

HkCms tp6版 使用手册《免授 权，永久商用》

HkCms开源内容管理系统《永久商用》



目 录

简介

下载与安装配置

下载

虚拟主机安装

虚拟主机必读

百度云虚拟主机

创业云虚拟主机

Linux下宝塔安装

宝塔一键部署

Windows下宝塔安装

phpEnv下安装

phpStudy下安装

伪静态 (url重写)

安装系统

调试模式与env文件

重新安装与安装文件说明

后台使用帮助

后台入口文件

首页面板

配置中心

站点配置

邮件配置

附件配置

应用中心配置

前台URL规则配置

SEO设置

SEO设置-静态生成

文档属性

站点模块

V2.2.2版本 -

V2.2.3版本 +

模型管理

栏目管理

字段类型与组件

前言

前端组件

动态下拉(selectpage)

验证规则

text 单行文本

textarea 多行文本
radio 单选
checkbox 复选
select 下拉列表
selects 下拉列表(多)
selectpage 动态下拉
editor 编辑器
image 图片
images 图片(多)
number 数字
date 日期
datetime 日期和时间
downfile 文件上传
downfiles 文件上传(多)
array 数组(键值对)

内容管理

自定义URL

多语言

界面多语言

内容多语言

多语言URL地址

附件管理

应用中心

标签语法

标签使用必读

列表标签

content 内容列表标签

contentpage 内容列表分页

wfpage 瀑布流分页

filter 筛选标签

order 排序标签

内容标签

guestbook 留言(表单)标签 (废弃)

form 表单标签

pre 上一篇标签

next 下一篇标签

prenext 上一篇下一篇(高级)标签

fileinfo 获取附件信息(文件下载)

内容分页

raw 格式化编辑器文本(即将废弃)

全局标签

seo 标签

channel 栏目标签
arcone 获取单条文档
breadcrumb 面包屑导航标签
adv 站点模块标签
date 日期格式化标签
substr字符(标题)截取标签
volist 循环标签
language 获取语言列表标签
lang 特定语言下显示内容标签
query 查询数据库内容
include 导入模板文件标签
原生PHP与php标签
map生成位置地图
thumb 生成缩略图
addons 插件检测

标签管理

taglist标签(获取标签列表)
tagarclist根据标签获取文章内容列表

导入标签

模板开发

1. 创建模板
2. 模板目录结构
3. 模板描述文件(重要) info.ini
4. 变量输出格式
4. 模板变量
5. 获取站点配置
6. 模板界面多语言
7. 模板静态目录及调用静态文件
8. 模板定义插件事件
9. 模板搜索页
10. 模板配置JSON
11. 导入演示数据
12. 卸载删除演示数据
13. 模板文件打包
14. 整站源码打包

插件开发

1. 创建插件
2. 目录结构
3. 插件名称类
4. 插件信息(info.ini)
5. 插件事件
6. 配置文件

- 7. install 安装目录
- 8. 创建菜单
- 9. 插件控制器
- 10. 插件视图
- 10-1 插件主题化
- 11. 插件模型
- 12. 模型事件(已废弃)
- 13. 插件内置函数
- 13-1 插件函数库
- 14. 插件多语言
- 15. 插件中间件
- 16. 插件路由
- 17. 插件验证器
- 18. 标签库
- 19. 第三方类库
- 20. 数据库文件导入
- 插件打包
- 后台前端使用帮助
- 二次开发帮助(更新中)
- 后台前端开发帮助
- 路由(链接地址、访问规则)
 - URL访问
 - 路由定义
 - URL生成
- 会员管理
 - 前台限制页面登录或权限控制
- 附录
 - 数据字典
 - 助手函数库
- API接口

简介

HkCms开源内容管理系统

HkCms开源内容管理系统是一款基于ThinkPHP6.0开发的CMS系统，可用来快速建设一个PC，手机，微信都可以访问的企业站点。以代码开源、免费商业授权、系统易安装升级、界面功能简洁轻便、易上手、插件与模板在线升级安装、建站联盟扶持计划等优势为一体的CMS系统，我们一直致力于为企业建站提供一款国内真正免费开源的内容管理系统，放心使用。

HkCms免费开源，免授权，永久商用

近期有很多网友咨询我们，HkCms是否会收取授权费用，我们特此声明，**HkCms免费开源，免授权，永久商用**，在您遵守《[HkCms软件许可使用协议](#)》下可将HkCms开源内容管理系统用于商业。

从ShuipFCms到LvYeCms，再到HkCms，我们从未停下为建站行业提供优秀的免费开源内容管理系统的初心，我们希望将免费开源进行到底。

各位网友务必在遵守国家法律法规的前提下使用HkCms开源内容管理系统，禁止使用HkCms开源内容管理系统进行任何违法犯罪的活动，使用HkCms开源内容管理系统过程中产生的任何版权纠纷及法律责任由使用方承担。一个优秀的开源内容管理系统离不开网友的建议、帮助和支持，我们希望与广大网友一起完善我们的开源内容管理系统，在使用过程中有任何的建议和思路都可以通过QQ群：[808251031](#) 向我们反馈。

软件架构

- 基于ThinkPHP6.0.9开发
- PHP版本： $\geq 7.2.5$
- MySql版本： ≥ 5.6

使用说明

前往安装使用章节下载并安装，安装后前往后台清理缓存。

主要目录架构说明

www 根目录

├─addons 插件模块目录

├─app 应用目录

| └─admin 后台目录

| └─api 接口

| └─common 通用公共目录

| └─index 首页目录

- | └─install 安装初始化目录
- | └─common.php 公共函数文件
- | └─event.php 事件定义文件
- | └─ExceptionHandler.php 应用异常处理
- | └─middleware.php 全局应用中间件
- | └─provider.php 容器定义文件
- | └─Request.php 请求类
- | └─service.php 系统服务定义文件
- └─config 配置目录
 - | └─app.php 应用配置
 - | └─cache.php 缓存配置
 - | └─console.php 控制台配置
 - | └─cookie.php Cookie配置
 - | └─database.php 数据库配置
 - | └─filesystem.php 文件磁盘配置
 - | └─lang.php 多语言配置
 - | └─log.php 日志配置
 - | └─middleware.php 中间件配置
 - | └─route.php URL和路由配置
 - | └─session.php Session配置
 - | └─trace.php Trace配置
 - | └─view.php 视图配置
- └─extend 扩展类库目录
- └─public WEB目录（对外访问目录）
 - | └─index.php 入口文件
 - | └─nginx_conf.conf nginx 参考配置文件
 - | └─.htaccess 用于apache的重写
- └─runtime 应用的运行时目录（可写，可定制）
- └─template 模板安装目录
- └─vendor Composer类库目录，核心底层代码
- └─.env 数据库等配置信息
- └─composer.json composer 定义文件
- └─LICENSE.txt 授权说明文件
- └─[REMADME.MD](#) README 文件
- └─think 命令行入口文件

线上交流

官方QQ群：808251031(HkCms交流②群)

官方微信群：联系微信号(cn-MrHua)，备注来源。

官方网站：<http://www.hkcms.cn>

社区问答：<http://ask.hkcms.cn>

下载与安装配置

- [下载](#)
- [虚拟主机安装](#)
- [Linux下宝塔安装](#)
- [phpEnv下安装](#)
- [phpStudy下安装](#)
- [伪静态](#)
- [安装系统](#)
- [调试模式与env文件](#)

下载

下载

下载

git仓库地址

https://gitee.com/Hk_Cms/HkCms

官网下载地址

<http://www.hkcms.cn/index/index/download.html>

更新日志

<https://www.hkcms.cn/index/index/uplogs>

虚拟主机安装

[虚拟主机必读](#)

[百度云虚拟主机](#)

[创业云虚拟主机](#)

虚拟主机必读

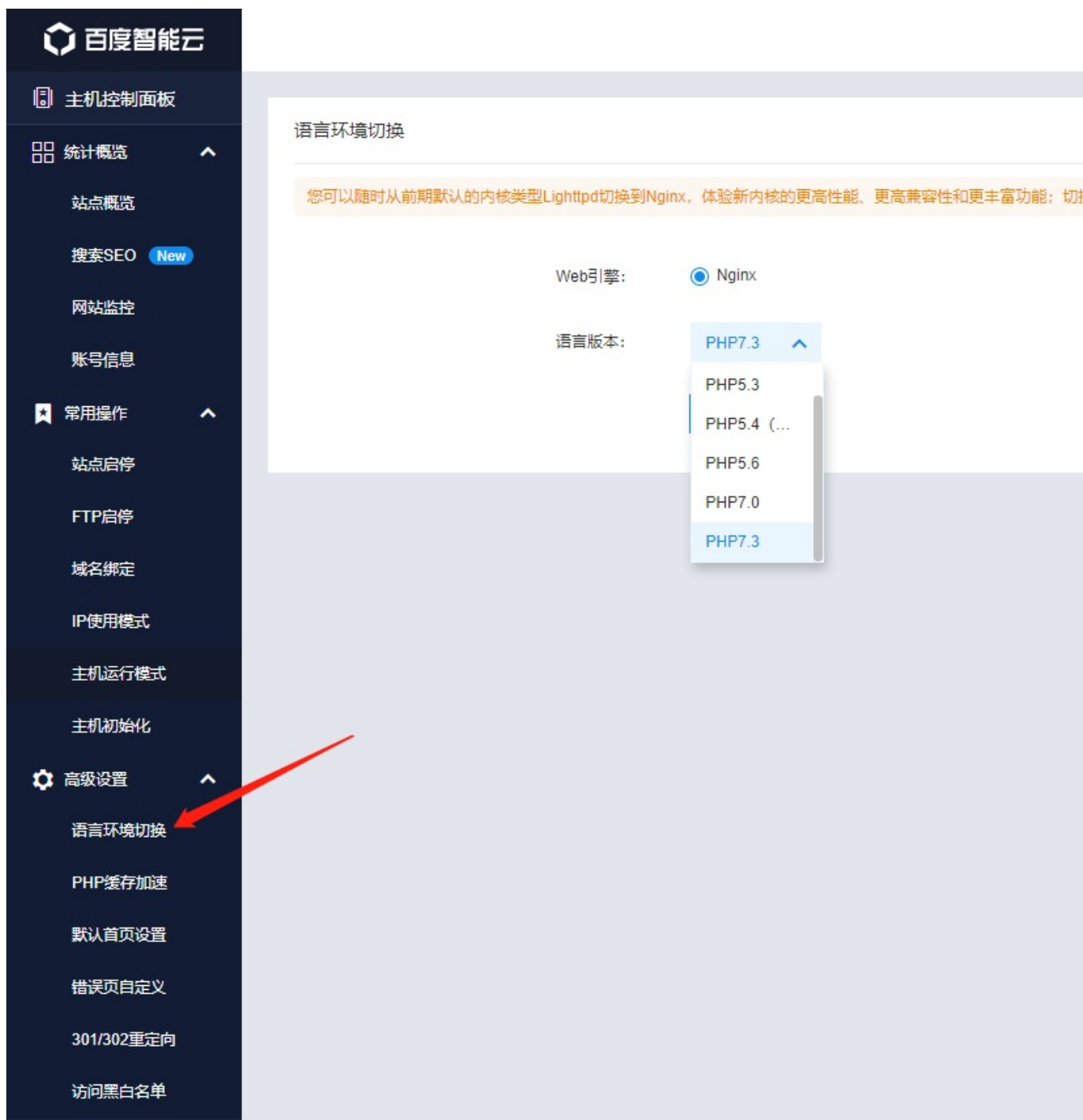
注意事项

HkCms需要支持设置运行目录的虚拟主机才能正常使用，目前测试可以用的虚拟主机包含在当前章节的虚拟主机，其他未测试虚拟主机并不保证完全适配。

百度云虚拟主机

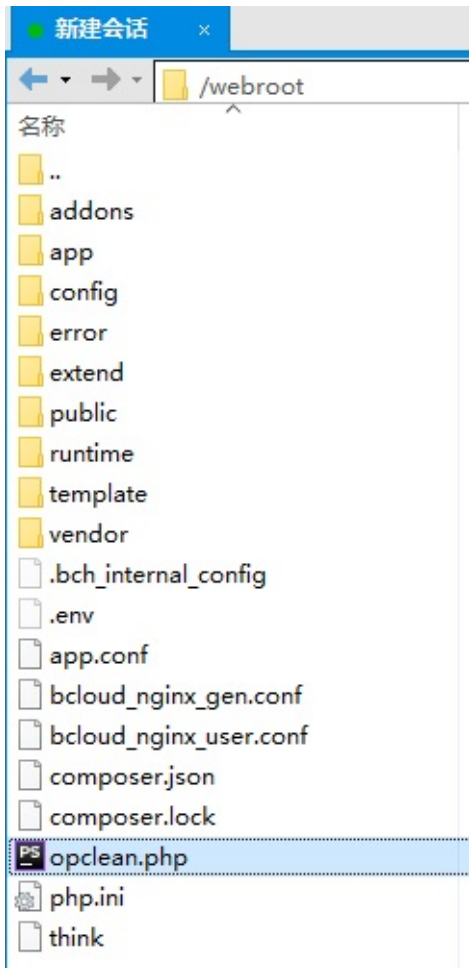
百度虚拟主机安装

- 切换PHP版本，PHP版本必须大于等于7.2



- 下载解压

1. 用FTP把项目存放到webroot目录下即可
2. 确保runtime目录有读写权限
3. webroot目录下的index.html删除掉。



- 配置nginx，在webroot目录下创建bcloud_nginx_user.conf文件，复制以下代码保存文件。这一步主要目的是配置到public目录下，而有些虚拟主机商有直接提供后台配置到public目录下，从而并不需要配置。

```
location / {
    root /home/bae/app/public/;

    if (!-e $request_filename){
        rewrite ^(.*)$ /index.php?s=$1 last; break;
    }
}
location ~ .php {

    root /home/bae/app/public/;

    set $path_info "";
    set $real_script_name $fastcgi_script_name;

    if ($fastcgi_script_name ~ "^(.+?\.\php)(/.)+$") {
        set $real_script_name $1;
        set $path_info $2;
    }
    fastcgi_pass 127.0.0.1:9000;
```

```
fastcgi_index index.php;  
fastcgi_param SCRIPT_FILENAME $document_root$real_script_name;  
fastcgi_param SCRIPT_NAME $real_script_name;  
fastcgi_param PATH_INFO $path_info;  
fastcgi_param SERVER_PORT 80;  
include fastcgi_params;  
}
```

