

EduaskCMS建 站手册

eduaskcms



目 录

前言

系统基础

环境需求

系统安装

开发目录

URL说明

后台操作

栏目操作

内容发布

广告发布

系统设置

字典操作

数据提取

数据字典

前台开发

callback使用

模板引擎

前台模板继承

默认css、js和添加

栏目数据

首页数据

广告数据

列表数据

详情数据

系统设置数据

列表风格扩展

图集风格扩展

单独模板扩展

网站地图

项目部署

URL重写

目录改造

前言

认识

很高兴可以在这里因为EduaskCMS这个内容管理系统和大家能够在这里沟通！

首先，我还是先简单的自我介绍下吧！大家可以叫我老伍，职业是一名PHP讲师。开发这个系统的主要目的算是有个3个吧：

- 1、没有在第一线做开发了，还是想动动代码，算是爱好和手痒吧；
- 2、自己在web开发这个行业那么多年，还是希望可以留下点足迹吧，算是为这个行业做那么点点贡献；
- 3、为了我那一群又一群爱过、伤过、感动过的可爱的“孩子们”，在他们工作之际最后为他们留下一点点什么吧！

EduaskCMS我会一直更新、维护下去，所以使用的朋友们不用担心后续版本终断的问题。不认识的朋友，欢迎你提出你高贵的意见；认识的朋友，多半就是可爱的你们了，以后就要靠你们自己了哦，加油吧！ps：/尴尬，文档中误打、错打的文字可能有点多，如果发现了我会修改，请谅解！

关于

EduaskCMS是基于ThinkPHP5.0.11框架开发的一个免费开源的内容管理系统，开发的理念不光注重后台，同样注重后台功能的扩展；更加注重的是前台的快速开发和便捷。系统希望实现的目标是“让你的代码，能省则省”。你可以使用EduaskCMS直接进行你自己的建站、学习、功能开发，**但不可以在系统基础上更新发布子版系统。**

传送门

系统的手册有3本：

[EduaskCMS培训手册](#)：

主要介绍系统后台的相关操作和要点，使用EduaskCMS开发的网站可以使用它做为客户的网站操作培训手册。

[EduaskCMS建站手册](#)：

也是你现在正在看的这本手册，主要介绍如何使用EduaskCMS开发一个网站。使用该手册意味着你几乎不用写PHP代码，你主要的任务是前段页面的制作，然后“套”网站。所以该手册不会介绍功能如何扩展，只会告诉你模板的继承、数据的调用等，完成一个“套”网站的任务。

[EduaskCMS开发手册](#)：

表示系统现有的功能不够支撑到你这个项目，你需要扩展、修改功能。而该手册详细的介绍了利用EduaskCMS进行功能的扩展和开发。同时该手册的介绍也是系统的核心所在，你可以利用它来学习ThinkPHP5、学习CMS开发、学习EduaskCMS系统、快速便捷开发一些定制功能。

[EduaskCMS官网地址](#)：

可以查看系统最新动态和下载最新系统。

[EduaskCMS后台演示地址](#)：

用户名和密码都是：test123，可能有时会卡，请见谅！主要是用的9块9的服务器。

[码云下载地址](#)：

如果系统对你有所帮助，请举手之劳点个赞！感谢！

[ThinkPHP5.0完全开发手册](#)：

如果你仅仅需要建立一个简单的项目，大概了解下ThinkPHP5也就够了。

[LayUI开发文档](#)：

系统后台是基于LayUI前端框架的。

广告

小广告插播，成都IT培训请到[成都华育国际](#)。如果有朋友需要介绍来的，请直接联系作者 老伍哦，是有回报的哦！

开始

最后需要确认下，你已经至少了解过ThinkPHP5.0框架、简单明白EduaskCMS后台操作，你需要利用EduaskCMS开发一个企业、个人、博客型的简单项目，如果需要定制开发一些功能或修改功能请换一本手册。现在我们开始来如何快速的开发一个网站吧：

系统基础

[环境需求](#)

[系统安装](#)

[开发目录](#)

[URL说明](#)

环境需求

环境要求

ThinkPHP5 的环境要求如下

- PHP >= 5.4.0
- PDO PHP Extension
- MBstring PHP Extension
- CURL PHP Extension

环境安装

下载[phpstudy](#)并安装，phpstudy集成环境包的使用请自行查阅对应手册，里面集成了不同版本的PHP可以轻松满足系统环境要求。

- 1、安装成功以后，找到你安装目录phpstudy下面的www目录，将里面除了phpMyAdmin文件夹，其他的文件都删除；
- 2、浏览器地址栏输入127.0.0.1或localhost即可访问到www目录了；
- 3、安装的时候需要涉及到导入数据库文件，可以通过www目录下面的phpMyAdmin完成导入任务。

