

# 互联网软件开发技术与实践

李梦好 1801210847 第四次作业

## 1、 使用 HTTPS 协议，用户登录功能，在提交时，分析数据包前端加密和后端加密的优缺点。

HTTPS(Secure Hypertext Transfer Protocol 安全超文本传输协议)：在 http 协议基础上增加了使用 SSL（安全套接字层）加密传送信息的协议。所以理论上来说，前端与后端之间的通信是被 HTTPS 加密过的，这个过程密码泄露风险不在讨论范围之内。

数据包前端加密：

也就是说密码在前端传送之前就处理成密文，在发送过程中即是以密文的方式，当然传送过程中本来就是被 HTTPS 加密处理过的，但是它的优点在于，数据在到达后台的时刻起就是被加密的，即使发生后台服务器被恶意使用、控制，黑客们获取到的也只是加密后的数据。如今很多用户会在不同系统上设置相同的密码，在这种情况下，一旦明文密码被获取，则用户面临着多个账号失窃的危险，这一点是可以被数据包前端加密所避免的风险。前端加密的另一个缺点是前端进行加密会造成页面需要 js 脚本才能运行，那么假设你的系统需要兼容不能运行 js 的客户端，就必须再设计一个使用原文的登录接口。

数据包后端加密：

这种方式的支持者们不赞成前端加密的必要性。由于前端加密所使用的 javascript 语言是脚本语言，在浏览器上所有人都是可以看到明文代码的。故而在前端加密的意义并不大。且，即使密码在前端进行加密了，后台服务器被恶意控制的情况下，黑客还是可以利用加密后的数据跟数据库中的数据（也是被加密的）直接进行匹配，这样，即使他们不知道原始密码也仍然可以登录进系统进行攻击。<sup>[1]</sup>

## 2、 使用你熟悉的开发框架（例如 ThinkPHP），实现一个用户注册和登录的功能。

### 2.1 ThinkPHP 框架

之前并没有用过 ThinkPHP 框架，但是据了解，ThinkPHP 是为了简化企业级应用开发和敏捷 WEB 应用开发而诞生的。最早诞生于 2006 年初，2007 年元旦正式更名为 ThinkPHP，并且遵循 Apache2 开源协议发布。ThinkPHP 从诞生以来一直秉承简洁实用的设计原则，在保持出色的性能和至简的代码的同时，也注重易用性。并且拥有众多原创功能和特性，在社区团队的积极参与下，在易用性、扩展性和性能方面不断优化和改进。

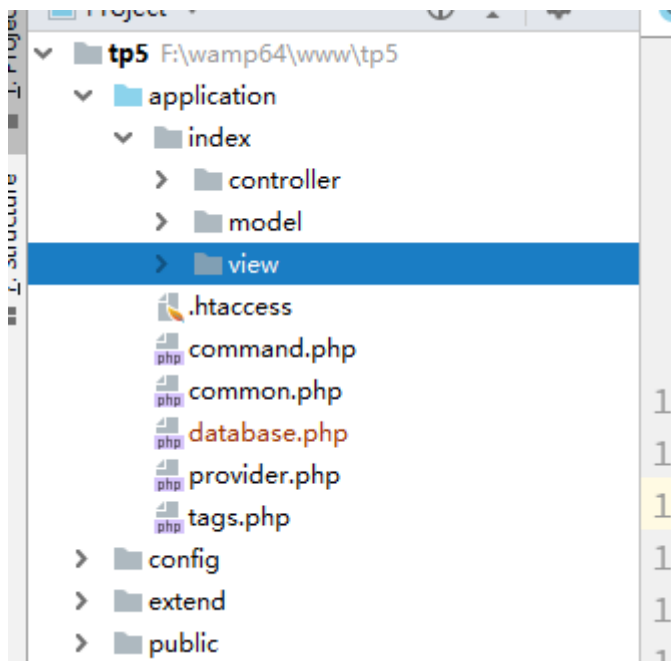
#### 2.1.1 环境搭建

于是开始 ThinkPHP 框架的学习。依照网上的教程进行环境搭建，这里我下载安装了 WAMPServer 的集成开发环境，从 github 上安装了 5.0 版本的 ThinkPHP，在其模板的基础上实现登录与注册的功能。