- 配置完以上步骤，进行重启站点服务。



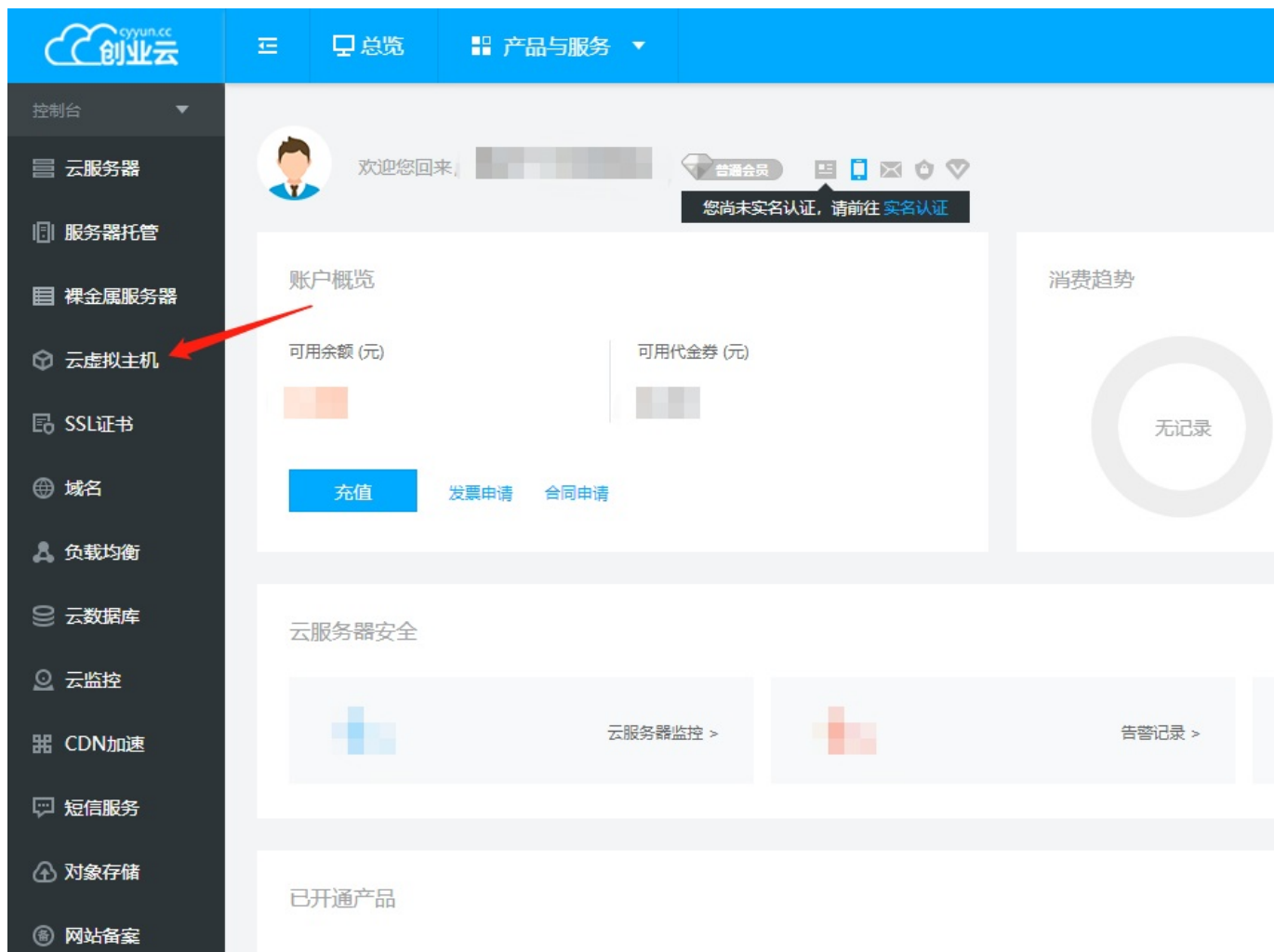
- 之后找到你的数据库连接信息，在安装页配置进去即可

创业云虚拟主机

创业云

网址：<https://www.cyyun.cc/>

1. 账号登录后，去到控制台，如下



2. 找到你的虚拟主机，点管理

类型/到期日期	操作
包年包月 2022-08-14 17:58:43	管理 续费 升级
包年包月 2024-06-14 17:11:26	管理 续费 升级

3. 找到下面的按钮，运行目录设置

The screenshot shows the '云虚拟主机详情' (Cloud Virtual Host Details) page. The left sidebar contains navigation options like '云服务器', '服务器托管', and '云虚拟主机'. The main content area is divided into sections: '网站信息' (Website Information), '常用操作' (Common Operations), '高级设置' (Advanced Settings), and '文件管理' (File Management). In the '文件管理' section, the '运行目录设置' button is highlighted with a red arrow and the text '来到这里点击' (Click here).

4. 选择public目录即可，如果选择不了，手动输入即可

注意：你需要先上传代码保证目录下有public目录



5. 点击确定即可

Linux下宝塔安装

安装宝塔

宝塔官方地址：<https://www.bt.cn/bbs/thread-19376-1-1.html>，按官方教程安装

安装PHP环境

去到应用商店安装PHP、nginx、mysql

旧版不在支持PHP7.1，请安装PHP 版本 ≥ 7.2 、MySQL ≥ 5.6

应用搜索 支持应用名称、字段模糊搜索

应用分类 全部 已安装 运行环境 系统工具 宝塔插件 专业版插件 企业版插件 第三方应用 一键部署 更新软件列表

宝塔运维直播答疑 企业版999元/年 精品软件推荐

专业版可以免费使用专业版插件，企业版可以免费使用专业版及企业版插件。 [立即购买](#)

软件名称	开发商	说明	价格	到期时间	位置	状态	首页显示	操作
Nginx 1.18.0	官方	轻量级，占有内存少，并发能力强	免费	--		▶	<input checked="" type="checkbox"/>	设置 卸载
Apache	官方	世界排名第一，快速、可靠并且可通过简单的API扩充	免费	--		▶	<input type="checkbox"/>	安装
MySQL 5.6.50	官方	MySQL是一种关系数据库管理系统!	免费	--		▶	<input checked="" type="checkbox"/>	设置 卸载
php PHP-8.0	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装
php PHP-7.4	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装
php PHP-7.3	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装
php PHP-7.2	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装
php PHP-7.1	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	设置 卸载
php PHP-7.0	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装
php PHP-5.6	官方	PHP是世界上最好的编程语言	免费	--		▶	<input type="checkbox"/>	安装

