# 序

## 1、适用用户

本框架适合：有面向对象的原生编程经验，至少使用过thinkphp，对mvc模式不陌生的程序猿，有基本的linux 指令和shell脚本操作能力。

目前实现大部分实用的框架功能，如：控制器、基本路由、数据模型、模版引擎，另外有简单的微信授权登录源码、和微信二级api。

## 2、环境

建议：php5.5、redis2.8(api环境)

## 3、待开发部分

1、模版引擎中 未实现 if else 条件标签。

2、虽然模版引擎（view.class.php）处理后能默认压缩html和显示，但跳过了对html页面内的js css 压缩。

3、模版引擎 未实现往静态目录输出静态页。

4、模版引擎 未实现智能转换路径的路由标签。

# 1、目录说明

## 【1公用目录】

api\bangju\weixin\

微信的基本授权目录，作为二级api，用redis做缓存，单独放在独立域名里，由计时器刷新7200秒失效的token，通过shell脚本实现。

wpi\bangju\shell\

计时器刷新api/bangju/weixin/index.php 的shell 脚本。

webtest\Common\app.php

公用文件，加载了webtest\Common\conf\ 三个设置文件，定义了路径变量，每个控制器入口页面都需要引入。

## 【2库类目录】

webtest\Library\Commons\ 公用类

webtest\Library\Debugs\ 测试类 包含框架测试 和 一般测试 webtest\indexTest.php 和控制器基类 中使用

webtest\Library\Https\ 调用 api\bangju\weixin\ 二级微信api 的请求类

webtest\Library\Thirds\ 第三方库类

webtest\Library\Controlers\ 控制器类

webtest\Library\Modelers\ 数据模型类 wpdb

webtest\Library\Views\ 视图类

## 【3系统目录】

webtest\Admin\ 管理后台 ( 使用boostrap ui 基本展示了 控制器类 和 视图类的使用 )

webtest\Home\ 前台 ( 无ui 基本展示了 微信授权登录 和 创建用户的 基本功能，使用了模型类 和 二级微信api请求类 )

# 2、使用方法

## 【1访问入口】

如：http://xxxx.com/Admin/

或：http://xxxx.com/Admin/index.php

对应文件：(webtest为网站根目录)/Admin/index.php ，该文件加载了控制器类 和 选择补充的公用类，还有一些基本设置。

## 【2控制器】

如：<http://xxxx.com/Admin/?/base/welcome>

或：<http://xxxx.com/Admin/index.php?/base/welcome>

对应目录：(webtest为网站根目录)/Admin/Controler/base.class.php ，(welcome)是其中的方法。

## 【3控制器类方法 中 使用视图类】

如：以上base.class.php 中的 admin 方法

先在该类头部 use \Views\view as V ; 前提是视图类在 app.php中已经加载 才能调用命名空间。

路径：/Admin/Controler/base.class.php –> admin

function admin(){

V::tamplate('index'); //路径：/Admin/Tamplate/index.html

$cArr['content']='change\_admin'; //路径：/Admin/Tamplate/change\_admin.html

V::asChangeArr($cArr);

$lArr[0]=array('aid'=>'1','admin'=>'admin1','auth'=>'超级','read'=>'说明');

$lArr[1]=array('aid'=>'2','admin'=>'admin2','auth'=>'普通','read'=>'说明');

$lArr[2]=array('aid'=>'3','admin'=>'admin3','auth'=>'普通','read'=>'说明');

$lArr[3]=array('aid'=>'4','admin'=>'admin4','auth'=>'普通','read'=>'说明');

$lArr[4]=array('aid'=>'5','admin'=>'admin5','auth'=>'普通','read'=>'说明');

V::forList('admin',$lArr); //路径：/Admin/Tamplate/change\_admin.html 替换列表标签

V::show();

}

路径：/Admin/Tamplate/index.html

加载index模版时，自动合并 include标签 指定的模版。

{{include::"include\_head"}}

{{include::"include\_nav"}}

{{include::"include\_foot"}}

路径：/Admin/Tamplate/change\_admin.html

按需要进行内嵌替换，把用不同”content”值区分的 change 标签 内嵌替换成指定模版

{{change::"content"}}

路径：/Admin/Tamplate/change\_admin.html 替换列表标签

{{forList::"admin"}}和{{/forList::"admin"}}之间的“[[key]]”将被php数组中同key的值替换。

{{forList::"admin"}}

<tr>

<th class="hidden-xs" id="[[aid]]" scope="row">[[aid]]</th>

<td>[[admin]]</td>

<td>[[auth]]</td>

<td class="hidden-xs">[[read]]</td>

<td>

<button type="button" class="btn btn-primary edit">编辑</button>

</td>

<td>

<button type="button" class="btn btn-danger">删除</button>

</td>

</tr>

{{/forList::"admin"}}

## 【4控制器类方法 中 使用数据模型类】

如：<http://xxxx.com/Home/?/weixin/index/>

加载Controler的文件：(webtest为网站根目录)/Home/Controler/weixin.class.php -> index

——————————

use \Modelers\model as model;//model装载器

use \Https\weiApi as weiApi;

class weixin extends baseControler {

function \_\_construct(){

model::init('weixin');//一个 controler 对应一个 同名 modeler

model::set();//加载 index modeler 可同时传参给 weixin.class.php的 构造函数

}

function index(){

$MS=model::set();//可随时传参给 weixin.class.php的 构造函数 改变整个执行结果

$user = $MS->rowSelect(USR, '\*', $whereArray);//筛选

}

}

——————————

加载的Modeler文件：(webtest为网站根目录)/Home/Modeler/weixin.class.php -> index

——————————

namespace Modelers;

use Modelers\baseModel;

class weixin extends baseModel {

function \_\_construct(){

new parent;//可以使用baseModel所有查询方法,包括wpdb的;

}

}

——————————

## 【5控制器类方法 中 使用路由序列类】

如：<http://xxxx.com/Home/?/weixin/index/sid=5/>

加载Controler的文件：(webtest为网站根目录)/Home/Controler/weixin.class.php -> index ? sid=5

——————————

use \Controlers\urlSerial as I;

class weixin extends baseControler {

function index(){

$I=I::get();//侦听url序列

$sid\_get=isset($I->sid)?$I->sid:'';//侦听url中的sid序列单元

}

}

——————————

# 3、其他