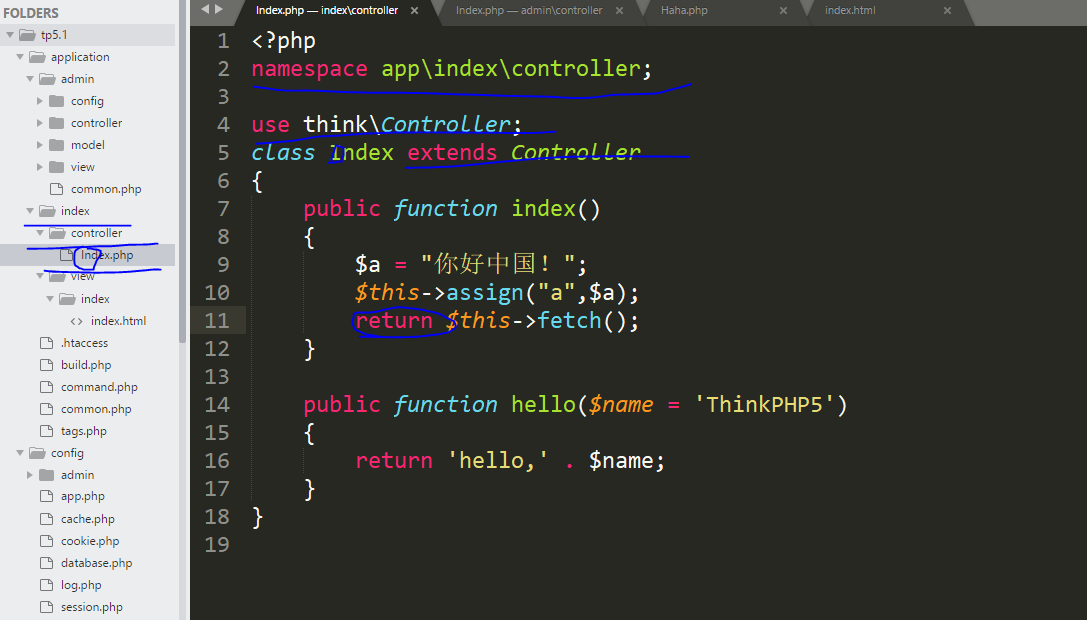
使用namespace进行定义。

使用时1：直接把命名空间加上。

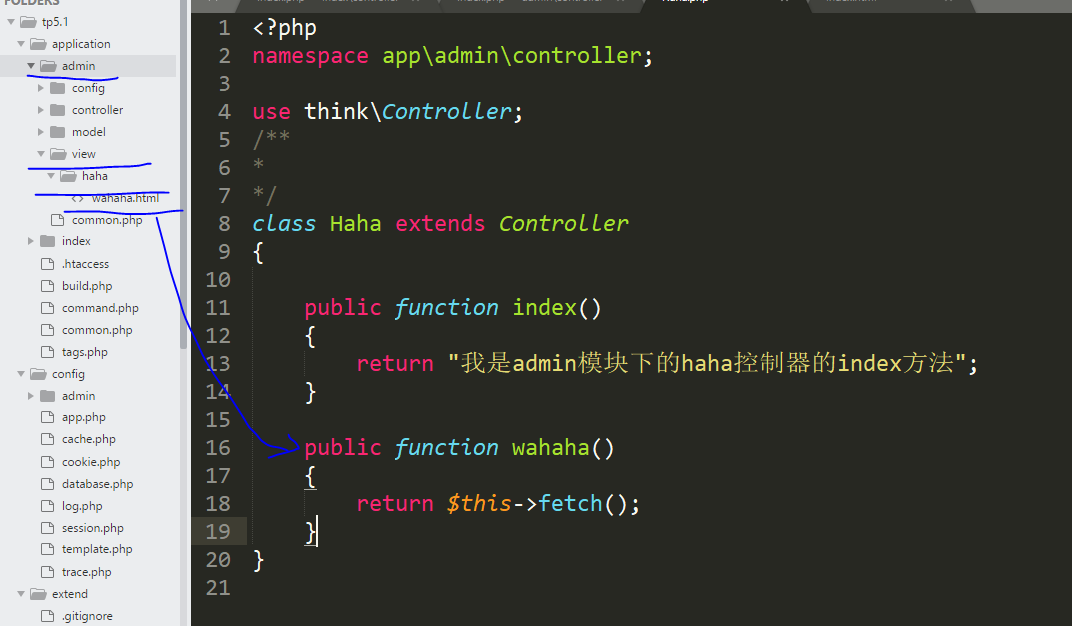


2：使用use关键字进行引入。

控制器的建立：



默认视图存放的位置：



视图赋值：3种方式。

在模板中使用url助手函数生成地址。

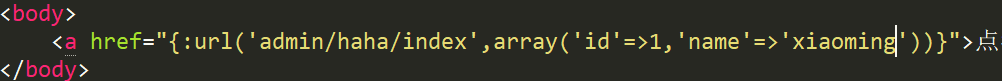
{:url()}参数为空，表示使用当前模块、当前控制器、当前方法

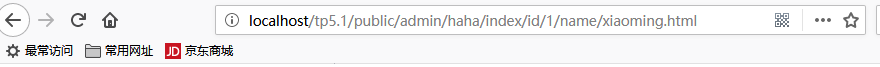
{:url('hello')}表示使用当前模块、当前控制器、hello方法

{:url('dx/dd')}表示使用当前模块、dx控制器、dd方法

{:url('admin/haha/index')}表示使用admin模块、haha控制器、index方法

使用数组增加参数和值





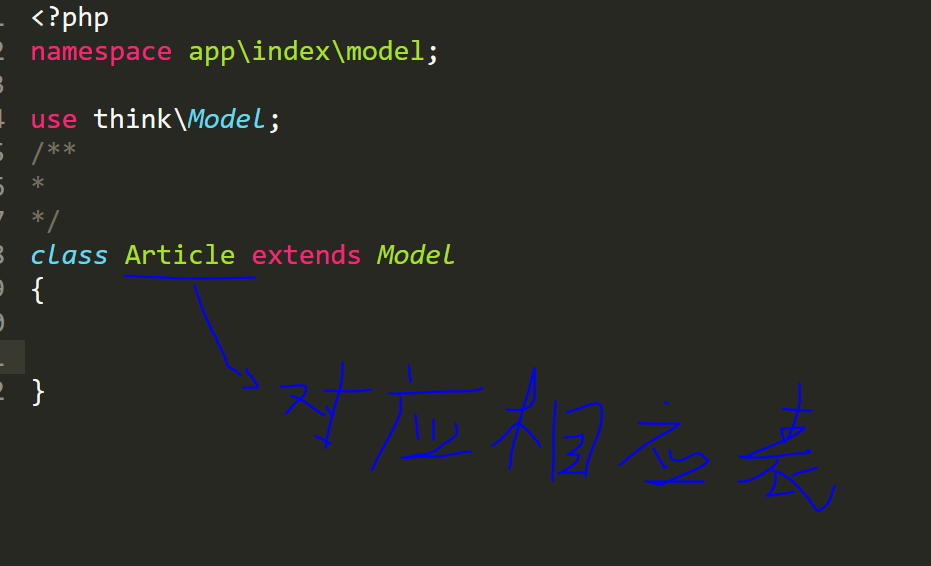
## 四、数据库

数据库配置

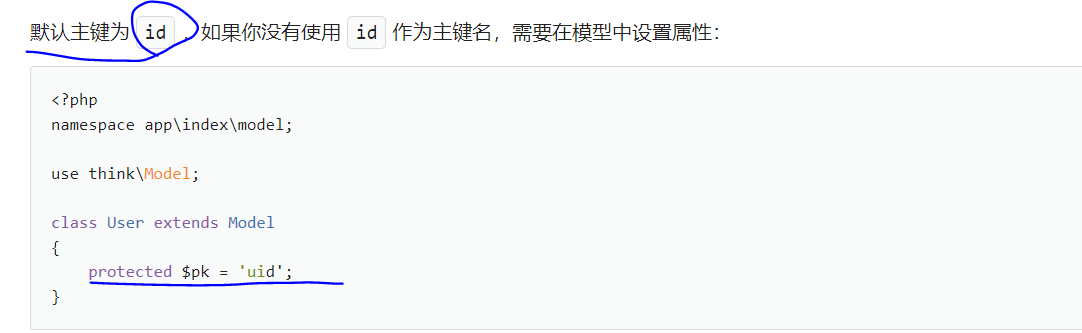
增删改查

## 模型层

模型层的定义：



主键的设置



数据添加：







批量添加数据：



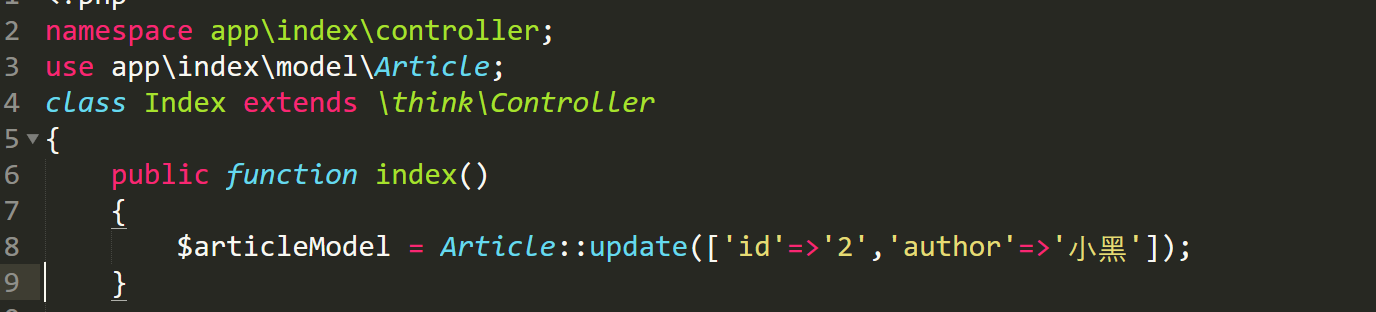
saveAll方法既可以增加也可以更新，



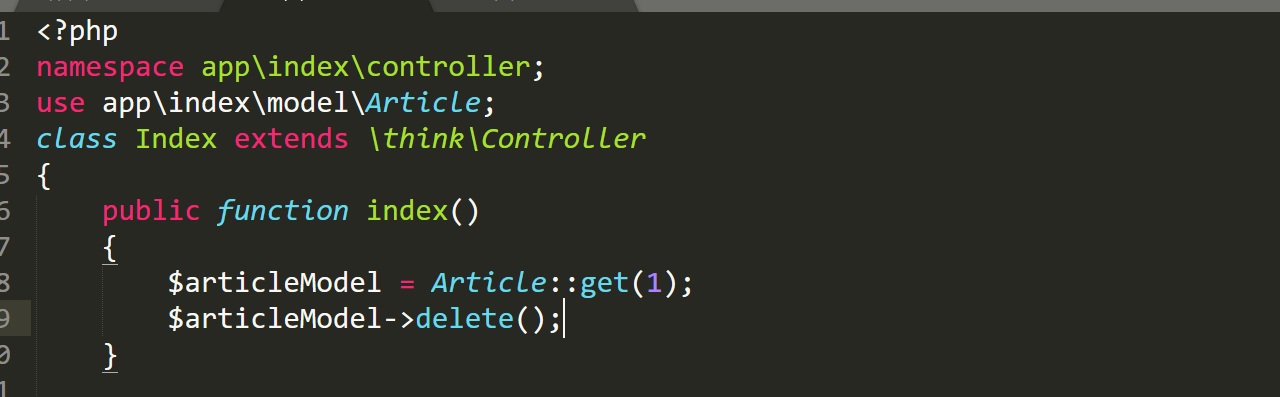
使用create静态方法：

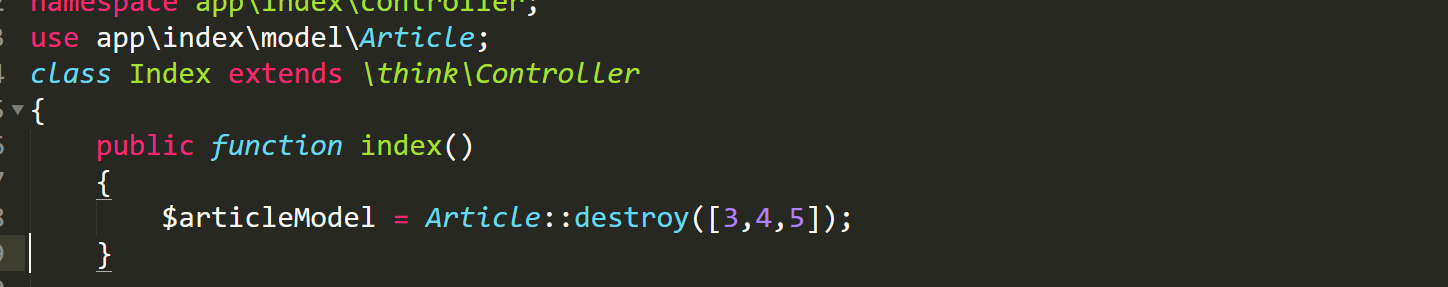
模型层的更新：





数据的删除：







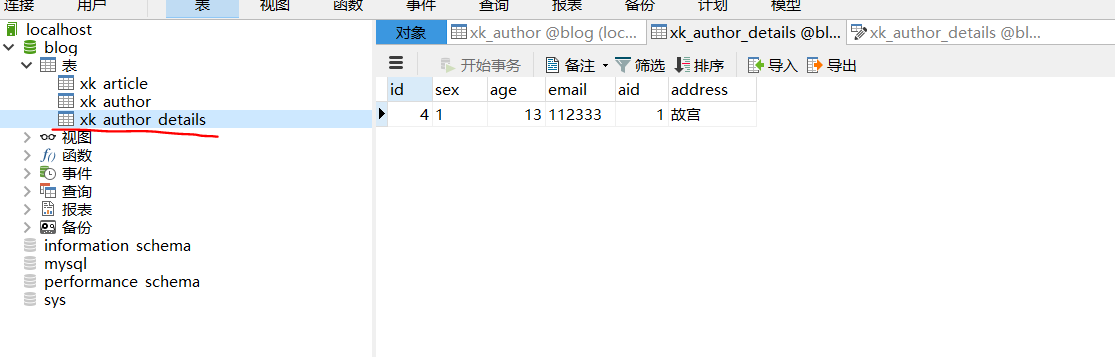


关联模型：

一对一关联：

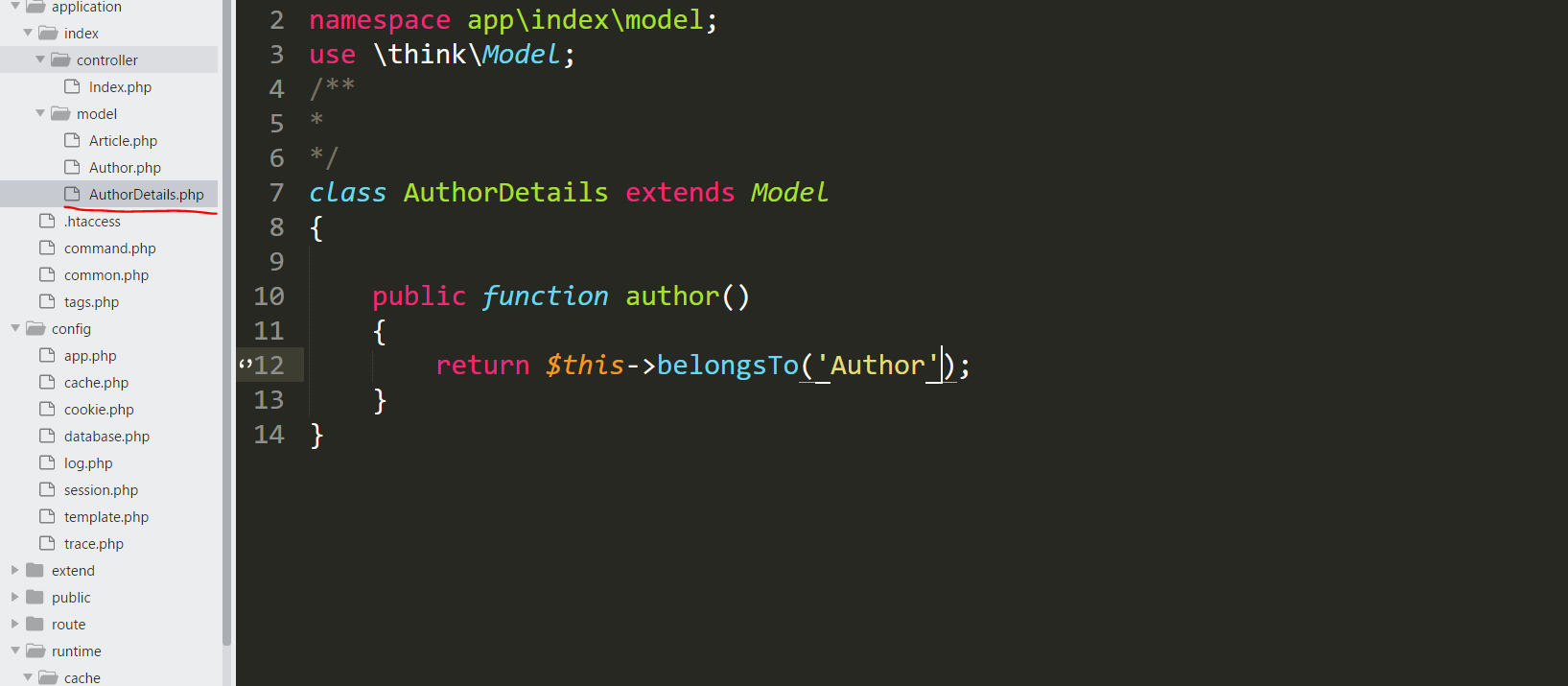
例如：一个作者只有一个资料表





建立两个模型：





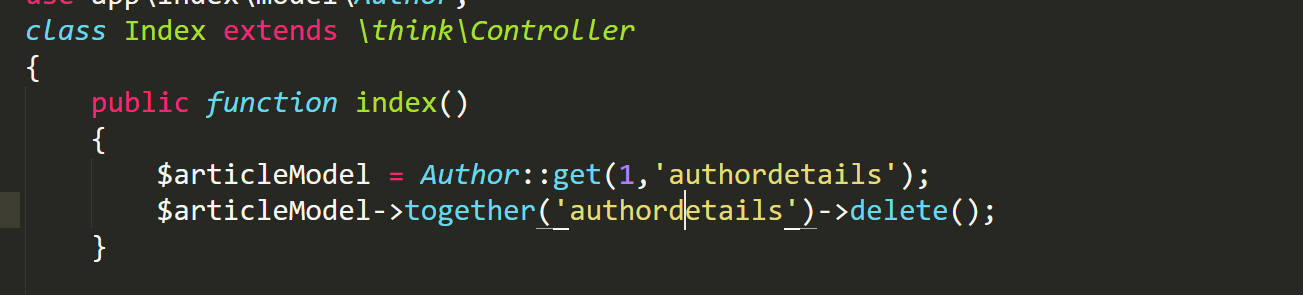