#### 2.1.2 ThinkPHP 中的 MVC

ThinkPHP 应用基于 MVC (Model-View-Controller)，即模型-视图-控制器架构。下载下来的 ThinkPHP 项目原型目录为：



application 是该项目的一个应用，其目录下即是该应用的各个模块，默认模块 index，index 下包含 controller,model,view 文件夹，分别装对应的 php 文件。controller 下 index.php 文件即是默认的项目入口。我们修改其 index 方法，并把上次做好的登录页面即 index.html 文件放在 view 文件夹下的 index 目录下。

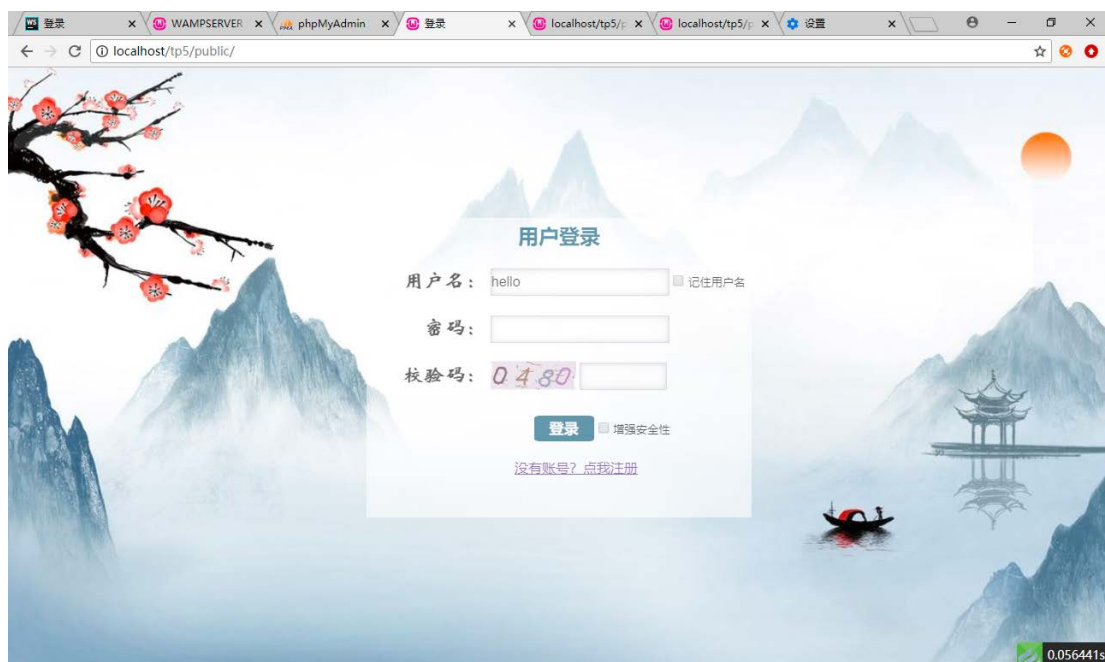
```
class Index extends Controller
{
    public function index(){
        return $this->fetch( template: 'index' );
    }
}
```

此时我们访问 url:http://localhost/tp5/public/即可看到自定义的登陆页面了。

注：

1. 访问一个方法的路径：localhost/项目名/入口路径/文件名/模块/控制器/方法名。例如我们在控制器 Index 的 index.php 中定义另一个 Hello 方法，其访问地址就会是 localhost/tp5/public/index.php/index/index/hello。tp5 是项目名称，ThinkPHP5.0 的入口路径是 public。
2. ThinkPHP 会对路径的大小写做相应的转化，默认情况下，URL 是不区分大小写的，也就是说 URL 里面的模块/控制器/操作名会自动转换为小写，控制器

