



福州大学至诚学院  
FUZHOU UNIVERSITY ZHICHENG COLLEGE

# 服务端开发技术



# 课程作业

## 基于ThinkPHP5的图书管理系统(产出:word文档)

环境的搭建

数据库搭建和数据库连接配置

图书管理功能的实现

列表功能的实现

编辑功能的实现

删除功能的实现

# 1. 框架基础知识

01

开发框架简介

02

创建虚拟主机

03

命名规范与目录结构

04

框架概述和术语

05

入口文件和URL访问

# Part.1

## 1.1 开发框架简介

# 1.1 开发框架简介

## 开发框架的概念

开发框架指的是为**规范开发流程、降低开发维度、提高开发效率**而制定的一整套供大家使用的**功能模块及编程约定**。

## 选择开发框架的原因

- (1) 代码更加规范，易于维护
- (2) 开发效率更高，节省时间
- (3) 项目更加健壮，安全可靠

## Part.2

### 1.2 创建虚拟主机

## 1.2 创建虚拟主机

### TP5虚拟主机的建立

通过**域名访问**时，可以通过在本地创建一个虚拟主机，为虚拟的主机绑定一个域名，然后这个域名指向的ip地址改为127.0.0.1

网站域名：tp5.com


根目录：xxx\WWW\tp5\public

网站端口：80



## 1.2 创建虚拟主机

### Tools

 [phpinfo\(\)](#)

 [phpmyadmin](#)

 [Add a Virtual Host](#)

Apache Virtual Hosts `c:/wamp64/bin/apache/apache2.4.41/conf/extra/httpd-vhosts.conf`

VirtualHost already defined:

ServerName : [localhost](#) - Directory : C:\wamp64\www

Windows hosts `C:/Windows/system32/drivers/etc/hosts`

Name of the **Virtual Host** No space - No underscore(\_) **Required**

tp5.com

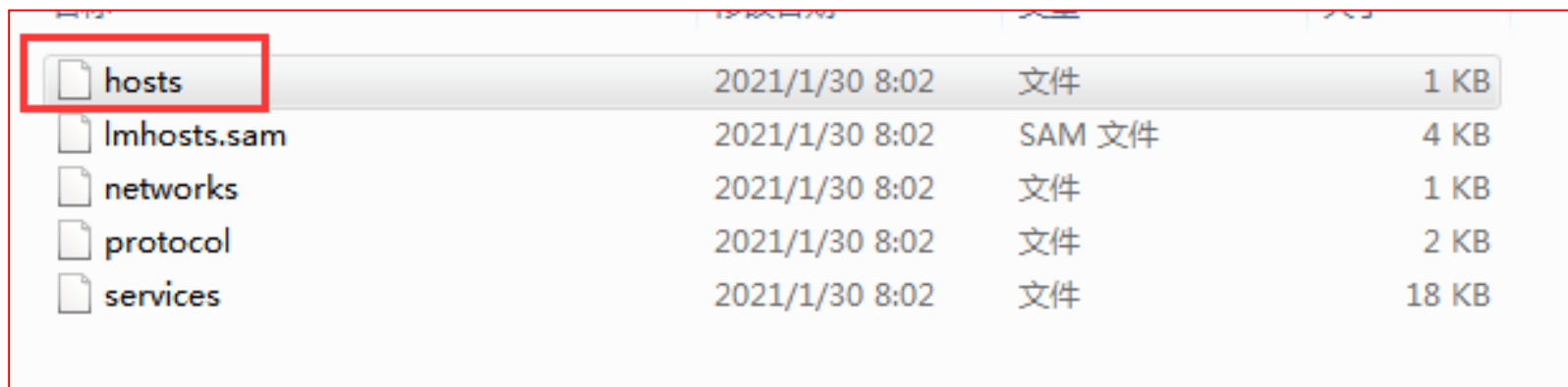
Complete absolute **path** of the VirtualHost **folder** Examples: C:/wamp/www/projet/ or E:/www/site1/ **Required**