系统安装

下载EduaskCMS

目前可以下载到最新版系统主要有2个方式：

- 1、官网下载：<http://www.eduaskcms.xin>
- 2、码云下载：<https://gitee.com/eduaskcms/eduaskcms>

解压

下载系统以后解压，你会得到2个文件夹：

eduaskcms-1.x.x：cms系统文件夹，文件夹名字可以随便改。它就是项目文件夹了。我们每做一个项目，可以复制一个该文件夹出来搭建项目。

libs：系统核心引入文件夹，比如thinkphp5核心。

这两个目录在后面的开发目录中会详细介绍。

现在将这2个目录 放置与服务器根目录即可。

安装

1、修改数据库配置文件：eduaskcms-1.x.x/app/database.php

```

14  'type'           => 'mysql',
15  // 服务器地址
16  'hostname'       => '127.0.0.1',
17  // 数据库名
18  'database'       => 'tpcms', 输入你想要的数据库名
19  // 用户名
20  'username'       => 'root',
21  // 密码
22  'password'       => 'root', 输入你的数据库用户名和密码
23  // 端口
24  'hostport'       => '',
25  // 连接dsn
26  'dsn'            => '',
27  // 数据库连接参数
28  'params'         => [],
29  // 数据库编码默认采用utf8
30  'charset'        => 'utf8',
31  // 数据库表前缀
32  'prefix'         => 'eduask_', 输入数据表前缀，可以不修改
33  // 数据库调试模式

```

其中database的库名可以是不存在的，其他的根据具体情况配置

2、通过浏览器访问eduaskcms-1.x.x下的install

如果是本地localhost访问大概是这样的：

```

http://127.0.0.1/eduaskcms/public/install
http://localhost/eduaskcms/public/install

```

如果是虚拟主机一定要绑定到public目录，大概是这样的：

本文档使用 [看云](#) 构建


```
http://www.eduaskcms.xin/install
```

这样就安装成功了！

强烈建议，为了项目安全，安装成功以后将：eduaskcms-1.x.x/app/home/controller/Install.php文件重命名或删除。

登录后台

在系统中home是前台模块，run是后台模块，下一节有说明。

后台访问url：domain/run

默认用户名：eduask

默认密码：eduask

开发目录

下载最新版系统后，解压缩到web目录下面，可以看到初始的目录结构如下：

Libs	系统核心引入目录（可配置）
—smarty	Smarty模板引擎引入核心
—thinkphp	ThinkPHP框架引入核心
—Vendors	第三方类库引入核心（短信、邮件、优图...其它待
开发)	
eduaskcms	应用部署目录（每新建一个项目，添加一个该目录）
—app	应用目录（可设置）
—common	公共模块目录（可更改）
—api	第三方接口类库（短信、邮件、优图...其它待开发
)	
—behavior	行为实现类库
—controller	控制器目录
—helper	助手类库
—lib	扩展类库（目前只有callback类）
—model	模型目录
—utility	工具类库（Hash、TpText）
—home	前台模块目录
—controller	控制器目录
—view	视图目录
—...	更多类库目录
—run	后台模块目录
—controller	控制器目录
—view	视图目录
—...	更多类库目录
—command	命令行工具配置文件
—common	应用公共（函数）文件
—config	应用（公共）配置文件
—database	数据库配置文件
—route	路由配置文件
—tags	应用行为扩展定义文件
—public	WEB 部署目录（对外访问目录，可修改；如西部数
码服务器一般为wwwroot)	
—css	CSS文件目录
—editor-4.5.6	Ckeditor4.5.6版本
—files	插件文件目录
—font	字体文件目录
—images	图片文件目录
—js	JS文件目录
—tempfile	临时文件目录（可清空）
—upload	上传文件目录
—htaccess	用于 apache 的重写
—index.php	应用入口文件
—web.conf	用于 IIS7+ 的重写
—...	
—runtime	应用的运行时目录（可写，可设置）
—appcache	应用静态缓存文件目录（Menu、AdminMenu、Set
ting...)	

—cache	应用缓存目录
—log	应用日志目录
—temp	模板编译文件目录
—build.php	自动生成定义文件（参考）
—index.php	本地127.0.0.1访问让url跳转到public目录
—think	命令行入口文件

这里Libs目录是系统核心引入目录，通常是存放每个应用（项目）都会使用到的核心文件，比如每个应用都需要引入的ThinkPHP核心、Smarty核心、第三方接口SDK等。然后每个应用（项目）都和Libs并列放置：

www	服务器根目录
—Libs	系统核心引入目录（可配置）
—项目目录	不同项目目录
—项目目录	
—项目目录	
—...	