增加数据：



数据更新：

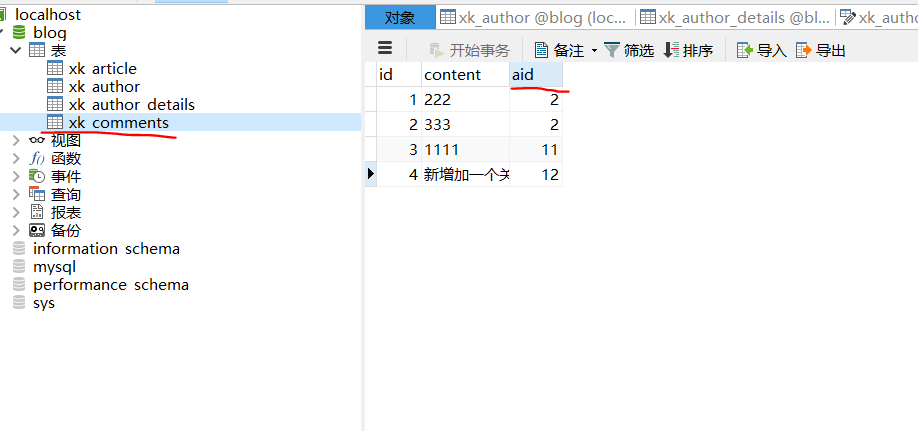


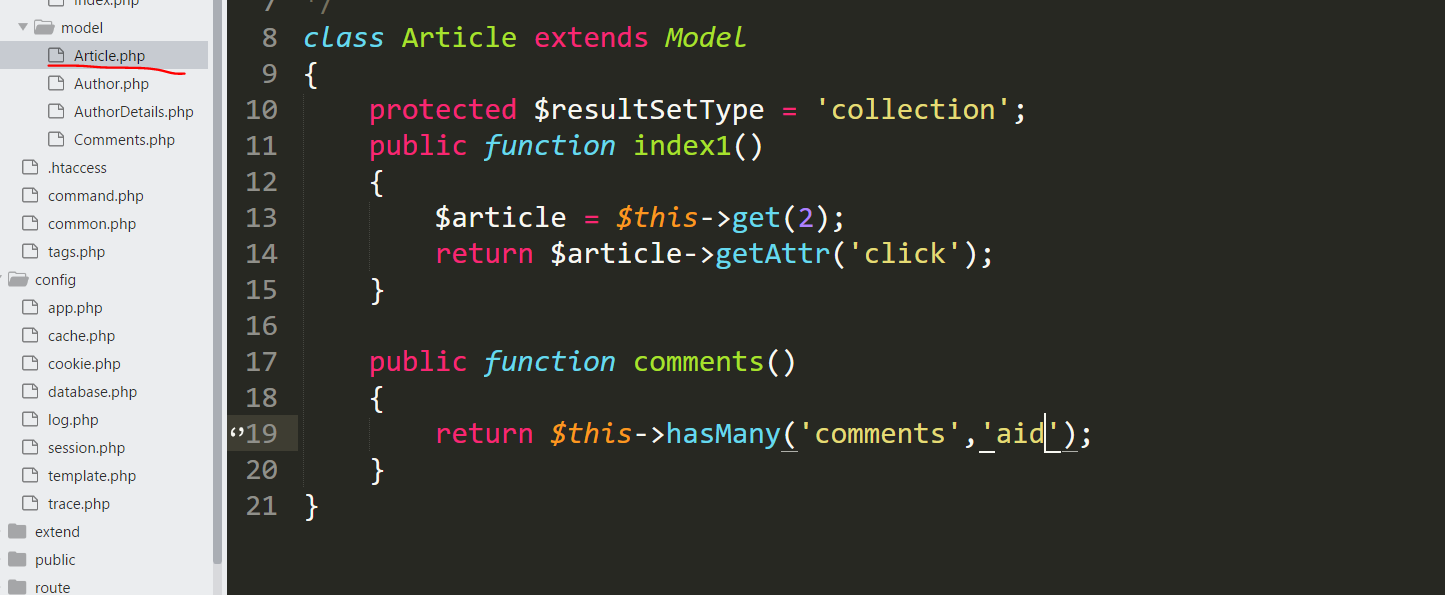
数据删除：



一对多关联模型：

例如一片文章有多个评论，一个评论只属于一篇文章。







关联查询：



关联新增：



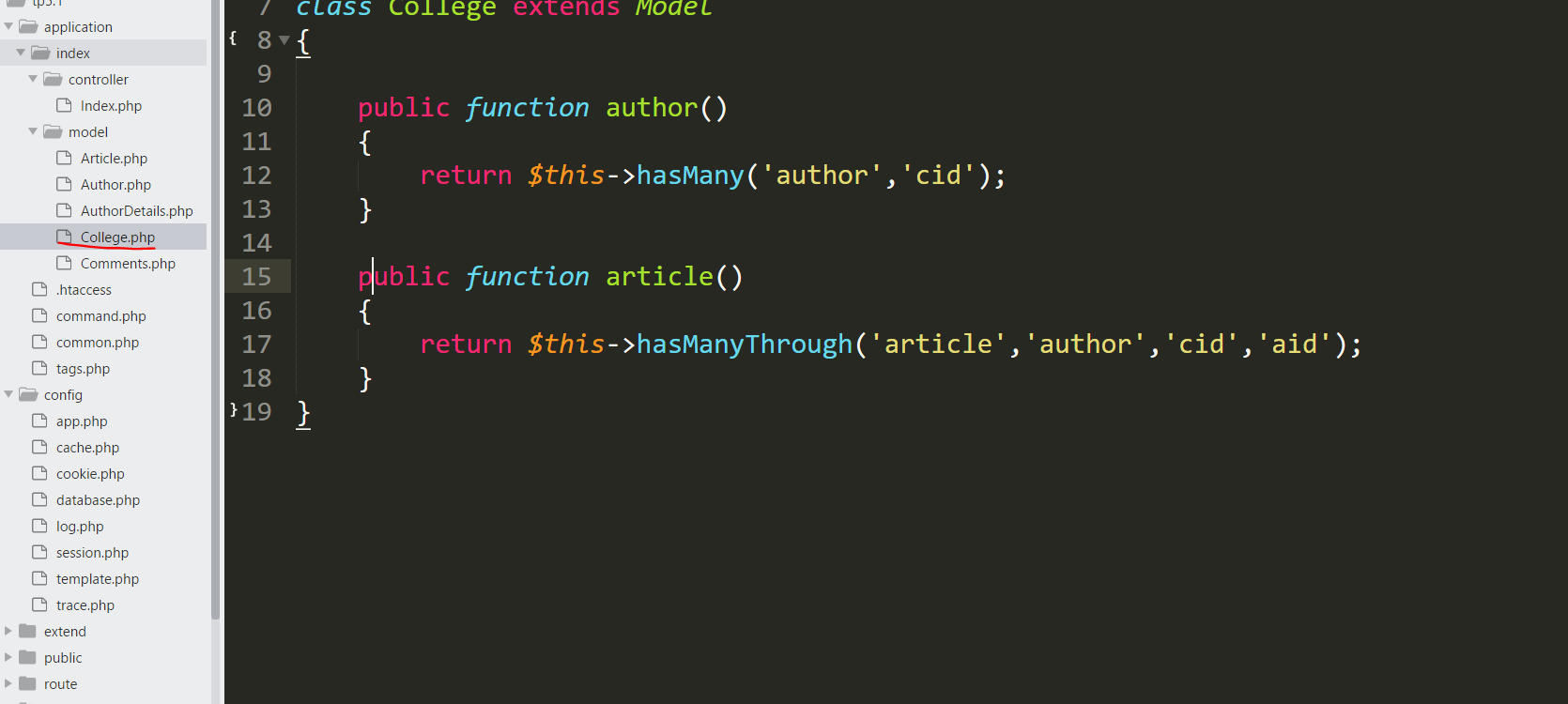


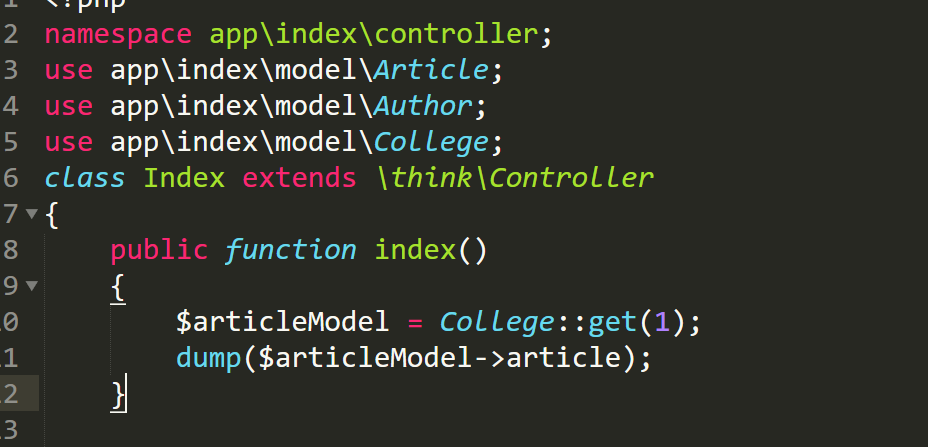
关联删除



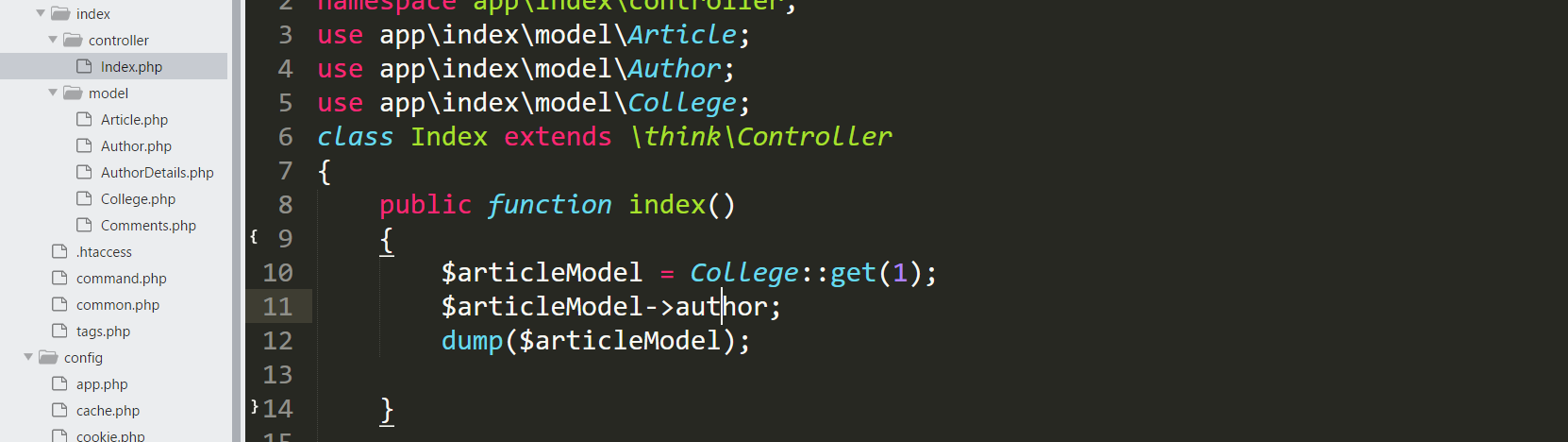
远程一对多：

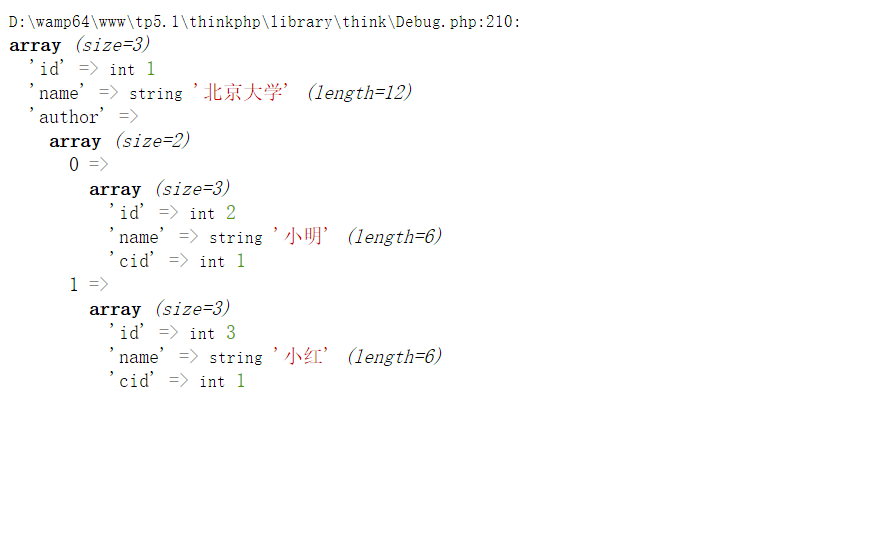
例如一个学校有多个老师，一个老师有多篇文章。



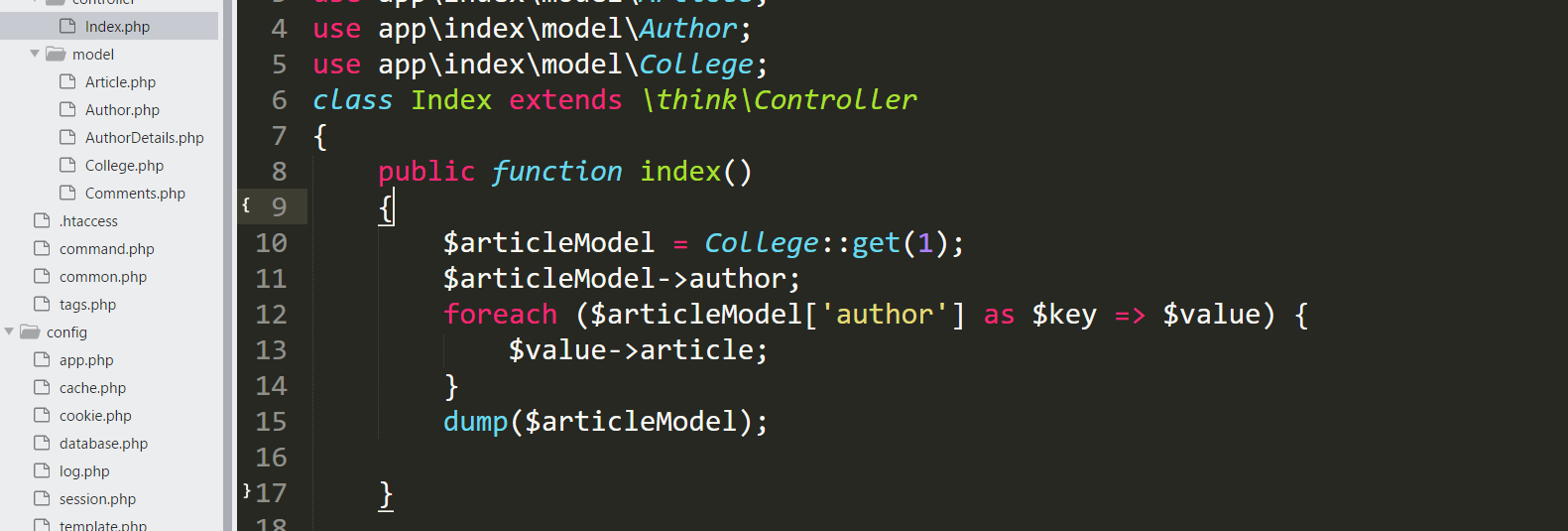


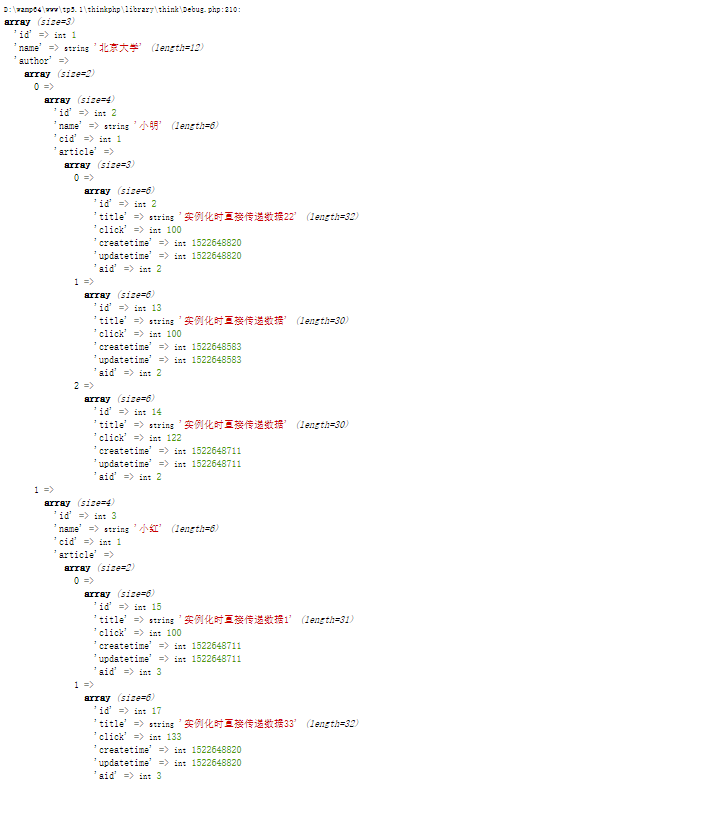
高级用法（手册未讲）：



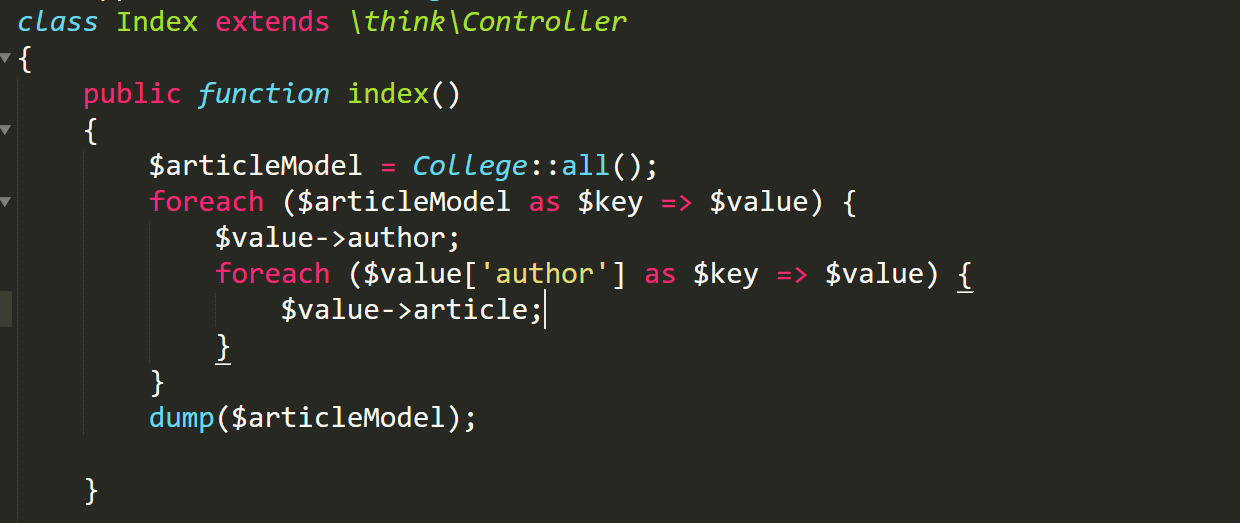


xdebug.var\_display\_max\_depth=10





—————————————————终极用法——————————————————



**array** *(size=4)*

0 =>

**array** *(size=3)*

'id' => int 1

'name' => string '北京大学' *(length=12)*

'author' =>

**array** *(size=2)*

0 =>

**array** *(size=4)*

'id' => int 2

'name' => string '小明' *(length=6)*

'cid' => int 1

'article' =>

**array** *(size=3)*

0 =>

**array** *(size=6)*

'id' => int 2

'title' => string '实例化时直接传递数据22' *(length=32)*

'click' => int 100