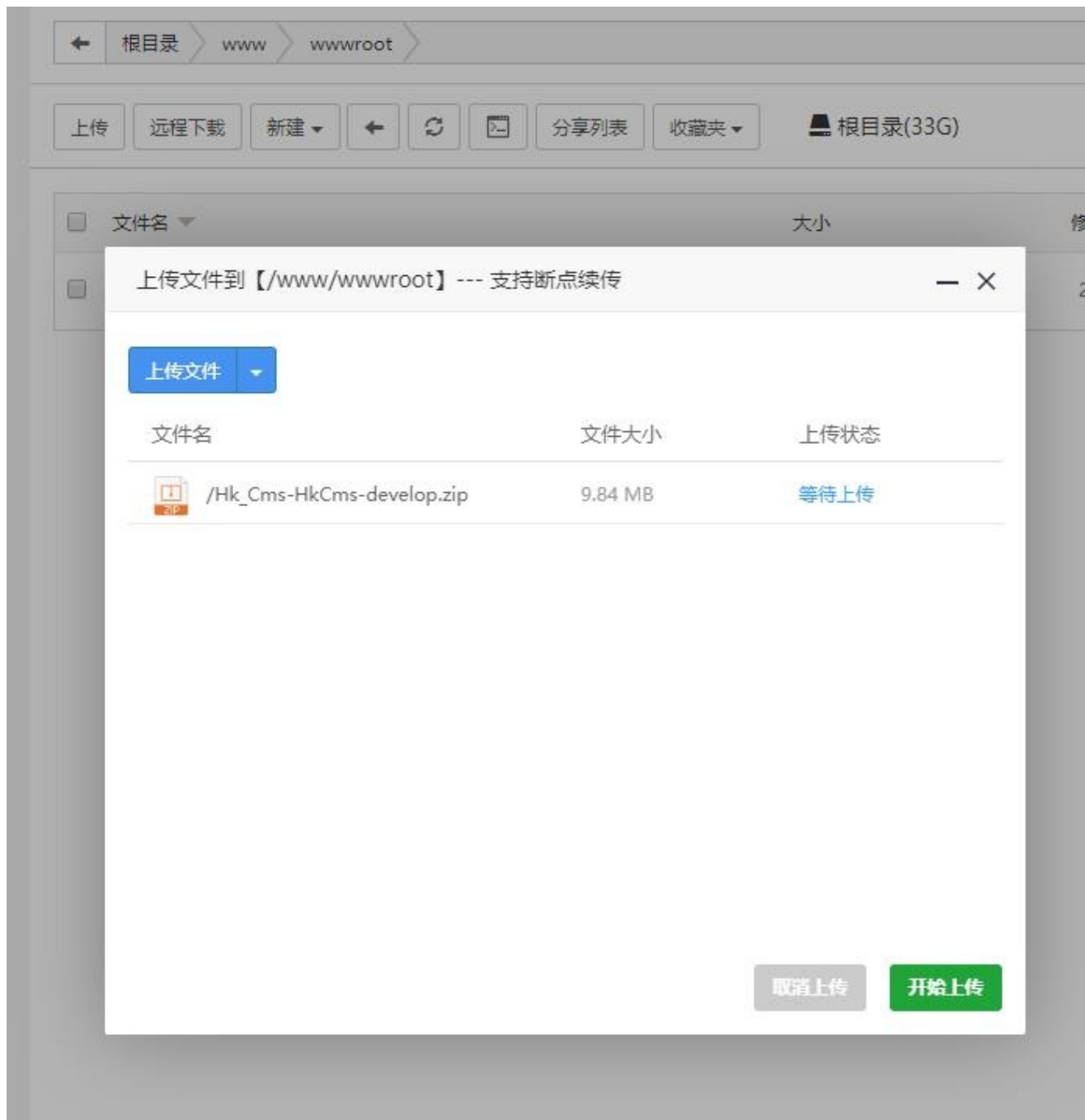
oms要求PHP版本必须大于等于7.1以上版本

宝塔部署

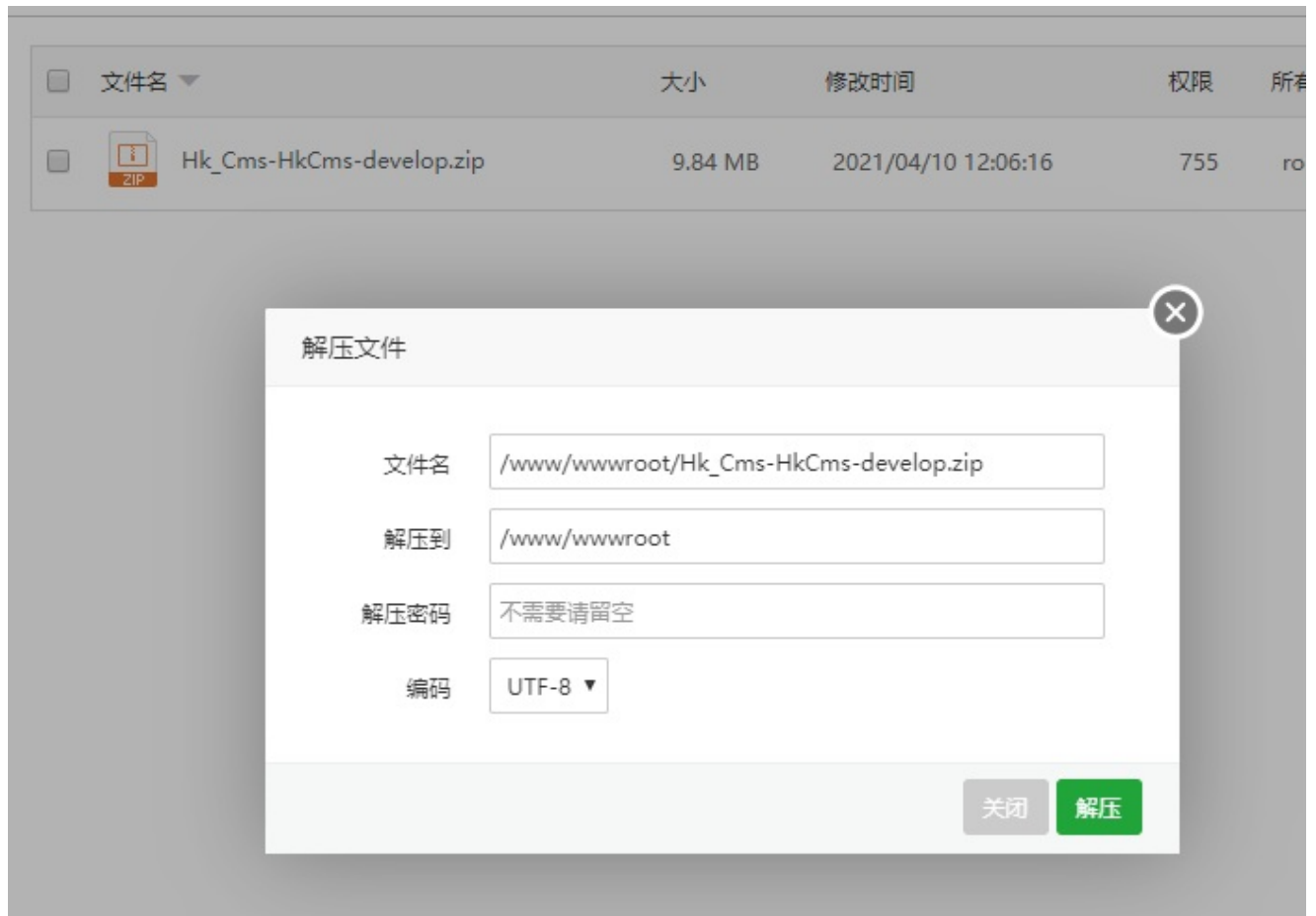
1. 下载解压

登录宝塔，在后台进入文件菜单，点击上传，上传下载好的压缩包。

步骤如下图所示，上传下载的压缩包。



解压



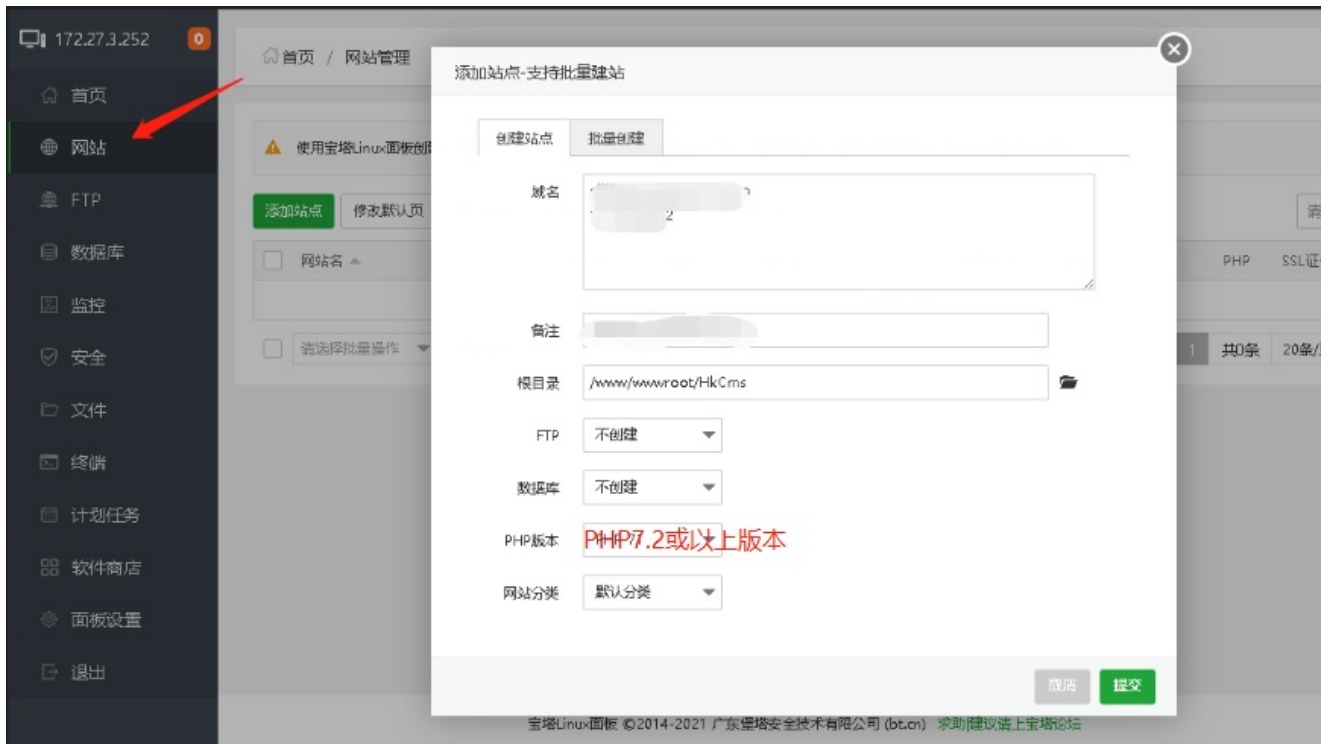
解压成功后文件夹名字HkCms，你可以根据实际情况重命名文件夹。



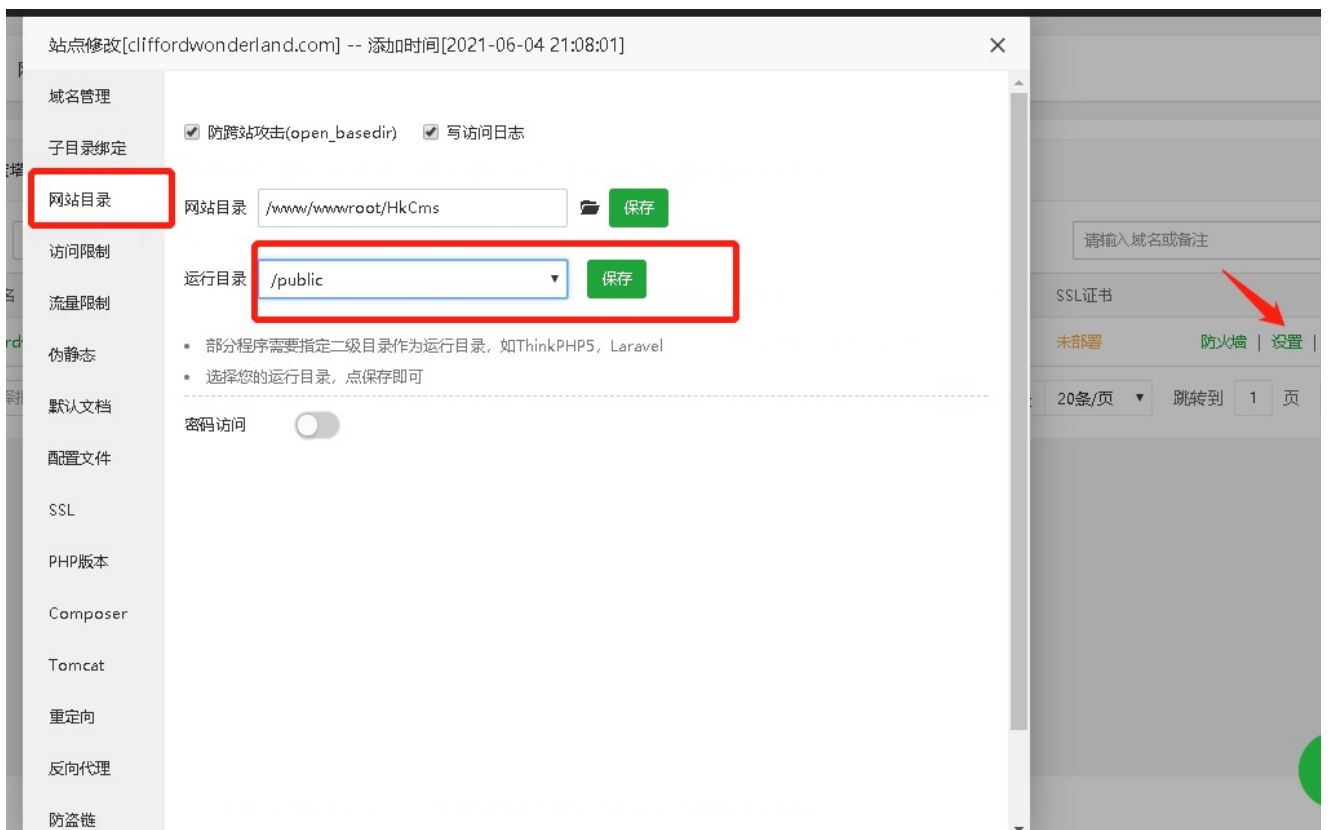
至此解压完成

2. 部署

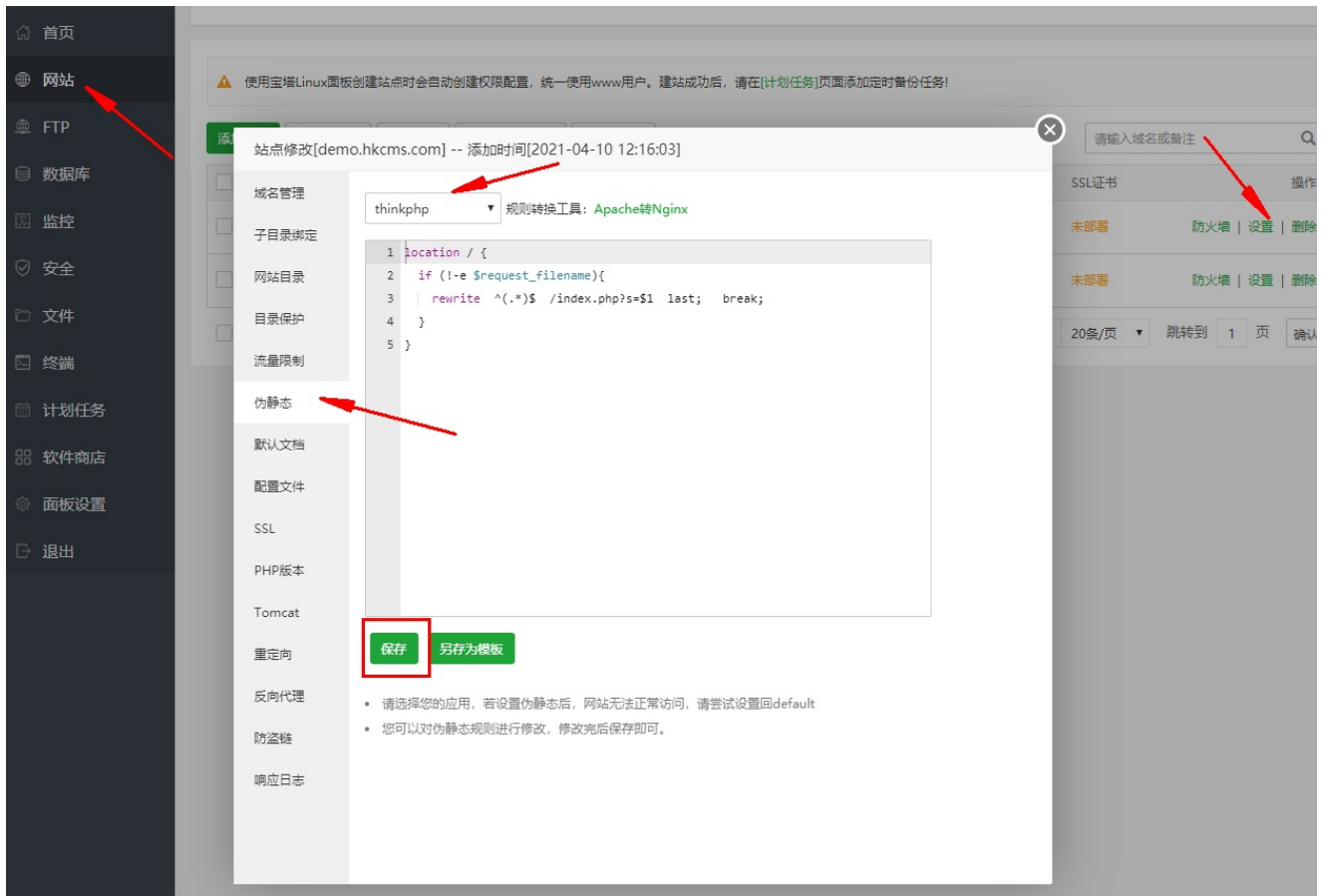
去到网站菜单、新增站点、路径指向到你项目的根目录，如下图所示。



添加网站后，我们需要把运行目录指向到public下，点击设置，出现如下界面，按图指向即可



如果安装的是nginx，需要设置伪静态



3. 文件权限

需要开启以下目录读写权限

```
.env // 配置文件 (必须)  
runtime/ // 运行是的缓存目录 (必须)
```

宝塔一键部署

安装宝塔

宝塔官方地址：<https://www.bt.cn/bbs/thread-19376-1-1.html>，按官方教程安装

教程步骤

(1) 登录宝塔后台找到软件商店-》搜索HkCms->点击一键部署，如下图所示



(2) 填写上你的域名，点击提交即可



(3) 创建成功看到以下信息，保存好你的数据库名、账号、密码，等会安装的时候用。点击访问站点链接



(4) 点击上面的链接、进入安装教程，在安装页写上上面图片中的数据库信息、点下一步即可安装成功。

数据库信息	
数据库服务器	127.0.0.1
数据库端口	3306
数据库用户名	linux_demo_com
数据库密码	czicxGcQPX
数据库名	linux_demo_com
数据库表前缀	hkcms_
网站配置	
网站名称	HkCms开源内容管理系统

输入刚刚弹出的用户信息

注意事项、说明

这是第一个一键部署版本，若遇到问题，请联系我们，我们将持续完善这块。

一键部署的版本通常会比官网更新会慢一点，在部署成功后在后台可以更新版本（自动更新需注意文件是否有限写入），也可以手动更新，前往官网下载对应补丁或更新包覆盖安装即可。

Windows下宝塔安装

安装宝塔

打开[链接](#)，下载windows宝塔，双击安装即可。

安装PHP环境

登录宝塔后台，会弹出推荐安装的集成环境，这里选择了Nginx环境。环境要求PHP \geq 7.2，MySQL \geq 5.6

安装系统

1. 上传HkCms安装包，并解压文件



2. 新建站点



3. 配置目录

站点创建后，点击设置，配置运行目录



4. 开启伪静态

下拉选择后点保存



5. 更改Nginx默认的path_info设置

默认的设置是没法访问 `index.php/ceshi/ceshi` 这种地址，配置文件路径如下，注意看你安装的位置以及使用的PHP版本，例如PHP7.3是73.conf，以下仅供参考

```
D:/BtSoft/nginx/conf/php/72.conf
```

配置文件内容如下

```

location ~ /\.php(.*)$ {
    try_files $uri =404;
    fastcgi_pass 127.0.0.1:20072;
    fastcgi_index index.php;

    fastcgi_split_path_info ^((?U).+\.php)(/?.+)$;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;

    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}

```

修改后的配置，fastcgi_pass不用动，看效果图，主要是删掉那4行，增加以下代码

```

location ~ /\.php(.*)$ {
    try_files $uri =404;
    fastcgi_pass    127.0.0.1:20072;
    fastcgi_index  index.php;

    set $fastcgi_script_name2 $fastcgi_script_name;
    if ($fastcgi_script_name ~ "^(.+\.php)(/?.+)$") {
        set $fastcgi_script_name2 $1;
        set $path_info $2;
    }
    fastcgi_param    PATH_INFO $path_info;
    fastcgi_param    SCRIPT_FILENAME    $document_root$fastcgi_script_name2;
    fastcgi_param    SCRIPT_NAME    $fastcgi_script_name2;

    include          fastcgi_params;
}

```

效果图

```

G 72.conf x
1 Location ~ /\.php(.*)$ {
2     try_files $uri =404;
3     fastcgi_pass    127.0.0.1:20072;
4     fastcgi_index  index.php;
5
6     # 注释4行
7     #fastcgi_split_path_info    ^((?U).+\.php)(/?.+)$;
8     #fastcgi_param    SCRIPT_FILENAME    $document_root$fastcgi_script_name;
9     #fastcgi_param    PATH_INFO    $fastcgi_path_info;
10
11     #fastcgi_param    SCRIPT_FILENAME    $document_root$fastcgi_script_name;
12
13     set $fastcgi_script_name2 $fastcgi_script_name;
14     if ($fastcgi_script_name ~ "^(.+\.php)(/?.+)$") {
15         set $fastcgi_script_name2 $1;
16         set $path_info $2;
17     }
18     fastcgi_param    PATH_INFO $path_info;
19     fastcgi_param    SCRIPT_FILENAME    $document_root$fastcgi_script_name2;
20     fastcgi_param    SCRIPT_NAME    $fastcgi_script_name2;
21
22     include          fastcgi_params;
23 }

```

6. 重启服务



然后输入你的域名即可体验HkCms系统的安装了。

phpEnv下安装

安装phpEnv

下载地址：<https://www.phpenv.cn/download.html>

部署流程

phpEnv 新版下载下来默认不是apache，进行如下设置



之后在“网站管理”->新增站点来到如下图页面，路径选择到public下，保存。返回主页启动服务即可。



phpStudy下安装

安装phpStudy

下载地址：<https://www.xp.cn/download.html>

部署流程

第一步启动apache、mysql服务，然后去到网站菜单新建网站保存即可。

套件	状态	操作
Apache2.4.39	▶ (A)	停止 重启 配置
FTP0.9.60	■ (A)	启动 重启 配置
MySQL5.7.26	■ (A)	启动 重启 配置
Nginx1.15.11	■ (A)	启动 重启 配置

