# 基于thinkphp框架的简单应用实现

傅广播

目录

[基于thinkphp框架的简单应用实现 1](#_Toc6075448)

[项目简介 3](#_Toc6075449)

[环境搭建 3](#_Toc6075450)

[购买腾讯云服务器 4](#_Toc6075451)

[搭建web开发环境 5](#_Toc6075452)

[PhpStorm连接远程项目 8](#_Toc6075453)

[数据库创建 16](#_Toc6075454)

[登录注册的简单实现 19](#_Toc6075455)

[编写公用模块 28](#_Toc6075456)

[编写管理界面功能 32](#_Toc6075457)

## 项目简介

本项目部署在华为云服务器上，访问入口地址如下，用户名密码都为admin

<http://114.116.12.238/test/public/index.php/login/index/index.html>

主要使用lnmp一键脚本搭建web开发环境（linux+nginx+mysql+php），thinkphp框架实现内容。

使用的工具：

Xshell6，Xftp：ssh方式连接远程服务器，上传文件等（<https://xshell.en.softonic.com/>）

PhpStorm:远程连接服务器代码，本地开发（<https://www.jetbrains.com/phpstorm/>）

Lnmp脚本：一键搭建开发环境（<https://lnmp.org/>）

其他：如果对lnmp脚本不熟悉的可以使用宝塔面板（<http://www.bt.cn/>），官网有详细文档

软件版本：

Mysql：5.5

PHP：5.6

Thinkphp框架：5.0.24

Bootstrap:4.0.0

JQuery:3.4.0

## 环境搭建

**注：在本地开发的可跳过这一步**

### 购买腾讯云服务器

登录腾讯云网站（<https://cloud.tencent.com/>），使用代金券以按量计费方式购买服务器，操作系统选用centos，内存选1核1GB即可，（成都上海等地的服务器收费标准默认是0.14元/小时，流量0.8元/GB，北京的较贵0.25/小时）

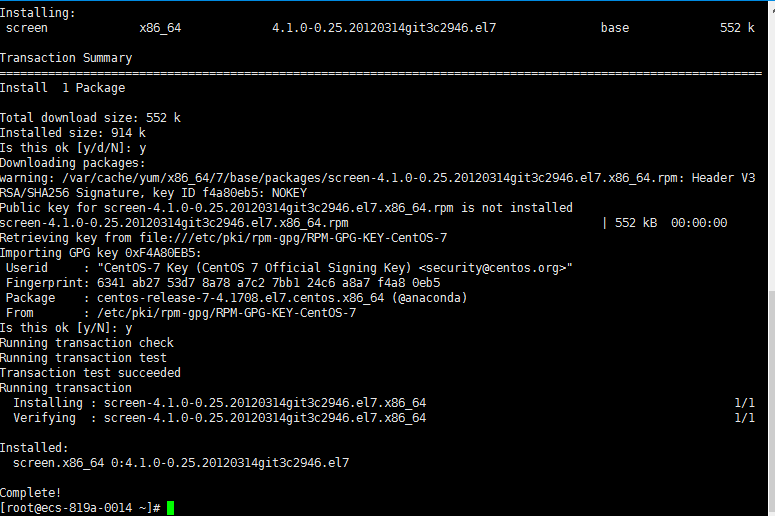


安全组使用新建安全组即可。



### 搭建web开发环境

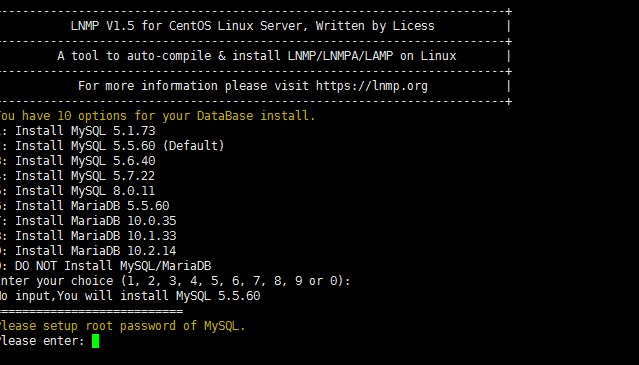
1. 用XShell连接上服务器，执行yum install screen 安装screen



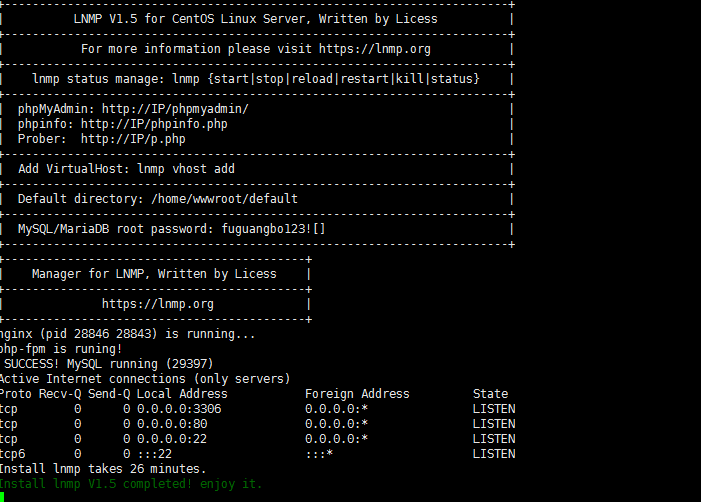
1. 安装完成后运行screen -S lnmp打开新窗口后执行

wget http://soft.vpser.net/lnmp/lnmp1.5.tar.gz -cO lnmp1.5.tar.gz && tar zxf lnmp1.5.tar.gz && cd lnmp1.5 && ./install.sh lnmp

