

2-12mvc与前端模板的套用

| | |
|-----------------------|---|
| 2-12mvc与前端模板的套用 | 1 |
| 1. 课堂案例 | 2 |
| 课堂讲解 | 2 |
| 一、上节回顾 | 2 |
| 新闻的添加 | 2 |
| 新闻的修改 | 2 |
| 新闻的删除 | 2 |
| 二、学习目标 | 2 |
| mvc..... | 2 |
| 公用部分分离 | 2 |
| 视图加载封装 | 2 |
| 三、教学过程描述 | 2 |
| 1、公用部分的分离 | 2 |
| 2、MVC | 2 |
| 3、单入口 | 3 |
| 4、PHP规范..... | 4 |
| 5、数据库常量 | 5 |
| 6、header | 6 |
| 7、视图函数的封装 | 6 |
| 8、url封装..... | 6 |
| 9、success封装 | 7 |
| 10、error封装 | 7 |
| 11、百度编辑器的重新定义 | 7 |
| 12、前端模板的套用 | 7 |
| 四、小结 | 8 |
| 自由主题 | 8 |
| 2. 课堂练习 | 8 |
| 3. 课后练习 | 8 |
| 4. 资料扩展 | 8 |

1. 课堂案例

课堂讲解

一、上节回顾

新闻的添加

新闻的修改

新闻的删除

二、学习目标

mvc

公用部分分离

视图加载封装

三、教学过程描述

1、公用部分的分离

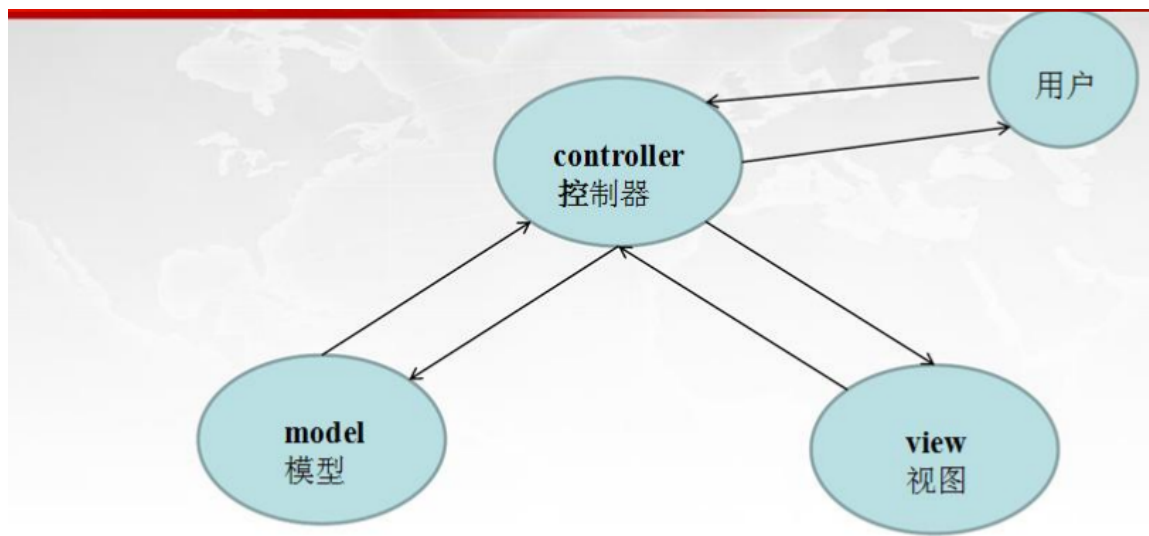
把公用部分分离出来, 方便以后维护管理

2、MVC

MVC全名是Model

View

Controller, 是模型(model)-视图(view)-控制器(controller)的缩写, 一种软件设计典范, 用一种业务逻辑、数据、界面显示分离的方法组织代码, 将业务逻辑聚集到一个部件里面, 在改进和个性化定制界面及用户交互的同时, 不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。



2-1、controller

业务逻辑层, 负责程序业务逻辑, 接收用户请求, 输出结果及调度各

2-2、model

数据层, 负责处理与数据相关的逻辑, 数据库等的处理

2-3、view

视图层, 负责处理前端页面数据, 如Html、Js等, 生成前端页面, 但不负责输出, 输出由Controller负责

3、单入口

系统只有一个入口, 所有的程序访问都必须先经过该程序, 这段程序就是这个系统的唯一入口。由于所有的程序都必须经过此入口, 所以一切的初始化工作都可以交给入口来做。