每做一个项目复制一份eduaskcms来进行开发，这样多个项目可以共用同一份核心目录（Libs）。当然这样的目录放置一般是本地或拥有自己服务器的；有些客户可能使用的是虚拟空间或无法满足这样的Libs和项目并列放置的需求，这个时候也是可以将Libs目录配置到项目里面去的，让一个项目单独使用一份Libs，这个需求后面“项目部署”中会提到。

URL说明

后台URL规则

后台url都是按标准url进行方法：模块/控制器/方法/参数1/参数值1.html

e.g：domain/run/menu/lists.html 栏目列表

e.g：domain/run/menu/sort/parent_id/1.html 一级栏目排序

后台控制器方法对照：

create	添加方法
modify	修改方法
delete	删除方法
batch_delete	批量删除方法
sort	排序方法
lists	列表页方法
ajax_switch	列表页“开关”字段方法
ajax_set_field	列表页字段值修改方法

前台URL规则

前台url同样安装标准url进行方法：模块/控制器/方法/参数1/参数值1.html。但目前，前台页面基本上都是和栏目相关联的，每个栏目类型有主要的2个方法，分别是：show列表页方法和view详情页面方法，而且都经过了路由进行美化。

列表页：domain/控制器（栏目类型）/show/栏目id.html

详情页：domain/控制器（栏目类型）/数据id.html

美化后访问url：domain/article/show/3.html 栏目id为3的文章列表页
对用完整url： domain/home/article/show/menu_id/3.html

美化后访问url：domain/article/8.html id为8的文章详情页
对应完整url： domain/home/article/view/id/8.html

从例子中可以看出：

列表页方法为show，栏目id参数变量为menu_id

详情页方法为view，详情id参数变量为id

目前有的栏目关联类型控制器名称对照：

Article	文章
Page	单页
Product	产品
Link	链接
Download	下载

Feedback
Album

留言
图集

后台操作

使用这本手册的人们都是专业人士了，这里就不详细的介绍具体操作了，主要介绍一些规则或使用。

[后台操作传送门](#)

Ps：写手册比写代码还累....

栏目操作

后台栏目操作传送

- 1、标题 和 副标题：可用于中英文标题
- 2、别名：最后栏目自动生产的url，可用于url美化
- 3、列表页条数：默认为0，将自动获取系统设置中的列表条数；大于0以后该栏目的列表分页的条数，将以该数字为准
- 4、导航显示：控制该导航是否在前台导航中显示
- 5、自体跳转：如果该栏目有子栏目，将自动跳转到它下面的第一个子栏目
- 6、地图显示：用于“网站地图”中是否显示
- 7、栏目类型：

栏目-- 默认显示该导航子级导航或同级导航的连接，引导用户进入导航的作用，可能会比较少用

文章-- 顾名思义。用于发布一些需要文字介绍的内容，如：新闻、资讯、问答、信息类的栏目

单页-- 可以理解成只能同时显示一篇文章，所以没有列表页，如：公司介绍、联系、致辞等栏目

产品-- 用于发布一些图文介绍的内容，如：公司产品、案例等栏目

外链-- 用于填写一个链接，方便用户通过栏目到该链接中，显示内容有该链接页面决定；可以是站外链接，也可以是自己站内的某个链接

链接-- 用于发布很多链接，如：友情链接、底部链接

下载-- 用于后台发布一些文件，供前台下载。如：资料下载

留言-- 用于用户与网站进行交互

图集-- 用于相册等图片展示内容，如：团队介绍、案例、风采等

8、缩略图尺寸是设置该栏目的内容在上传图片是自动生成的缩略图尺寸。默认为0，将自动获取系统设置中的数据。

9、单独模板：每个栏目主要是列表--默认模板是show.html；详情--默认模板是view.html。如果你某个栏目想自定义一个模板可以设置。

举例：

单独模板：	show	news	删除 插入 
	view	view1	删除 插入

表示该栏目的show方法（列表页）使用news.html模板；view方法（详情页面）使用view1.html模板。这样你就可以自己创建2个模板单独写该栏目的页面。
比如有些网站的某个栏目是需要单独设计页面，该功能就比较实用。

