## 规范

* 目录使用小写+下划线
* 类库、函数文件统一.php作为后缀
* 命名空间与类库文件所在的路径一致
* 类文件采用驼峰法，其他文件采用小写+下划线命名；
* 类名和类文件名保持一致，统一采用驼峰法；
* 与tp3不同，类的命名不需要加Controller后缀
* 函数命名：小写+下划线；
* 方法命名：小驼峰；
* 属性命名：小驼峰；
* \_\_开头的是魔术方法或属性；
* 常量命名：大写+下划线
* 配置参数：小写+下划线
* 数据库字段：小写+下划线，不要以下划线开头；

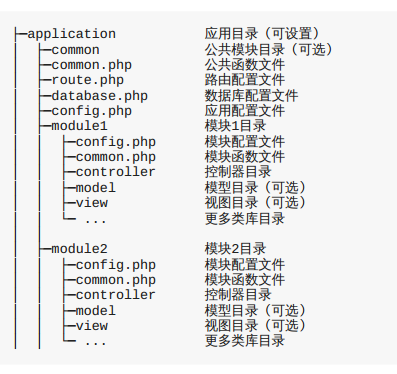
## URL

默认情况，URL不区分大小写，URL 里面的模块/控制器/操作名会自动转

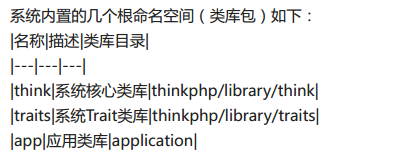
换为小写，控制器在最后调用的时候会转换为驼峰法处理

此时，要访问驼峰法的控制器需要使用<http://localhost/index.php/Index/blog_test/read>

## 目录结构



## 根命名空间



如果需要增加新的根命名空间，有两种方式：注册新的根命名空间或者放入 EXTEND\_PATH

目录（自动注册）：

1自动注册：将自己的类库包目录放入EXTEND\_PATH目录，默认为extend。

例：类文件位于extend/my/Test.php

|  |
| --- |
| namespace my;  class Test  {  public function sayHello()  {  echo 'hello';  }  }  我们就可以直接实例化和调用：  $Test = new \my\Test();  $Test->sayHello(); |

2 手动注册：

在应用改的公共文件中（application/commom.php）定义：

|  |
| --- |
| \think\Loader::addNamespace('my','../application/extend/my/');  多个的话：  \think\Loader::addNamespace([  'my' => '../application/extend/my/',  'org' => '../application/extend/org/',  ]); |

3 config.php中定义：

'root\_namespace' => [  
'my' => '../application/extend/my/',  
'org' => '../application/extend/org/',

]

.

## 类库导入（不推荐）

如果你不需要系统的自动加载功能，又或者没有使用命名空间的话，那么也可以使用

think\Loader 类的 import 方法手动加载类库文件

|  |
| --- |
| 示例  // 引入 extend/qrcode.php  Loader::import('qrcode', EXTEND\_PATH);  // 助手函数  import('qrcode', EXTEND\_PATH);  // 引入 extend/wechat-sdk/wechat.class.php  Loader::import('wechat-sdk.wechat', EXTEND\_PATH, '.class.php');  // 助手函数  import('wechat-sdk.wechat', EXTEND\_PATH, '.class.php'); |

## 路由

定义：

Route::rule('路由表达式','路由地址','请求类型','路由参数（数组）','变量规则（数组）');

可选参数：

'blog/:year/[:month]'=>'Blog/archive'

可选参数只能放到路由规则的最后，如果在中间使用了可选参数的话，后面的变量都会变

成可选参数。