依次选择安装数据库版本，设置数据库密码，设置存储引擎，PHP版本等，这里我全都选择默认选项（MYSQL5.5.60，PH,5.6.36）进行安装



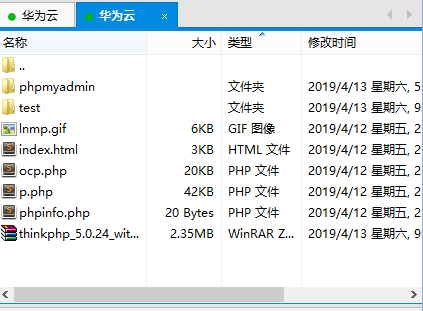
大约需要半小时以上，安装完毕



1. 访问服务器ip地址，显示如下



1. 安装完后默认访问home/wwwroot/default目录下的网页文件，可以看到同目录下有phpmyadmin文件夹，访问ip地址/phpmyadmin即可进入图形化数据库管理界面



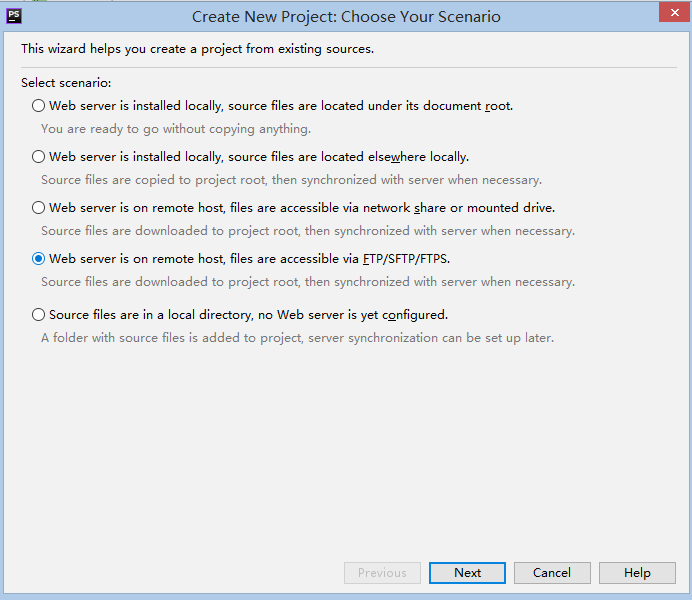


1. 下载thinkphp框架的压缩文件，使用xftp上传至default目录下并解压至test文件夹，访问相应路径，显示如下，表明框架安装成功

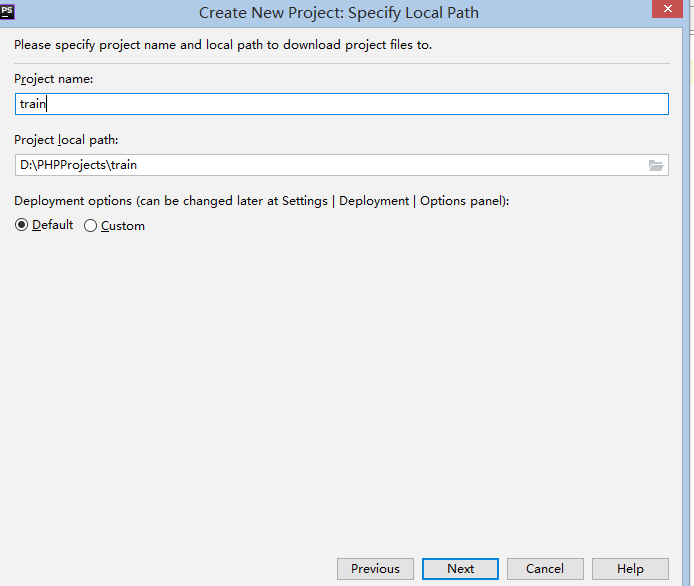


### PhpStorm连接远程项目

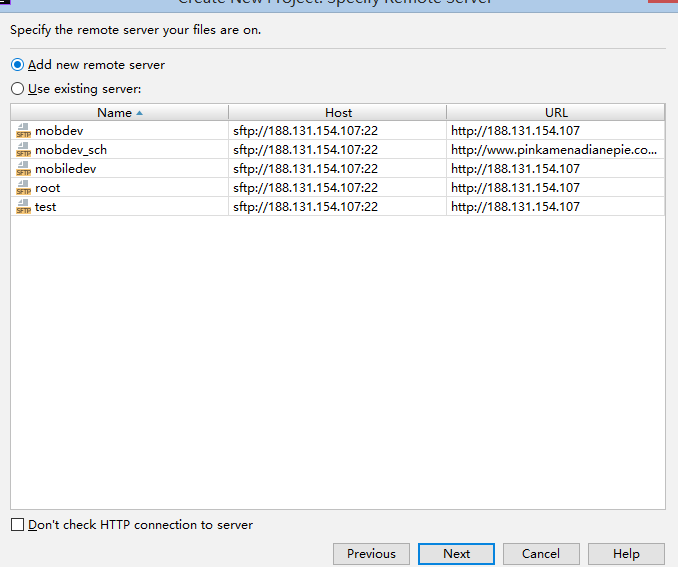
1. 点击菜单里的File->New Project From Existing Files，选择连接远程服务器上的项目