10、列表风格：可以选择当前栏目列表页列表的样式（如果列表页设置了单独页面无效），里面的样式可以自行添加定义，后续文档中会单独介绍这个功能。

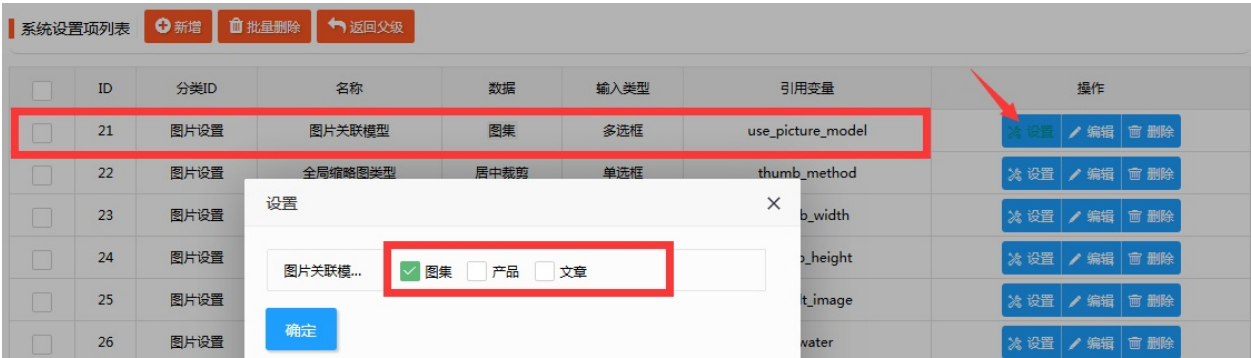
11、文字简介：有些网页可能会显示一些栏目文字，但又不是该栏目中某个具体内容中的数据，就可以使用该字段设置数据。

12、seo关键词和描述：用于对应栏目meta中的数据。

内容发布

内容发布传送

内容管理感觉没有什么需要特别介绍的。
有一点就是，如果你想文章、产品一条数据关联多张图片，就想图集功能一样。
你可以在系统设置中：



将你需要关联图片的模型打勾。

广告发布

系统设置

字典操作

数据提取

数据字典

前台开发

[callback使用](#)

[模板引擎](#)

[前台模板继承](#)

[默认css、js和添加](#)

[栏目数据](#)

[首页数据](#)

[广告数据](#)

[列表数据](#)

[详情数据](#)

[系统设置数据](#)

[列表风格扩展](#)

[图集风格扩展](#)

[单独模板扩展](#)

[网站地图](#)

callback使用

特点

这是一个特殊的类，我们在建站的过程当中，几乎所有的代码都可以写在这里面，避免去核心控制器中添加或修改代码。同时，这里面也可以方便我们获取到所有准备传输到页面上的数据并可以重新进行修改。如果项目没有自己的功能修改，我们也建议前台的你需要添加的操作在这个文件中进行，以避免系统有更新和你自己新增代码冲突。

介绍

本手册主要针对企业型建站，所以这个文件你需要关注的方法是：

appBeforeHome：每次访问Home模块的某个方法之前执行

appAfterHome：每次访问Home模块的某个方法后、渲染页面之前（这个时候所有需要assign到页面的数据都已经准备好）执行

模板引擎

模板引擎

系统默认采用的模板引擎是：Smarty，所以你最好是熟悉这个模板引擎的。

[在线Smarty3手册](#)

当然你也可以自己修改home模板的配置，配置成thinkphp5自己的模板引擎。这样的话home模块的模板将完全由你自己去搭建，这里我们暂时不多做介绍（后期考虑再搭建一版Think引擎的出来，目前只有Smarty引擎模板）。

Smarty重要笔记

A、模板中定义变量 -- 定义变量主要有两种方式

```
{assign var="变量名" value="值"}
```

或者像php中一样

```
{ $变量名="值" }
```

不管是哪一种定义的变量或者是控制器中传递出来的变量，如果要输出变量，可以

```
{ $变量名 }
```

B、判断语句

if分支

```
{if 条件}满足条件的任意代码{/if}  
{if $a>10}<div>变量$a的值是{$a}，它大于了10哟</div>{/if}
```

if...else分支

```
{if 条件}满足条件的任意代码{else}不满足条件的任意代码{/if}  
{if $a>10}  
a变量的值大于10  
{else}  
a变量的值小于等于10  
{/if}
```

if...elseif...else多分支

```
{if 条件1}
满足条件1的任意代码
{elseif 条件2}
满足条件2的任意代码
{elseif 条件3}
满足条件3的任意代码
{else}
不满足任何条件的任意代码
{/if}
```

C、变量修饰器

变量修饰器类似于函数的作用，让某个值或变量实现某种目的。

先来看格式：

```
{ $变量 | 修饰器函数 : 参数 }
{ $变量 | 修饰器函数 : 参数 | 修饰器函数 : 参数 }
可以使用多个修饰器
```

重点几个修饰器

```
1、cat：连接多个变量
{ $var | cat : ' yesterday.' }
2、date_format：格式输出日期
{ $var | date_format : 'Y-m-d H:i:s' : date }
3、default：为变量设置默认值。当变量是unset或者empty的字符串时，默认值将显示
{ $var | default : '这个家伙什么也没有填' }
4、nl2br：将变量值中的"\n"回车全部转换成HTML的 <br />
{ $var | nl2br }
5、replace：对变量进行简单的搜索和替换
{ $var | replace : 'Garden' : 'Vineyard' }
6、strip：转换连续空格，回车和tab到单个空格或是指定字符串
{ $var | strip : '&nbsp;' }
7、strip_tags：去除标记等任何包含在< 和 >中间的字符
{ $var | strip_tags }
8、truncate：字符串截取，指定字符代替
{ $var | truncate : 长度 : '...' : true }
```