****根目录选择到public目录下。****



常见问题

1. 默认创建网站后，如果没有填写伪静态，会覆盖默认的伪静态。文件路径为 `public` 下的 `.htaccess` 文件，导致异常。可复制以下代码到 `.htaccess` 文件。

```
<IfModule mod_rewrite.c>
  Options +FollowSymLinks -Multiviews
  RewriteEngine On

  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^(.*)$ index.php [L,E=PATH_INFO:$1]
</IfModule>
```

伪静态 (url重写)

apache 伪静态

在public下新建.htaccess文件，复制以下代码。注意：public目录下是自带该文件的。

```
<IfModule mod_rewrite.c>
  Options +FollowSymlinks -Multiviews
  RewriteEngine On

  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^(.*)$ index.php [L,E=PATH_INFO:$1]
</IfModule>
```

nginx 伪静态 参考

1. 宝塔服务器参考[Linux下宝塔安装](#)

找到nginx网站配置文件，打开后在合适位置增加以下配置

```
#伪静态配置，重写规则引用，修改后将导致伪静态规则失效
location / {
    if (!-e $request_filename){
        rewrite ^(.*)$ /index.php?s=$1 last; break;
    }
}
```

完整nginx配置参考

```
server
{
    listen 80;
    server_name 192.168.0.118;
    index index.php index.html;
    root /www/data/HkCms/public;

    #PHP引用配置
    location ~ [^/]\.php(/|$)
    {
        try_files $uri =404;

        # Socket管道方式
        # fastcgi_pass unix:/tmp/php-cgi-71.sock;
```

```

# tcpip方式, 端口根据PHP-FPM配置文件
fastcgi_pass 127.0.0.1:10711;
fastcgi_split_path_info ^((?U).+\.php)(/?.+)$;

fastcgi_index index.php;

# 引用fastcgi_param参数
include fastcgi.conf;

set $real_script_name $fastcgi_script_name;
if ($fastcgi_script_name ~ "^(.+?\.php)(/?.+)$") {
    set $real_script_name $1;
    set $path_info $2;
}
fastcgi_param SCRIPT_FILENAME $document_root$real_script_name;
fastcgi_param SCRIPT_NAME $real_script_name;
fastcgi_param PATH_INFO $path_info;
}

#伪静态配置, 重写规则引用, 修改后将导致伪静态规则失效
location / {
    if (!-e $request_filename){
        rewrite ^(.*)$ /index.php?s=$1 last; break;
    }
}

#禁止用户上传目录下所有.php文件的访问, 提高安全性
location ~ ^/uploads/.*\.(php|php5)$ {
    deny all;
}

#禁止访问的文件或目录
location ~ ^/(\.user.ini|\.htaccess|\.git|\.svn|\.project|LICENSE|README.md)

{
    return 404;
}

location ~ .*\.(\.gif|jpg|jpeg|png|bmp|swf)$
{
    expires 30d;
    error_log /dev/null;
    access_log off;
}

location ~ .*\.(\.js|css)?$
{
    expires 12h;
    error_log /dev/null;
    access_log off;
}

```


伪静态 (url重写)

```
}

# 访问日志
access_log /www/wwwlogs/192.168.0.118.log;
# 错误日志
error_log /www/wwwlogs/192.168.0.118.error.log;
}
```

IIS参考

主要添加的内容

```
<rewrite>
  <rules>
    <rule name="OrgPage" stopProcessing="true">
      <match url="^(.*)$" />
        <conditions logicalGrouping="MatchAll">
          <add input="{HTTP_HOST}" pattern="^(.*)$" />
          <add input="{REQUEST_FILENAME}" matchType="IsFile"
negate="true" />
          <add input="{REQUEST_FILENAME}" matchType="IsDirect
ory" negate="true" />
        </conditions>
        <action type="Rewrite" url="index.php/{R:1}" />
      </rule>
    </rules>
  </rewrite>
```

添加的位置

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="OrgPage" stopProcessing="true">
          <match url="^(.*)$" />
            <conditions logicalGrouping="MatchAll">
              <add input="{HTTP_HOST}" pattern="^(.*)$" />
              <add input="{REQUEST_FILENAME}" matchType="IsFile"
negate="true" />
              <add input="{REQUEST_FILENAME}" matchType="IsDirect
ory" negate="true" />
            </conditions>
            <action type="Rewrite" url="index.php/{R:1}" />
          </rule>
```

伪静态 (url重写)

```
        </rules>
    </rewrite>
    <handlers>
    </handlers>
</system.webServer>
</configuration>
```

安装系统

安装系统

通过以上章节任意安装环境搭建好后，直接访问配置好的域名或IP，即可进入安装页面。

前言（重要）

在进入安装页面前，请保证网站目录runtime有读写权限。

安装界面

输入你配置的域名访问，即可出现如下界面，阅读完协议没问题就可以下一步了。



检查环境

目录说明

- .env 文件，在安装时，你填写的数据库配置将写入到这里。需要读写权限，位于根目录下。
- template\，模板目录，在线安装模板时需要开启读写权限。
- addons\，插件目录，在线安装插件需要开启的读写权限。

- public\uploads\, 文件上传目录。
- public\static\module\, 模板静态目录。
- public\static\addons, 插件静态目录

配置系统

在配置系统这里，可以选择勾选测试数据，有完善的栏目数据、产品模型等演示数据。

管理员信息	
管理员帐号	<input type="text" value="admin"/>
管理员密码	<input type="password" value="请输入6位或6位以上的密码"/>
重复密码	<input type="password"/>
邮箱	<input type="text" value="admin@admin.com"/>
测试数据	<input type="checkbox"/> 默认测试数据，了解HkCms开源内容管理系统!

调试模式与env文件

env文件

env文件主要包含数据库配置与调试模式等信息，在项目的根目录可以看到，打开后会看到以下内容

```
-- 关闭调试模式,
APP_DEBUG = false,
-- 开启调试模式
APP_DEBUG = true

[APP]
DEFAULT_TIMEZONE = Asia/Shanghai

[DATABASE]
TYPE = mysql
HOSTNAME = 127.0.0.1 -- 数据库地址
DATABASE = hkcms -- 数据库名称
USERNAME = root -- 数据库用户名
PASSWORD = root -- 数据库密码
HOSTPORT = 3306 -- 数据库端口
CHARSET = utf8
PREFIX = hkcms_ -- 数据库表前缀
DEBUG = true -- 开启数据库调试模式

[LANG]
default_lang = zh-cn
```

注意：HkCms_v2.2.1.20220603 版本起不在env里面开启调试模式

- 新版开启方式

在后台-》配置中心-》站点配置，找到开发者模式选择开启即可

第三方代码电脑端：
代码自动放在前台网页底部，无需手动添加，常用于站点统计、百度商

第三方代码手机端：
代码自动放在前台网页底部，无需手动添加，常用于站点统计、百度商

开发者模式： 开启 关闭

站点状态： 开启 关闭

- 数据库执行日志依然是在env文件里面DEBUG属性，如上面代码开启所示。

重新安装与安装文件说明

安装文件标识

系统安装完后会生成安装标识，路径如下。删除该文件即可重新安装

```
/app/install/install.lock
```

说明

安装系统后你可以重命名public目录下的install.php入口文件，或直接删除即可。

后台使用帮助

- [后台入口文件](#)
- [首页面板](#)
- [站点配置](#)
 - [邮件配置](#)
 - [附件配置](#)
 - [应用中心配置](#)
 - [前台URL规则配置](#)
- [模型管理](#)
- [栏目管理](#)
- [内容管理](#)
- [站点模块](#)
- [应用中心](#)

后台入口文件

后台入口文件

后台入口文件默认是 `public` 目录下的 `admin.php`。访问后台时加上 `admin.php` 访问，您可根据需要，重命名后台入口文件。

重命名后需要在 `config/app.php` 文件中修改配置，配置如下，如果没有以下配置那么该版本无需理会。

```
// 入口文件绑定, 无需写index
'app_file' => [
    // 应用名称 => 文件名, 无需写后缀, 例如: 'admin'=>'demo', 那么对应的文件名为demo.php
    'admin' => 'admin',
    'api' => 'api',
],
```

首页面板

首页面板

首页界面

The screenshot displays the HkCms admin dashboard. On the left is a dark sidebar with navigation options: 内容管理, 栏目管理, 站点模块, 设置 (with sub-items: 会员管理, 配置中心, 附件管理, 模型管理, 权限管理), 更多, and 应用中心. The main content area has a top bar with '控制台', '缓存', '语言', '首页', and a user profile 'admin'. Below this are three sections: '快速导航' with a '+ 添加' button, '系统信息' with a table of system details, and '服务器信息' with a table of server specifications.

站点名称	HkCms开源内容管理系统	当前版本	v2.2.3.221012
系统更新	已是最新版	商业授权	免授权, 永久商用

服务器系统	WINNT	服务器域名	hkcmsdev.com
服务器环境	Apache/2.4.43 (Win64) OpenSSL/1.1.1g mod_fcgid/2.3.10-dev	PHP 版本	7.4.28
数据库版本	5.7.28	GD 版本	bundled (2.1.0 compatible)

配置中心

[站点配置](#)

[SEO设置](#)

[SEO设置-静态生成](#)

[文档属性](#)

站点配置

站点配置

配置站点功能，配置站点信息。如下图，可根据需要调整或新建扩展配置。

The screenshot displays the '站点配置' (Site Configuration) page in the HKCMS admin system. The left sidebar contains navigation options like '内容管理', '栏目管理', '站点模块', and '配置中心'. The main content area is divided into tabs: '基础配置', '邮件配置', '附件配置', '多语言', '高级配置', and '分组配置'. The '基础配置' tab is active, showing various site settings such as '网站名称', '首页标题', '首页关键字', '首页描述', '备案号', '公安网备案号', 'CDN', '网站LOGO', and '地址栏图标'. To the right of these settings is a list of template variables: `{${site.title}}`, `{${site.home_title}}`, `{${site.keyword}}`, `{${site.description}}`, `{${site.icp}}`, `{${site.psrn}}`, `{${site.cdn}}`, `{${site.logo}}`, and `{${site.favicon}}`. A red box highlights this list, with a red annotation '可在模板中直接调用' (Can be directly called in the template).

邮件配置

邮件配置

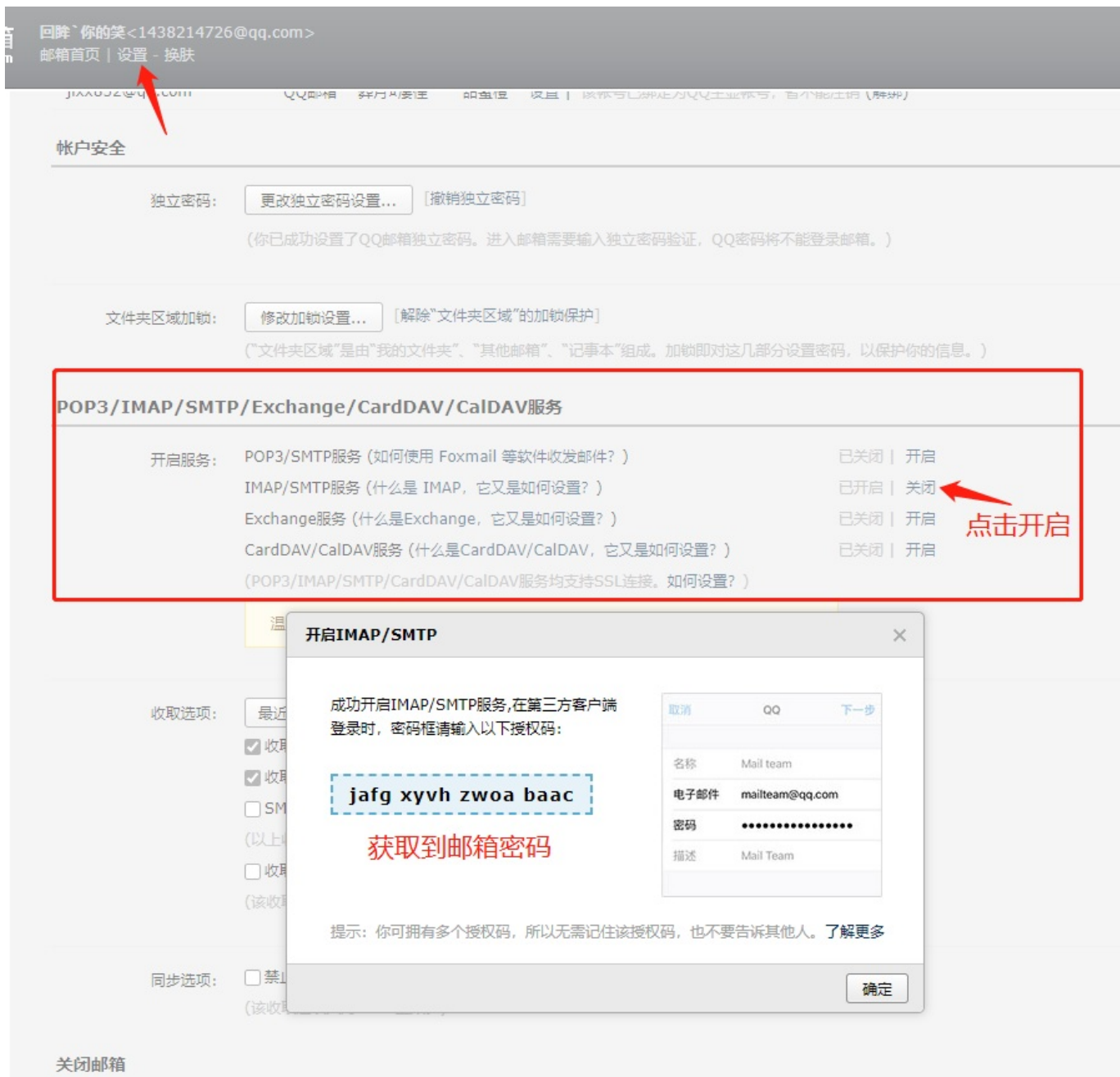
基础配置 **邮件配置** 附件配置 多语言 高级配置 分组配置 +

邮箱开关:	<input type="radio"/> 开启 <input checked="" type="radio"/> 关闭		{site.mail_on}
邮件发送模式:	SMTP	?	{site.mail_type}
SMTP服务器:	smtp.qq.com	?	{site.mail_server}
SMTP发送端口:	465	?	{site.mail_port}
发件人邮箱:	(填写完整邮箱)	?	{site.mail_from}
发件人名称:	(发件人名称标题)	?	{site.mail_fname}
SMTP验证:	SSL	?	{site.mail_auth}
SMTP用户名:	(填写完整邮箱)	?	{site.mail_user}
SMTP密码:	(密码)	?	{site.mail_password}

示例

以1438214726@qq.com 邮箱为例

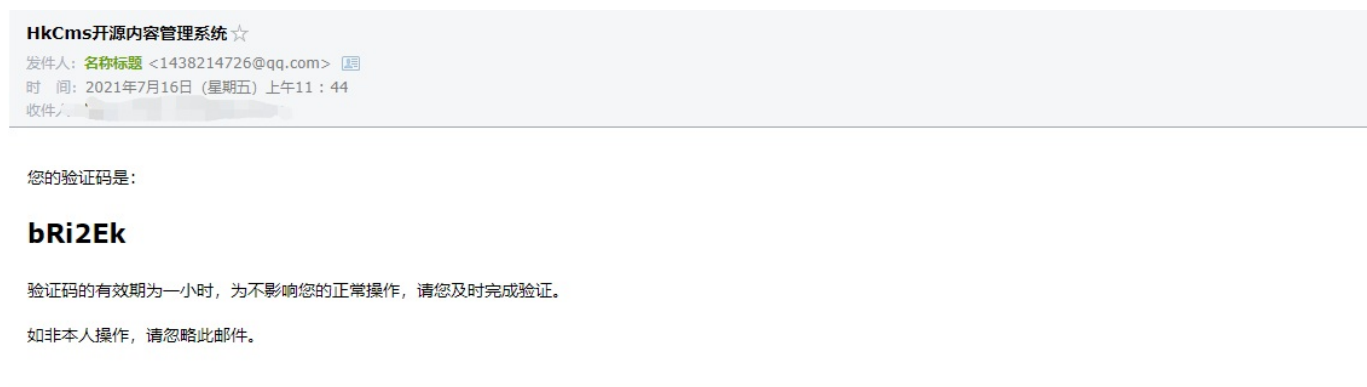
1. 去到邮箱账号设置



2. 填写到邮件配置

基础配置	邮件配置	附件配置	高级配置	分组配置	+
邮件发送模式:	<input type="text" value="SMTP"/>	<input type="button" value="i"/>			{site.mail_type}
SMTP服务器:	<input type="text" value="smtp.qq.com"/>	<input type="button" value="i"/>			{site.mail_server}
SMTP发送端口:	<input type="text" value="465"/>	<input type="button" value="i"/>			{site.mail_port}
发件人邮箱:	<input type="text" value="1438214726@qq.com"/> 你的邮箱	<input type="button" value="i"/>			{site.mail_from}
发件人名称:	<input type="text" value="名称标题"/>	<input type="button" value="i"/>			{site.mail_fname}
SMTP验证:	<input type="text" value="SSL"/>	<input type="button" value="i"/>			{site.mail_auth}
SMTP用户名:	<input type="text" value="1438214726@qq.com"/>				{site.mail_user}
SMTP密码:	<input type="text" value="jafgyvhzwoabaac"/>				{site.mail_password}

3. 效果预览



附件配置

附件配置

基础配置
邮件配置
附件配置
多语言
高级配置
分组配置
+

上传附件类型:	<input type="text" value="jpg gif png bmp jpeg ico zip gz rar iso txt doc xls ppt wps swf mpg mp3 rm rmvb wmv wma wav mid mov mp4"/>	?	{\$site.file_type}
上传附件大小:	<input type="text" value="10"/> <input type="text" value="MB"/>	?	{\$site.file_size}
上传地址:	<input type="text" value="/common/upload"/>	?	{\$site.upload_url}
附件CDN:	<input type="text"/>	?	{\$site.cdn_url}
保存格式:	<input type="text" value="/uploads/{year}{month}{day}/{md5}{suffix}"/>	?	{\$site.savename}
开启分片:	<input type="radio"/> 开启 <input checked="" type="radio"/> 关闭	?	{\$site.chunk}
分片大小:	<input type="text" value="2"/> <input type="text" value="MB"/>	?	{\$site.chunk_size}

开启分片

1. 开启分片，默认对大文件系统是没有开启分片上传的，如果需要在这里开启。
2. 分片大小是指每个分片最大上传大小，如果文件小于分片，依然是普通上传。

保存路径格式

1. 保存的文件格式，默认路径是在public/uploads文件夹下，建议保存在这个文件夹下，命名规则支持以下变量，注意suffix后缀是必须的。