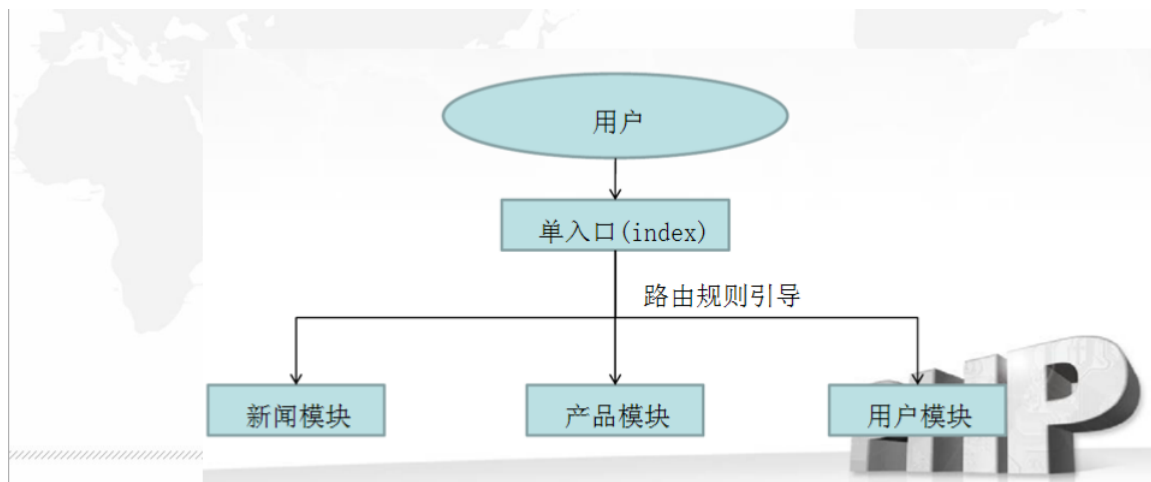
例如: 一幢房子只有一个大门, 通过这个大门我们就可以更好的管理出入的人员

3-1、优点

代码复用率高, 例如初始化的代码可以写在单入口, 其他模块不需要重复写
安全性较高, 可以把所有验证安全的代码都写在入口中

保证所有程序的访问目录是一致的, 一切的路径都以index.php为准

单入口只能放置系统初始化与公共必须的逻辑, 如验证登录并非每个程序都需要, 因此验证登录就不能放在单入口执行。



3-2、架构

1、建立单入口程序，在该程序中加载系统所需的所有初始化程序或类库等。

2、建立路由规则，通过url，以参数的方式告诉入口需访问的程序。

入口根据不同参数访问不同模块

路由规则：

m:module (模块获取变量admin:后台home:前台)

c:controller(控制器获取变量)

a:action(操作获取变量)

例：访问后台的登录：

`http://localhost/mvc/index.php?m=admin&c=index&a=login`

4、PHP规范

类的方法命名规范

* 包括修饰符(public,private,protected)

*

2.1 基本编码准则

代码必须符合 PSR-1 中的所有规范。

PHP代码文件必须以 `<?php` 或 `<?=` 标签开始；

PHP代码文件必须以 不带BOM的 UTF-8 编码；

PHP代码中应该只定义类、函数、常量等声明，或其他会产生

从属效应

的操作(如：生成文件输出以及修改.ini配置文件等)，二者只能选其一；

类的命名必须遵循 StudlyCaps 大写开头的驼峰命名规范；`AdminbaseController`

方法名称必须符合 camelCase 式的小写开头驼峰命名规范：`add_post => addPost`

类中的常量所有字母都必须大写，单词间用下划线分隔；`define("GREETING", "xiaoxia");`

PHP代码必须使用 `<?php ?>` 长标签 或 `<?= ?>` 短输出标签；

一定不可使用其它自定义标签。

PHP代码必须且只可使用不带BOM的UTF-8编码。

属性

类的属性命名 可以 遵循:小写开头的驼峰式 (\$camelCase)