'createtime' => int 1522648820

'updatetime' => int 1522648820

'aid' => int 2

1 =>

**array** *(size=6)*

'id' => int 13

'title' => string '实例化时直接传递数据' *(length=30)*

'click' => int 100

'createtime' => int 1522648583

'updatetime' => int 1522648583

'aid' => int 2

2 =>

**array** *(size=6)*

'id' => int 14

'title' => string '实例化时直接传递数据' *(length=30)*

'click' => int 122

'createtime' => int 1522648711

'updatetime' => int 1522648711

'aid' => int 2

1 =>

**array** *(size=4)*

'id' => int 3

'name' => string '小红' *(length=6)*

'cid' => int 1

'article' =>

**array** *(size=2)*

0 =>

**array** *(size=6)*

'id' => int 15

'title' => string '实例化时直接传递数据1' *(length=31)*

'click' => int 100

'createtime' => int 1522648711

'updatetime' => int 1522648711

'aid' => int 3

1 =>

**array** *(size=6)*

'id' => int 17

'title' => string '实例化时直接传递数据33' *(length=32)*

'click' => int 133

'createtime' => int 1522648820

'updatetime' => int 1522648820

'aid' => int 3

1 =>

**array** *(size=3)*

'id' => int 2

'name' => string '南京大学' *(length=12)*

'author' =>

**array** *(size=1)*

0 =>

**array** *(size=4)*

'id' => int 4

'name' => string '小刚' *(length=6)*

'cid' => int 2

'article' =>

**array** *(size=2)*

0 =>

**array** *(size=6)*

'id' => int 19

'title' => string '使用create静态方法写入数据' *(length=36)*

'click' => int 100

'createtime' => int 1522649296

'updatetime' => int 1522649296

'aid' => int 4

1 =>

**array** *(size=6)*

'id' => int 20

'title' => string '使用create静态方法写入数据' *(length=36)*

'click' => int 100

'createtime' => int 1522649327

'updatetime' => int 1522649327

'aid' => int 4

2 =>

**array** *(size=3)*

'id' => int 3

'name' => string '东京大学' *(length=12)*

'author' =>

**array** *(size=1)*

0 =>

**array** *(size=4)*

'id' => int 5

'name' => string '小白' *(length=6)*

'cid' => int 3

'article' =>

**array** *(size=0)*

*empty*

3 =>

**array** *(size=3)*

'id' => int 4

'name' => string '西京大学' *(length=12)*