```

'{year}' => 年份,
'{month}' => 月份,
'{day}' => 日,
'{hour}' => 小时,
'{minute}' => 分,
'{second}' => 秒,
'{md5}' => md5值,
'{sha1}' => sha1值,
'{random}' => 随机16位字符,
'{suffix}' => 后缀是必须，一般写在后面，不要变更位置,
    
```


例如：/uploads/{year}-{month}-{day}/{random}{suffix}

上传附件与类型

附件类型使用 "|" 符合分隔开即可

文件大小设置，请确保您的PHP配置支持设定的文件上传数值。

应用中心配置

应用中心配置

在这里设置应用中心账号以访问在线的插件与模板。当然你也可以去到应用中心直接输入账号密码登录也是可以的。

配置项	值	备注
应用中心账号:	应用中心登录账号	前往注册
应用中心密码:	应用中心登录账号的密码	前往注册
后台模板:	adminlte	{site.admin_theme}
前台模板:	default	{site.index_theme}
后台语言:	中文	{site.admin_lang}
前台语言:	中文	{site.index_lang}
URL 模式:	动态	{site.url_mode}

[提交](#) [重置](#)

前台URL规则配置

动态地址说明

动态访问的url地址是固定的，通过变化参数来访问栏目、文档，并且使用动态地址则不需要配置URL重写。

地址示例：<http://网址/index.php/index/lists?catname=news>

伪静态说明：

伪静态使用前请先配置URL重写：[伪静态（url重写）](#)

伪静态开启后效果：

<http://网址/abc/>，<http://网址/abc/1.html>

伪静态是可以把文档、栏目按照目录结构一样访问，实现静态化，像是真实存在的目录，实际上并不存在该目录文件，这样做的好处是你每个文章、栏目都是一个独立的URL地址，对搜索引擎友好。

配置URL规则

前往 [站点配置](#) 找到 [高级配置](#) 将URL模式改为 [伪静态](#)。

如下图所示，默认带四个URL规则定义

```
search/index: 搜索页地址
guestbook/index: 留言表单提交地址
index/lists: 栏目地址
index/show: 详情页
```

URL 模式: 伪静态 动态

{site.url_mode}

URL 伪静态:

URL地址

规则

search/index

/search\$.html



guestbook/index

/guestbook\$.html



index/lists

/:catname/\$/,/:catname/list_;p



index/show

/:catname/:id\$.html



追加

[帮助](#)

提交

重置

URL地址唯一，一个URL地址支持多个规则，多个规则之间英文逗号分隔，在匹配时只会匹配其中一个规则。

结尾符号 \$ 说明

我们看到默认的规则里面有 \$ 符号，这个符号的意思是完全匹配，例如定义栏目目录名称规则为：

```
/:catname$
```

```
http://网址/new
```

加上\$符号后上面地址会匹配成功，而下面地址不会成功，如果没加\$，这两个地址都会匹配成功，通常用于多个规则区分开来，防止覆盖其他规则。

```
http://网址/new/info
```

注意：规则使用的是正则形式，像下划线_之类的关键字符注意使用反斜杠转义

支持的变量列表

变量格式：:变量名称或者 [:变量名称]，加中括号是可选变量

变量名称	变量说明	使用方式
catdir	取当前栏目的父级目录名称的值	:catdir 或 [:catdir]
catname	栏目目录名称，添加栏目时填写	:catname 或 [:catname]
catid	栏目ID	:catid 或 [:catid]
id	文章ID	:id或 [:id]
model	模型标识，栏目页：需要与栏目同时使用	:model 或 [:model]
year	栏目更新日期年，栏目与详情页需要栏目ID或目录同时存在	:year 或 [:year]
month	栏目更新日期日	:month或 [:month]
day	栏目更新日期天	:day 或 [:day]
page	页码	:page或 [:page]
lang	语言标识	:lang或 [:lang]

参考示例

1. 列表页 (index/lists)

假设栏目目录名称为：news，该栏目的父级为：dir，模型标识：archives

规则	效果	格式说明	备注

/:catname/\$,/:catname/list_:\$page\$.html	首页 : http://网址/news/ 分页 : http://网址/news/list_2.html	http://网址/栏目目录名称/	系统默认规则
/[:catdir]/:catname/\$	首页 : http://网址/dir/news/ 分页 : http://网址/dir/news/?page=2	http://网址/父目录/目录名称/	加了中括号表示 : 栏目没有父级时也可以访问, 为可选变量
/[:catdir]/:catname/\$,[:catdir]/:catname/:page\$.html	首页 : http://网址/dir/news/ 分页 : http://网址/dir/news/2.html	http://网址/父目录/目录名称/页码.html	同上, 只是这个加了页码变量, 2为当前页码, 注意文件后缀.html加在最后。
/list/:catname/\$	首页 : http://网址/list/news/	http://网址/list/目录名称/	使用固定值
/:model/:catname/\$	http://网址/archives/news/	http://网址/模型/目录名称/	
/:lang/:catname/\$	http://网址/zh-cn/news/ http://网址/en/news/	http://网址/语言/目录名称/	多语言

2. 文档详情页

假设栏目目录名称为 : news , 该栏目的父级为 : dir , 模型标识 : archives , 文章ID为 : 10

由于不同模型不同表, 所以栏目、模型必须存在一个才能定位文档

规则	效果	格式说明	备注
/:catname/:id\$.html	http://网址/news/10.html	http://网址/目录名称/文章ID.html	
/[:catdir]/:catname/:id\$.html	http://网址/dir/news/10.html	http://网址/父目录/当前目录/文章ID.html	没有父级时自动省略

SEO设置

seo设置 - URL配置

用于配置前台的url地址格式与标题显示格式，url链接地址自定义请前往：[前台URL规则配置](#) 了解界面示例

URL 模式: 动态 伪静态

注意：勾选伪静态后您的服务器必须配置URL重写才能正常访问

栏目标题格式: 栏目名称 - 网站名称

栏目名称 - 第N页 - 网站名称

栏目名称_第N页

栏目名称_第N页_网站名称

自定义:

栏目名称: \$name; 网站名称: \$site; 页码: \$page

内容标题格式: 内容标题

内容标题 - 网站名称

内容标题 - 栏目名称 - 网站名称

内容标题_网站名称

内容标题_栏目名称_网站名称

自定义:

栏目名称: \$name; 网站名称: \$site; 内容标题: \$title

前台入口文件: 显示(index.php) 隐藏(index.php)

注意：勾选隐藏后您的服务器必须配置URL重写才能正常访问

提交

重置

url模式 在这里切换动态或静态，静态需要配置服务器上的URL重写，如果是apaceh服务器则无需配置。

栏目标题格式 配置栏目列表页网页标题格式，除了上图4个标题格式外，你可以通过\$name、\$site、\$page来自定义标题格式。

内容标题格式 配置前台详情页的网页标题格式

伪静态的URL链接格式，系统提供了以下几个规则可以选择，若不能满足你可前往自定义提供的链接地址根据规则配置。

自定义规则配置参考：[前台URL规则配置](#)

URL 模式: 动态 伪静态

注意: 勾选伪静态后您的服务器必须配置URL重写才能正常访问

Url 格式: 当前目录

多语言/当前目录

父目录/当前目录

多语言/父目录/当前目录

模型/栏目

自定义, [点我前往](#)

SEO设置-静态生成

SEO设置-静态生成

用于生成静态页面，支持首页、栏目页、内容页生成。

开启

配置中心-》SEO设置-》URL配置，选择静态生成

URL 模式: 动态 伪静态 静态页面

1. 启用后栏目与文章在新增或修改都会按照规则生成静态页面。
2. 多语言情况下自动增加语言标识
3. 修改模板后请重新生成
4. 暂不支持非响应模板
5. 确保public目录下有写入的权限

栏目规则: 父目录/子目录 目录名称 自定义: [帮助](#)内容规则: 父级目录/子目录/文档ID 目录名称/文档ID 自定义: [帮助](#)栏目标题格式: 栏目名称 - 网站名称 栏目名称 - 第N页 - 网站名称 栏目名称_第N页 栏目名称_第N页_网站名称 自定义:

栏目名称: \$name; 网站名称: \$site; 页码: \$page

内容标题格式: 内容标题 内容标题 - 网站名称 内容标题 - 栏目名称 - 网站名称 内容标题_网站名称 内容标题_栏目名称_网站名称 自定义:

栏目名称: \$name; 网站名称: \$site; 内容标题: \$title

保存目录:

使用说明

• 栏目规则说明

1. 父目录/子目录

父目录: 如果生成的当前栏目有父级则取父级的目录名称

子目录: 取当前生成栏目的目录名称

例如: 栏目名称: 公司新闻, 目录名称: company_news, 父级栏目: 新闻中心, 父级栏目名称: news

生成新闻中心：/news/index.html

多语言效果：/zh-cn/news/index.html，/en/news/index.html，这里的zh-cn与en取决多语言里面的设置

生成公司新闻：/news/company_news/index.html

多语言效果：/zh-cn/news/company_news/index.html

2. 目录名称

同上一样，只是不含父级栏目

生成新闻中心：/news/index.html

生成公司新闻：/company_news/index.html

- 内容规则

基本与栏目规则一致，假设文档ID为1，所属栏目为公司新闻

生成文档静态：/company_news/1.html

- 保存目录

默认生成的静态文件是放在public目录下，若想在public目录下新建html文件夹然后生成的栏目页、内容页放置于此，那么在保存目录里面填写html即可。

自定义栏目规则与内容规则

目前支持的规则变量有：

```
[catdir]：父栏目栏目名称  
[list]：栏目名称  
[id]：文档ID  
[year]：文档发布的年分  
[month]：文档发布的月  
[day]：文档发布的日  
[page]：分页
```

例如栏目规则（父目录/子目录）完整规则如下：

```
/[catdir]/[list]/index.html|[catdir]/[list]/index_[page].html
```

我们可以看到中间有“|”符合隔开，意思是“|”左边的是首页规则右边的则是分页规则。效果如下：

第一页(首页)：采用左边的首页规则

```
/news/company_news/index.html
```

第二页: 开始采用右边的规则

```
/news/company_news/index_2.html
```

第三页

```
/news/company_news/index_3.html
```

所以在自定义规则时需要写好左边与右边的规则，注意注意注意使用“|”分隔

内置的规则参考：

```
/父级栏目/子栏目/index.html|父级栏目/子栏目/index_2.html  
/[catdir]/[list]/index.html|/[catdir]/[list]/index_[page].html
```