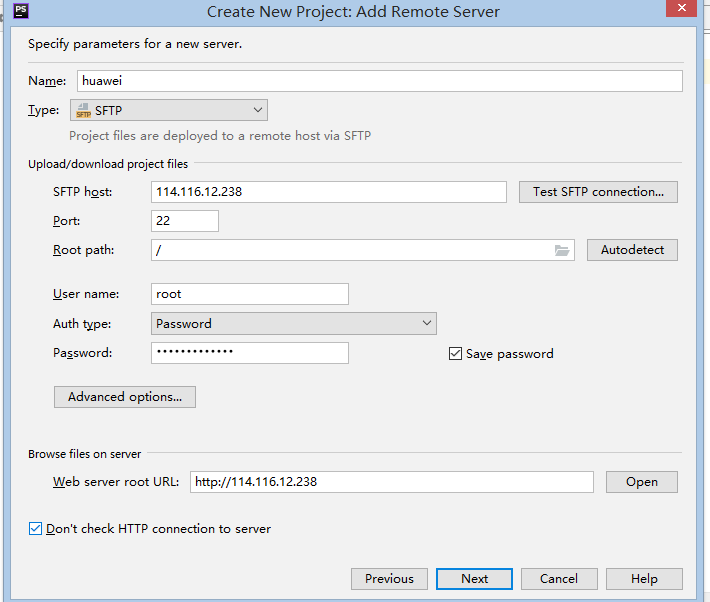
1. 选择本地项目名及本地保存路径

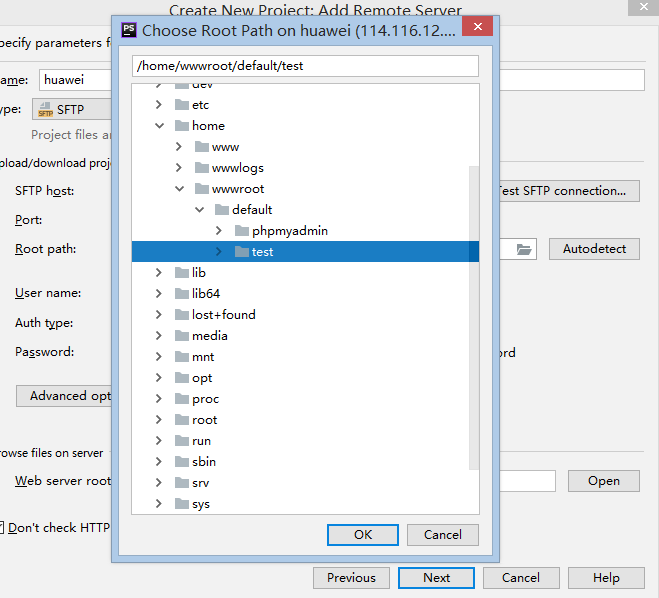


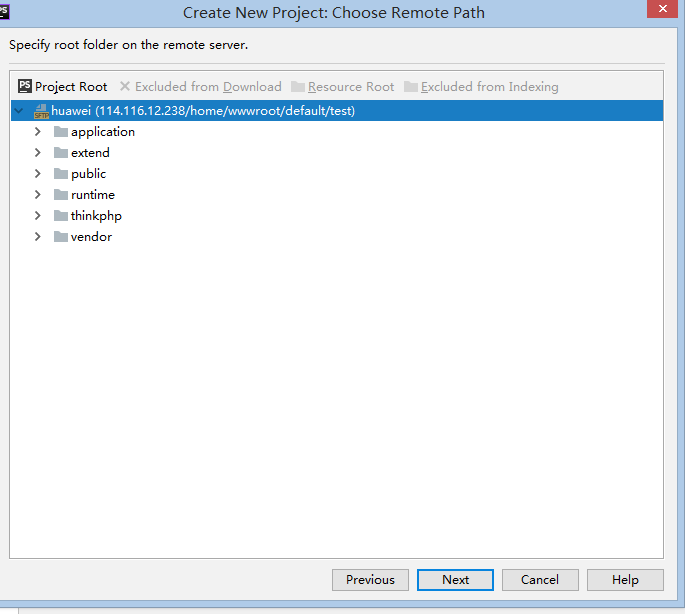
1. 选择新增一个服务器



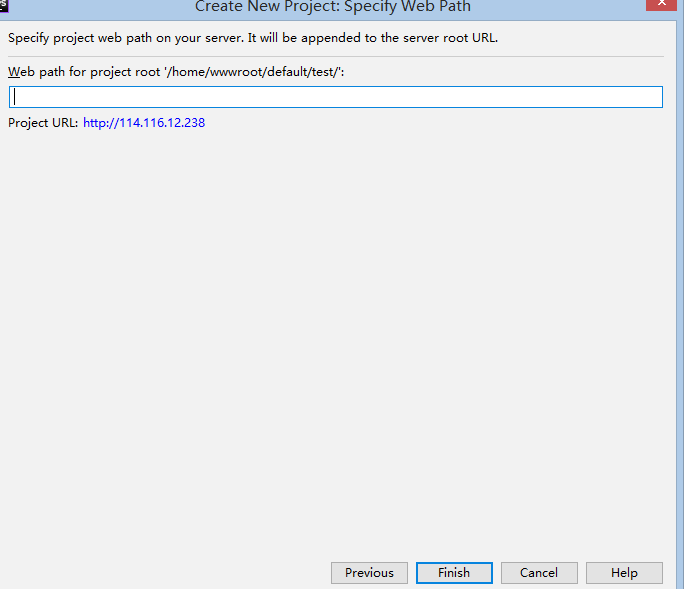
1. 填写服务器信息，并在根目录（Root Path）处选择服务器上的项目目录，这里就是home/wwwroot/default/test