'author' =>

**array** *(size=1)*

0 =>

**array** *(size=4)*

'id' => int 6

'name' => string '小黑' *(length=6)*

'cid' => int 4

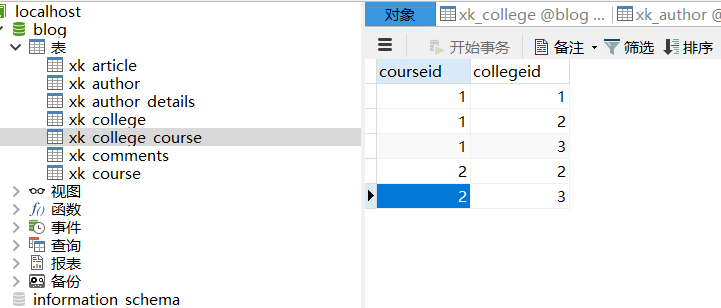
'article' =>

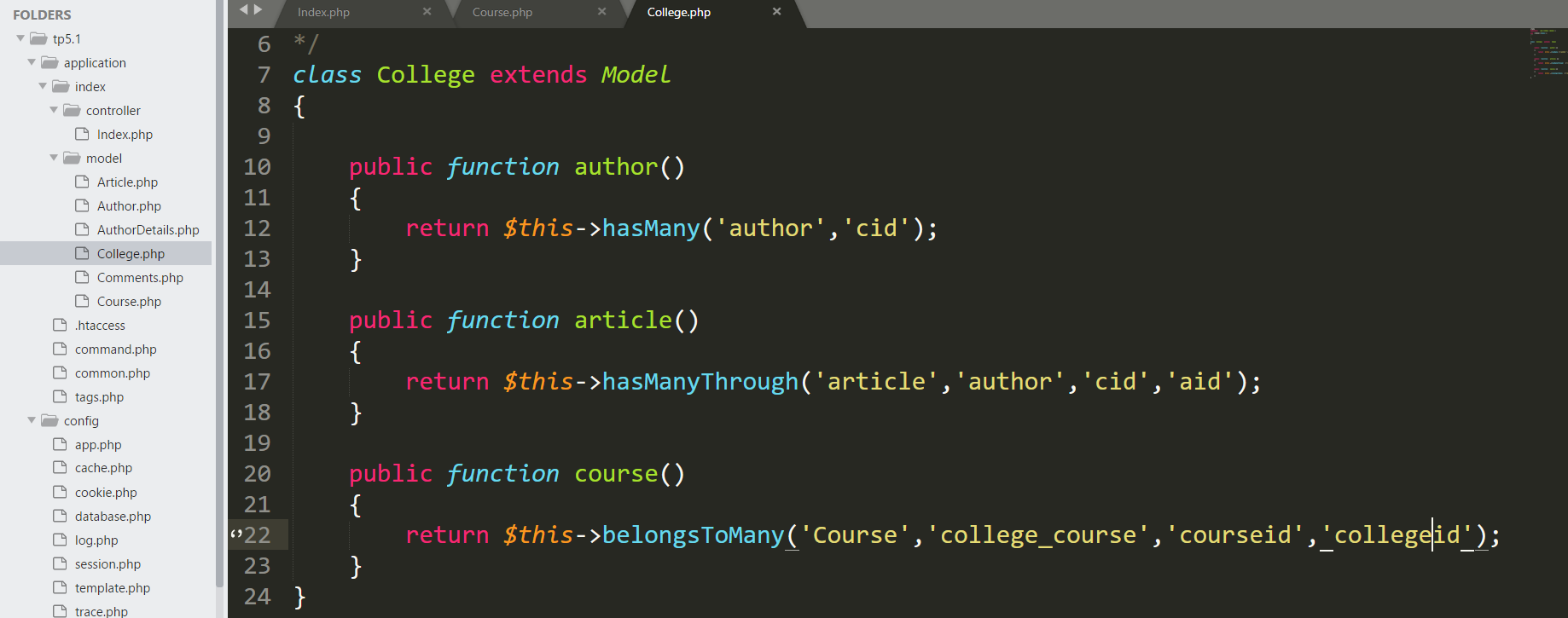
**array** *(size=0)*

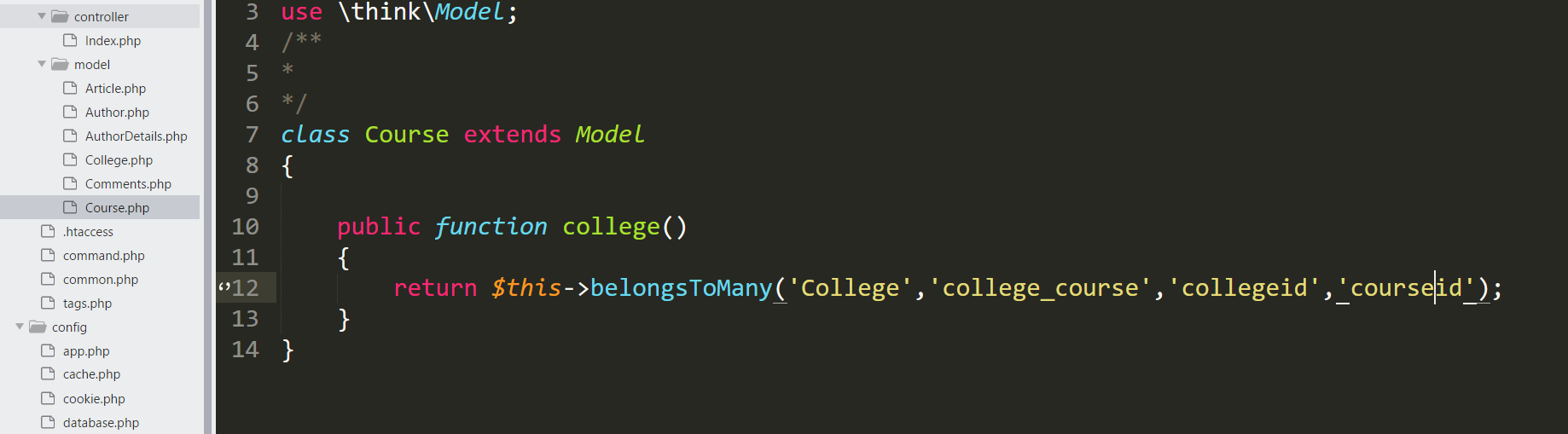
*empty*

多对多关联模型：

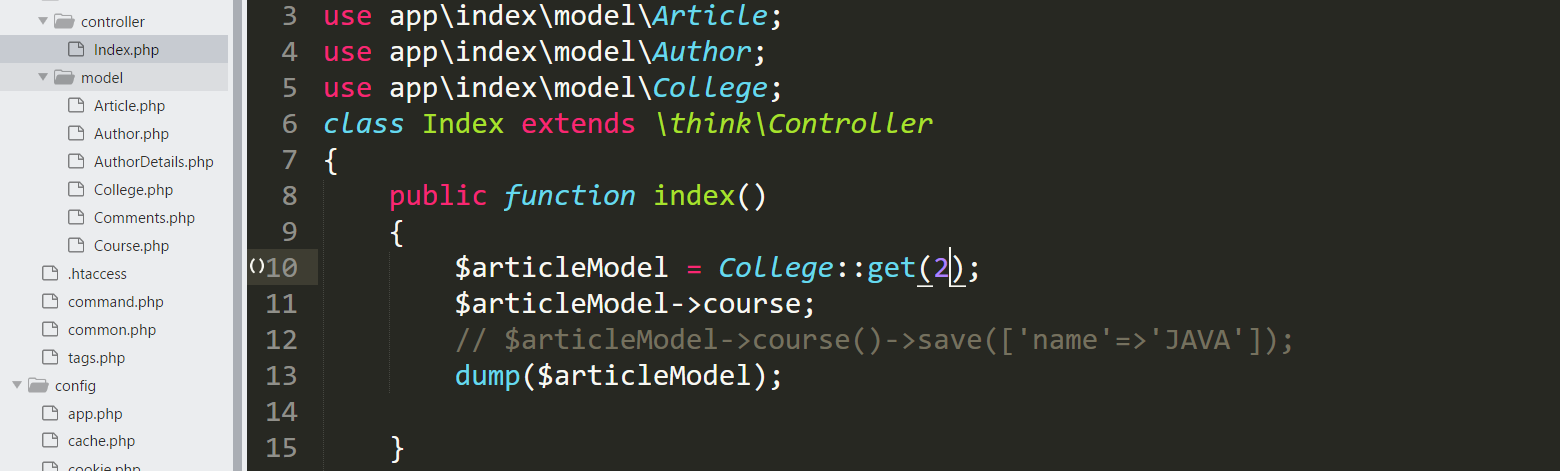
例如一个大学有多门课程，一个课程可以被多个大学教授；







关联查询：



**array** *(size=3)*

'id' => int 2

'name' => string '南京大学' *(length=12)*

'course' =>

**array** *(size=2)*

0 =>

**array** *(size=3)*

'id' => int 1

'name' => string '高等数学' *(length=12)*

'pivot' =>

**array** *(size=2)*

'courseid' => int 1

'collegeid' => int 2

1 =>

**array** *(size=3)*

'id' => int 2

'name' => string '大学英语' *(length=12)*

'pivot' =>

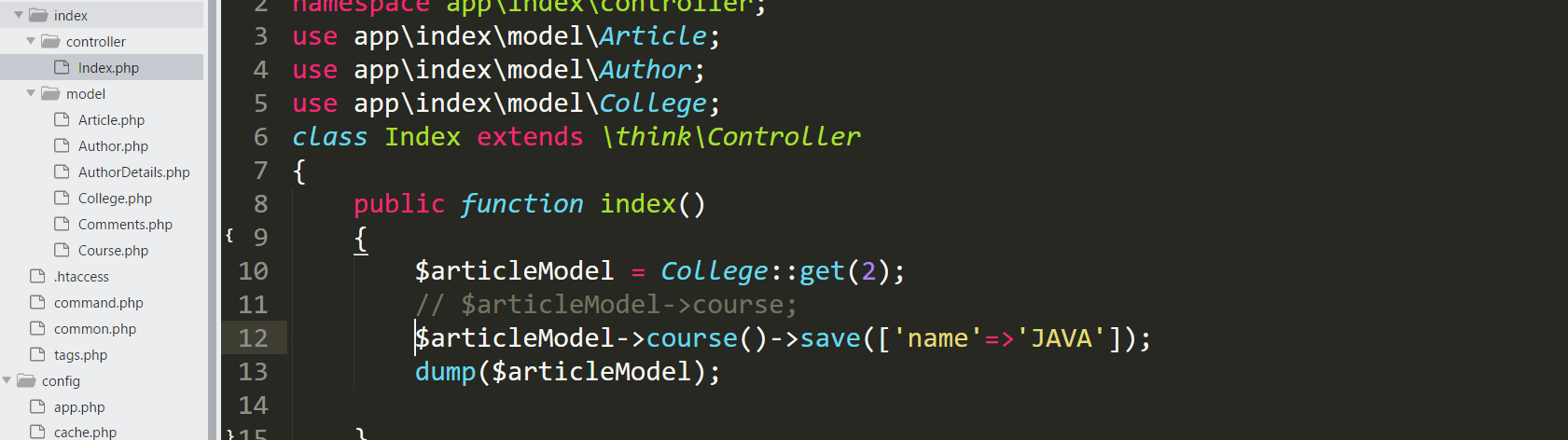
**array** *(size=2)*

'courseid' => int 2

'collegeid' => int 2

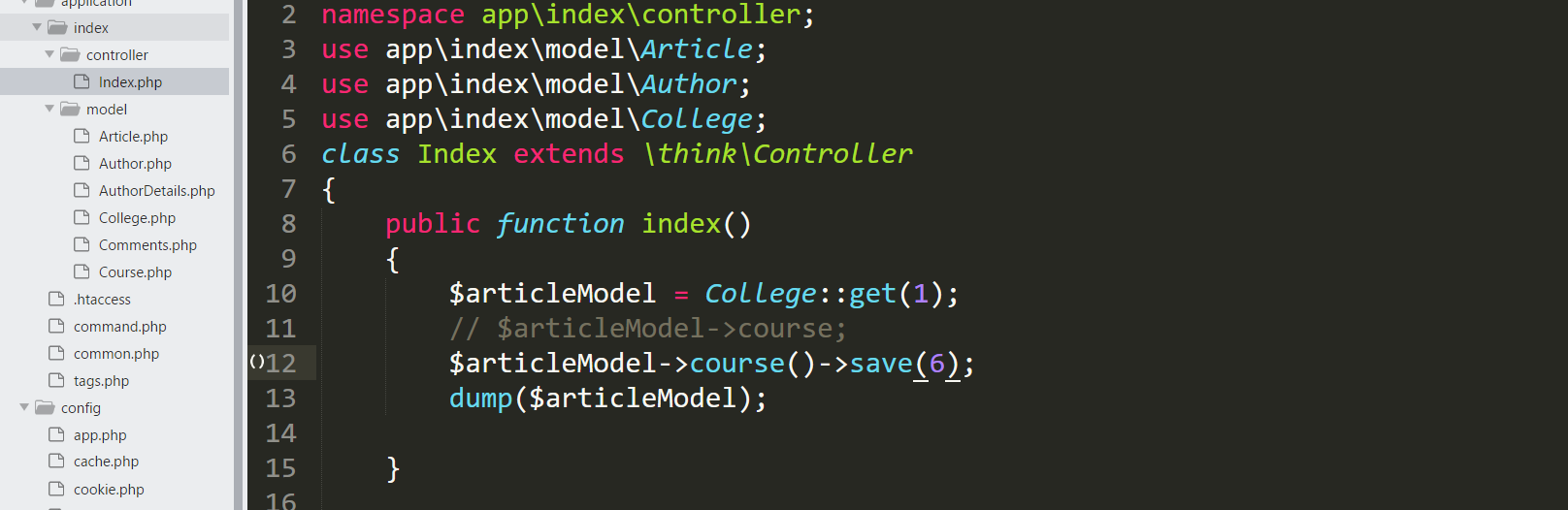
关联增加：

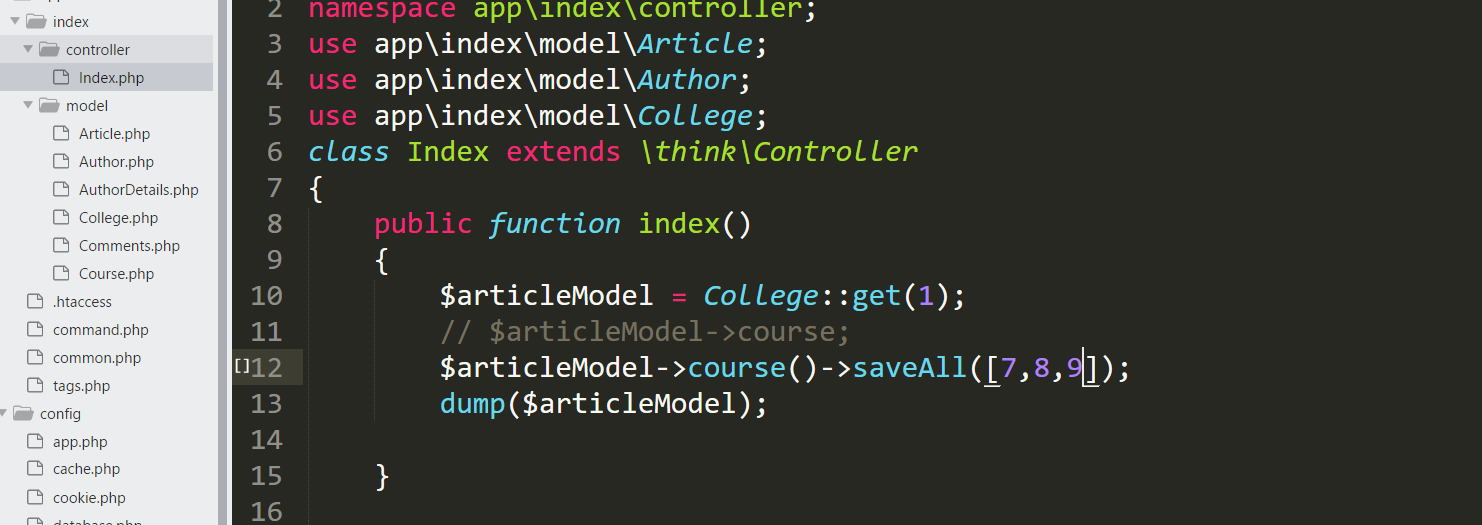
增加数据的同时，建立起关联关系。