在最后调用的时候会转换为驼峰法处理。



3. css/js/img 等静态文件需要放在主目录下的 static 目录下。在引用到这些文件的文件中写相应的绝对路径地址。如下图，页面中的静态 css 文件和图片都没有载入，页面报错找不到相应的资源。查看发现用的相对路径“/static/...”经计算后与实际并不相符。需换为“/tp5/public/...”开头的绝对路径。

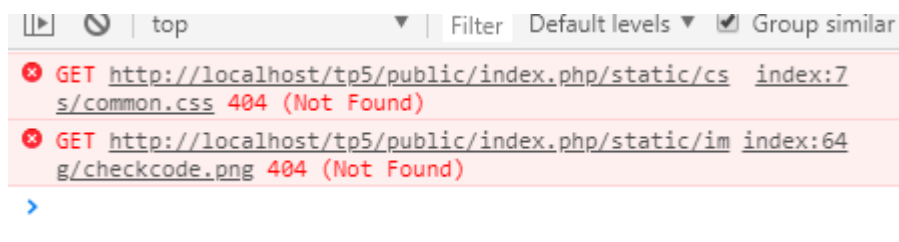
**用户登录**

用户名:  
  ☐ 记住用户名

密码:

校验码:  


☐ 增强安全性



```

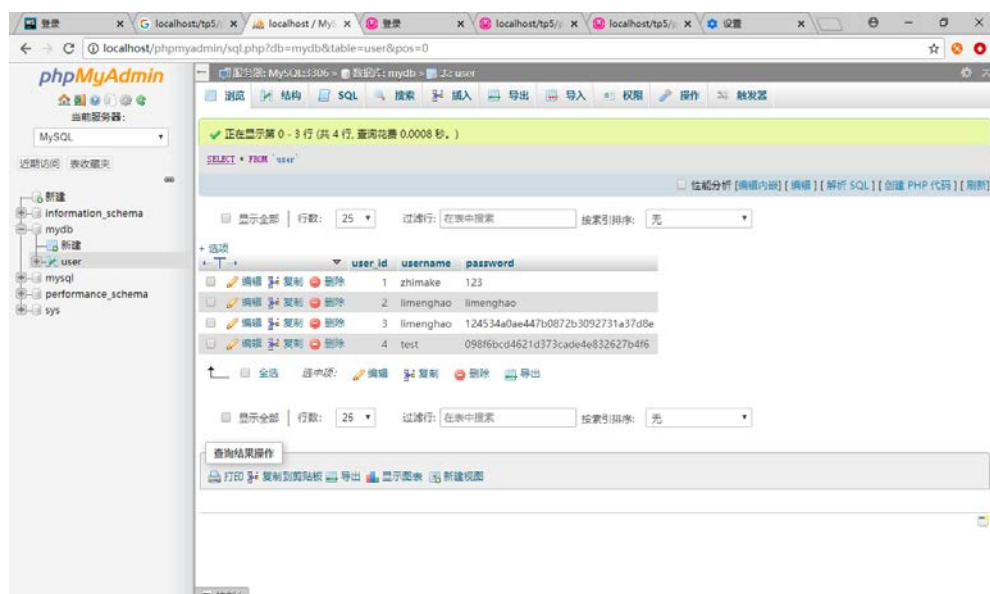
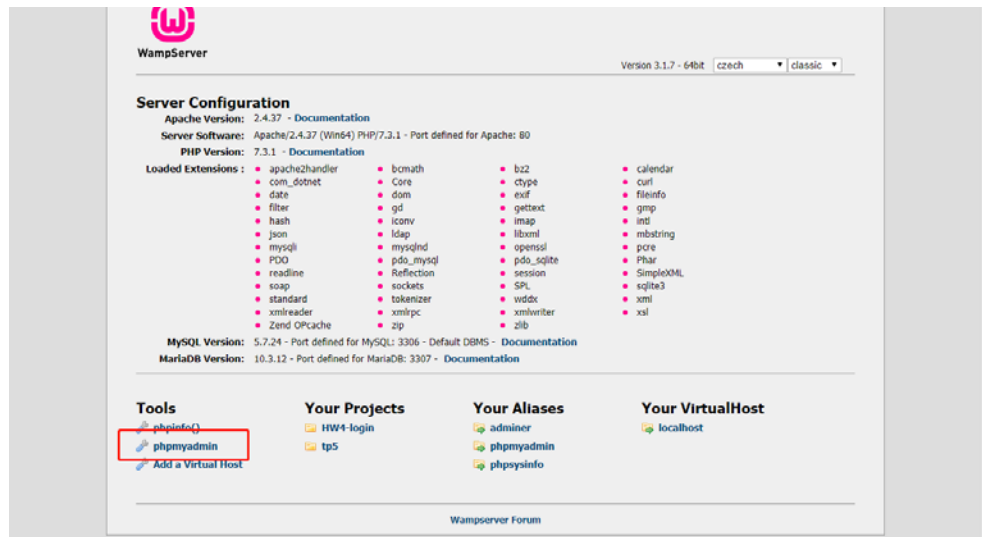
</title>豆浆</title>
<meta charset="UTF-8">
<link href="css/common.css" rel="stylesheet">
</head>
<title>豆浆</title>
<meta charset="UTF-8">
<link href="/tp5/public/static/css/common.css" rel="stylesheet">
</head>
<script src="https://cdn.staticfile.org/jquery/1.10.2/jquery.min.js"

