

主讲教师：传智.神龙教主

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 回顾

1. cookie：将数据存放在请求头和响应头中
2. cookie中只能存放字符串和数字
3. setcookie(名，值，过期时间，有效目录，子域名)
4. cookie分为临时性cookie和永久性cookie
   1. 临时性cookie:关闭浏览器就消失
   2. 永久性cookie:可以将信息永久保存在客户端，关闭浏览器不消失
   3. 永久性cookie比临时性cookie多了过期时间
5. 删除cookie
   1. Setcookie(‘名字’,false)
   2. Setcookie(‘名字’,’’);
   3. Setcookie(‘名字’) 清空cookie
6. Cookie中不能存放数组，但是支持下标
7. 默认情况下，cookie只能在当前目录和子目录中生效。一般用”/”设置在整个站点下有效
8. 默认Cookie分区子域名。可以设置在同一级域名下有效
9. Cookie的缺点
   1. 安全性差
   2. 数据量只能有4K大小
   3. 可控性比较差
10. Session：session是基于cookie的技术
11. 要使用session，默认要显式的开启会话
12. 要自动开启会话，在php.ini中session.auto\_start=1。
13. Session\_id()：用来获取会话编号
14. Session\_name()：读取会话名称
15. Session存储的是除了资源类型以外的数据
16. 重复开启会话会提示一个提示性错误
17. 通过session.save\_path=‘N；路径’设置会话保存的路径
18. Session.name 设置会话名字
19. Session.hash\_bits\_per\_character 每个会话名字的字符占用的字节
20. Session.cookie\_lieftime=0; sessionID在客户端保存的时间
21. Session.cookie\_path=/; 整站有效
22. Session.save\_handler=files; 会话按文件来存储
23. Session.gc\_maxlifetime=1440; 会话在服务器保存的时间
24. Session\_destroy()：销毁会话，销毁存储介质上的文件
25. 自定义会话存储，session\_set\_save\_handler(open,close,read,write,destroy,gc)
26. Session.gc\_probability=1 概率
27. Session.gc\_divisor=1000 除数
28. 如果禁用cookie，默认情况下基于cookie的技术都无法使用
29. 可以通过配置更改
    1. Session.use\_only\_cookie=0 ;session不仅仅依赖cookie
    2. Session.use\_trans\_sid=1; 允许通过其他形式传递sid