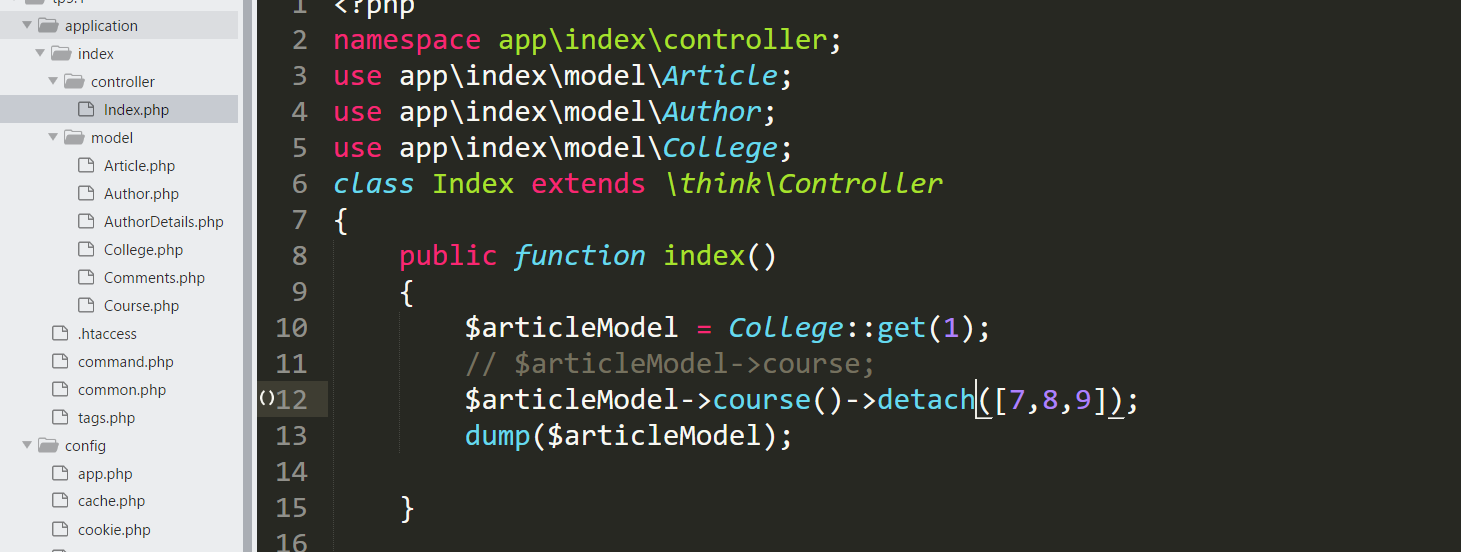


数据已经存在，只增加关联关系。



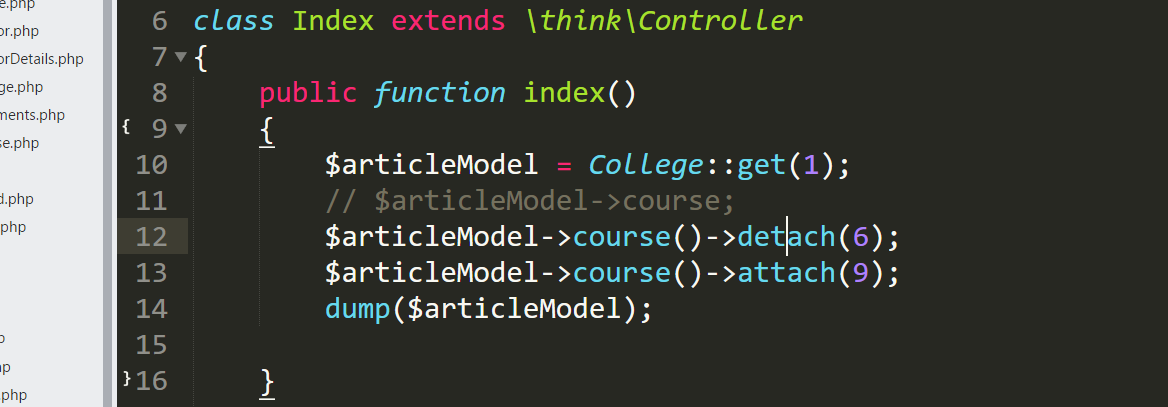


关联删除：



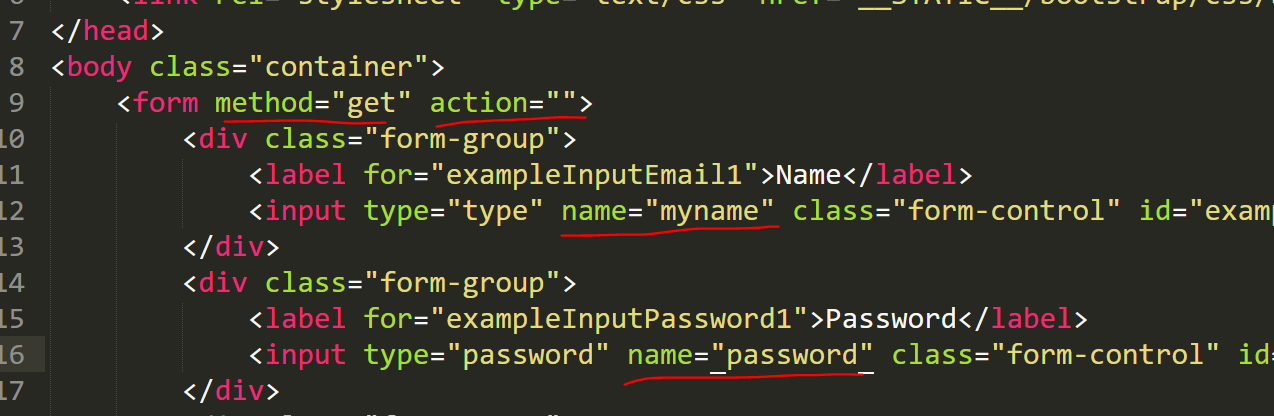
关联更新：

其实是先删除后增加



## 请求操作

请求对象：



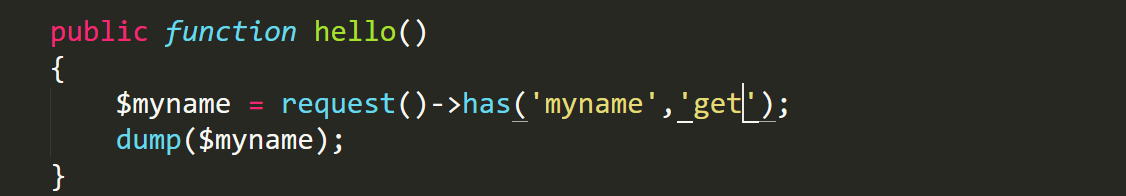


使用助手函数：



输入变量：

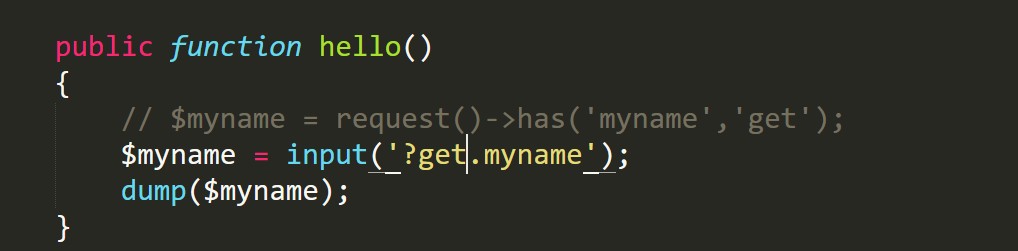
Has检测变量是否设置



注意：Pathinfo的参数已经不能使用get方式获取，需要使用param或者route方式进行获取。

Input助手函数

判断变量是否定义：



## 七、验证

1、html5(浏览器兼容性问题)；

2、js、jquery、vue…

3、服务器端验证.THINKPHP自带的验证规则.

4、数据库验证.

验证器：  
构造验证器：



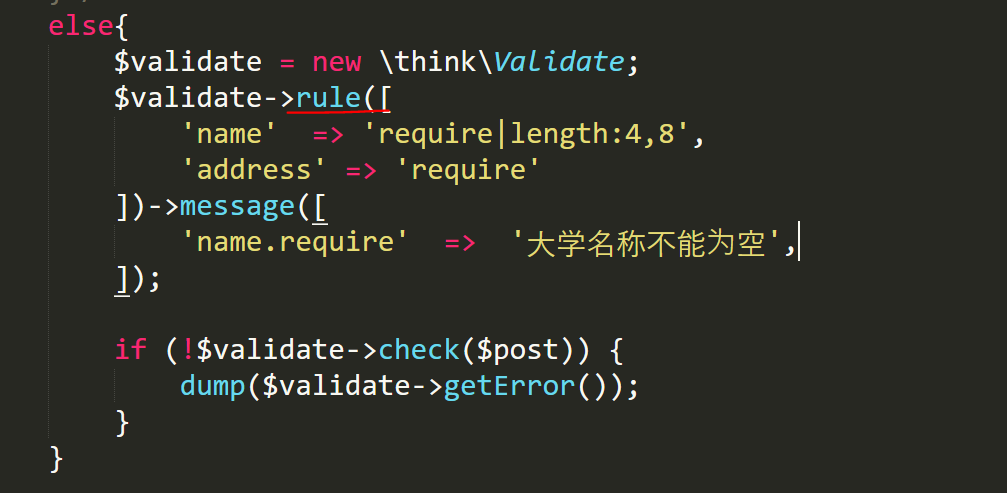


Tp5.1提供的更为简单的方法:



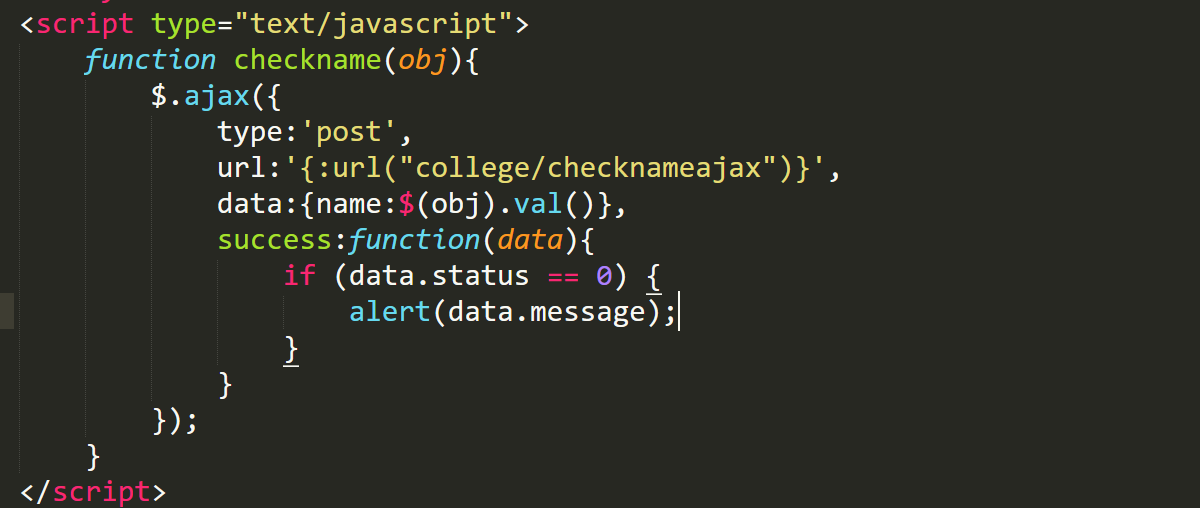
验证规则：

1. 在验证器里定义；
2. 单独使用：



验证场景：

例如ajax验证：







内置规则：

如果验证规则没有添加require就表示没有值的话不进行验证

'num'=>'between:1,10'（包含1和10有效）

'num'=>'notBetween:1,10'（1和10无效）

'name'=>'length:4,25'（包含4和25）

'name'=>'max:25'（包含25）

'name'=>'min:5'（包含5）

'begin\_time' => 'after:2016-3-18',（2016-3-18有效）

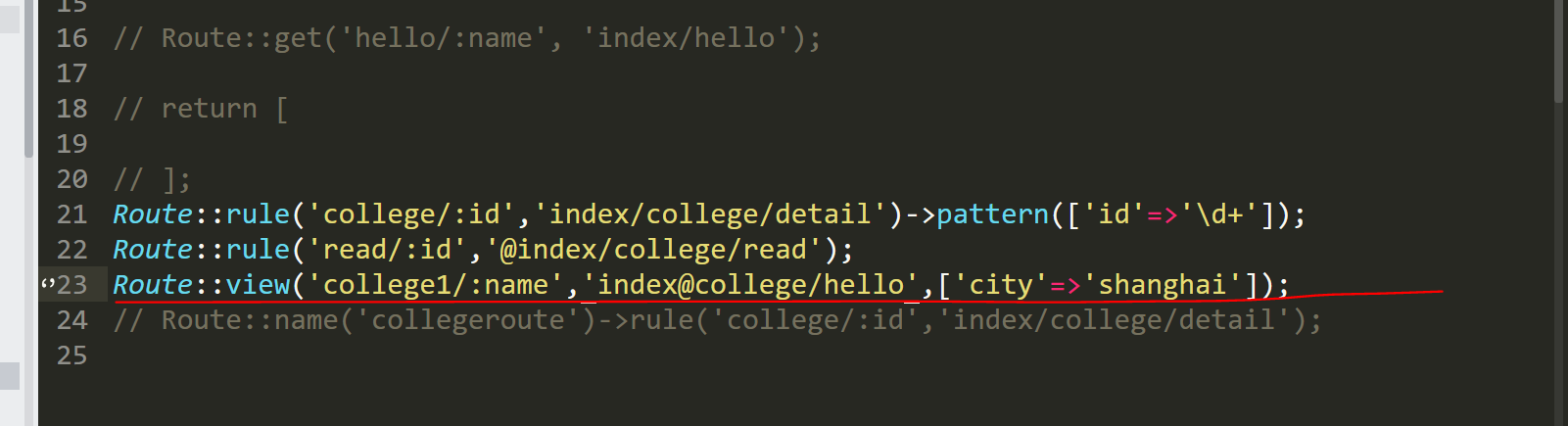
'end\_time' => 'before:2016-10-01',（2016-3-18有效）