```

## 2.2 数据库操作

### 2.2.1 phpmyadmin

安装好 WampServer 之后，在浏览器目录中选择 phpmyadmin 工具就可以直接进入数据库管理页面，而且是图形化用户界面版的，可以方便进行数据表的增删改查等操作。



## 2.2.2 ThinkPHP 中的数据库访问方式

1. 直接在方法中连接数据库进行数据库操作。

```
class Index extends Controller
{
    public function index()
    {
        Db::connect('mysql://root:@127.0.0.1:3306/mydb#utf8');
        $result = Db::query('select * from user where user_id=?',[1]);
        print_r($result);
        //echo $result[username];
    }
}
```

Array ( [0] => Array ( [user\_id] => 1 [username] => zhimake [password] => 123 ) )

2. 建立 model 后直接用模型实例进行增删改查。

对应数据库中的 User 表，建立一个 User 的 model:

```
/** Created by PhpStorm. ... */

namespace app\index\model;

use think\Model;

class User extends Model
{
    protected $table = 'user';
    protected $pk = 'user_id';
}
```

关联对应的表名和主键。则我们在控制器中可以通过这个 model 来操作数据实例:

```
class Index extends Controller
{
    public function index()
    {
        $user = User::get( data: 1 ); // 获取主键为1的实例user
        print_r($user);
    }
}
```

app\index\model\User Object ( [data] => Array ( [user\_id] => 1 [username] => zhimake [password] => 123 ) [relation] => Array ( ) )

注：

a) 如何设置数据库主键自增：

在利用模型 Model 来往数据库表中添加数据时，我们往往不会为该条数据的主键赋值（因为我们并不知道哪些主键已经被用过了），一个好的解决办法是直接设置数据库表的主键自增。



## 2.3 登录功能实现

直接在 index 模块的 index 控制器中新增 logIn 方法，登录表单以 post 方式提交到这个方法。



方法中根据用户名查询数据库表 user 中有无此用户名，不存在返回错误“用户名不存在”，存在则进一步判断其密码经 md5 加密后的结果与数据库中的密码加密值是否匹配，匹配则登录成功，并跳转到主页面去。其他错误情况下页面会报错，并返回到上层页面即登录页面去。