## Ecshop介绍

ecshop是一个开源的商城系统。

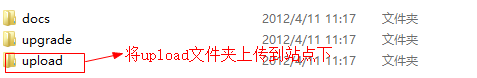
### 获取安装包

到ecshop官方网站上下载安装包

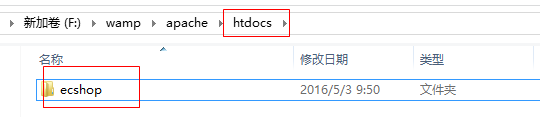


### 解压

解压后会发现3个文件夹，我们只需要upload文件夹的内容



将upload文件夹拷贝到站点下，并改名为ecshop。



### 安装ecshop

安装ecshop，就是创建数据库和初始化配置文件信息的过程

在地址中输入http://www.php.com/ecshop



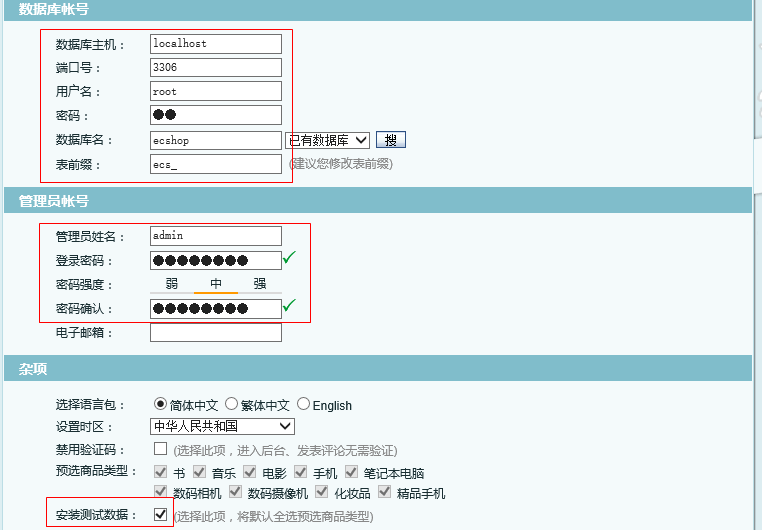
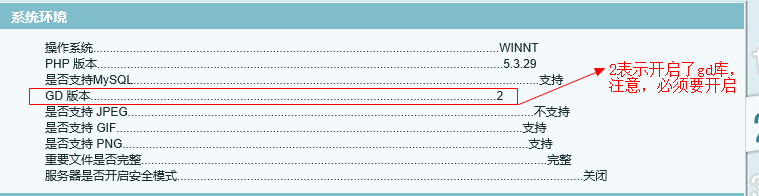
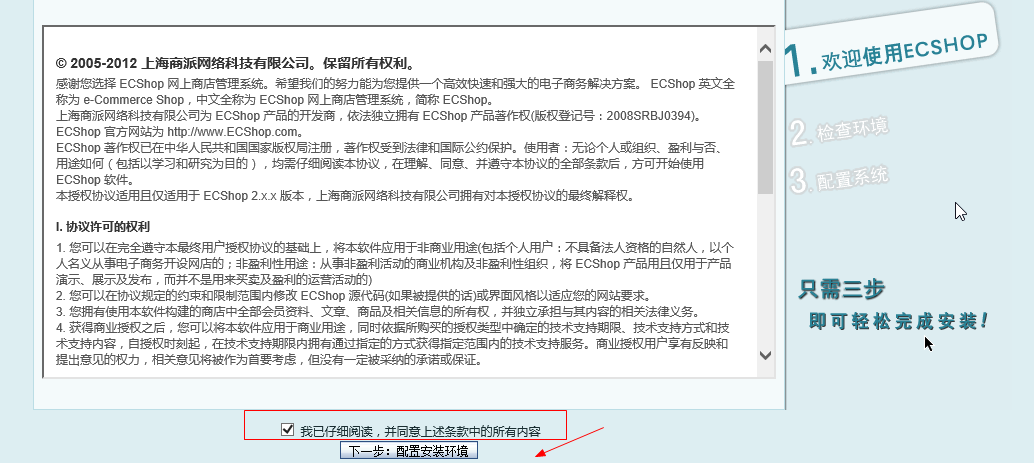
如果没有安装过ecshop，就跳转到install/index.php页面



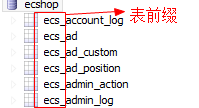
脚下留心：在安装ecshop之前必须要开启gd2库扩展，此扩展用于处理图片，如果没有开启，则不能使用验证码等与图片相关的功能。



步骤：

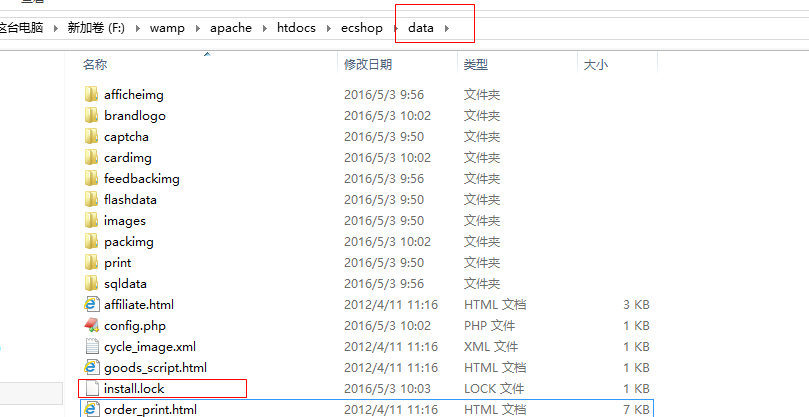


多学一招：为什么数据库中的表要添加表前缀？



一个数据库中如果有多个项目的表，为了区分每个表属于哪个项目就通过表前缀来区分。

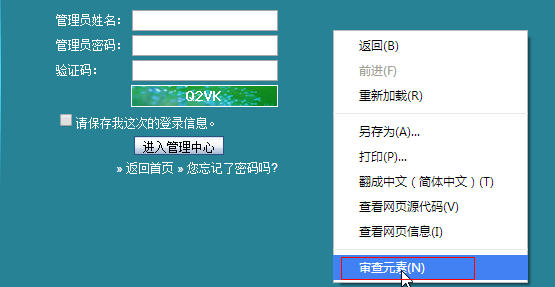
多学一招：ecshop安装完毕后，会在data文件夹下多一个install.lock文件，如果要重新安装ecshop就将此文件删除即可。