因为Smarty的底层代码就是php，最终也会被编译成php代码，所以Smarty中可以使用php的函数来处理变量。

```
{ trim($str) }
{ substr($str, 0, 1) }
{ if in_array($str, [5,6,7,8]) } ... { /if }
```

D、注释

```
{*注释的内容*}
```

E、文件加载

```
{include file="文件路径"}
```

F、数组使用

定义：

```
{$array=[5,6,7,8]} //smarty中[]就代表数组，定义方式和php中一致
```

循环：

```
{foreach $array as $key=>$value}
    //这里面是循环体，可以是任意代码
    <li>
        当前循环次数：{$value@iteration}
        索引数字数组访问：{$value[0]}
        关联数组访问：{$value.title}
        混合数组访问：{$value[0].title}
    </li>
{/foreach}
在循环体中：
{$value@index} : 返回当前循环索引
{$value@iteration} : 返回当前循环次数和index相差1
{$value@first} : 如果是第一次循环返回true；否则返回false
{$value@last} : 如果是最后一次循环返回true；否则返回false
{$value@total} : 返回总循环数
```

G：{}转义

我们知道smarty所有的相关语法都要放置到{}中，但在模板当中经常出现{...}而不是smarty的时候肯定报错，我们怎么处理呢？

简单的css和js代码左{后有空格或换行即可

```
<style>
div{ color:red;}
</style>
<script>
var obj={ "attr":"val"};
```

```

if (...) {
    ....
}
</script>

```

但模板中并不是所有的地方都可以通过空格或回车解决这个问题，比如正则中

```

<script>
var val='test';
if (/^\s{0,}$/.test(val)) {
    ....
}
</script>
//这个时候肯定是报错的，而正则中的{}是不可以通过空格解决的。

{ldelim} //可以转义左边的花括号
{rdelim} //可以转义右边的花括号

<script>
if (/^\s{ldelim}0,}$/.test(val)) {
    ....
}
</script>
//这样写放到模板中将不会有问题

//如果一段代码中，有大量的花括号但又没有smarty语法，我们可以使用literal标签将js代码
包住
{literal}
<script>...</script>
{/literal}
//这样里面的代码所有的{}都将视为文本，不会做任何解析。

```

H、自定义function

定义：

```

{function 函数名 参数名=默认值}
{$参数名}
{/function}

```

调用：

```

{call 函数名 参数名=值}

```

I、模板继承

继承是从面向对象编程而来的概念，模板继承可以让你定义一个或多个父模板，提供给子模

板来进行扩展。扩展继承意味着子模板可以覆盖部分或全部父模板的块区域。

模板继承

```
{extends file="父级模板路径"}
```

模板重写

```
{block name=""}
```

这里的代码可以覆盖父级同名block里面的代码

```
{/block}
```

前台模板继承

默认css、js和添加

CSS和JS文件引用

我们在实际开发中，有的css、js可能需要整站每个页面都引用，有些css、js可能需要某个页面单独使用。

在控制当中，我们只需要调用Assign对象的addJs方法即可添加一条js引用

默认js文件应该存放在public/js/下面，加载在head里面，文件后缀会自动补全：

```
$this->assign->addJs('test');
//<script type="text/javascript" src="http://domain/js/test.js"></script>
//head里面引用；位置在public/js/test.js
$this->assign->addJs('home/test.js');
//<script type="text/javascript" src="http://domain/js/home/test.js"></script>
//head里面引用；位置在public/js/home/test.js
$this->assign->addJs('/home/test', true);
//<script type="text/javascript" src="http://domain/home/test.js"></script>
//body最后引用；位置在public/home/test.js
//说明第二个参数true时，将在body结尾引用，否则head中引用；路径以 / 开始表示定义public/后面的路径，否则表示定义public/js/后面的路径
$this->assign->addJs('http://www.外网.com/jquery.js');
//引起外网js文件使用绝对网络路径
```