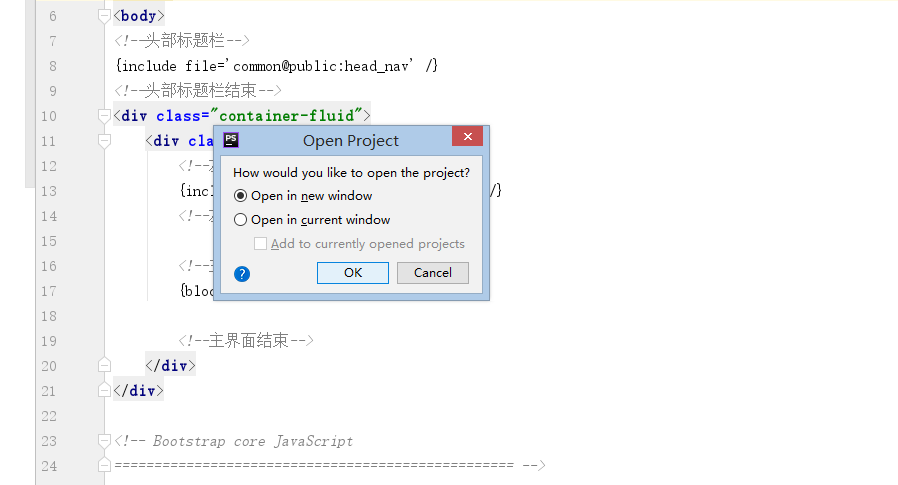




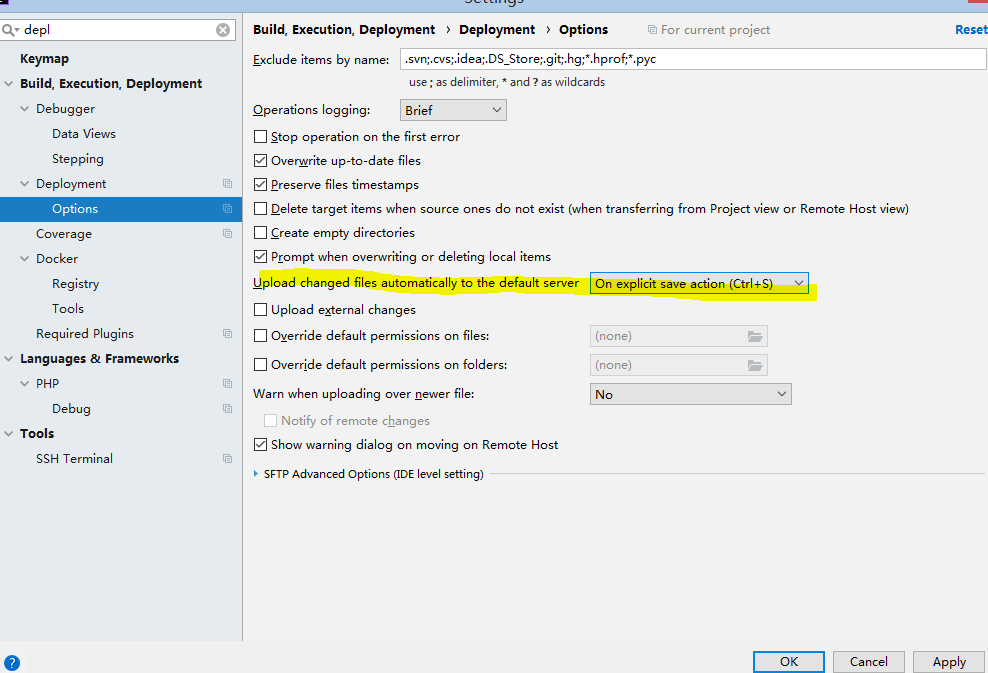
1. 这里填写项目网站的访问路径（不填也可以,影响不大）



1. 点击完成后会自动开始下载项目到本地



1. 最后在菜单栏File->Settings处打开设置，搜索deploy，修改如下设置，这样每次保存后修改的文件会自动上传到服务器（也可以选always，这样就会一直保持同步）



