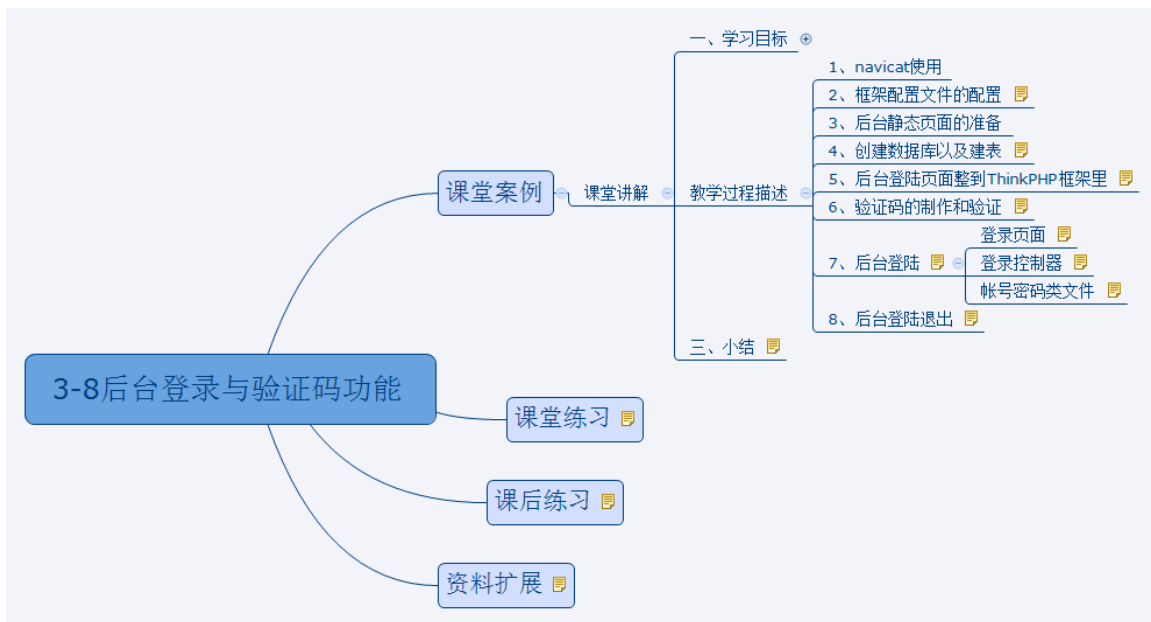


3-8后台登录与验证码功能

3-8后台登录与验证码功能	1
1. 课堂案例	2
课堂讲解	2
一、学习目标	2
掌握配置文件的配置	2
熟练掌握建库、建表	2
制作后台登陆	2
制作验证码	2
教学过程描述	2
1、navicat使用	2
2、框架配置文件的配置	2
3、后台静态页面的准备	3
4、创建数据库以及建表	3
5、后台登陆页面整到ThinkPHP框架里	3
6、验证码的制作和验证	5
7、后台登陆	7
8、后台登陆退出	9
三、小结	9
2. 课堂练习	10
3. 课后练习	10
4. 资料扩展	10



1. 课堂案例

课堂讲解

一、学习目标

掌握配置文件的配置

熟练掌握建库、建表

制作后台登陆

制作验证码

教学过程描述

1、navicat使用

2、框架配置文件的配置

参考框架开发手册配置

```

<?php
return array(
    // '配置项'=>'配置值'
    'MODULE_ALLOW_LIST'    => array('Home','Admin'),    //
    允许访问的模块列表
    'DEFAULT_MODULE'       => 'Home',
    //连接数据库的设定
    'DB_TYPE'              => 'mysql',                // 数据库类型
    'DB_HOST'              => 'localhost',            // 服务器地址
    'DB_NAME'              => 'shop0428',            // 数据库名
    'DB_USER'              => 'root',                // 用户名
    'DB_PWD'               => '',                    // 密码
    'DB_PORT'              => '3306',                // 端口
    'DB_PREFIX'            => 'rl_',                // 数据库表前缀
);

```

也可以在核心文件中ThinkPHP--->conf---

>convention.php找到需要修改默认的配置复制到应用配置文件中

3、后台静态页面的准备

4、创建数据库以及建表

字段	索引	外键	触发器	选项	注释	SQL 预览
名					类型	长度 小数点 不是 null
id					int	20 0 <input checked="" type="checkbox"/> 1
username					varchar	32 0 <input checked="" type="checkbox"/>
pass					varchar	100 0 <input checked="" type="checkbox"/>
sex					tinyint	1 0 <input type="checkbox"/>
phone					varchar	16 0 <input type="checkbox"/>
email					varchar	50 0 <input type="checkbox"/>
status					tinyint	1 0 <input type="checkbox"/>
remark					text	0 0 <input type="checkbox"/>
addtime					int	11 0 <input type="checkbox"/>
updatetime					int	11 0 <input type="checkbox"/>