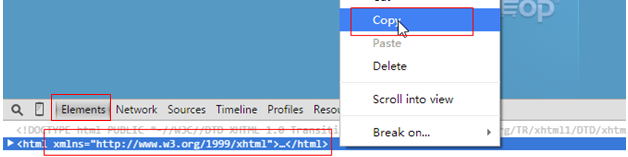
## 将登陆界面扒下来

在crome浏览器上右键——审查元素

’

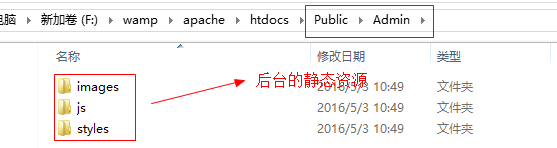


在Elements下选择HTML代码右键——copy



将拷贝的HTML代码粘贴到项目中（login.html）。

将后台的静态资源（ecshop/Admin）拷贝到Public/Admin文件中



### 显示登陆页面

控制器

<?php

/\*

\* 登陆的控制器

\*/

class LoginController extends Controller{

//登陆页面

public function loginAction(){

require \_\_VIEW\_\_.'login.html';

}

}

视图：注意更改静态资源的路径



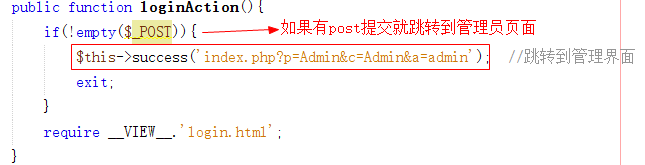
//注意更改静态资源路径



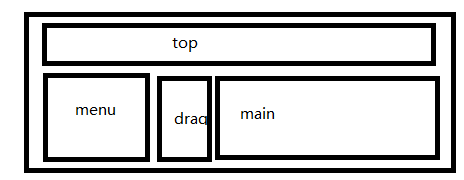
模型

没有

### 点击“进入管理中心”跳转到管理页面



管理界面由四个框架组成



问题：一个页面有N个框架组成，请问保存几个页面？

答：由N+1个文件组成。

控制器

class AdminController extends Controller{

public function adminAction(){

require \_\_VIEW\_\_.'admin.html';

}

public function topAction(){

require \_\_VIEW\_\_.'top.html';

}

public function menuAction(){

require \_\_VIEW\_\_.'menu.html';

}

public function dragAction(){

require \_\_VIEW\_\_.'drag.html';

}

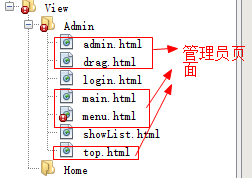
public function mainAction(){

require \_\_VIEW\_\_.'main.html';

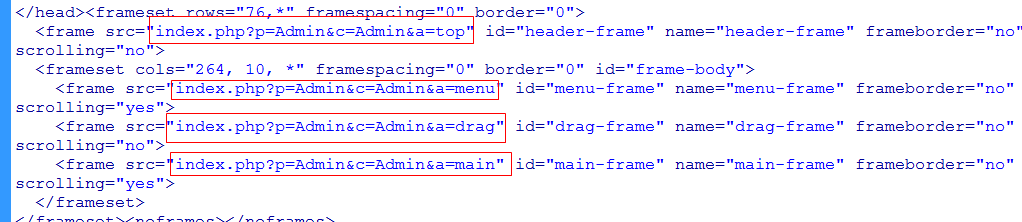
}

}

视图（要注意更改静态资源的路径）



admin.html中的框架中包含其他页面



模型

无

## 实现登陆功能

### 管理员表（admin）

drop table if exists admin;

create table admin(

id int unsigned auto\_increment primary key comment '主键',

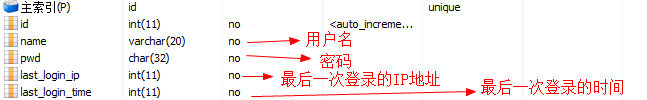
name varchar(30) not null comment '用户名',

pwd char(32) not null comment '密码 md5加密',

last\_login\_ip int not null comment '最后一次登录的地址',

last\_login\_time int not null comment '最后一次登录的时间'