C:\wamp64\www\tp5\public



## 1.2 创建虚拟主机

在C:\Windows\System32\drivers\etc下面



hosts	2021/1/30 8:02	文件	1 KB
lmhosts.sam	2021/1/30 8:02	SAM 文件	4 KB
networks	2021/1/30 8:02	文件	1 KB
protocol	2021/1/30 8:02	文件	2 KB
services	2021/1/30 8:02	文件	18 KB

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost
127.0.0.1   tp5.com
127.0.0.1   www.tp5.com
```

保存该文件，如果不能保存则保存在桌面，将该

再复制粘贴到C:\Windows\System32\drivers\etc进行替换

## 1.2 创建虚拟主机

**httpd-vhosts.conf中修改为如下代码：**

```
<VirtualHost *:80>
    ServerName tp5.com
    DocumentRoot "c:/wamp64/www/tp5/public"
    <Directory "c:/wamp64/www/tp5/public/">
        Options +Indexes +Includes +FollowSymLinks +MultiViews
        AllowOverride All
        Require local
    </Directory>
</VirtualHost>
```

**重启Apache服务，在URL中访问:www.tp5.com或tp5.com**



## Part.3

### 1.3 命名规范与目录结构

# 1.3 命名规范与目录结构

## 命名规范

### 一、匈牙利命名法：

标识符中的每个单词之间用下划线分隔：user\_name;

### 二、驼峰命名法：

标识符中的每个单词的首字母大写，用大写字母来分隔，根据首字母是否大写又分为：

1.大驼峰:首字母大写,常用作类名: ClassName,又称:帕斯卡命名法

2.小驼峰:除第一个单词外，每个单词首字母大写,常用作类属性和方法: \$userName

# 1.3 命名规范与目录结构

## 目录、类文件和其他文件的写法

类型	命名规范	实例
目录	小写字母 / 小写字母+下划线	user / user_type
类文件 (驼峰法)	命名空间与文件真实路径完全一致	app\index\controller
	类名与文件名保持一致(首字母大写)	User.php中只有User类
其它文件	小写字母 / 小写字母+下划线	list.php / list_user.php

# 1.3 命名规范与目录结构

## 函数与类属性的写法

类型	命名规范	实例
函数	匈牙利命名法(小写字母+下划线)	get_user_type()
类的方法	驼峰法(首字母小写)	getName()
类的属性	驼峰法(首字母小写)	userType
魔术方法	双下划线开头+驼峰命名法	__set(),__callStatic()



## 1.3 命名规范与目录结构

### 常量与参数的配置

类型	命名规范	实例
常量	(匈牙利命名法)大写字母+下划线	APP_PATH