验证当前操作（注意不是某个值）是否在某个有效日期之内，例如：

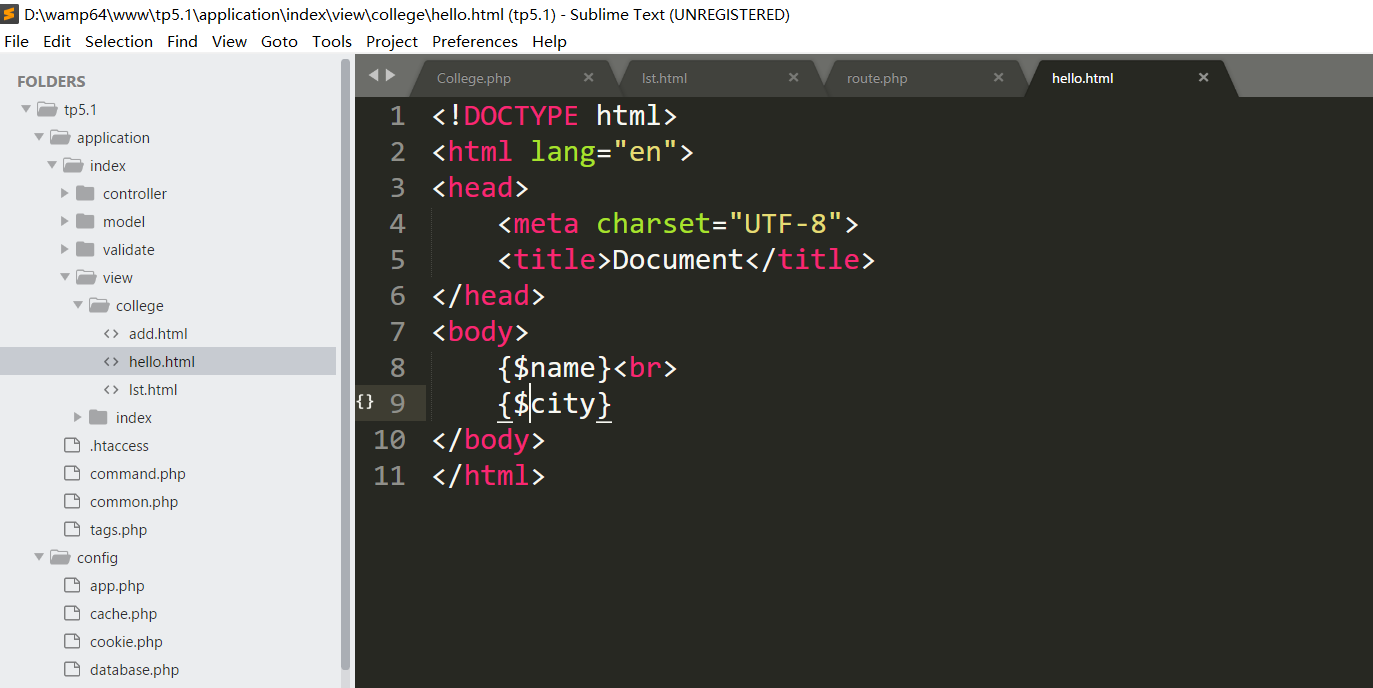
'expire\_time' => 'expire:2016-2-1,2016-10-01',