5、后台登陆页面整到ThinkPHP框架里

1、后台静态页面的js、css等配置文件整入public/admin公共文件目录里

D:\wamp64\www\shop\Public\admin		
名称	修改日期	类型
lib	2017/5/5 14:30	文件夹
static	2017/5/5 14:30	文件夹
temp	2017/5/5 14:30	文件夹

然后搭建一个这样的控制器和视图

```

1 <?php
2 namespace Admin\Controller;
3 use Think\Controller;
4
5 class ManagerController extends Controller
6 {
7     function login()
8     {
9         $this->display();
10    }
11 }

```

再开启服务器，打开浏览器访问刚才建立的路径后，结果显示如下：

后台登录

localhost/shop/admin/manager/login

应用 360安全导航 百度一下 闻 今日头条 一秒提现

账户

密码

验证码: 看不清，换一张

☐ 使我保持登录状态

登录 取消

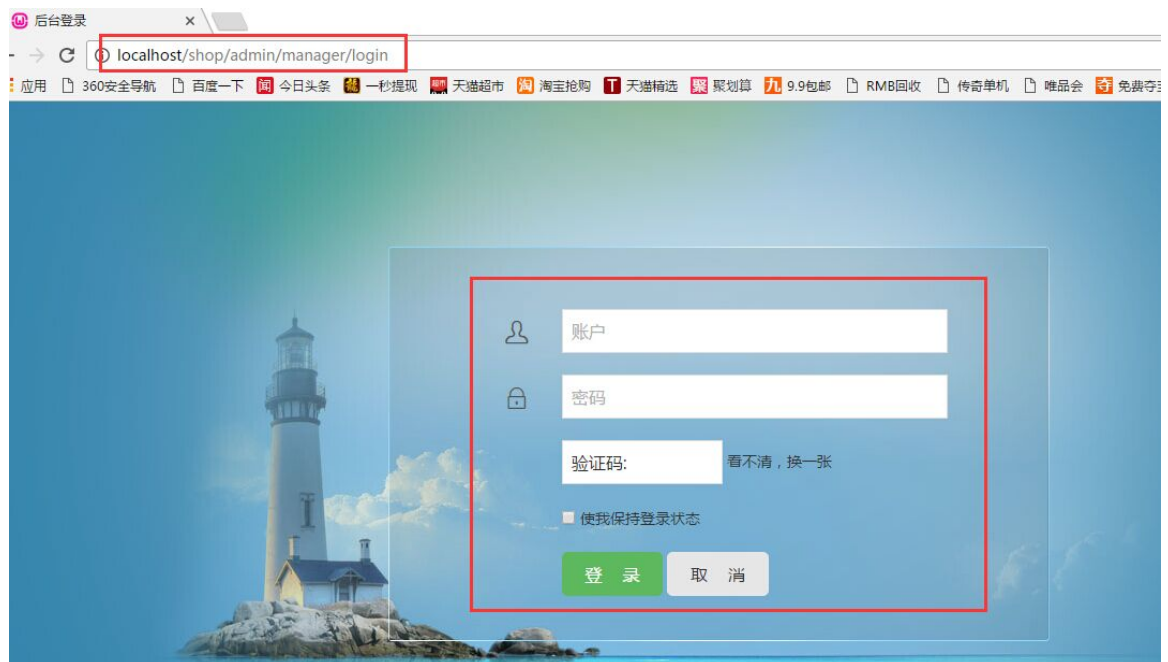
哎，说明访问路径是对的，但是css、js路径不对。

```
// 后台js、css路径
define('RL_URL', '/shop/Public/admin/');
```

我们在入口文件配置了一个常量路径,为什么要这样做呢,因为我们现在的项目是在本地根目录下shop文件夹里,等我们调试完毕后,这个项目放到线上根目录上,项目里面所有的配置路径等,只需要在入口文件改一下就可以了。不使用相对路径的一个原因也是担心linux、window可能会出现路径上的差距。使用绝对路径是不会出现这个问题的。

```
<link href="{Think.const.RL_URL}static/h-ui/css/H-ui.min.css" rel="stylesheet" type="text/css" />
<link href="{Think.const.RL_URL}static/h-ui.admin/css/H-ui.login.css" rel="stylesheet" type="text/css" />
<link href="{Think.const.RL_URL}static/h-ui.admin/css/style.css" rel="stylesheet" type="text/css" />
<link href="{Think.const.RL_URL}lib/Hui-iconfont/1.0.8/iconfont.css" rel="stylesheet" type="text/css" />
```