## 数据库创建

1. 访问ip地址/phpmyadmin并登录进入，新建一个名为test的数据库



**注：之后的表可用直接使用上方菜单栏的导入功能将sql文件直接导入进数据库，不用建表**

1. 新建menu表作为侧边栏菜单的数据表





1. 新建title表作为存储网站标题栏文字的数据



1. 新建user表作为网站用户的数据表



1. 修改thnkphp框架内的配置文件

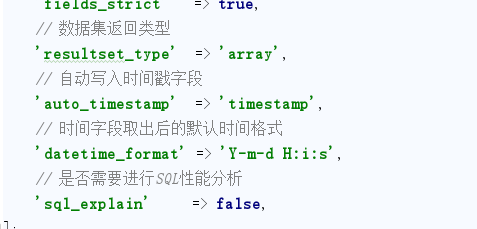
修改application/config.php，应用调试模式方便查错



修改application/database.php，填写数据库用户名，密码和端口（3306）

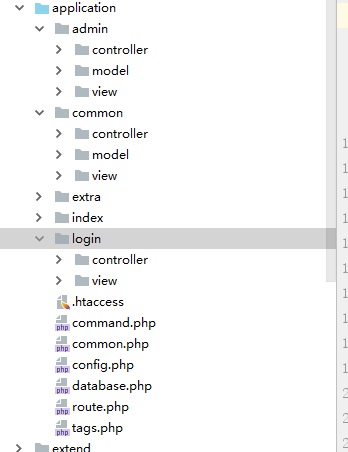


自动写入时间戳,修改默认时间格式

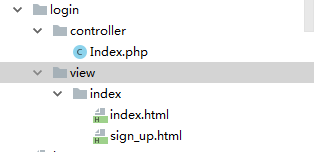


## 登录注册的简单实现

1. 在application目录下新建三个模块：login登录模块，admin管理界面模块，common公用模块



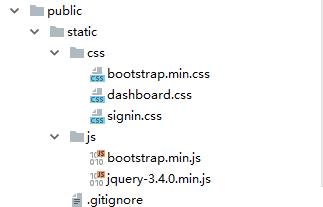
1. 在login模块下新增如下文件,index.html作为登录界面，sign\_up.html作为注册界面，Index.php作为后台处理的控制器



1. 这里为了方便直接使用了bootstrap的网页模板()，在public/static目录下放置相应的样式文件和js文件

例子下载地址：<https://getbootstrap.com/docs/4.3/examples/>

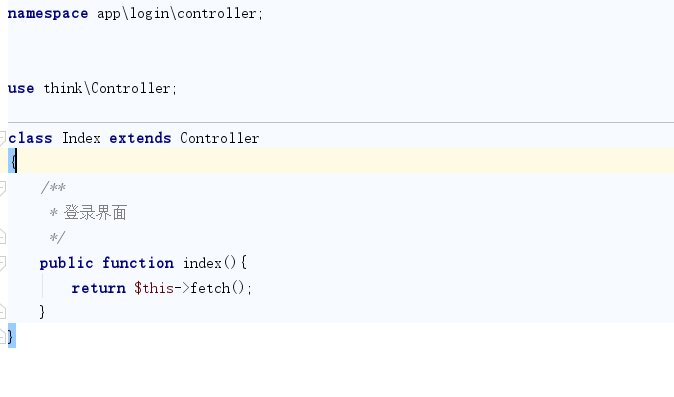
解压后在site\docs\4.3\examples目录下的sign-in和dashboard例子