在控制当中，我们只需要调用Assign对象的addCss方法即可添加一条引用

默认css文件应该存放在public/css/下面，只能加载在head里面，文件后缀会自动补全：

```
$this->assign->addCss('test');
//<link rel="stylesheet" type="text/css" href="http://domain/css/test.css" />
//位置在public/css/test.css
$this->assign->addCss('home/test.css');
//<link rel="stylesheet" type="text/css" href="http://domain/css/home/test.css" />
//位置在public/css/home/test.css
$this->assign->addCss('/home/test.css');
//<link rel="stylesheet" type="text/css" href="http://domain/home/test.css" />
//位置在public/home/test.css
//路径以 / 开始表示定义public/后面的路径，否则表示定义public/css/后面的路径
$this->assign->addCss('http://www.外网.com/test.css');
//引起外网css文件使用绝对网络路径
```


栏目数据

保存路径

所有的栏目数据缓存成静态PHP数组，该文件存放在：

项目/runtime/appcache/Menu.php，该文件每当栏目数据有改变的时候都会重新生成；

所有如果你想重新生成栏目数据，随便一个栏目点击“编辑”，再点击“提交”即可。

注意该目录：项目/runtime/appcache/ 其实所有的静态数据（非数据缓存）都存放在该目录的。

栏目数据调用

在系统中，我们给大家准备了一个menu()函数用来获取栏目数据，该函数你可以在PHP代码中使用；也同样可以在写页面的时候，模板中使用。

```
menu('nav') //获取到所有导航结构，你在生成主导航遍历的时候，就会使用到它
e.g:
pr(menu('nav'));//执行该代码，这里pr函数是数据输入函数，方便调试，输出：
Array
(
    [3] => Array
        (
            [7] => Array
                (
                )
            [8] => Array
                (
                )
        )
    [4] => Array
        (
        )
    [5] => Array
        (
        )
    [6] => Array
        (
        )
)
//这里我们有4个一级栏目，id为3的栏目有2个二级栏目；结构中的数字就是每个栏目的ID
e.g:
pr(menu('nav', 3));//执行该代码，输出：
Array
(
    [7] => Array
        (
        )
    [8] => Array
        (
        )
)
```

```
(
)
)
```

```
menu(栏目ID) //已知栏目ID，获取该栏目数据
e.g:
pr(menu(3)); //我想获取ID为3的栏目数据，结果输出：
Array
(
    [id] => 3
    [parent_id] => 1
    [level] => 1
    [child_count] => 2
    [title] => 关于
    ....
    [keywords] =>
    [description] =>
) //这里整个ID为3的栏目数据都可以得到了，每个字段是什么意思，可以在后台 > 开发工具 >
模型管理 > 找到Menu模型 > 数据字典中查阅
```

```
menu(栏目ID, '字段') //已知栏目ID，想获取该栏目某个字段的值（字段可以按上述方法查
阅数据字典）
e.g:
pr(menu(3, 'title')); //输入 ID为3栏目的标题，最后输入“关于”
```

```
menu('list') //一个二维数字，得到所有栏目的详细数据
```

下面我们在模板中，做一个生成主导航的示范：

```
<ul class="grid">
    <li class="{if !$top_id}current{/if}"><a href="{call index_link}">首页
</a></li>
    {foreach menu('nav') as $level1_id=>$level1} ##遍历一级栏目
    <li class="{if $top_id==$level1_id}current{/if}"> ##给当前栏目加一个curr
ent类名
        <a href="{call menu_link id=$level1_id}">{menu($level1_id, 'title'
)}</a>##输出一级栏目 和 生成链接
        {if $level1} ##判断是否有二级栏目
        <dl class="layui-nav-child">
            {foreach $level1 as $level2_id=>$level2} ##遍历二级栏目
            <dd><a href="{call menu_link id=$level2_id}">{menu($level2_id
, 'title')}</a></dd>
            {/foreach}
        </dl>
        {/if}
    </li>
    {/foreach}
</ul>
```

如果要复制到模板中测试，请自行将 `##` 注释删除

栏目链接生成

在“系统基础”的“URL说明”中，我们已经明白，每个栏目都有列表和详情2个页面

show：是栏目列表

view：是具体数据详情

如果已知栏目ID，我们要生成栏目列表页链接：

```
{call menu_link id=栏目id}  
e.g :  
{call menu_link id=3} //可以得到如下：  
/page/show/3.html
```

首页数据

广告数据

广告数据获取

后台操作中，我们已经知道每个广告位都有一个“变量”。这个变量，就是该广告的唯一识别码。

如果想在控制器中获取某个广告位的数据，通过该“变量”即可。在Home控制器中，我们已经给大家准备好了一个方法 `getAdData` 用来做这件事。

```
//控制器中：
//$this->getAdData(string 广告位变量, int 广告数量);
$this->ts->getAdData('index_banner', 0); //0表示不限制，但受广告位“广告限制”字段限制
```