## 七、路由

路由到模板：

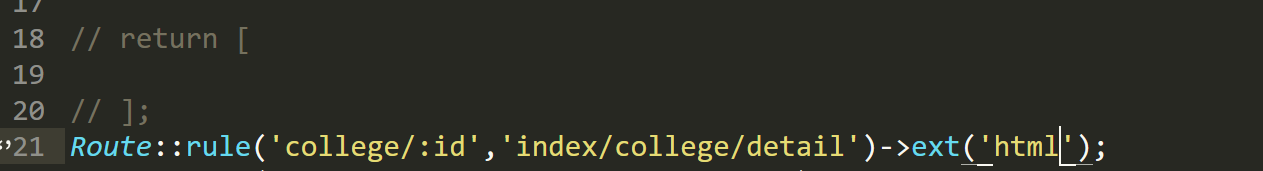


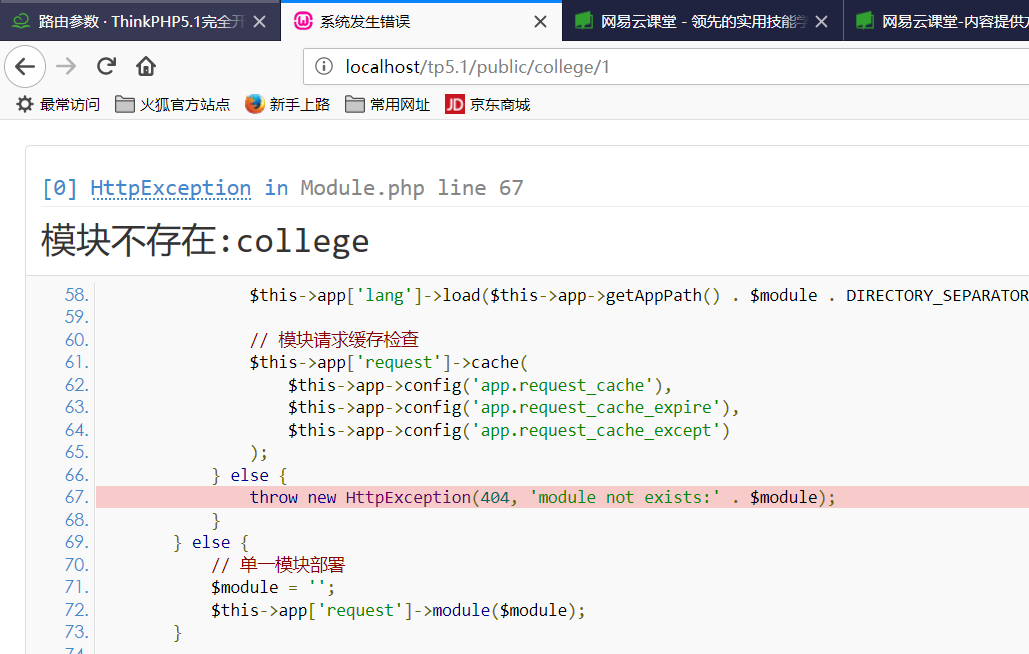






路由参数：

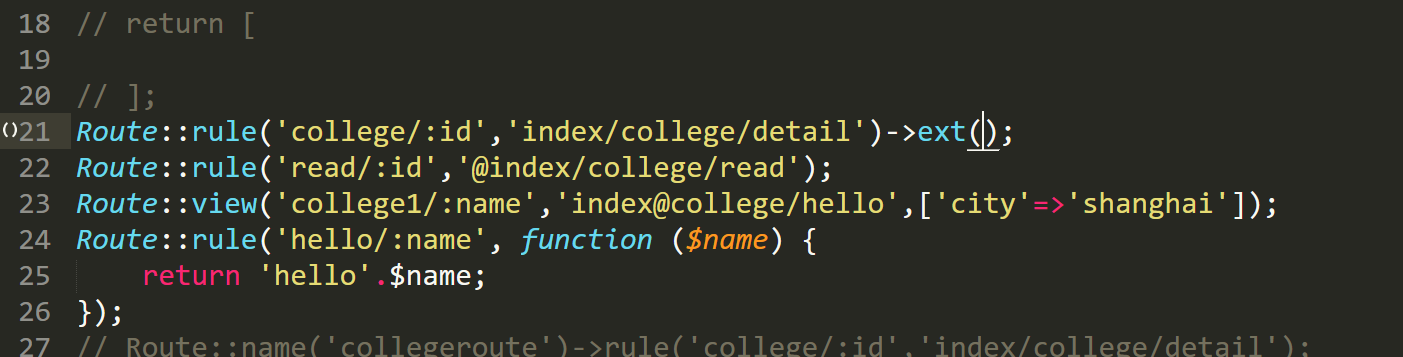


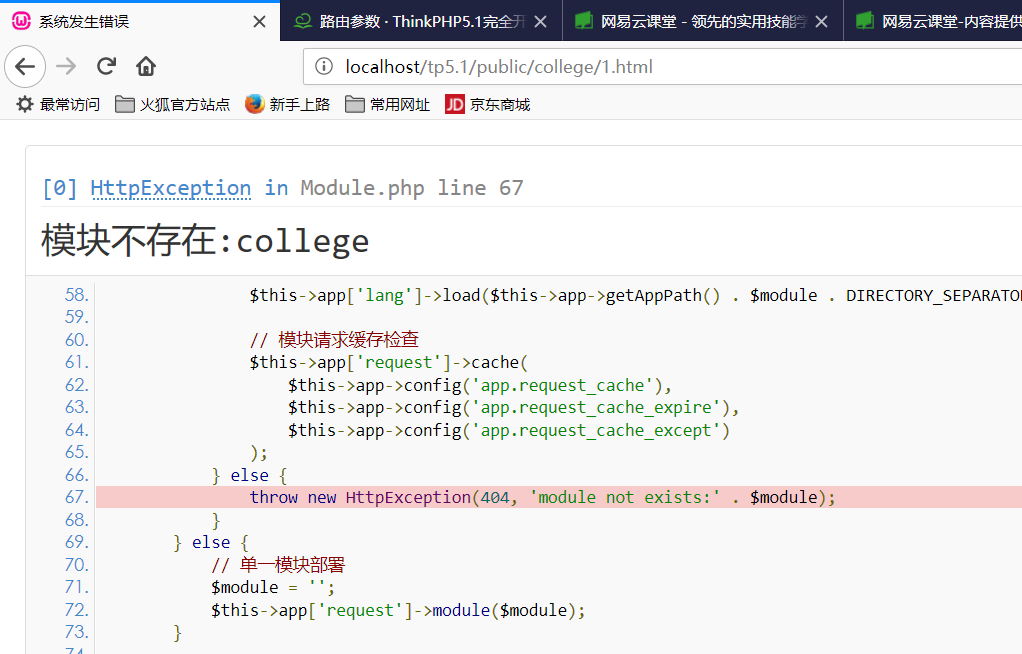




如果不设置ext那么就根据设置的后缀进行

如果ext不传入任何值，那么不允许后缀





生成路由简化后对应的url地址：

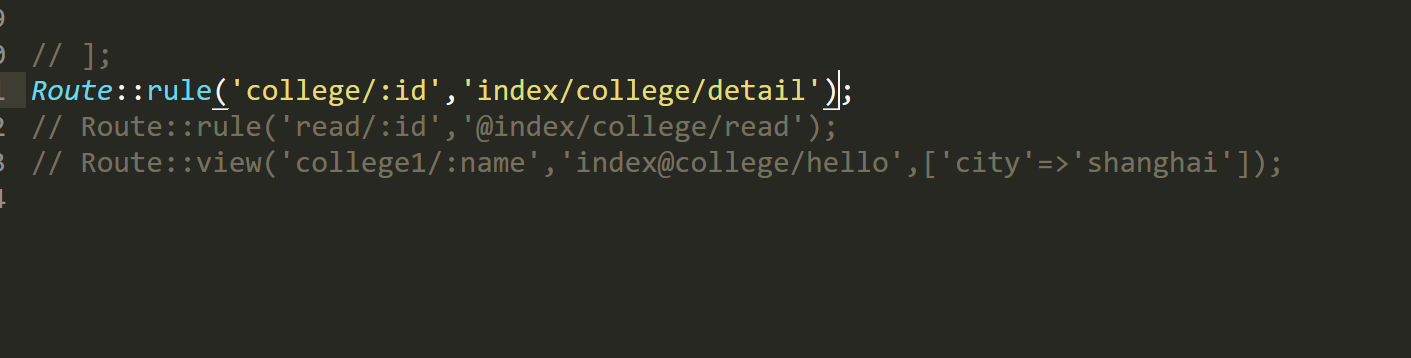
生成url有四种方式：

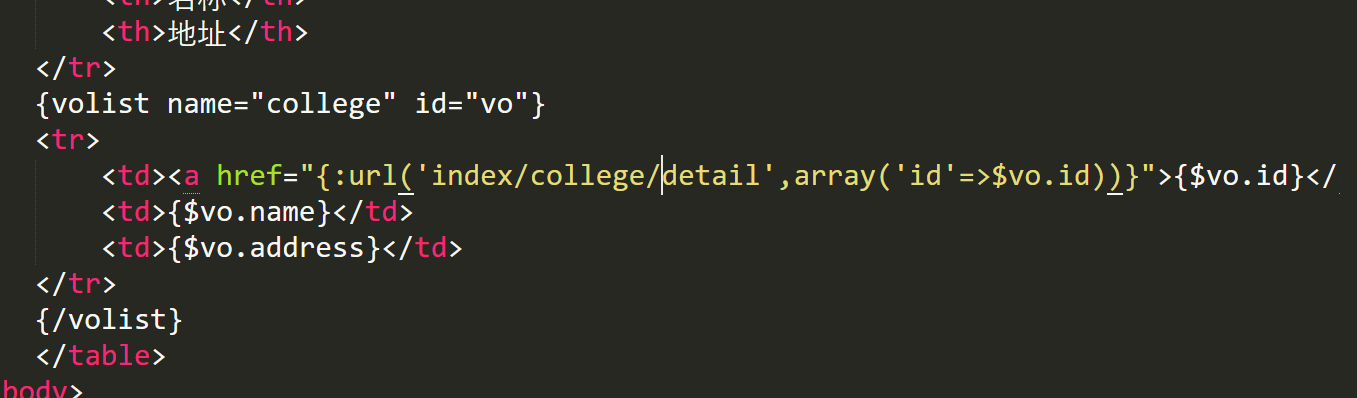
Url::build('index/blog/read', 'id=5&name=thinkphp');

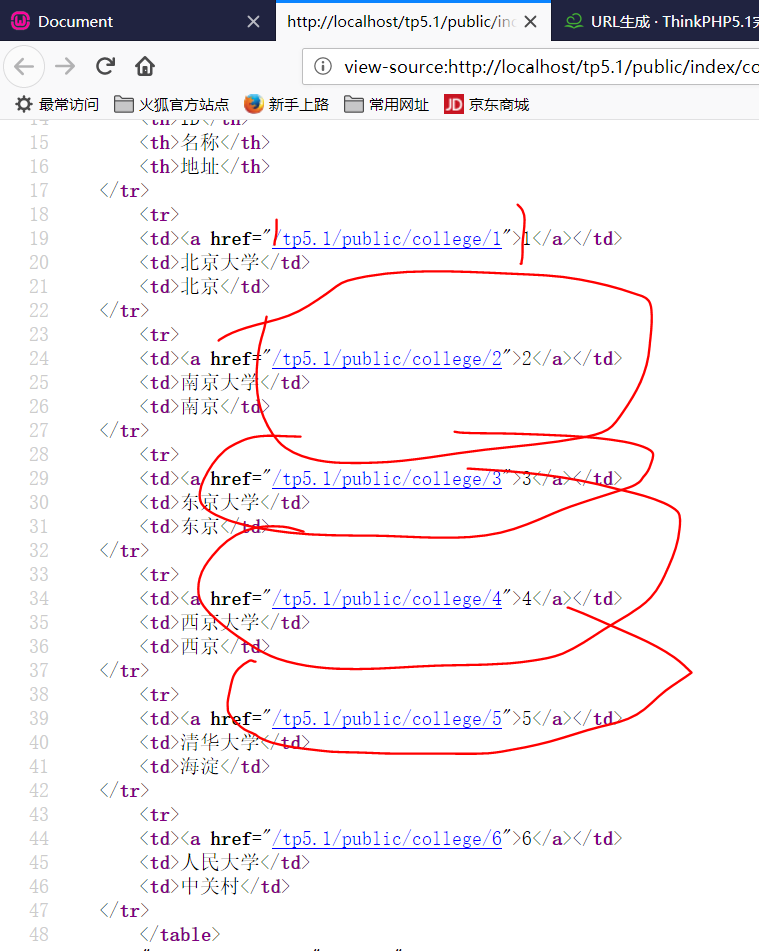
Url::build('index/blog/read', ['id' => 5, 'name' => 'thinkphp']);

url('index/blog/read', 'id=5&name=thinkphp');

url('index/blog/read', ['id' => 5, 'name' => 'thinkphp']);（推荐）







## 杂项

文件缓存：

Session

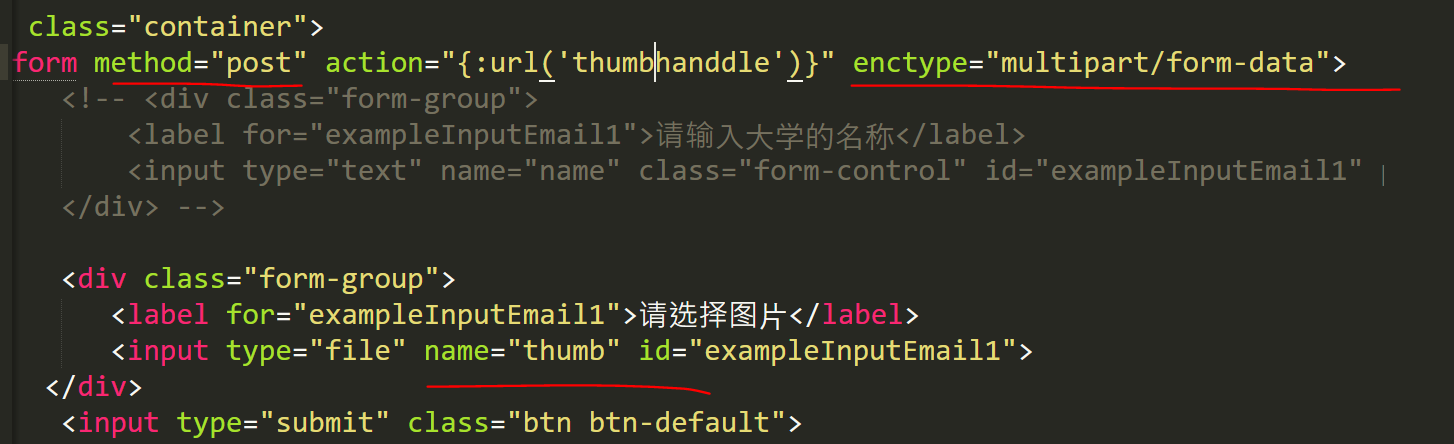
Session::set('name','thinkphp');第三个参数是作用域，也没有返回值。

Session::get('name');第二个参数指明作用域

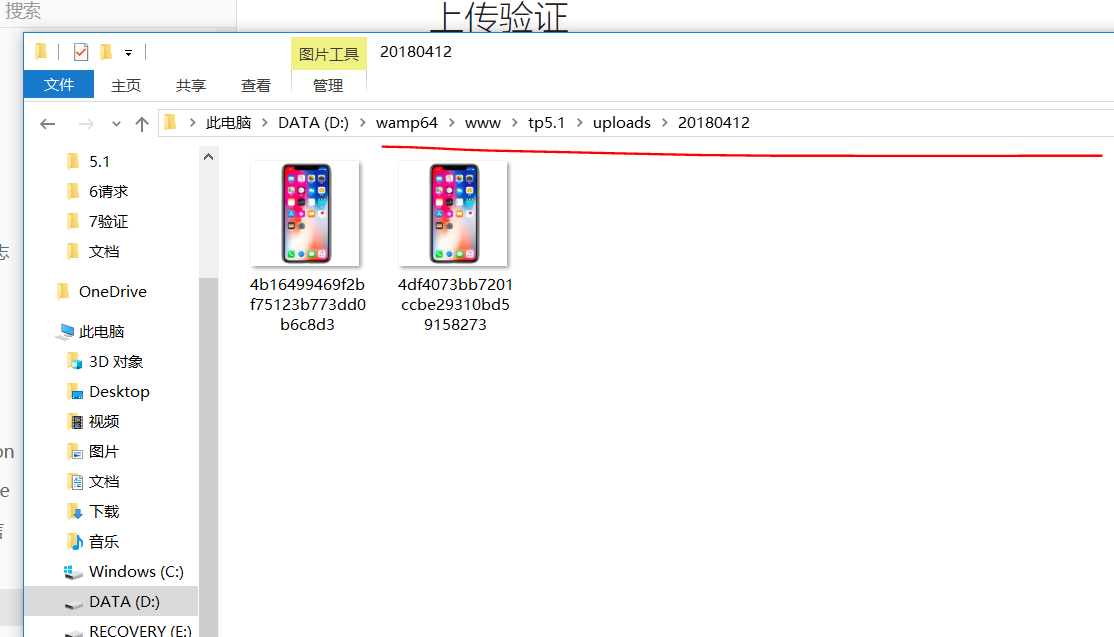
如果你没有使用Session类进行Session操作的话，例如直接操作$\_SESSION,必须使用上面的方式手动初始化或者直接调用session\_start()方法进行session初始化。

Cookie：Cookie::set('name','value',3600);第三个参数为时间。

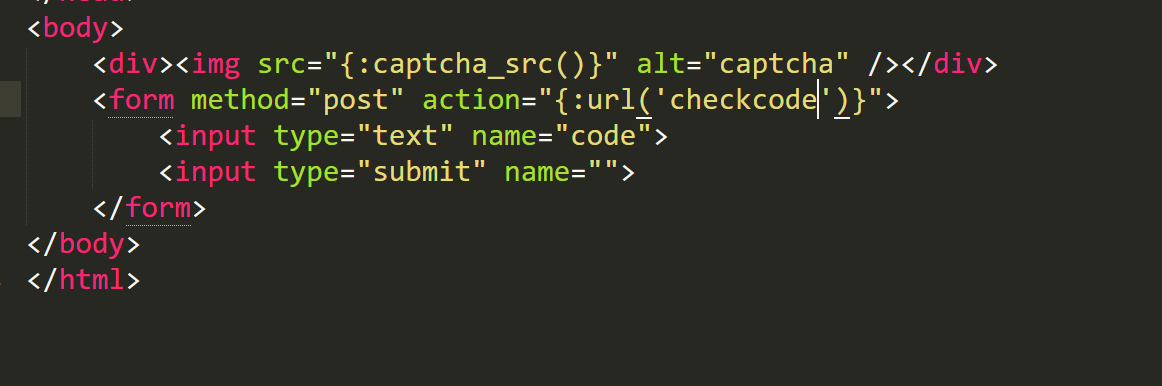
上传：







验证码：





图片处理：

$image = \think\Image::open('./image.png');

打开的是位于public文件夹下的图片。

缩略图位置：

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

## 个人博客项目

前台；

数据库的设计：

Article:

ID：id

作者:author

标题：title。

图片：img

发布时间：create\_time

修改时间：update\_time

文章具体内容:contents

(标签)多对多关系

Tags：

Id：

Name：标签名称

Article\_tags(中间表)

Article\_id

Tags\_id

后台：

Admin：

Id:

Name:

Password:

文章添加流程图：

将信息写入到数据库当中

跳转到文章添加界面

N

N

Y

Y

上传图片是否成功

Y

将图片上传到指定文件夹

跳转到文章添加界面

N

Y

Y

Y

是否有上传

N

跳转到文章添加界面

Y

Post请求数据是否通过验证

跳转到文章列表界面

N

是否是post请求

开始