配置参数	(匈牙利命名法)小写字母+下划线	url_route_on



# 1.3 命名规范与目录结构

## 数据库与字段

类型	命名规范	实例
数据表	小写字母+下划线,前缀用数据库名	tp5_staff
字段表	小写字母+下划线,前缀用表名	staff_name

## 1.3 命名规范与目录结构

### 目录结构

序号	名称	说明
1	application	应用目录(可自定义)
2	extent	扩展类库目录(可自定义)
3	public	外部可访问的Web目录
4	runtime	应用运行时目录
5	thinkphp	框架核心库目录
6	vendor	第三方类型目录

## 1.3 命名规范与目录结构

### 目录结构

序号	名称	说明
7	build.php	自动生成定义文件（参考）
8	composer.json	composer 定义文件
9	LICENSE.txt	版权文件
10	README.md	readme，即框架说明文件
11	think	命令行入口文件

## Part.4

### 1.4 框架概述和术语

## 1.4 框架概述和术语

### MVC简介

- think PHP5.0它是基于MVC方式来组织的
- MVC指的就是模型、视图和控制器
- 控制器先从模型中获取数据，然后再选择合适的视图进行输出

ThinkPHP5.0 基于 MVC（模型-视图-控制器）方式来组织；



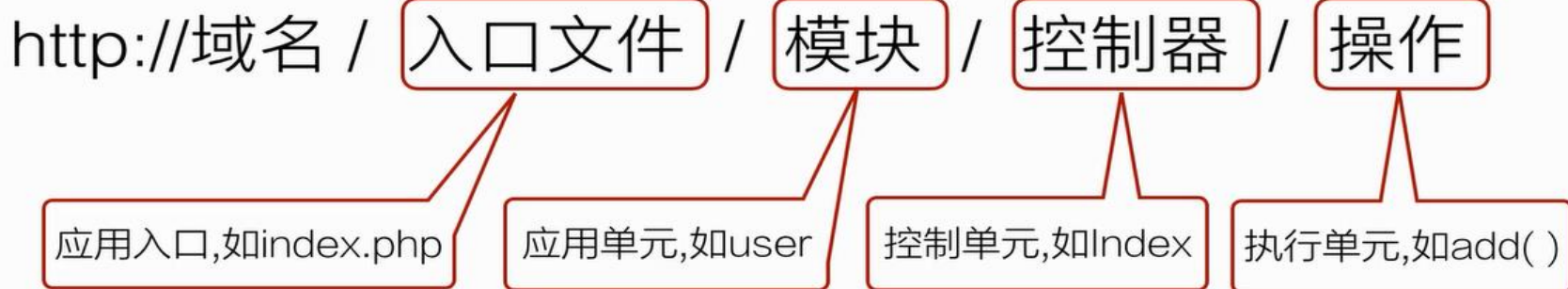
模型与视图，用控制器强制分离，数据请求与展示由控制器统一调配！

## 1.4 框架概述和术语

### URL默认采用PATH\_INFO方式

- PATH\_INFO就是将URL的访问地址用分割符分为几个相对独立的部分

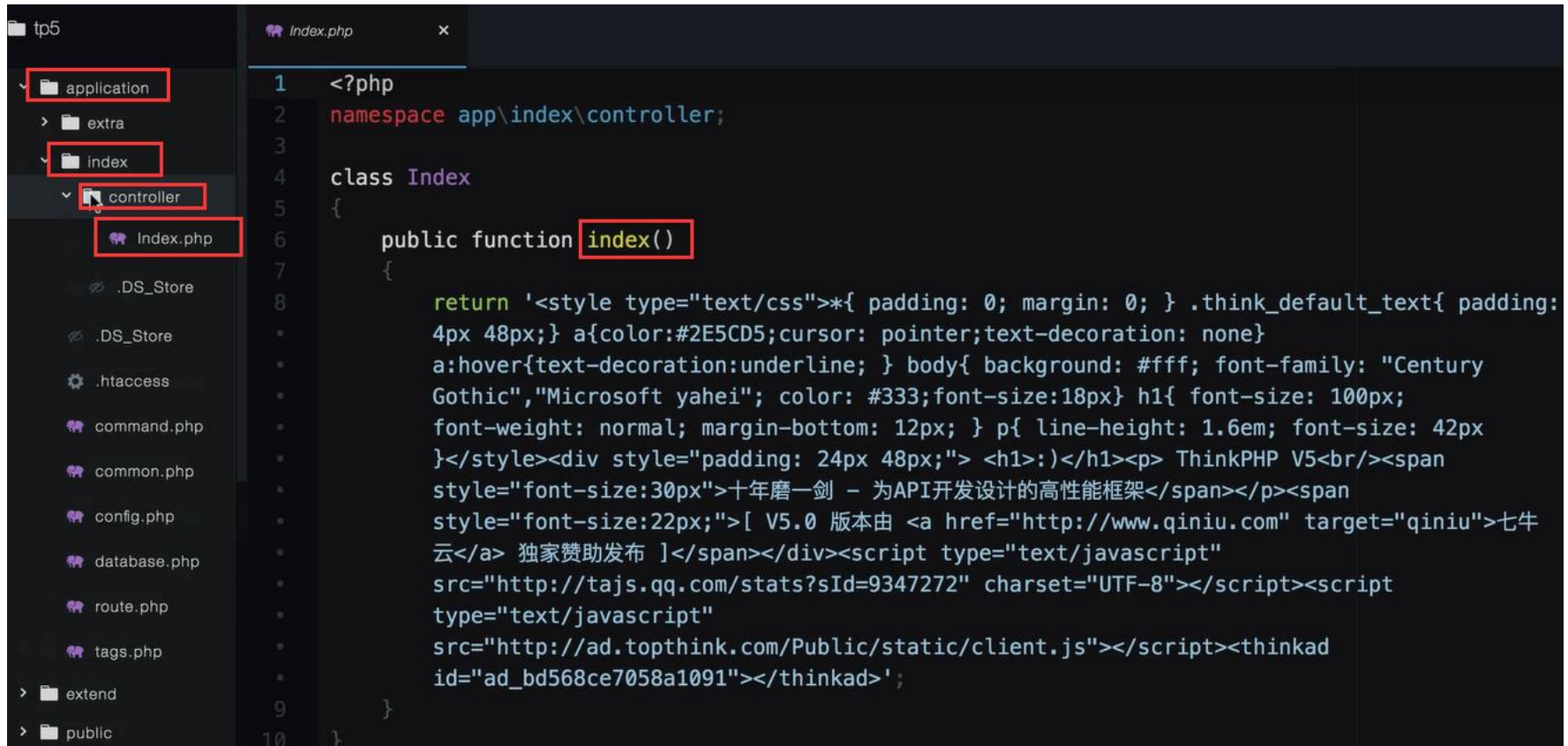
URL访问默认采用PATH\_INFO方式:



举例:http://tp5.com/index.php/index/user/list/ id/10/name/peter 参数列表



# 1.4 框架概述和术语



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the directory structure of a project named 'tp5'. The 'application' directory is expanded, showing 'extra', 'index', and 'controller' subdirectories. The 'controller' directory is further expanded, showing 'Index.php'. The 'Index.php' file is selected, and its contents are displayed in the code editor. The code is a PHP script that defines a class 'Index' with a public function 'index()'. The function returns a string of HTML code that includes a CSS style, a header, a paragraph, and a script tag.