该方法没有返回值，但数据已经帮你自动的assign到模板中了。

模板中获取到数据方式：

```
{$ad.广告位变量}
```

现在我们在首页模板中打印index_banner广告位的数据：

```
{pr($ad.index_banner)}
e.g :

Array
(
    [id] => 1
    [title] => 首页Banner广告位
    [ex_title] =>
    [vari] => index_banner
    [width] => 1920
    [height] => 600
    [mobile_width] => 600
    [mobile_height] => 450
    [limit] => 0
    [is_thumb] => 1
    [is_text] => 0
    [created] => 2017-10-10 11:22:25
    [modified] => 2017-10-10 11:22:25
    [ad_count] => 2
    [Ad] => Array
        (
            [0] => Array
```

```

(
    [id] => 2
    [ad_position_id] => 1
    [title] =>
    [ex_title] =>
    [image] => upload/Ad/201710/201440365559eebaed42d70.j
pg
    [thumb] => upload/thumbs/201710/201440365559eebaed42d
70_1920_600_3.jpg
    [mobile_image] => upload/Ad/201710/201440365559eebaed
42d70.jpg
    [mobile_thumb] => upload/Ad/201710/201440365559eebaed
42d70.jpg
    [link] =>
    [content] =>
    [is_verify] => 1
    [created] => 2017-10-24 12:00:45
    [modified] => 2017-10-24 12:00:45
    [list_order] => 0
)
[1] => Array
(
    [id] => 1
    ...
    [list_order] => 0
)
)
)

```

从打印的结果中，我们可以看到广告位和广告数据都已经给你了。这个时候我们需要遍历广告：`$ad.index_banner.Ad` 即可（具体每个字段什么意思，请查看后台模型数据库字典）。

如果查询广告位后，你想在控制器也得到查询到的数据：

`$this->assign->ad['广告位变量']` 即可。

默认广告位数据处理

列表数据

列表页数据，是指访问show方法后 由控制器assign(系统默认)出来的数据。

`$list` : 当前访问的列表数据
`$page` : 当前访问的页码数据
`$render` : bootstrap风格翻页html结构

`$menu_data` : 当前访问栏目的数据
`$top_id` : 当前访问栏目所在的一级栏目ID
`$path` : 当前访问栏目所经过的所有栏目ID，可用于做"当前位置"
`$side_menu['top_menu']` : 当前访问栏目所在的一级栏目数据
`$side_menu['menus']` : 内页侧栏目ID列表

详情页面连接生成

{call view_link item=数据 mdl=模型}

列表遍历示例：

```
ul class="list">
    {foreach $list as $item}
    <li><span>[{ $item.date|date_format:'Y-m-d'}]</span><a href="{call view_link item=$item}">{$item.title}</a></li>
    {/foreach}
</ul>
```

首页遍历示例：

```
<ul class="list">
    {foreach $query_data[3] as $item}
    <li><span>[{ $item.date}]</span><a href="{call view_link item=$item mdl=menu(3,'type')}">{$item.title}</a></li>
    {/foreach}
</ul>
```


详情数据

详情页数据

详情页数据，是指访问view方法后 由控制器assign(系统默认)出来的数据。
为了方便理解，建议看的时候打开一个view.html进行参照。

`$data` : 当前访问的详情数据

e.g :

/Article/5.html : 文章模型ID为5的这条数据

/Product/6.html : 产品模型ID为6的这条数据

`$prev` : 当前访问数据的上一条数据

`$next` : 当前访问数据的下一条数据

`$menu_data` : 当前访问栏目的数据

`$top_id` : 当前访问栏目所在的一级栏目ID

`$path` : 当前访问栏目所经过的所有栏目ID，可用于做"当前位置"

`$side_menu['top_menu']` : 当前访问栏目所在的一级栏目数据

`$side_menu['menus']` : 内页侧栏目ID列表

侧栏目示例

```
<div >
  <h2 class="reset">{$menu_data.title}</h2>
  <ul class="list">
    {foreach $side_menu.menus as $menu_id}
      <li class="{if $menu_id==$menu_data.id}current{/if}"><a href="{ca
ll menu_link id=$menu_id}">{menu($menu_id, 'title')}</a></li>
    {/foreach}
  </ul>
</div>
```

系统设置数据

保存路径

所有的系统设置数据缓存成静态PHP数组，该文件存放在：
项目/runtime/appcache/Setting.php
如果想重新生成该文件，随便一个系统设置点“设置”，然后“确定”即可。