1. Index.html大致代码如下，这里的引用路径\_\_STATIC\_\_在thinkphp框架中默认指向public/static文件夹

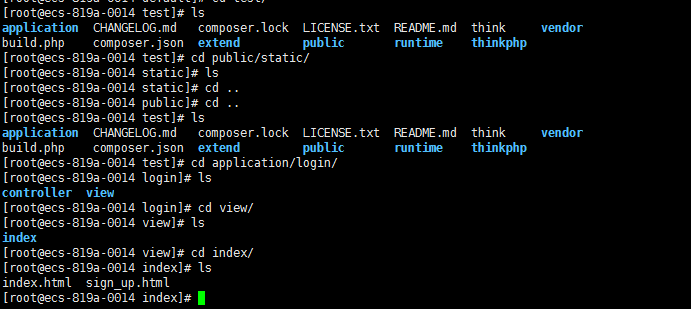


Index.php代码如下

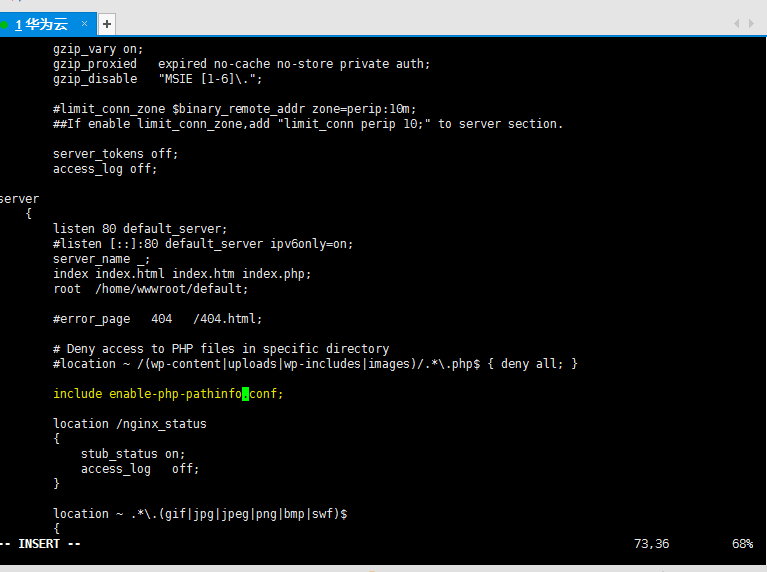


1. 尝试访问该路径，出现404错误

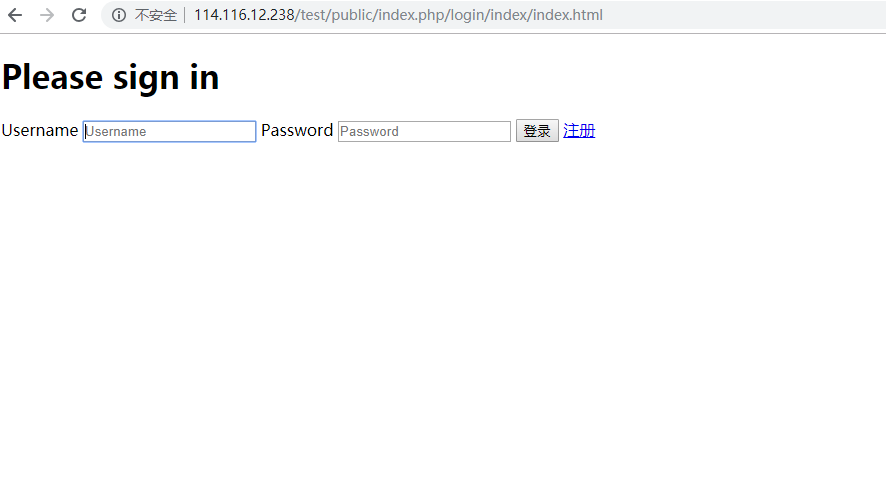
在XShell上查看服务器文件是否存在,发现模板文件都已经上传

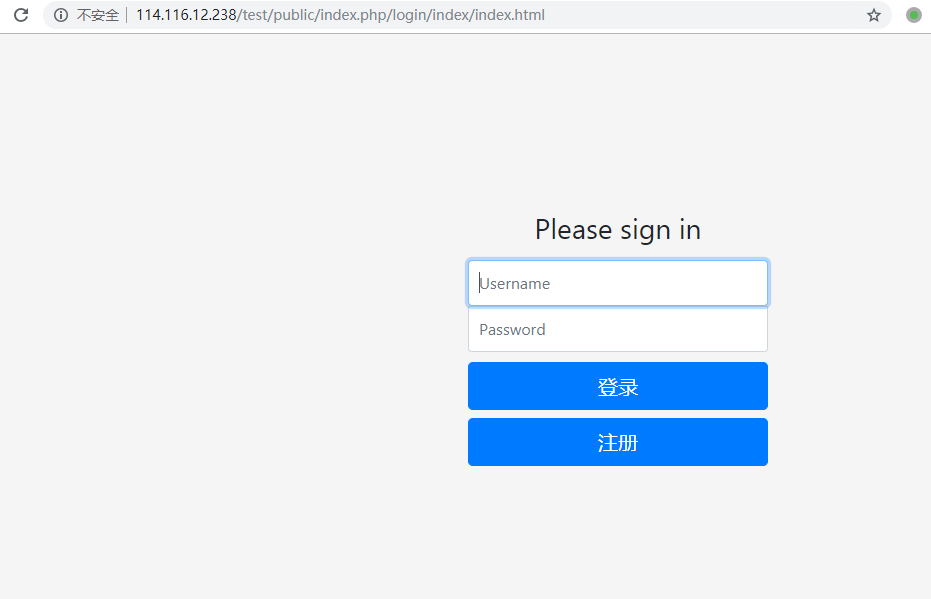


发现是nginx默认不支持pathinfo，在usr/local/nginx/conf目录下修改nginx.conf配置文件（vim nginx.conf），修改成标黄处即可

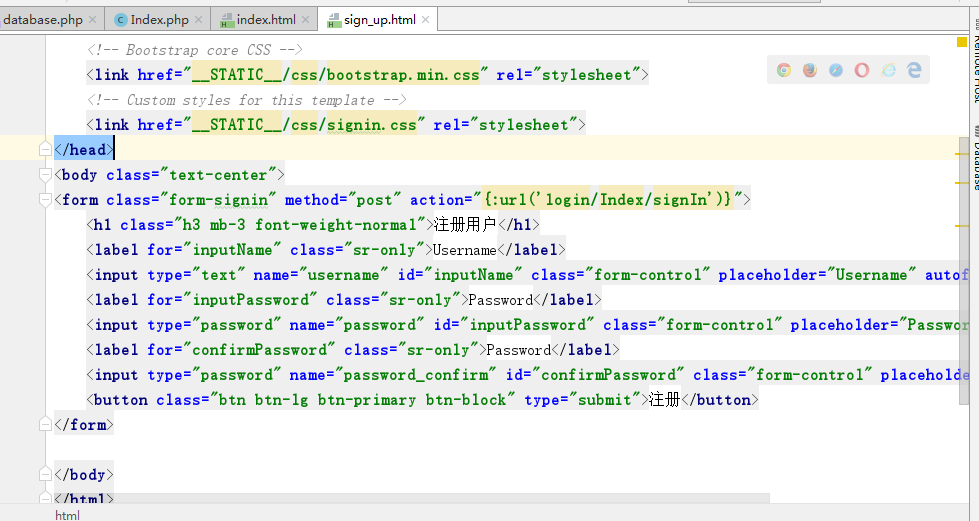


之后保存退出编辑并重启nginx服务器（service nginx restart），重新访问该路径出现页面，但没有样式，发现static文件夹下的文件没上传到服务器，上传完后出现完整页面（选中文件夹右键，Deployment->upload to …）