```
1 <?php
2 namespace app\index\controller;
3
4 class Index
5 {
6     public function index()
7     {
8         return '<style type="text/css">*{ padding: 0; margin: 0; } .think_default_text{ padding:
9             4px 48px;} a{color:#2E5CD5;cursor: pointer;text-decoration: none}
10            a:hover{text-decoration:underline; } body{ background: #fff; font-family: "Century
11            Gothic","Microsoft yahei"; color: #333;font-size:18px} h1{ font-size: 100px;
12            font-weight: normal; margin-bottom: 12px; } p{ line-height: 1.6em; font-size: 42px
13            }</style><div style="padding: 24px 48px;"> <h1>:</h1><p> ThinkPHP V5<br/><span
14            style="font-size:30px">十年磨一剑 - 为API开发设计的高性能框架</span></p><span
15            style="font-size:22px;">[ V5.0 版本由 <a href="http://www.qiniu.com" target="qiniu">七牛
16            云</a> 独家赞助发布 ]</span></div><script type="text/javascript"
17            src="http://tajs.qq.com/stats?sId=9347272" charset="UTF-8"></script><script
18            type="text/javascript"
19            src="http://ad.topthink.com/Public/static/client.js"></script><thinkad
20            id="ad_bd568ce7058a1091"></thinkad>';
21     }
```

- <http://www.tp5.com/index.php/index/Index/index>
- <http://www.tp5.com/index.php/index/index/index>



## 1.4 框架概述和术语

