学习thinkphp有半个月多了，首先入门的是做一个[博客系统](http://weibingsheng.cn/blog),跟着老师的教学，把原生的blog改成tp版的blog系统，之后用tp做了一个类似商业的网站（[六堡茶官网](htttp://lbc.wgscf.com)），对thinkphp更深入的学习了一番。

总的来说这个框架学起来还是蛮容易接受的，学习框架对于程序的开发来说缩短了周期，简化了操作步骤。

比如输入直接是：

I('post.name','','htmlspecialchars')

单字母函数比如I、D、U、A、M等，使用起来非常方便。

链接数据库可以用数组直接：

'DB\_TYPE' => 'mysql', // 数据库类型

'DB\_HOST' => 'localhost', // 服务器地址

'DB\_NAME' => 'thinkphp', // 数据库名

'DB\_USER' => 'root', // 用户名

'DB\_PWD' => '123456', // 密码

'DB\_PORT' => 3306, // 端口

'DB\_PREFIX' => 'think\_', // 数据库表前缀

'DB\_CHARSET'=> 'utf8', // 字符集

'DB\_DEBUG' => TRUE, // 数据库调试模式 开启后可以记录SQL日志 3.2.3新增

简单设置就可以愉快的进行网站开发了。

实例化数据表可以直接用D函数，模型调用方便。

对数据进行分页是一个专题，当时卡了将近一天，后来发现了问题所在。

$User = M('User'); // 实例化User对象

$count = $User->where('status=1')->count();// 查询满足要求的总记录数

$Page = new \Think\Page($count,25);// 实例化分页类 传入总记录数和每页显示的记录数(25)

$show = $Page->show();// 分页显示输出

// 进行分页数据查询 注意limit方法的参数要使用Page类的属性

$list = $User->where('status=1')->order('create\_time')->limit($Page->firstRow.','.$Page->listRows)->select();

$this->assign('list',$list);// 赋值数据集

$this->assign('page',$show);// 赋值分页输出

$this->display(); // 输出模板

这里注意$Page这个变量不可以和系统冲突，否则不会成功。

然后上传功能也是一个主题，当时还是得出现了很多问题，要么是没有提示，要么是提示目录问题，提示非法问题。

发现需要在form里面添加属性

<form action="\_\_URL\_\_/upload" enctype="multipart/form-data" method="post" >

<input type="text" name="name" />

<input type="file" name="photo" />

<input type="submit" value="提交" >

</form>

实例化上传类

$upload = new \Think\Upload();// 实例化上传类

$upload->maxSize = 3145728 ;// 设置附件上传大小

$upload->exts = array('jpg', 'gif', 'png', 'jpeg');// 设置附件上传类型

$upload->rootPath = './Uploads/'; // 设置附件上传根目录

$upload->savePath = ''; // 设置附件上传（子）目录

// 上传文件

$info = $upload->upload();

if(!$info) {// 上传错误提示错误信息

$this->error($upload->getError());

}else{// 上传成功

$this->success('上传成功！');

}

这里对传过来的数据进行判断，个性化修改提示信息，用在后台修改图片和其他文件时非常有必要。

获取客户端的IP地址也非常简单

$ip = get\_client\_ip();

如果要支持IP定位功能，需要使用扩展类库Org\Net\IpLocation，并且要配合IP地址库文件一起使用，例如：

$Ip = new \Org\Net\IpLocation('UTFWry.dat'); // 实例化类 参数表示IP地址库文件

$area = $Ip->getlocation('203.34.5.66'); // 获取某个IP地址所在的位置

​

登录和注册需要验证码，直接实例化使用内置的类即可：

$Verify = new \Think\Verify();

$Verify->entry();

设置SESSION值：

session('name','value'); //设置session

$value = session('name');//获取session

session('name',null); // 删除name

设置COOKIE值：

cookie('name','value'); //设置cookie

cookie('name','value',3600); // 指定cookie保存时间

$value = cookie('name');//取值

cookie('name',null);//删除

控制器的定义控制器的名称采用驼峰法命名（首字母大写），控制器文件位于 Home/Controller/IndexController.class.php。

<?php

namespace Home\Controller;

use Think\Controller;

class IndexController extends Controller {

public function hello(){

echo 'hello';

}

public function test(){

echo 'test';

}

}

跳转和重定向

$User = M('User'); //实例化User对象

$result = $User->add($data);

if($result){

//设置成功后跳转页面的地址，默认的返回页面是$\_SERVER['HTTP\_REFERER']

$this->success('新增成功', 'User/list');

} else {

//错误页面的默认跳转页面是返回前一页，通常不需要设置

$this->error('新增失败');

}

// 操作完成3秒后跳转到 /Article/index

$this->success('操作完成','/Article/index',3);

// 操作失败5秒后跳转到 /Article/error

$this->error('操作失败','/Article/error',5);

//重定向到指定的URL地址

redirect('/New/category/cate\_id/2', 5, '页面跳转中...')

数据的CURD操作：

创建用create();

插入用add();

更新用save();

删除用delete();

自动验证表单合法性：

$rules = array(

array('verify','require','验证码必须！'), //默认情况下用正则进行验证

array('name','','帐号名称已经存在！',0,'unique',1), // 在新增的时候验证name字段是否唯一

array('value',array(1,2,3),'值的范围不正确！',2,'in'), // 当值不为空的时候判断是否在一个范围内

array('repassword','password','确认密码不正确',0,'confirm'), // 验证确认密码是否和密码一致

array('password','checkPwd','密码格式不正确',0,'function'), // 自定义函数验证密码格式

);

$User = M("User"); // 实例化User对象

if (!$User->validate($rules)->create()){

// 如果创建失败 表示验证没有通过 输出错误提示信息

exit($User->getError());

}else{

// 验证通过 可以进行其他数据操作

}

模板中变量输出：

$name = 'ThinkPHP';

$this->assign('name',$name);

$this->display();

如果是数组可以这样：

Name：{$data.name}

Email：{$data.email}

使用函数时可以这样：

{$data.name|md5}

也可以在模板中直接用php代码：

<?php echo (date("y-m-d",$create\_time)); ?>

定义模型，名字一定要和数据表名字一致，否则提示改模型下无此方法

<?php

namespace Home\Model;

use Think\Model;

class SettingsModel extends Model {

public function t(){

echo 3243;

}

}}

​

还有很多细节的东西，只是偶尔记录一下。学习thinkphp还需要不断接触，不断解决遇到的问题。

我一直坚信“解决问题的过程就是进步的过程”所以，不断学习才是真理。