

糖果PHP框架为第二版框架,对原来框架进行了代码优化,结构优化,并且兼容URL重写和普通框架版本目前处于测试阶段,遇到有想法的朋友可以给我邮箱:nease@163.com 官网:http://www.wv3c.com

首先了解下项目目录结构:

- **system** 主文件路径,抱歉对vender不是太感冒
  - **bootstrap.php**->主框架核心
  - **autoload.php**->一个简化版的文件加载和报错处理
  - **core/template.php**->参考ZBLOG的模块编译库,蛮好用的.
  - **core/database.php**->一个改良的数据库类
- **app**->主项目目录文件夹
  - **config**->配置文件夹,包括数据库配置和其他配置选项
  - **controllers**->由Controller结尾的控制器类(逻辑视图调用)
  - **http->router.php** 其实就定义了一个Rewrite路由
  - **libs**->库文件目录
  - **models**->由Model结尾的模块类(数据调用)
  - **views**->一大堆HTML模版文件
  - **cache**->这个默认是没有的由views模版进行编译

以上的结构和第一代框架几乎差不多,做项目足够了.

假如你的服务器支持URL重写,例如阿里云,美橙等服务器,那么按照如下顺序往下看:

首先确认下.htaccess文件是否存在,如果不存在创建一个:

```
<IfModule mod_rewrite.c>
  <IfModule mod_negotiation.c>
    Options -MultiViews
  </IfModule>

  RewriteEngine On
  RewriteRule ^system/(.*)$ - [F]
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)/$ /$1 [L,R=301]
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^ index.php [L]
  RewriteCond %{HTTP:Authorization} .
  RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
</IfModule>
```

**index.php:**

```
<?php
require 'system/bootstrap.php';
bootstrap::start();
```

**app/http/router.php**

```
<?php
map("/helloworld",function(){
    echo "helloworld";
});
```

在浏览器中输入http://localhost/helloworld 就会出现helloworld了,是不是很神奇. 整个框架分为7个函数:

```
map    路由
model  模块
view   视图
route  控制器
import 库文件导入
```

```
dao    数据库控制函数
config 配置文件读取函数
```

**bootstrap::map(\$\_name,方法或者字符串形式)**

```
bootstrap::map('/helloworld',function(){echo "helloworld";})
bootstrap::map("/(\d+)",function($num){
    echo $num;
});
bootstrap::map("/(\w+)",function($str){
    echo $str;
});
bootstrap::map("/say","Helloworld@say");
```

**bootstrap::model(\$\_name)** 目录映射 app/models/名称Model.php

app/model/SampleModel.php

```
<?php
class SampleModel
{
    function getList(){
        return array("a","b","c");
    }
}
```

app/controllers/HelloworldController.php

```
<?php
class HelloworldController
{
    function say(){
        var_dump(bootstrap::model("sample")->getList());
    }
}
```

http://localhost/say

```
array(3) {
  [0]=>
  string(1) "a"
  [1]=>
  string(1) "b"
  [2]=>
  string(1) "c"
}
```

**bootstrap::renderer(\$viewname,\$params=array(),\$mode=0)** 视图调用

```
bootstrap::renderer("/helloworld",array("name"=>"小张"));
```

app/view/helloworld.html

```
<html><body>hello{$name}</body></html>
```

浏览器输出:

```
hello小张
```

**bootstrap::route(\$controlname,\$method)**手动路由模式,如果你有特别需要

```
bootstrap::route("Helloworld","say");
```

**import(\$dot\_name)** 导入libs下的文件

```
import excel.PHPExcel
import microMsg.MicroMsgProxy
```

**dao(\$name)**数据库连接器,单独作为类使用用的PDO驱动,目前的话基本都支持PDO驱动的.

```
bootstrap::dao("config");
```

对应配置文件:app/config/config.php

```
<?php
return array(
    "dsn"=>"mysql:host=localhost;dbname=sample",
    "user"=>"root",
    "passwd"=>"root"
);
```

有如下方法:

```
bootstrap::dao()->fetch()
bootstrap::dao()->fetchAll();
bootstrap::dao()->lastInsertId();
...
```

如果要使用简拼的方法名称:

```
index.php
<?php
require 'system/bootstrap.php';
require 'system/short_func.php';
bootstrap::start("app");

ww_view=bootstrap::renderer
ww_route=bootstrap::route
ww_import=bootstrap::import
ww_config=bootstrap::config
ww_dao=bootstrap::dap
ww_map=bootstrap::map
```

<http://blog.csdn.net/flashdevelop/article/details/54845739>

更多功能等着你去发现.