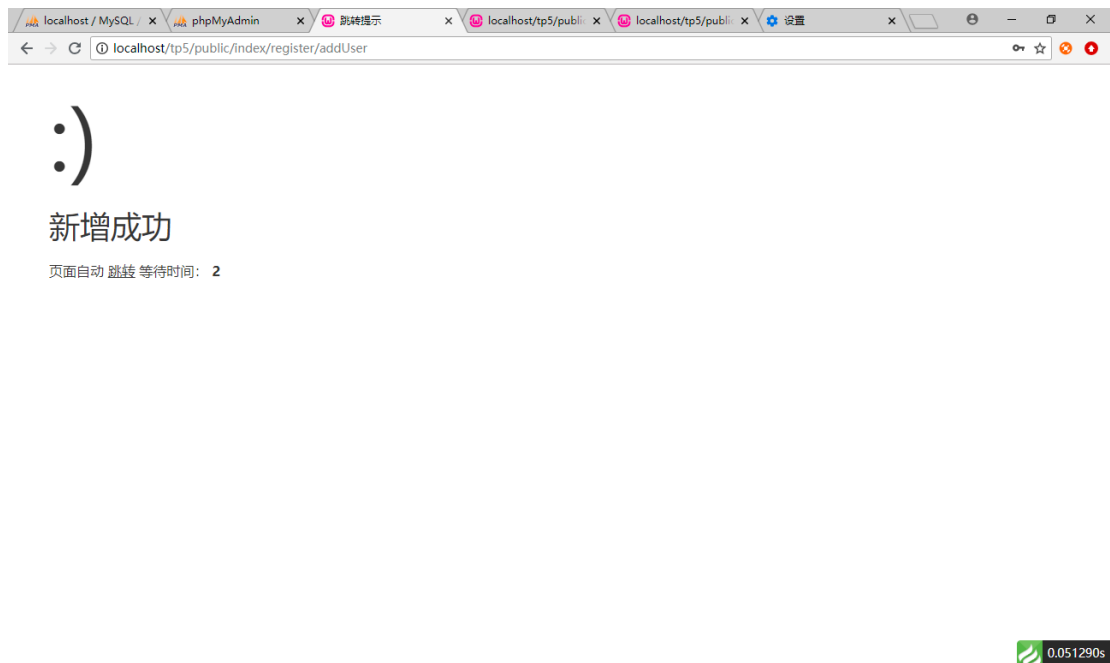
## 2.4 注册功能实现

新建控制器 Register，写方法 addUser，同样的此方法会被注册表调用，提交后，方法中先判断用户名是否存在，然后将密码经过 md5 加密后储存到数据库表中。注册成功后跳转到登录页面。

```
<div class="container">
  <div id="modal_dialog">
    <form action="/tp5/public/index/register/addUser"
      enctype="multipart/form-data" method="post">
      <div class="title">用户注册</div>
      <div class="line">
        <div class="myTit"><label>用户名: </label></div>
        <input type="text" name="username" style="width:
      </div>
      <div class="line">
        <div class="myTit"><label>密码: </label></div>
        <input type="password" name="password" style="wi
```

```
class Register extends Controller
{
  public function index()
  {
    return $this->fetch( template: 'register' );
  }
  public function addUser($username,$password){
    // 首先判断用户名是否存在
    $user = User::get(['username' => $username]);
    if(!empty($user)){
      $this->error( msg: '新增失败，用户名已存在' );
    }
    $newUser = new User;
    $newUser->username = $username;
    $newUser->password = md5($password);
    $newUser->save();
    //echo $newUser->user_id;
    if($newUser){
      // 设置成功后跳转页面的地址，默认返回页面是$_SERVER['HTTP_
      $this->success( msg: '新增成功', url: 'index/index' );
    } 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( msg: '新增失败' );
    }
  }
}
```





注册成功后在 phpmyadmin 查看 user 表中数据，发现的确新插入了一个用户 admin，且其密码是加密过的 md5 长字符串。

+ 选项				user_id	username	password
<input type="checkbox"/>	编辑	复制	删除	1	zhimake	123
<input type="checkbox"/>	编辑	复制	删除	2	limenghao	limenghao
<input type="checkbox"/>	编辑	复制	删除	3	limenghao	124534a0ae447b0872b3092731a37d8e
<input type="checkbox"/>	编辑	复制	删除	4	test	098f6bcd4621d373cade4e832627b4f6
<input type="checkbox"/>	编辑	复制	删除	5	admin	21232f297a57a5a743894a0e4a801fc3

【1】 [https://blog.csdn.net/xl\\_lx/article/details/79075386](https://blog.csdn.net/xl_lx/article/details/79075386)

【2】 <https://baike.baidu.com/item/thinkphp/3521908?fr=aladdin>