```
/栏目名称/index.html|/栏目名称/index_2.html  
/[list]/index.html|/[list]/index_[page].html
```

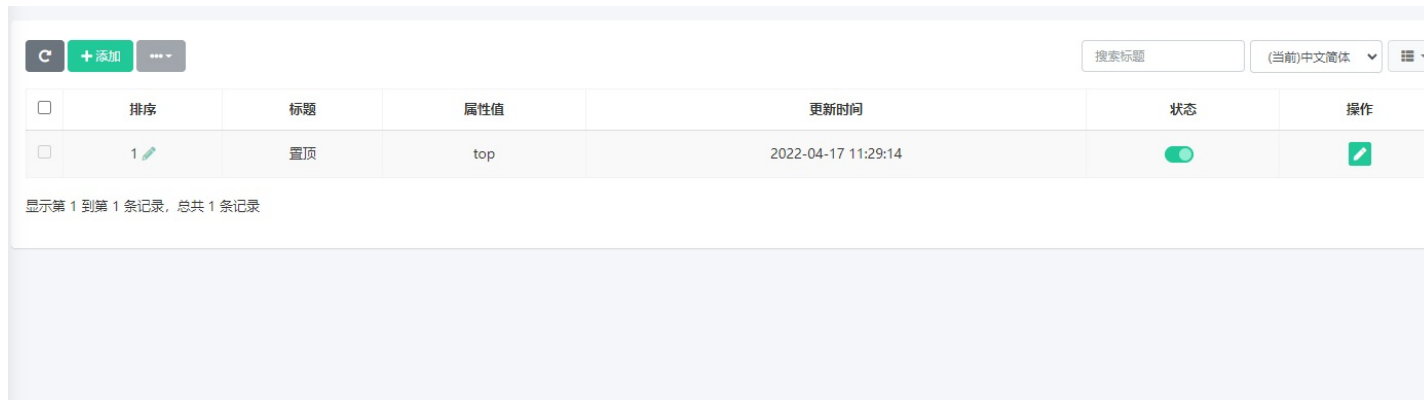
```
/父级栏目/子栏目/文档ID.html|父级栏目/子栏目/文档ID_1.html  
/[catdir]/[list]/[id].html|/[catdir]/[list]/[id]_[page].html
```