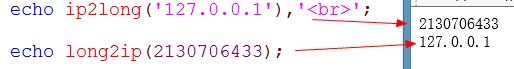
)engine=innodb charset=utf8;



插入测试数据

insert into admin values (null,'aa',md5('aa'),0,0);

### 将IP地址和整数的转换



思考：为什么IP用整形存储？

答：如果用IP地址直接存储，需要15个字节长度，如果转成整形只需要4个字节的长度；另外字符串的运算速度没有整数快。

### 登陆的业务逻辑

控制器

public function loginAction(){

if(!empty($\_POST)){ //登陆的业务逻辑

$username=$\_POST['username']; //获取用户名

$pwd=$\_POST['password']; //获取密码

$model=new AdminModel();

$info=$model->getAdminByUserNameAndPwd($username, $pwd); //通过用户名和密码获取管理员信息

if($info)

$this->success('index.php?p=Admin&c=Admin&a=admin','登陆成功',1); //跳转到管理界面

else

$this->error ('index.php?p=Admin&c=Login&a=login', '登陆失败');

exit;

}

require \_\_VIEW\_\_.'login.html';

}

模型

<?php

class AdminModel extends Model{

/\*

\* 通过用户名和密码获取用户信息

\* @param $username string 用户名

\* @param $pwd string 密码

\* @return array 用户的信息

\*/

public function getAdminByUserNameAndPwd($username,$pwd){

$pwd=md5($pwd); //将密码进行md5加密

$sql="select \* from admin where name='$username' and pwd='$pwd'";

return $this->db->fetchRow($sql); //返回用户信息

}

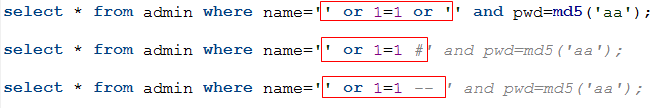
}

视图

无

## 防止SQL注入

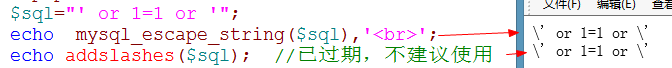
利用拼接SQL语句来达到绕过正常的SQL查询的技术称为SQL注入



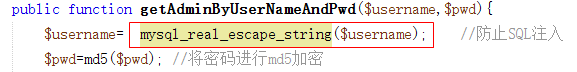
因为上面的SQL语句有1=1，所以肯定能获取到表中的数据

### 解决

给单引号加上转义字符



代码



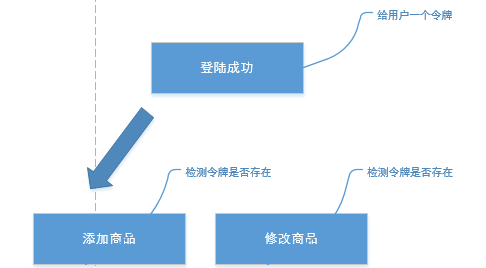
md5加密后不需要在防止SQL注入，因为md5加密后将单引号转成了其他字符。

## 防止翻墙

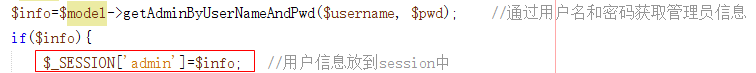
直接通过URL地址进入后台

### 解决

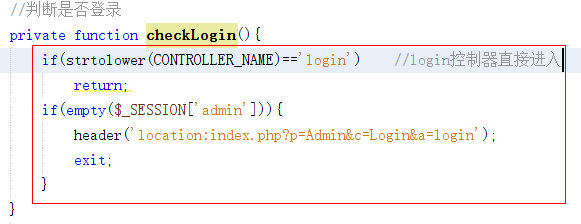
登陆成功后，给用户一个令牌，执行每个操作之前先判断令牌是否存在，如果不存在就跳转到登陆页面