1. 同理修改sign\_up.html，编写注册界面



1. 处理注册登录逻辑，由于之后管理界面展示内同时也要访问用户表，为了方便我在admin/model下新建了User模型类处理用户表相关的数据处理,第一行定义保护变量table = “user”，thinkphp框架会默认该模型类使用数据库内名为user的表。

添加findUser方法，查找是否有相同用户名密码的有效用户



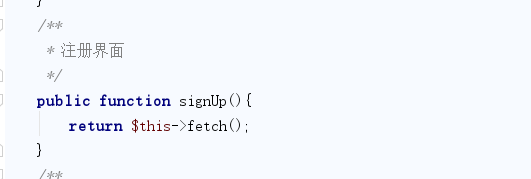
添加addNewUser方法，实现添加不重名的有效用户



1. login下的Index控制器，添加登录方法login，处理前端传来的用户名密码，若登录成功则在Session内写入相关信息



1. 添加注册界面的显示方法signUp(注：驼峰法写的方法调用this->fetch，框架默认会去寻找view/控制器名/下划线切分后的方法名对应的html模板文件，这里signUp方法对应的模板文件命名应为sign\_up)



1. 实现注册方法signIn

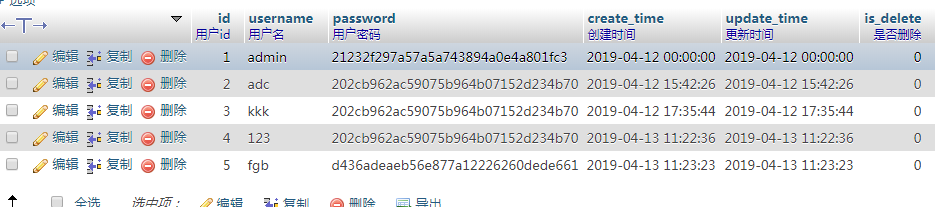


1. 尝试注册用户



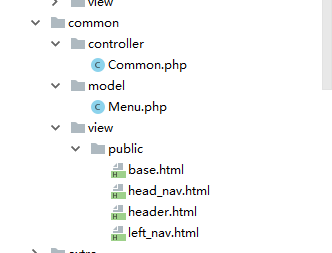


数据库中数据成功插入

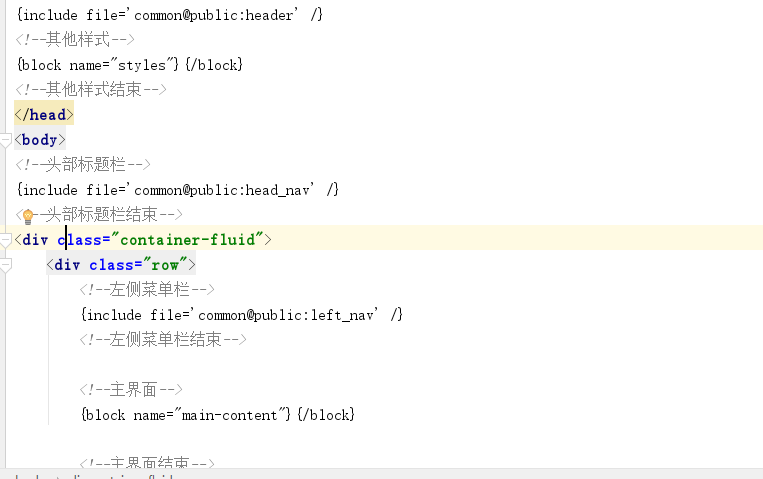


## 编写公用模块

1. 首先观察dashboard例子的主界面html，可以把一个界面分为如下几部分：
   1. 头部描述及公用的css样式引用
   2. 个别界面独有的样式引用
   3. 头部标题栏
   4. 左侧菜单栏
   5. 右侧对应的内容
   6. 公用的js引用
   7. 独有的js处理部分
2. 在common/view/public下新建如下html，base.html作为整体模板，head\_nav作为标题栏模板，header作为公用头部引用，left\_nav作为左侧菜单栏模板