```
/栏目名称/文档ID.html|/栏目名称/文档ID_2.html  
/[list]/[id].html|/[list]/[id]_[page].html
```

文档属性

文档属性

文档属性支持多语言，用于给文档标志额外的属性，例如热门推荐、置顶，根据你的情况来新增文档属性



The screenshot shows a management interface for document attributes. At the top, there is a search bar labeled '搜索标题' and a language dropdown set to '(当前)中文简体'. Below this is a table with the following columns: '排序' (Sort), '标题' (Title), '属性值' (Attribute Value), '更新时间' (Update Time), '状态' (Status), and '操作' (Action). A single row is visible with the following data: '1' in the sort column, '置顶' (Pin) in the title column, 'top' in the attribute value column, '2022-04-17 11:29:14' in the update time column, and a green toggle switch in the status column. Below the table, it says '显示第 1 到第 1 条记录, 总共 1 条记录'.

排序	标题	属性值	更新时间	状态	操作
1	置顶	top	2022-04-17 11:29:14	<input checked="" type="checkbox"/>	

默认是置顶属性不能删除，新增界面如下。属性值必须是字母开头，可以是字母数字混合。



The '添加' (Add) form contains the following fields and options:

- 标题: 热门文章
- 属性值: hot
- 排序: 1
- 状态: 正常 隐藏

添加之后，在内容管理文章添加就可以看到刚刚新建的文档属性

添加

常规
SEO
其他

文档属性:

作者:

置顶
热门文章

浏览量:

发布时间:

内容模板: show.html X

如果你想把文档属性移到常规等位置可以去字段管理里面调整文档属性的位置，如下：

+ 添加
删除
设置

排序	字段	标题	字段类型	长度	默认值	主表字段	分组	允许投稿	是否排序	筛选	
1	seotitle	seo标题	text	200		主表	SEO	✔	✖	✖	✔
0	keywords	关键词	text	200		主表	SEO	✔	✖	✖	✔
0	description	描述	textarea	250		主表	SEO	✔	✖	✖	✔
10	author	作者	text	250	小编	主表	其他	✖	✖	✖	✔
18	thumb										✔
19	flags										✔
3	views										✔
2	publish_time	发									✔
1	show_tpl	内									✔
20	title										✔
6	content										✔
1	weigh										✔
0	status										✔

修改

字段类型: 动态下拉

分组: 常规 X

主表字段: 主表

字段: flags

标题:

配置: 数据源: URL 关联表

数据表: flags

内容管理添加界面

添加

常规 SEO 其他

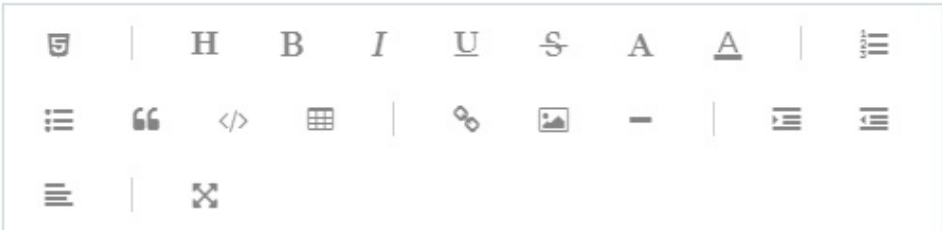
所属栏目: 卜公司荣誉

标题:

文档属性:

封面图:

内容:



站点模块

[V2.2.2版本 -](#)

[V2.2.3版本 +](#)

V2.2.2版本 -

站点模块

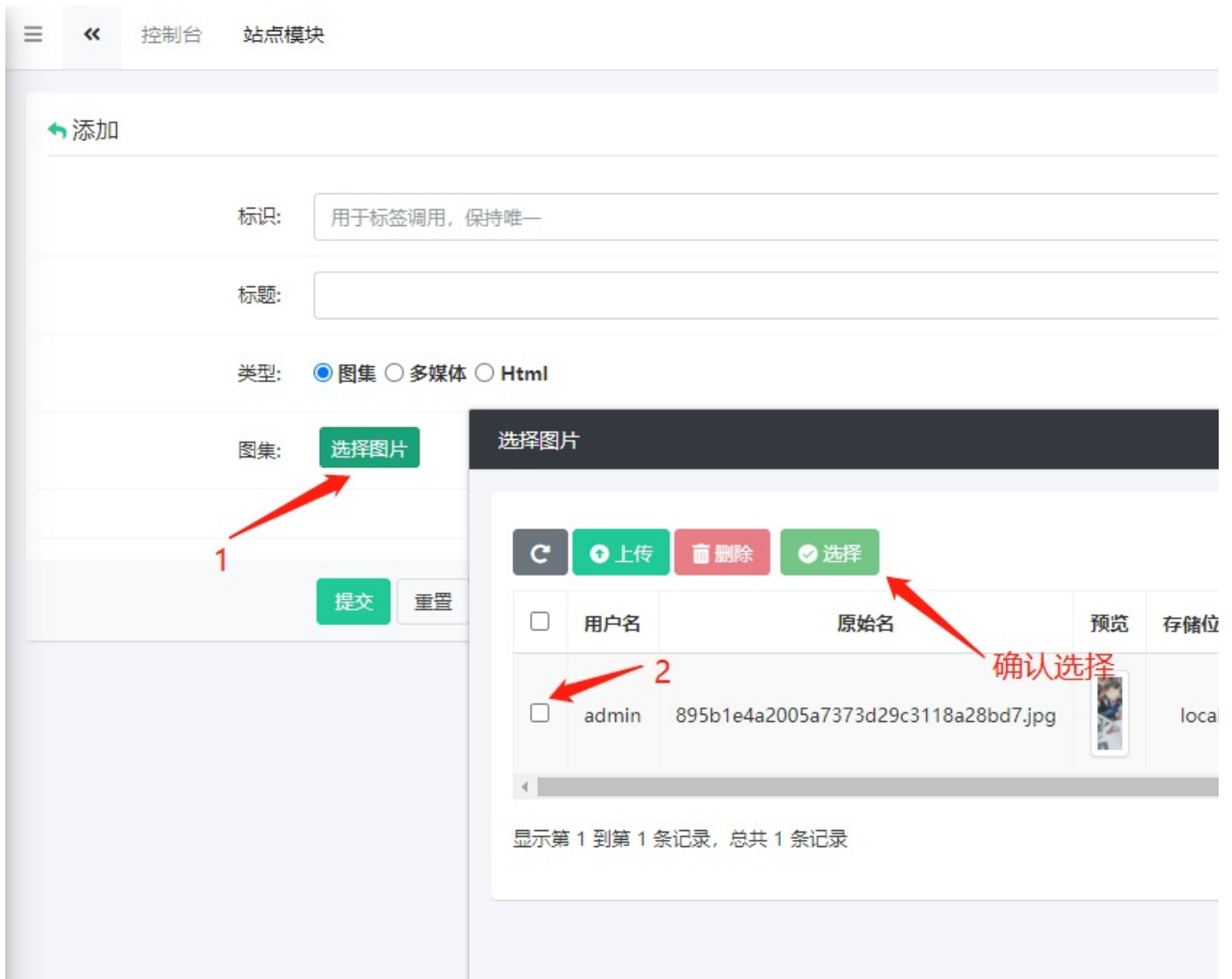
站点模块，可以创建图片、视频、html代码为一体的模块。通常用于广告位，或者站点轮播图。模板调用方式，请前往标签语法“站点模块标签”查看。

创建图集

1. 点击添加



2. 进入添加页，选择图片，勾选或上传图片，点击选择即可。



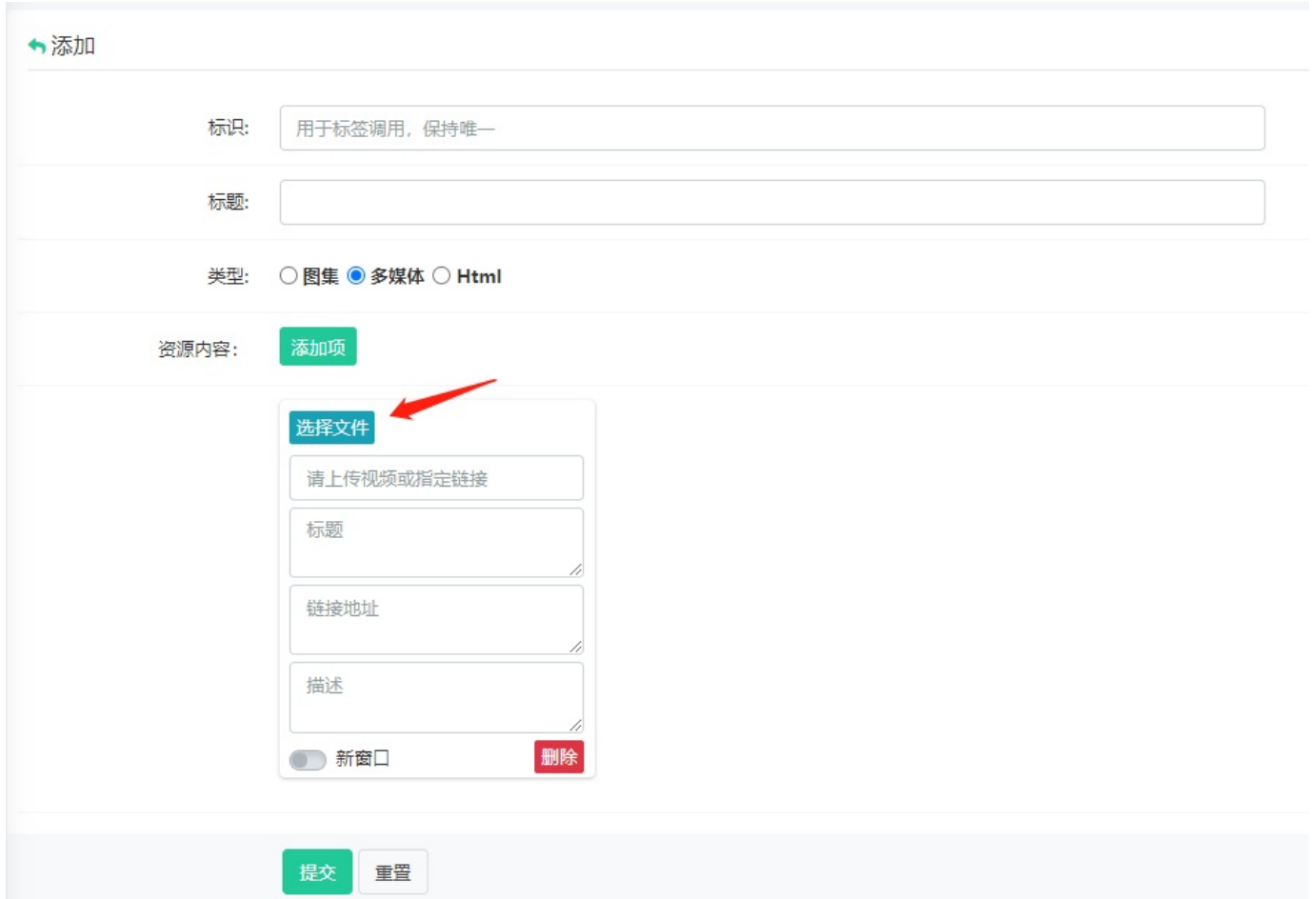
3. 选择后的效果



创建多媒体

1. 类型选择多媒体

支持选择本地的视频文件，你也可以指定链接地址。




← 添加

标识:

标题:

类型: 图集 多媒体 Html

资源内容:



标题

链接地址

描述

新窗口

创建HTML内容模块

如果图集，多媒体都不符合你，你可以完全插入HTML标签的形式创建站点模块



← 添加

标识:

标题:

类型: 图集 多媒体 Html

HTML 内容:

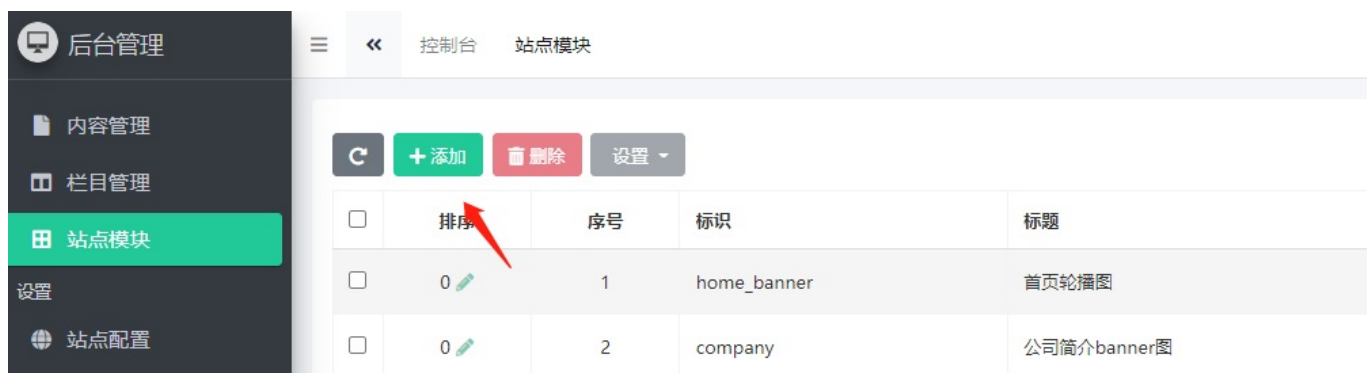
V2.2.3版本 +

站点模块

站点模块，可以创建图片、视频、html代码为一体的模块。通常用于广告位，或者站点轮播图。模板调用方式，请前往标签语法“站点模块标签”查看。

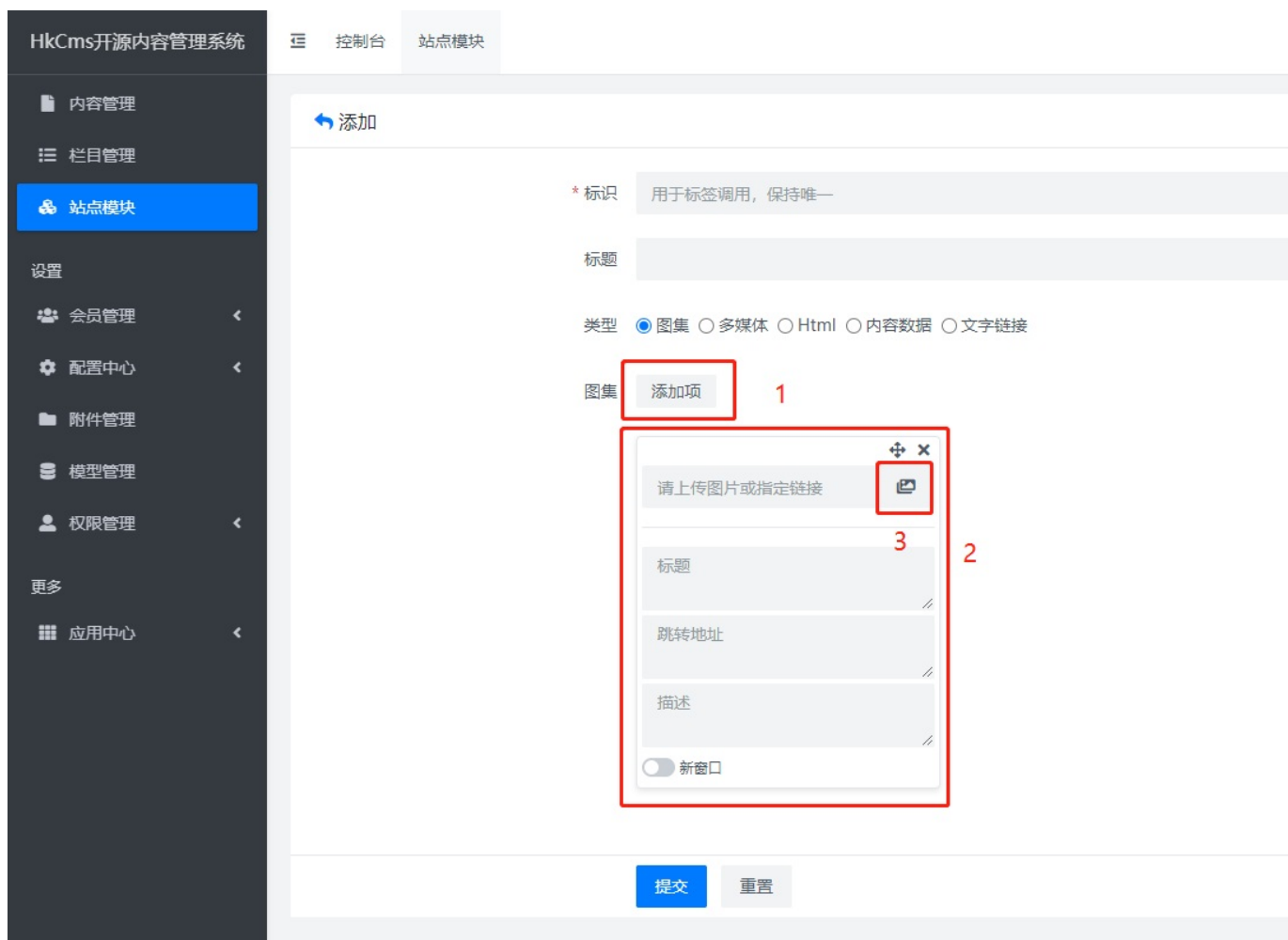
创建图集

1. 点击添加



2. 添加图片集

- (1) 点击 框1 会自动出现新的 框2
- (2) 点击 框3 选择图片，勾选或上传图片，点击选择即可。



3. 选择后的效果



创建多媒体

操作跟图集一样

支持选择本地的视频文件，你也可以指定链接地址。

类型 图集 多媒体 Html 内容数据 文字链接

资源内容 添加项

请上传视频或指定链接

标题

跳转地址

描述

新窗口

创建HTML内容模块

如果图集，多媒体都不符合你，你可以完全插入HTML标签的形式创建站点模块



The screenshot shows a web form for creating an HTML content module. At the top left, there is a blue arrow icon and the text '添加'. Below this, there are several input fields and a radio button group. The first field is labeled '* 标识' and contains the text '用于标签调用, 保持唯一'. The second field is labeled '标题'. Below these is a radio button group with options: '图集', '多媒体', 'Html' (which is selected with a blue dot), '内容数据', and '文字链接'. The main part of the form is a large text area labeled 'HTML 内容' containing the text 'HTML 内容'. At the bottom of the form, there are two buttons: a blue '提交' button and a grey '重置' button.

创建HTML内容模块

选择模型

栏目可多选

条数：显示多少个文章

← 添加

* 标识

标题

类型 图集 多媒体 Html 内容数据 文字链接

模型 文章模型 单页模型 留言表单 产品模型 图集模型 招聘模型

栏目 公司荣誉 新闻中心 公司新闻 行业资讯 其他资讯 媒体报道

条数

排序

创建文字链接

不用上传图片，操作跟图集一样

← 添加

* 标识 用于标签调用, 保持唯一

标题

类型 图集 多媒体 Html 内容数据 文字链接

文字链接 添加项

<input type="text" value="跳转地址"/>	<input type="text" value="跳转地址"/>
<input type="text" value="标题"/>	<input type="text" value="标题"/>
<input type="text" value="描述"/>	<input type="text" value="描述"/>
<input type="checkbox"/> 新窗口	<input type="checkbox"/> 新窗口









提交

重置

模型管理

模型管理

在安装的时候，勾选演示数据，默认会有这些模型。模型好比数据表，负责整个内容的字段信息。

ID	模型名称	表名	控制器	备注	全局搜索	状态	操作
5	留言板	guestbook	Guestbook		-	<input checked="" type="checkbox"/>	字段管理  
4	产品模型	product	Archives		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	字段管理  
2	单页模型	single	Single		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	字段管理  
1	文章模型	archives	Archives		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	字段管理  

显示第 1 到第 4 条记录，总共 4 条记录

添加模型

默认有以下模型类别，后续可通过插件安装其他类别的模型。这些类别好比一个模板，按这些模板创建模型数据表。

文章类：包含常见的文章所需的字段数据

单页类：包含默认的单页类的字段数据，一个栏目绑定一条记录，在内容管理则直接是一个单页添加。

留言表单：常用于留言表单、报名表单等前台表单提交的。

添加
—

模型名称:

表名:

控制器:

文章类
文章类
单页类
留言表单

备注:

全局搜索: 允许 不允许

状态: 正常 禁用

字段管理

在这里可以禁用某个字段，同时你也可以配置栏目，只用于什么栏目下显示。

☰
控制台
模型管理 ⊗
留言板.. ⊗

...

 admin

 网站首页

C
+ 添加
删除
设置 ▾

<input type="checkbox"/>	排序	字段	标题	字段类型	长度	默认值	主表字段	分组	状态	操作
<input type="checkbox"/>	0	name	姓名	text	100		主表	常规	<input checked="" type="checkbox"/>	栏目
<input type="checkbox"/>	0	email	邮箱	text	250		主表	常规	<input checked="" type="checkbox"/>	栏目
<input type="checkbox"/>	0	phone	手机	text	20		主表	常规	<input checked="" type="checkbox"/>	栏目
<input type="checkbox"/>	0	content	内容	textarea	250		主表	常规	<input checked="" type="checkbox"/>	栏目

添加字段

选择你需要的字段类型添加即可。字段类型说明请参考字段类型章节。

添加✕

字段类型:	<input type="text" value="单行文本"/>
分组:	单行文本 多行文本 单选
字段:	复选 下拉列表 下拉列表(多)
标题:	编辑器 图片
长度:	图片(多) 数字
默认值:	日期 日期和时间 文件上传
规则:	文件上传(多)
提示信息:	<input type="text" value="提示信息"/>
错误信息:	<input type="text" value="错误信息"/>
附加属性:	<input type="text" value="追加或覆盖html属性, 多个属性使用英文逗号分隔, 值用双引号。示例: data-title='标题',class='form-control selecpage'"/>
排序:	<input type="text" value="0"/>
状态:	<input checked="" type="radio"/> 正常 <input type="radio"/> 禁用

栏目管理

栏目管理

如图所示，进入栏目管理。



栏目类型

栏目首页：只支持一个栏目首页模板可选。

列表栏目：支持列表页、详情页模板可选。

其他链接：如果不选模型，则作为跳转链接，选了模型，则支持单页页面模板。

添加页说明

点击添加：默认栏目名称，支持批量添加，如文本框提示

栏目管理

后台管理

控制台中 站点配置 模型管理 应用中心 留言板字段 栏目管理

+ 添加 删除 设置 栏目权限 扩展字段 回收站

排序	标题	栏目类型	所属模型	导航显示	状态	操作
0	首页	其他链接	-	开启	开启	+子栏目 编辑 删除
0	行业资讯	列表栏目	文章模型	开启	开启	+子栏目 编辑 删除

添加

常规 更多 扩展字段

栏目类型: 栏目首页 列表栏目 其他链接

所属模型: 文章模型 **模型选择**

上级栏目: 作为一级菜单

栏目名称(英文): 例如: 国内新闻|china 国际新闻|world

✖ 此处不能为空

链接: 指定栏目地址, "其他链接" 类型则是跳转的地址

列表模板: list.html

内容模板: show.html **页面选择**

排序: 0

提交 重置

扩展字段

栏目的扩展字段, 用于扩展额外栏目字段, 操作步骤如下。

+ 添加 删除 设置 栏目权限 扩展字段 回收站

排序	标题	栏目类型	所属模型	导航显示	状态	操作
0	首页	其他链接	-	开启	开启	+子栏目 编辑 删除
0	公司介绍	其他链接	单页模型	开启	开启	+子栏目 编辑 删除
0	服务	其他链接	单页模型	关闭	开启	+子栏目 编辑 删除
0	公司荣誉	列表栏目	文章模型	关闭	开启	+子栏目 编辑 删除
0	产品展示	列表栏目	产品模型	开启	开启	+子栏目 编辑 删除
0	色谱仪器	列表栏目	产品模型	开启	开启	+子栏目 编辑 删除
0	光谱仪器	列表栏目	产品模型	开启	开启	+子栏目 编辑 删除
0	电化学仪器	列表栏目	产品模型	开启	开启	+子栏目 编辑 删除
0	首页展示	列表栏目	文章模型	关闭	开启	+子栏目 编辑 删除

扩展字段

— ✕ ×

C + 添加 删除 设置 ▾ 刷新 重置 列表 ▾

<input type="checkbox"/>	排序	字段	标题	字段类型	长度	默认值	分组	状态	操作
没有找到匹配的记录									

2

添加

字段类型: 单行文本

分组: 单行文本

字段: 多行文本

标题: 单选

长度: 复选

默认值: 下拉列表

规则: 下拉列表(多)

提示信息: 编辑器

提示信息: 图片

提示信息: 图片(多)

提示信息: 数字

提示信息: 日期

提示信息: 日期和时间

提示信息: 文件上传

提示信息: 文件上传(多)

提示信息: 提示信息

提交 重置

选择你要的类型添加即可

添加完成后，在添加页面即可看到对应的字段

修改 — □ ×

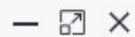
常规 更多 扩展字段

单行文本:

栏目权限

在栏目权限管理这里，需要提前创建好角色管理，超级管理员拥有所有栏目权限。

栏目权限



角色组:

内容管理员

栏目权限:

选中全部 展开全部

- 公司介绍
 - 服务
- 公司荣誉
- 产品展示
 - 色谱仪器
 - 光谱仪器
 - 电化学仪器
 - 首页展示
- 团队展示
- 客户案例
- 新闻中心
 - 公司新闻
 - 行业资讯
 - 其他资讯
- 联系我们
 - 在线留言

提交

重置

字段类型与组件

[前言](#)

[验证规则](#)

[text 单行文本](#)

[textarea 多行文本](#)

[radio 单选](#)

[checkbox 复选](#)

[select 下拉列表](#)

[selects 下拉列表\(多\)](#)

[selectpage 动态下拉](#)

[editor 编辑器](#)

[image 图片](#)

[images 图片\(多\)](#)

[number 数字](#)

[date 日期](#)

[datetime 日期和时间](#)

[downfile 文件上传](#)

[downfiles 文件上传\(多\)](#)

[array 数组\(键值对\)](#)

[selectpage\(动态下拉\)](#)

前言

前言

目前支持的字段类型有：

字段类型	字段名称	说明
text	单行文本	
textarea	多行文本	
radio	单选	
checkbox	复选	
select	下拉列表	
selects	下拉列表(多)	
selectpage	动态下拉	支持关联数据表，url链接
editor	编辑器	
image	图片	
images	图片(多)	
number	数字	
date	日期	
datetime	日期和时间	
downfile	文件上传	
downfiles	文件上传(多)	
array	数组(键值对)	

前端组件

动态下拉(selectpage)

动态下拉(selectpage)

说明

动态下拉使用的是 `selectpage` 多功能下拉分页选择插件。

文档地址：<https://terryz.gitee.io/selectpage/>

HkCms中使用方式

1. html 写法

在表单中创建 `input` 类型为 `text`，class里面增加 `selectpage`，同时增加如下属性。这里需要注意的是属性的写法例如文档中有个 `showField` 这种格式的属性的属性那么html的写法是

`data-show-field`。

```
<input type="text" class="form-control selectpage"
  data-select-only="true" // 选择模式，不允许输入
  data-pagination="false" // 不使用分页
  data-multiple="true" // 允许多选
  data-data="{:url('/auth.admin/getGroup')}">
```

一些常用的属性列表：

属性名	可选值	说明
<code>data-data</code>		url链接地址，即数据源
<code>data-select-only</code>	true/false	是否允许输入
<code>data-pagination</code>	true/false	是否开启分页
<code>data-multiple</code>	true/false	是否允许多选
<code>data-list-size</code>		默认10，规定分页大小
<code>data-show-field</code>	默认name	显示的字段
<code>data-key-field</code>	默认id	存储字段
<code>data-search-field</code>	默认同data-show-field 字段一致	需要提交的查询字段， 多个英文逗号分隔
<code>data-params</code>		附加查询参数，data-params='{"status":"normal"}'，使用系统自带的查询写法： <code>data-params='{"custom[status]":"normal"}'</code>

更多属性可参考上面的文档链接。

2. 初始化表单