在静态页面上配置好刚定义好的常量路径。再刷新浏览器



css、js等都加载进来了。其他页面的搭建也是类似一个这样的过程。接下来我们需要处理的是业务逻辑问题了。

6、验证码的制作和验证

可以参考手册-->专题-->验证码制作

```
// 验证码图片生成
public function verify()
{
    $config = array(
        'imageH'      =>    45,      // 验证码背景高度
        'fontSize'    =>    20,      // 验证码字体大小
        'length'      =>    4,       // 验证码位数
        'useNoise'     =>    false,   // 关闭验证码杂点
        'codeSet'      =>    '0123456789', //设置数字验证码
    );
    $Verify = new \Think\Verify($config);
    $Verify->entry();
}
```

应用到前台是这样的代码：

```
<div class="row cl">
    <div class="formControls col-xs-8 col-xs-offset-3">
        <input class="input-text size-L" type="text"
            placeholder="验证码" id="captcha" style="width:150px;"
            >
        
    </div>
```



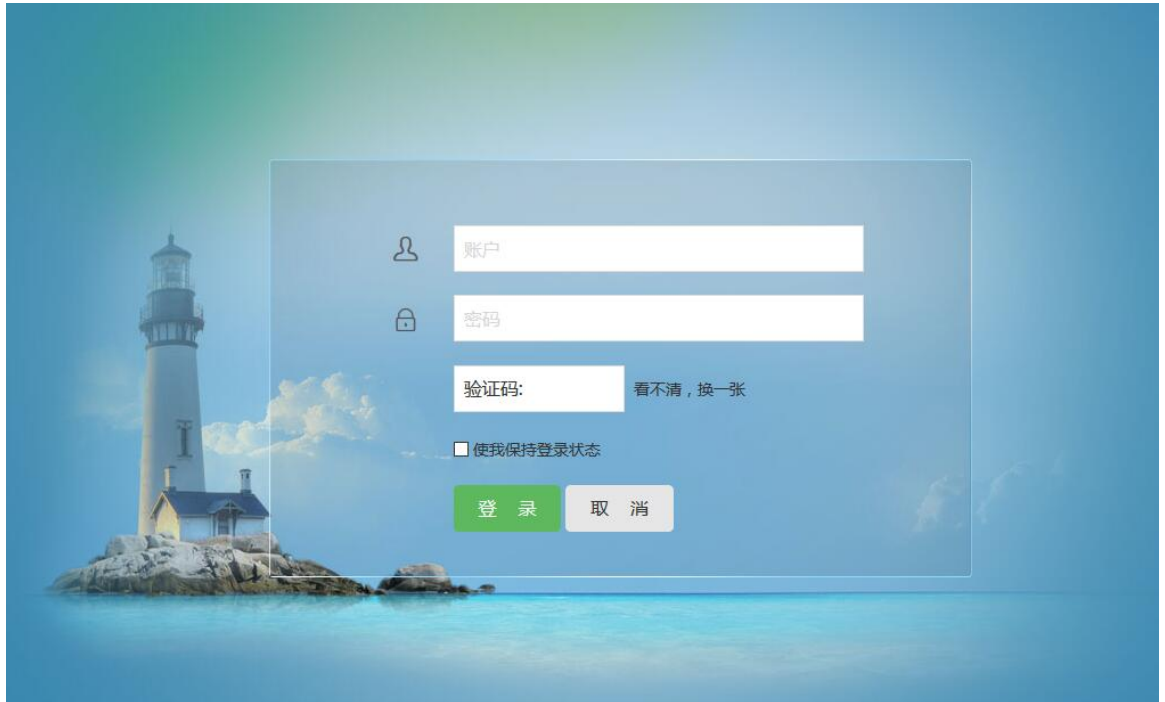
下面是验证验证码的代码

```
$verify = new \Think\Verify();  
$res = $verify->check(I('captcha'));  
// 测试得出匹配输出是1(true),不匹配没内容输出(false)  
//file_put_contents('cao.txt', $res);
```