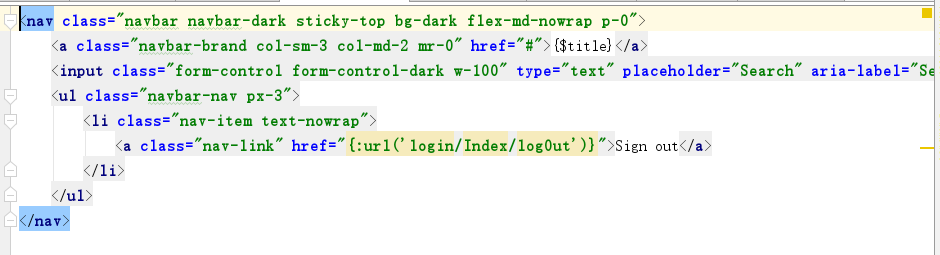


Base.html，组合其他三个html文件，开发人员只要编写styles,main-content和scripts三个block的内容即可

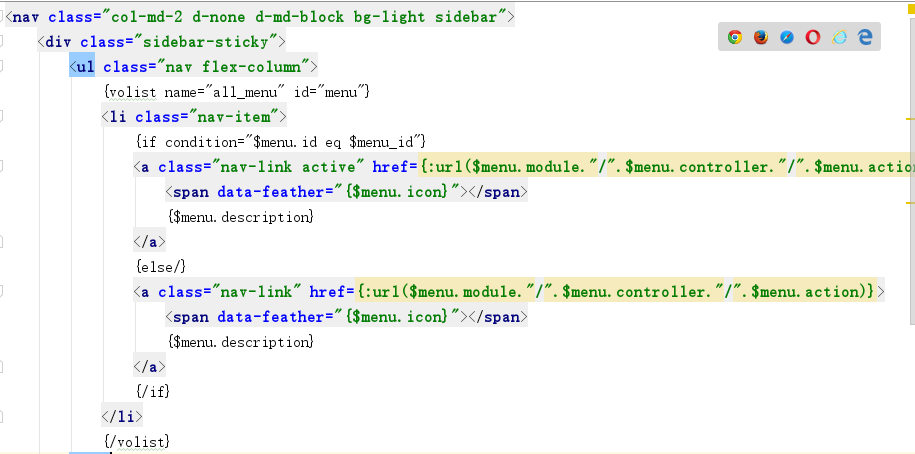




Head\_nav.html，放置标题栏标题及退出按钮



Left\_nav.html，放置统一的左侧菜单栏



1. 在model下新建Menu模型类，处理menu表，添加getAllByOrder方法，返回按number排序的所有数据



添加getIdByMvc方法，根据传入的模块/控制器/方法名返回相应的id



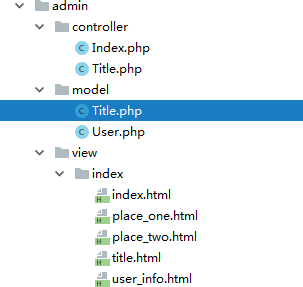
1. 在controller下新建Common类，实现初始化方法，初始化时根据menu表和title表的数据向前端赋值，实现点击侧边栏菜单后跳转后同时高亮当前选项以及显示标题栏标题。





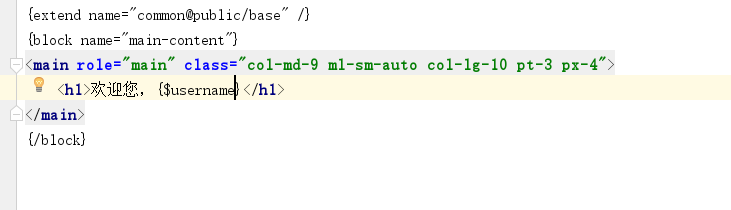
## 编写管理界面功能

1. Admin模块整体结构如下



1. 在admin/view/index目录下新建5个菜单选项对应的html,这里仅关注title.html和user\_info.html，前者实现修改标题栏标题的功能，后者展示所有已注册的有效用户信息。

Index.html，直接引用base模板，仅修改main-content这块即可，同时显示从Session中取到的用户名，place\_one/place\_two类似



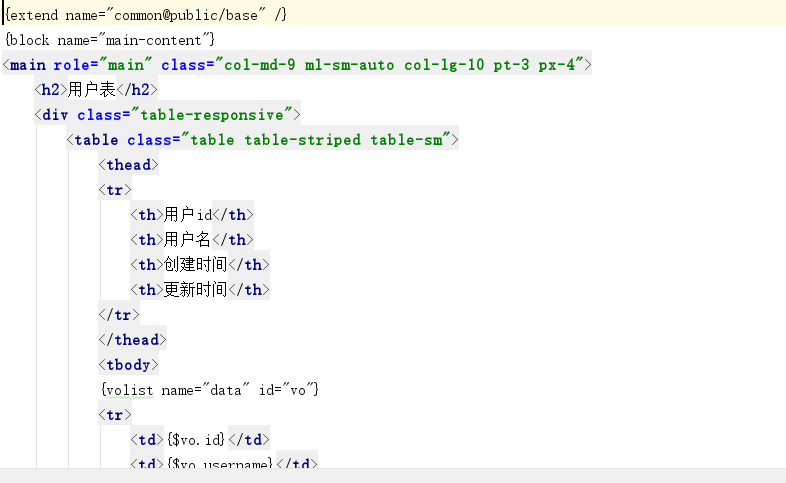
Title.html,显示修改标题内容的界面

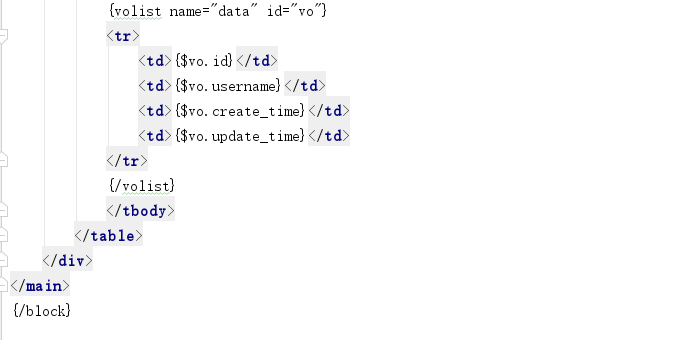


Js部分，向Title/updateTitle方法发送ajax请求，传输更新后的标题值



User\_info.html 显示所有用户信息的界面，用volist标签循环输出所有用户信息





1. 新建Index控制器继承自Common类，处理菜单栏部分的点击方法

从session中取出用户名





1. 新建Title控制器和模型类，处理修改标题事件

Title控制器类



Title模型类，实现更新标题内容