```
application > index > controller > Index.php
1  <?php
2  namespace app\index\controller;
3
4  class Index
5  {
6      public function index()
7      {
8          return 'Hello World';
9      }
10 }
11
```

← → ↻ ⓘ 不安全 | tp5.com/index.php/index/Index/index

Hello World

# 1.4 框架概述和术语

## 常用术语

### 一、入口文件(index.php)

1. 入口文件是整个Web应用的起点;
2. 入口文件最常用的是:index.php;
3. 入口文件并不唯一, 例如可为后台单独设置入口:admin.php;
4. 入口文件可以通过URL重写功能将其隐藏。

# 1.4 框架概述和术语

## 二、应用(application)

1. 应用是URL请求到完成的(生命周期)处理对象，由`\think\App`类处理；
2. 应用必须在入口文件(如`index.php`)中调用并执行；
3. 可为不同的应用创建多个入口文件，如后台入口：`admin.php`；
4. 应用有自己独立的配置文件(`config.php`)和公共函数文件(`common.php`)；

## 三、模块(module)

1. 一个应用下有多个模块，对应着应用的不同部分，如前台，后台；
2. 每个模块都可以有完整的MVC类库，创建和管理这些类库是我们最主要的工作；
3. 每个模块可以有独立的配置文件(`config.php`)和公共函数文件(`common.php`)；

# 1.4 框架概述和术语

## 四、控制器(controller)

1. 每个模块下可以有多个控制器负责响应URL请求;
2. 每个控制器对应一个控制器类(class), 如User.php;
3. 控制器管理着模型与视图, 是系统资源的调度与分配中心;
4. ThinkPHP5的控制器不需要继承任何基类就可以工作。

```
application > index > controller > 🐞 Index.php
1  <?php
2  namespace app\index\controller;
3
4  class Index
5  {
6      public function index()
7      {
8          return '111';
9      }
10 }
```

```
application > index > controller > 🐞 Index.php
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          return '111';
9      }
10 }
```

## 1.4 框架概述和术语

### 五、操作(action)

1. 操作对应着控制器类中的方法，是URL请求的最小单元；
2. 任何URL请求，最终都是由控制器中的操作方法来完成；
3. 操作是整个应用的最终执行单元，是URL路由的核心与目标；

我才是真正干活的呀~~





## 1.4 框架概述和术语

Action对应着控制器类中的某一个方法，它是url请求的最小单元，任何的url请求最终都是由控制器中的操作方法来完成的。

```
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          return '111';
9      }
10
11     public function hello()
12     {
13         return '福州大学至诚学院';
14     }
15 }
```

← → ↻ ⓘ 不安全 | tp5.com/index/Index/hello

福州大学至诚学院

## 1.4 框架概述和术语

### 六、模型(model)

1. 模型通常对应整个应用，因此模型通常在应用(application)目录下创建；
2. 尽管模型主要是针对数据库进行CURD操作，但也可以不操作数据库；
3. 模型通常完成实际的业务逻辑和数据封装，并返回和格式无关的数据；
4. 模型返回的数据，通常是数组(array)或字符串(string)；
5. 模型支持分层操作，例如将模型层分为逻辑层/服务层/事件层。

### 七、视图(view)

1. 控制器调用模型返回的数据，是通过视图转换成不同格式输出的；
2. 视图根据请求，调用模板引擎确定是直接输出还是模板解析后再输出；
3. 视图由大量模板文件组成，这些文件对应着不同控制器中的操作方法；



## 1.4 框架概述和术语

TP5

application > index > view > index > hello.html > html

1 <!doctype html>  
2 <html lang="en">  
3 <head>  
4 <title>TP5</title>  
5 </head>  
6 <body>  
7 <h2>我是index模块下的Index控制器中的hello方法对应的模板文件</h2>  
8 </body>  
9 </html>

tp5.com/index/Index/hello

我是index模块下的Index控制器中的hello方法对应的模板文件

application > index > controller > Index.php