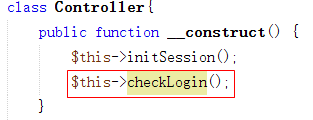
登陆成功后，将用户信息存放到会话中



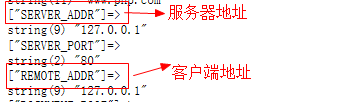
在基础模型类中，写一个方法用来判断是否登录



在构造函数中调用验证登陆的方法



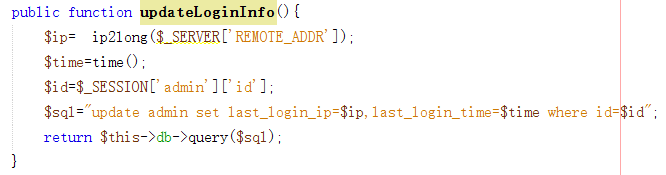
## 更新登陆信息



思考：为什么$\_SERVER['REMOTE\_ADDR']是客户端地址？

答：因为这段代码是PHP代码，PHP代码运行在服务器端，服务器的远程是客户端。

模型(AdminModel)



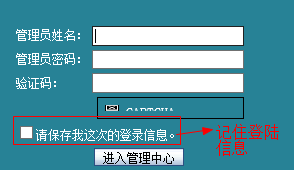
控制器(LoginController)



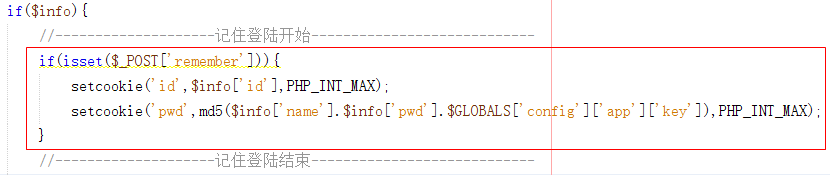
视图

没有

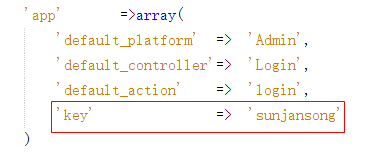
## 记住登陆



### 登陆成功后记住登陆信息



脚下留心：在配置中添加秘钥



### 打开登陆页面的时候如果从cookie中的信息登陆成功，直接跳转到管理界面

模型

/\*

\* 通过cookie获取用户信息

\*/

public function getAdminByCookie(){

if(!isset($\_COOKIE['id']) || !isset($\_COOKIE['pwd'])){

return false;

}

$id= mysql\_real\_escape\_string($\_COOKIE['id']);

$pwd=$\_COOKIE['pwd'];

$sql="select \* from admin where id=$id";

if($info= $this->db->fetchRow($sql)){

return md5($info['name'].$info['pwd'].$GLOBALS['config']['app']['key'])==$pwd?$info:false;

}

return false;

}

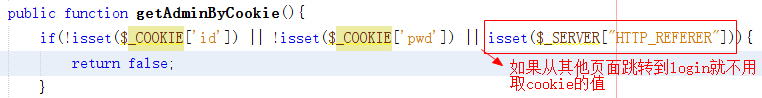
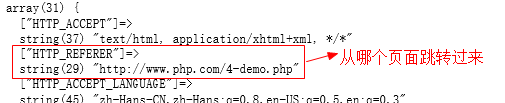
控制器



问题：点击“退出”，跳转到login页面，但是login页面又获取了cookie，又跳转到管理员页面，根本跳不到login页面上。



解决：如果直接登陆login页面就获取cookie,否则不用 取cookie



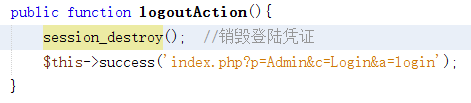
## 安全退出



退出：提出的时候不销毁登陆凭证

安全退出：退出的时候销毁登陆凭证

在LoginController控制器中



添加“退出”按钮