数据调用

在系统中，我们给大家准备了一个setting()函数用来获取栏目数据，该函数你可以在PHP代码中使用；也同样可以在写页面的时候，模板中使用。

使用方法：

```
setting('系统设置变量');  
//e.g:  
setting('site_title '); //获取网站名称
```

<input type="checkbox"/>	ID	分类ID	名称	数据	输入类型	引用变量	操作
<input type="checkbox"/>	34	站点信息	网站名称	EduaskCMS	文本框	site_title	设置 编辑 删除
<input type="checkbox"/>	35	站点信息	网站关键字	cms,eduaskcms,内容管理系统,thinkphp5	文本框	site_keywords	设置 编辑 删除
<input type="checkbox"/>	36	站点信息	网站描述	EduaskCMS是基于ThinkPHP5.0.11框架开发的一个免费开	文本域	site_description	设置 编辑 删除
<input type="checkbox"/>	37	站点信息	版权描述	© eduaskcms.xin 版权所有	文本框	site_copyright	设置 编辑 删除
<input type="checkbox"/>	38	站点信息	统计代码		文本域	site_code	设置 编辑 删除
<input type="checkbox"/>	39	站点信息	公司名称	公司名称	文本框	corp_title	设置 编辑 删除

列表风格扩展

列表风格

通过之前的文档和系统使用中，我们可以了解到，栏目的列表页的列表风格是可以在 后台栏目编辑 中选取的。目前系统默认的列表风格，我们只写了几种，后面的更新中我们会写出更多的列表样式（这里也希望广大使用者提供列表样式代码，我们集合到系统中）。当然，在实际开发中，有些栏目的列表样式可能需要我们自己定制开发。这里我们将介绍下，怎样将自己写的列表样式加入进去。

- 1、首页，你的html模板代码应该写在 `app\home\view\show.html` 中。
- 2、然后，在`{block name=function}`中写入自己的模板代码（可以参考系统自带的几个）：

```
{function name="自己的列表风格名字" list=[] mdl=$mdl}  
<ul class="list cms_case1">  
  {foreach $list as $item}  
    <li><span>[{$item.date|date_format:'Y-m-d'}]</span><a href="{call view_link item=$item}">{$item.title}</a></li>  
  {/foreach}  
</ul>  
{/function}
```

- 3、最后，将“自己的列表风格名字”添加到栏目列表风格选项中：找到文件 `common\model\Menu.php` 大概在158行把自己的列表风格添加进去。

图集风格扩展

图集风格

图集风格是指图集展示的详细页面以什么样的方式展示图集的许多图片。

图集风格扩展是指将自己的一些图片展示插件扩展到项目中使用。

1、在home\controller\Album 中的view方法指定自己插件的js和css代码：

```
switch ($viewStyle) {
    case 'simpleLide':
        $this->assign->addJs('/files/simple-lide/js/183.js');
        $this->assign->addJs('/files/simple-lide/js/simple.slide.
min.js');
        $this->assign->addCss('/files/simple-lide/css/simple.slid
e.css');
        break;
    case 'lightGallery':
        $this->assign->addJs('/files/lightGallery/js/lightGallery
.min.js');
        $this->assign->addCss('/files/lightGallery/css/lightGalle
ry.css');
        break;
    case '你自己的插件名字':
        请查阅“默认css、js和添加”章节添加自己的js和css引入文件
        break;
    default:
        break;
}
```

2、在home\view\Album\view.html中加入自己插件的html结构代码（大概在81行）：

```
{if $view_style == '你自己的插件名字'}
插件名字一点要一致，将自己的html代码写到一个if语句中
{/if}
```

所有的图片数据都在：\$data.AlbumPicture 中，你可以 {pr(\$data)} 或 {pr(\$data.AlbumPicture)} 看下数据结构（可以在后台模型数据字典中查看每个字段的意思）。

3、将自己的风格加入到图集的选项中：common\model\Album 大概86行

```
'view_style' =>[
    'type' => 'string',
```

```
'name' => '显示样式',
'elem' => 'select',
'options' => [
  'simpleLide' => '样式一',
  'lightGallery' => '样式二',
  '你自己的插件名字' => '随便取一个中文名给客户看的'
],
```

单独模板扩展

单独模板：每个栏目主要是列表--默认模板是show.html；详情--默认模板是view.html。如果你某个栏目想自定义一个模板可以设置。

举例：

单独模板：	show	news	- 删除 + 插入 
	view	view1	- 删除 + 插入

表示该栏目的show方法（列表页）使用news.html模板；view方法（详情页面）使用view1.html模板。这样你就可以自己创建2个模板单独写该栏目的页面。

比如有些网站的某个栏目是需要单独设计页面，该功能就比较实用。

网站地图

网站地图

网站地图控制器、方法为：Index/sitemap，而且进行了别名绑定。

访问URL为：你自己的域名/sitemap.html

项目部署

[URL重写](#)

[目录改造](#)

URL重写

URL重写

ThinkPHP5的url重写传送门：[走你](#)

在系统根目录 eduaskcms-1.x.x/public/ 中：

如果是apache服务器，.htaccss文件已经存在，可以直接运行；

如果是IIS7.0+服务器，web.Config文件以及存在，可以直接运行

目录改造