```
1 <?php
2 namespace app\index\controller;
3 use think\Controller;
4 class Index extends Controller
5 {
6     public function index()
7     {
8         return '111';
9     }
10
11     public function hello()
12     {
13         return $this->fetch();
14     }
15 }
```

## Part.5

### 1.5 入口文件和URL访问

# 1.5 入口文件和URL访问

## 入口文件

入口文件的定义，它主要用来定义框架路径或者项目路径。另外它还可以定义一些系统相关的常量，最后是载入框架的入口文件，那么这是必须的。

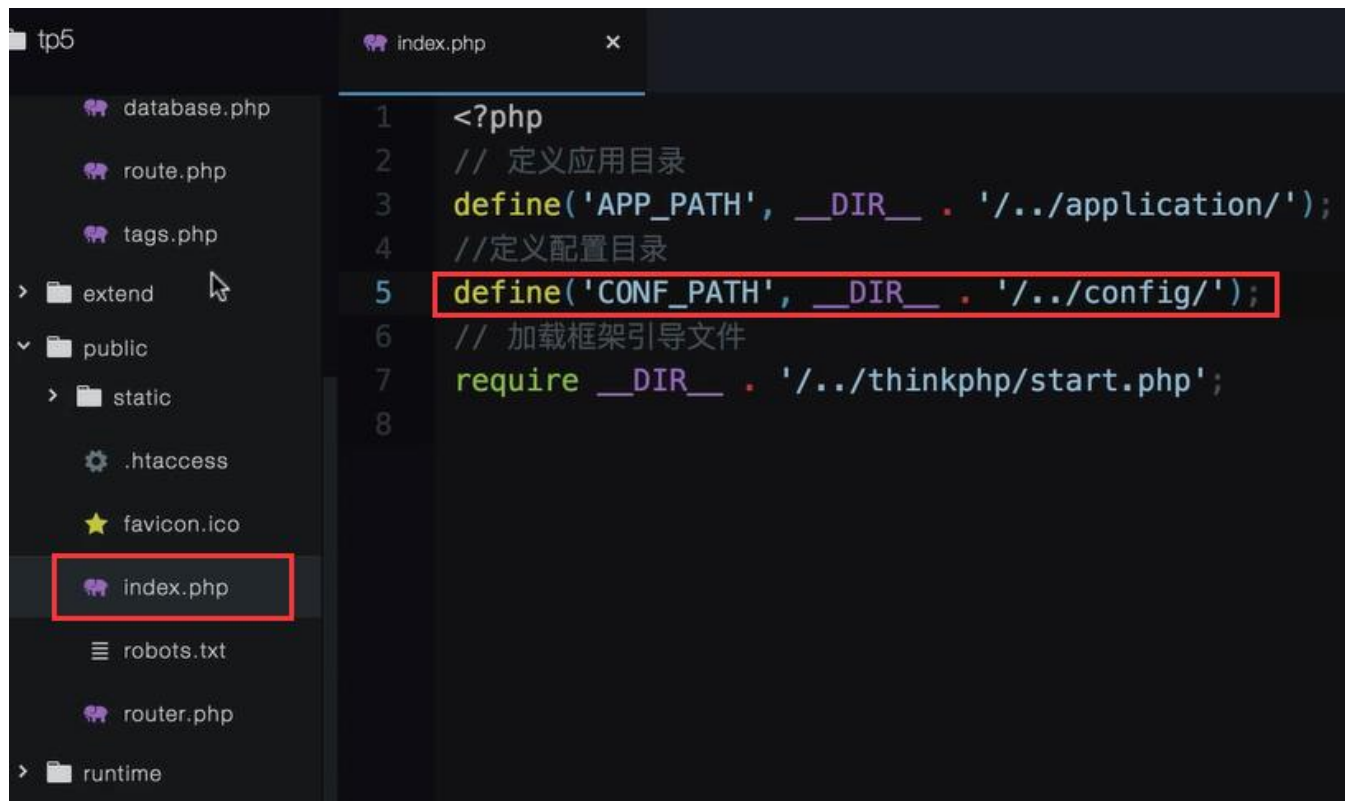
```
<?php
// 定义应用目录
define('APP_PATH', __DIR__ . '/../application/');
// 加载框架引导文件
require __DIR__ . '/../thinkphp/start.php';
```

应用的入口文件默认在:public/index.php

public目录为web可访问目录，其他的文件都可以放到非WEB访问目录下面

# 1.5 入口文件和URL访问

## 自定义一个配置目录