2.2 文件

所有PHP文件必须使用Unix LF (linefeed)作为行的结束符。

所有PHP文件必须以一个空白行作为结束。

纯PHP代码文件必须省略最后的 `?>` 结束标签。

2.3. 行

行的长度一定不能有硬性的约束。

软性的长度约束一定要限制在120个字符以内, 若超过此长度, 带代码规范检查的编辑器一定要发出警告, 不过一定不可发出错误提示。

每行不应该多于80个字符, 大于80字符的行应该折成多行。

非空行后一定不能有多余的空格符。

空行可以使得阅读代码更加方便以及有助于代码的分块。

每行一定不能存在多于一条语句。

2.4. 缩进

代码必须使用4个空格符的缩进, 一定不能用 tab键。

2.5. 关键字 以及 True/False/Null

PHP所有 关键字必须全部小写。

常量 true 、false 和 null 也必须全部小写。

3. namespace 以及 use 声明

namespace 声明后 必须 插入一个空白行。

所有 use 必须 在 namespace 后声明。

每条 use 声明语句 必须 只有一个 use 关键词。

use 声明语句块后 必须 要有一个空白行。

模板 assign里变量名用小写下划线格式:

```
$this->assign('server_info', $info);
```

model的limit用法统一为两个参数

```
limit(0,10)
```

5、数据库常量

把数据库登陆信息封装成常量方便以后的修改

6、header

header(string,replace,http_response_code)

参数 描述

string:必需。规定要发送的报头字符串。

replace :可选。指示该报头是否替换之前的报头, 或添加第二个报头。

默认是 true(替换)。false(允许相同类型的多个报头)。

http_response_code: 可选。把 HTTP 响应代码强制为指定的值。(PHP 4 以及更高版本可用)

例:

```
header('Location: http://www.baidu.com/')
```

倒计时跳转

```
header("Refresh: 3; url= http://www.baidu.com/")
```

7、视图函数的封装

封装视图加载函数

```
function view($theme, $data = array())
{
    extract($data);
    include('app/theme/' . $theme . '.html');
}
```

extract

extract()

函数从数组中将变量导入到当前的符号表。该函数使用数组键名作为变量名, 使用数组键值作为变量值。针对数组中的每个元素, 将在当前符号表中创建对应的一个变量。

8、url封装

地址跳转封装

```
function url($url)
{
    $info = explode("/", $url);
    $urls = 'index.php?m=' . $info[0] . '&c=' . $info[1] . '&a=' . $info[2];
    return $urls;
}
```

9、success封装

成功后提示封装

```
function success($data, $url)
{
    include('app/theme/admin/success.html');
    exit;
}
```

10、error封装

错误提示封装

```
function error($data, $url)
{
    include('app/theme/admin/error.html');
    exit;
}
```

11、百度编辑器的重新定义

```
1  * 百度编辑器上传文件路径 *
2  "imagePathFormat": "/demo214/data/ueditor/php/upload/image/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,可以自定义保存路径和
3  /* {filename} 会替换成原文件名,配置这项需要注意中文乱码问题 */
4  /* {rand:6} 会替换成随机数,后面的数字是随机数的位数 */
5  /* {time} 会替换成时间戳 */
6  /* {yyyy} 会替换成四位年份 */
7  /* {yy} 会替换成两位年份 */
8  /* {mm} 会替换成两位月份 */
9  /* {dd} 会替换成两位日期 */
10 /* {hh} 会替换成两位小时 */
```

12、前端模板的套用

新闻模块

12-1、列表

```

$pageSize = 10; //显示数目
$p = !empty($_GET['p']) ? $_GET['p'] : 1;
$offset = ($p - 1) * $pageSize;
//计算新闻总条数
$count = dbCount('mvc_news');

$result = dbAll('select *. "from mvc_news order by id desc limit $offset,$pageSize "');
//时间戳转换
foreach ($result as $key=>$value){
    $result[$key]['time'] = date('Y-m-d H:i:s',$value['time']);
}
$page = adminPage($count,$pageSize,$p);
view('home/news/index',array('result'=>$result,'page'=>$page));

```

12-2、详情

```

$id = $_GET['id'];
$sql = "select * ". "from mvc_news where id={$id}";
$result = dbOne($sql);
view('home/news/details',array('result'=>$result));

```

四、小结

- 1、mvc 就是为了更好的控制代码
- 2、封装好视图函数方便加载

自由主题

2. 课堂练习

- 1、了解mvc
- 2、命名规范
- 3、了解单入口

3. 课后练习

- 1、把以前的项目改成mvc模式

4. 资料扩展