7、后台登陆

前台ajax数据的提交到后端Manager控制器里, 后端把数据提交到Manager模型里去验证。后端经过处理把结果返回到前端。

登录页面



登录控制器

```

namespace Admin\Controller;
use Think\Controller;

class ManagerController extends Controller
{
    public function login()
    {
        if(IS_AJAX) {
            $verify = new \Think\Verify();
            $res = $verify->check(I('captcha'));
            // 测试得出匹配输出是1(true),不匹配没内容输出(false)
            //file_put_contents('cao.txt', $res);
            if($res) {
                // 当验证码输入正确的时候有来过这里
                //file_put_contents('cao.txt',123456789);

                $manager = new \Model\ManagerModel;
                $result = $manager->checkNamePwd(I('username'),I('pwd'));
                // 测试访问manager类后返回内容
                // file_put_contents('cao.txt',$result);
                if($result) {
                    session('admin_id',$result['id']);
                    session('admin_name',$result['username']);
                    // 验证成功的跳转
                    $this->ajaxReturn(2);
                } else {
                    // 账户或者密码错误
                    $this->ajaxReturn(1);
                }
            } else {
                // 验证码不匹配
                $this->ajaxReturn(0);
            }
        }
        $this->display();
    }

    // 验证码图片生成
    public function verify()
    {
        $config = array(
            'imageH' => 45, // 验证码背景高度
            'fontSize' => 20, // 验证码字体大小
            'length' => 4, // 验证码位数
            'useNoise' => false, // 关闭验证码杂点
            'codeSet' => '0123456789', //设置数字验证码
        );
        $Verify = new \Think\Verify($config);
        $Verify->entry();
    }
}

```

帐号密码类文件


```

<?php
namespace Model;
use Think\Model;

class ManagerModel extends Model
{
    function checkNamePwd($un,$pwd)
    {
        $info = $this->where(array('username'=>$un))->find();
        if($info) {
            if($info['rules']==0) {
                if($info['pass'] == $pwd) {
                    return $info;
                } else {
                    return null;
                }
            } else {
                return null;
            }
        } else {
            return null;
        }
    }
}

```

8、后台登陆退出

```

<li><a href="{:U('Manager/logout')}">退出</a></li>

```

```

public function logout()
{
    session('admin_name',null);
    session('admin_id',null);
    $this->redirect('login');
}

```

三、小结

- 1、框架config配置访问模块、连接数据库
- 2、建数据库和管理员登陆表格
- 3、验证码的制作以及验证
- 4、后台登陆制作
- 5、后台登陆后退出制作

2. 课堂练习

按照课堂讲解，配置框架config文件

按照课堂讲解，制作验证码

按照课堂讲解，制作后台登陆和登陆后退出

3. 课后练习

带着思考，把验证码制作和登陆代码敲三遍

4. 资料扩展

序号	列名	字段名	属性	非空	主键	默认	说明
1	用户id	id	int	not null			主键、自增
2	账号	username	varchar(32)	not null			唯一性约束
4	密码	pass	char(32)	not null			
5	性别	sex	tinyint(1)			0	0:表示男 1: 表示女
8	电话	phone	varchar(16)				
9	Email	email	varchar(50)	not null			
10	角色	role	varchar(255)			0	0:超级管理员 1:普通管理员
11	权限	rules	tinyint(1)			0	0:启用 1: 禁用
12	注册时间	addtime	int				

序号	列名	字段名	属性	非空	主键	默认	说明
1	分类id	id	int				主键、自增
2	分类名称	name	varchar(32)				
3	父类id	pid	int			0	
4	分类路径	path	varchar(255)				

序号	列名	字段名	属性	非空	主键	默认	说明
1	商品id	id	int				主键、自增
2	类别id	typeid	int				typeid表的外键
3	商品名称	goods	varchar(32)				
4	生产厂家	company	varchar(50)				
5	简介	descr	text				
6	单价	price	double(6,2)				
7	图片名	picname	varchar(255)				
8	状态	state	tinyint(1)			1	1: 新添加、2: 在售、3: 下架
9	库存量	store	int			0	
10	被购买数量	num	int			0	
11	点击次数	clicknum	int			0	
12	添加时间	addtime	int				

序号	列名	字段名	属性	非空	主键	默认	说明
1	id	id	int				主键、自增
2	会员id	uid	int				
3	联系人	linkman	varchar(32)				
4	地址	address	varchar(255)				
5	邮编	code	char(6)				
6	电话	phone	varchar(16)				
7	购买时间	addtime	int				
8	总金额	total	double(8,2)				
9	状态	status	tinyint				0:新订单；1: 已发货；2: 已收货，3 无效订单。

序号	列名	字段名	属性	非空	主键	默认	说明
1	id	id	int				主键、自增
2	订单id	orderid	int				(订单号的外键)
3	商品id	goodsid	int				
4	商品名称	name	varchar(32)				
5	单价	price	double(6,2)				
6	数量	num	int				