```
1 <?php
2 // 定义应用目录
3 define('APP_PATH', __DIR__ . '/../application/');
4 //定义配置目录
5 define('CONF_PATH', __DIR__ . '/../config/');
6 // 加载框架引导文件
7 require __DIR__ . '/../thinkphp/start.php';
8
```

# 1.5 入口文件和URL访问

```
TP5
├── application
├── config
│   └── config.php
├── extend
├── public
└── static
```

```
config > config.php
1  <?php
2  return
3  [
4      'site_name' => '福州欢迎您',
5  ];
```

```
application > index > controller > Index.php
1  <?php
2  namespace app\index\controller;
3  use think\Controller;
4  class Index extends Controller
5  {
6      public function index()
7      {
8          return config('site_name');
9      }
10
11
12 }
```



## 1.5 入口文件和URL访问

### 多入口文件

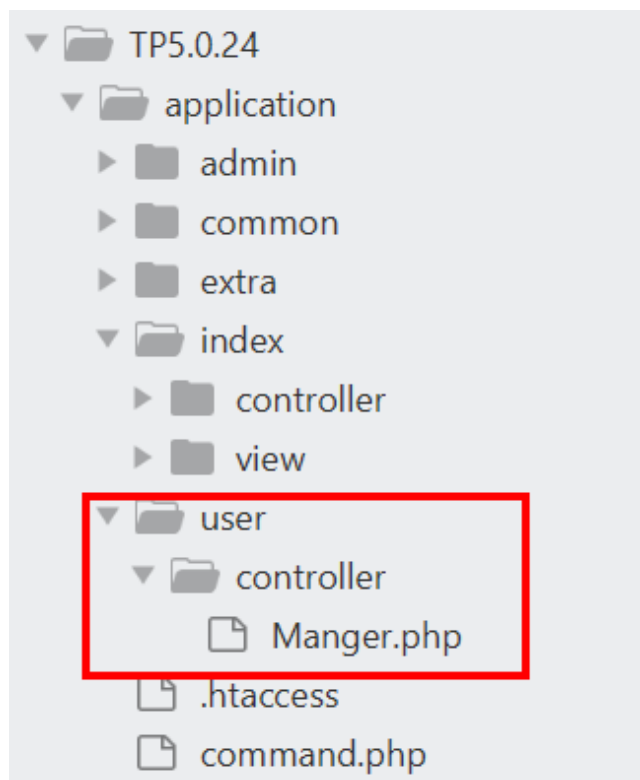
1. 一个应用通常只有一个入口文件，但也不是绝对的；
2. 例如为后台创建单独入口文件: `admin.php`, 与admin模块绑定；
3. 多入口文件，可以使应用项目层次更加清晰，管理更加灵活。

`index.php` 网站前台

`admin.php` 网站后台入口

## 1.5 入口文件和URL访问

### URL访问



在application选应用目录下，创建一个自定义的模块，新模块叫做user，在新的user模块下，再创建一个controller了，用它来存放控制器。在controller文件夹下面创建一个控制器文件，叫Manger.php。



## 1.5 入口文件和URL访问

### URL访问

```
<?php
//写上命名空间，控制器的命名空间和绝对路径一致
namespace app\user\controller;
//类名与控制器的文件名必须一致
class Manger
{
    public function add($n=0, $m=0)
    {
        return '$n + $m ='. ($n+$m);
    }
}
```

<http://tp5.com/index.php/user/manger/add/n/20/m/50>

# 1.5 入口文件和URL访问

## URL大小写

1. 默认情况下，URL是不区分大小写的；
2. URL里面的模块/控制器/操作名会自动转换为小写；
3. 控制器在最后调用的时候会转换为驼峰法处理。

```
class UserDemo
{
    public function list($id='')
    {
        //操作方法代码
    }
}
```

控制器文件名必须是UserDemo.php

访问控制器必须是user\_demo

例如：[http://tp5.com/index/user\\_demo/list/id/10](http://tp5.com/index/user_demo/list/id/10)