```
<script>
  require(['jquery', 'Form'], function ($, Form) {
    Form.api.init(); // 初始化表单组件生效
  })
</script>
```

附加参数

附加参数动态指定方式，提交效果：custom[status]: normal , custom[id]: 10

```
$("#frm-flags").data("params", function (e) {
  return {
    custom: {status:1, id:$('#id').val()}
  };
});
```

验证规则

验证规则

多个验证规则使用 “;” 隔开，更多内容可前往：<https://validator.niceue.com/docs/>

规则名称	说明
required	必须，必填
checkbox	checkbox/radio必选
digits	数字
integer	整数
letters	字母
date	日期
time	时间
email	邮箱
url	网址
qq	QQ号
IDcard	身份证号码
tel	电话号码
mobile	手机号
zipcode	邮政编码
chinese	中文字符
username	3-12位数字、字母、下划线
password	'6-16位字符，不包含空格

text 单行文本

text 单行文本

添加 — ✕ ×

字段类型: 单行文本 ▼

分组: 常规 ✕ -

主表字段: 主表 副表

字段: 仅支持字母、数字、下划线 ⓘ
✕ 此处不能为空

标题: 仅支持汉字、字母、数字和下划线 ⓘ
✕ 此处不能为空

长度: 250

默认值: 默认值

规则: ▼

提示信息: 提示信息

错误信息: 错误信息

附加属性: 追加或覆盖html属性，多个属性使用英文逗号分隔。示例: data-title="标题",class="form-control seleepage"

排序: 0

状态: 正常 隐藏

提交 重置

生成效果图，将生成单行文本框

测试:

分组

这里可以自定义字段分组，最后提现在标签切换，如下图所示

字段

字段名称对应数据库的字段，仅支持字母、数字、下划线

标题

生成控件时左边的名称

长度

长度在1~250文字内。对应数据库char类型。

默认值

生成控件时，填充你填写的默认值

规则

即提交时的验证规则，前往这里了解：[验证规则](#)

提示信息

在控件右侧会有感叹号，显示提示信息



错误信息

指定验证规则不通过时提示信息，无特殊要求使用默认的提示信息也可

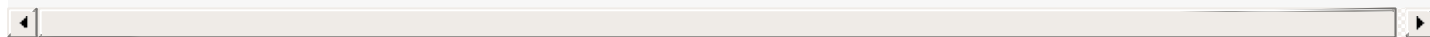
附加属性

用于生成html标签时，追加或覆盖标签属性，注意多个属性逗号分隔。

示例：`data-title="标题",class="form-control selectpage"`

以上信息会原样替换到标签上。结果示例

```
<input class="form-control selectpage" data-title="标题" name="row[ceshi]" type="text" value="我是默认值">
```



合理的使用附加属性可以实现调用动态下拉等组件，组件的使用可参考上面组件章节

模板获取值

假设变量为\$item，通过以下方式获取

```
{ $item.字段名称 } 或者 { $item['字段名称'] }
```

设置默认值

```
{ $item['字段名称']??'' } 或者 { $item['字段名称']|default="" }
```

textarea 多行文本

textarea 多行文本

添加 — ✕ ×

字段类型: 多行文本 ▼

分组: 常规 ✕ ▼

主表字段: 主表 副表

字段: ceshi|

标题: 测试

长度: 500

默认值: 默认值

规则: ▼

提示信息: 提示信息

错误信息: 错误信息

附加属性: 追加或覆盖html属性, 多个属性使用英文逗号分隔。示例: data-title="标题",class="form-control selecpage"

排序: 0

状态: 正常 隐藏

效果

测试:

长度

textarea 对应数据库varchar类型, 支持1~16000个字符。

textarea 多行文本

- 其他说明、参考text 单行文本

radio 单选

单选

添加 — ✕ ×

字段类型: 单选

分组: 常规

主表字段: 主表 副表

字段: ceshi3

标题: 测试3

长度: 250

默认值: 默认值

选项列表: key1|标题
key2|标题 

规则:

提示信息: 提示信息

错误信息: 错误信息

附加属性: 追加或覆盖html属性, 多个属性使用英文逗号分隔。示例: data-title="标题",class="form-control selecpage"

排序: 0

状态: 正常 隐藏

效果, 生成单选控件

测试3: 标题 标题2

长度

这里并不需要填写长度, 数据库将生成enum类型。

radio 单选

选项列表

格式如下：

key1|标题

1

key2|标题2

左边是保存到数据库的值，右边是单选控件显示的名称，以英文 “|” 符号分隔

默认值

默认值填写后，将会默认选，如上，默认值填写 key2，在进入页面后默认选中了key2

测试3: 标题 标题2



- 其他说明、参考text 单行文本

checkbox 复选

checkbox 复选

添加
— ✕ ✕

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

选项列表:

规则:

提示信息:

错误信息:

附加属性:

排序:

状态: 正常 隐藏

生成效果

复选测试: 标题1 标题2 标题3

长度

这里并不需要填写长度，数据库将生成set类型。

选项列表

checkbox 复选

格式如下：

key1|标题

1

key2|标题2

左边是保存到数据库的值，右边是单选控件显示的名称，以英文 “|” 符号分隔

默认值

默认值填写后，将会默认选，如上，默认值填写 key2，在进入页面后默认选中了key2，选中多个英文逗号分隔，key1,key2

模板获取

复选的值在模板是一维数组

假设变量是\$Info

1. 循环输出，字段名称是你增加的字段名

```
{hkcms:volist name="$Info['字段名称']" id="item"}  
<p>{$item}</p>  
{/hkcms:volist}
```

2. 给定元素输出

```
{$Info['字段名称'][0]}
```

3. 数组转字符，以逗号分隔

```
{:explode(',', $Info['字段名称'])}
```

- 其他说明、参考text 单行文本

select 下拉列表

select 下拉列表

添加 — ✕ ×

字段类型: 下拉列表 ▼

分组: 常规 ✕ ▼

主表字段: 主表 副表

字段: select1

标题: 下拉列表

长度: 250

默认值: 默认值

选项列表:
key1|标题1
key2|标题2
key3|标题3
key4|标题4

规则: ▼

提示信息: 提示信息

错误信息: 错误信息

附加属性: 追加或覆盖html属性, 多个属性使用英文逗号分隔。示例: data-title="标题",class="form-control seleepage"

排序: 0

状态: 正常 隐藏

提交 重置

生成效果

select 下拉列表

下拉列表:

- 标题1
- 标题2
- 标题3
- 标题4

长度

这里并不需要填写长度，数据库将生成enum类型。

选项列表

格式如下：

key1|标题

1

key2|标题2

左边是保存到数据库的值，右边是单选控件显示的名称，以英文 “|” 符号分隔

默认值

默认值填写后，将会默认选，如上，默认值填写 key2，在进入页面后默认选中了key2

- 其他说明、参考text 单行文本

selects下拉列表(多)

selects下拉列表(多)

添加 — ✕ ×

字段类型: 下拉列表(多) ▾

分组: 常规 ✕ ▾

主表字段: 主表 副表

字段: selects

标题: 下拉列表多选

长度: 250

默认值: key3,key1|

选项列表:
key1|标题1
key2|标题2
key3|标题3
key4|标题4

规则: ▾

效果

下拉列表多选:

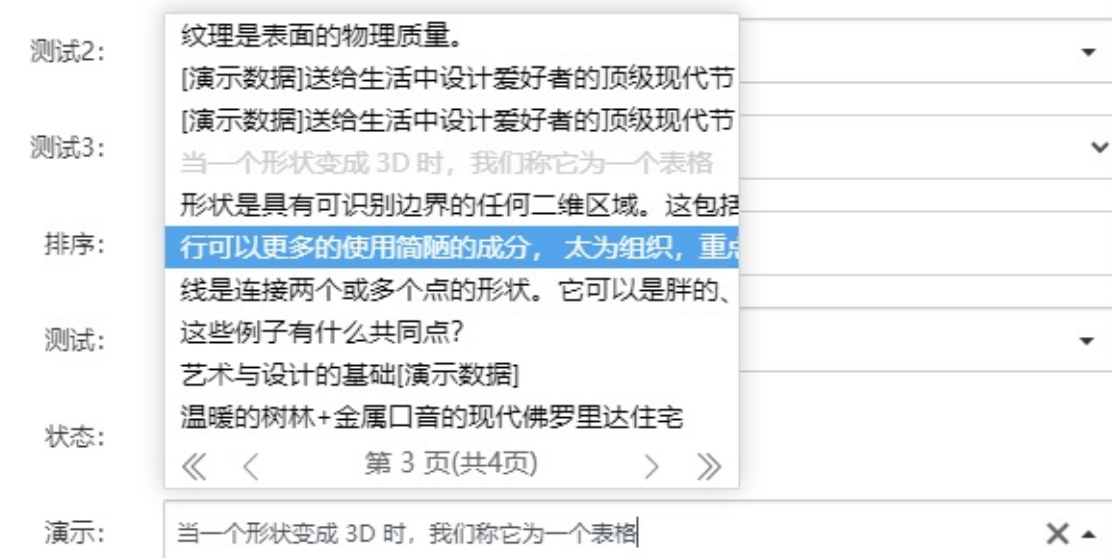
- 标题1
- 标题2
- 标题3
- 标题4

- 同复选一致

selectpage 动态下拉

动态下拉

里面的选项可来自于链接请求，在模型字段类型选择动态下拉，配置数据表的效果如下：



HkCms版本>2.1.220220

配置说明

数据源：配置数据源来实现动态数据。

URL 数据源是指以请求链接地址的方式来获取数据，常用于数据来源非数据库或无需关联的表的数据，系统有内置一些链接来支持 `selectpage` 组件，当然你也可以自定义控制器来继承 `BaseController` 控制器，里面有 `selectpage` 方法。

项	链接地址	说明
---	------	----

关联表 数据源这里就直接是选择数据库里的表。

显示字段 是指表单里面显示的字段。

存储字段 是指存储在数据库里面的数据

搜索字段 动态下拉支持输入检索，这里你可以指定要搜索的字段，支持多个字段搜索使用英文逗号分隔，例如：title,subtitle。

查询方式 是指在搜索字段多个的情况下是 **并且(and)** 关系还是 **或(or)** 关系

排序 定义排序的字段，例如：id desc,views desc

附加条件 增加筛选条件，字段=值

仅选择模式：不允许输入

多语言筛选：如果你希望按多语言区分显示数据，那么勾选它

模板输出

1. URL数据源

多选的情况下，使用[volist 循环标签](#)

2. 关联表数据源

关联表的输出是指定表的所有列输出例如你选择了 `archives` 表，存的是ID，那么前台拿到的是这个表的所有字段。

多选情况下返回的是数组使用volist循环标签即可，单选不是数组，单选就直接是`{info.demo.title}`

editor编辑器

editor编辑器

添加 - ✕ ×

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

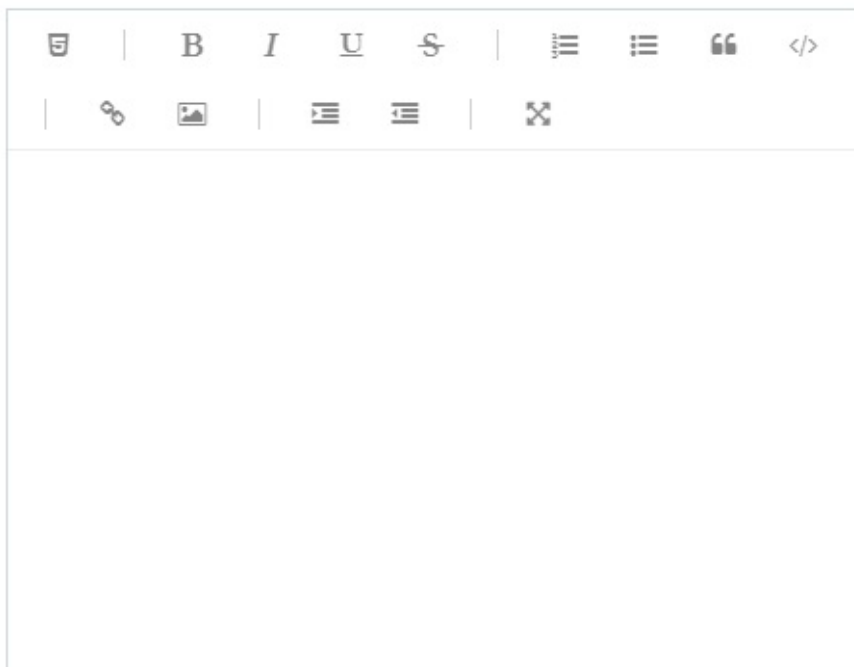
规则:

提示信息:

错误信息:

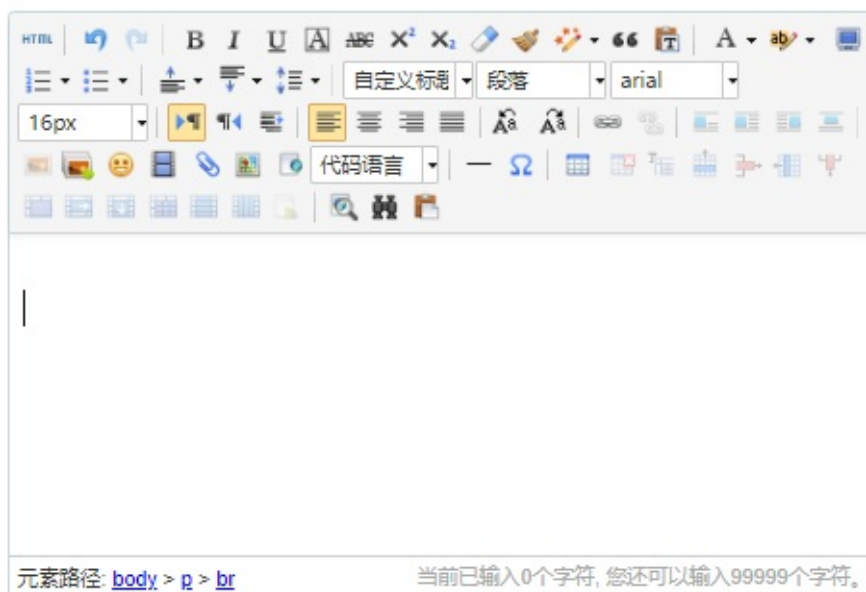
编辑器生成效果根据你安装的编辑器插件有关系，cms默认编辑器

编辑器:



百度编辑器插件演示效果

编辑器:



长度

这里并不需要填写长度，对应mysql text类型

- 其他说明、参考text 单行文本

image图片

image图片

添加

字段类型: 图片

分组: 常规

主表字段: 主表 副表

字段: img

标题: 图片

长度: 250

默认值: 默认值

规则:

提示信息: 提示信息

错误信息: 错误信息

提交 重置

生成效果

图片:

长度

对应数据库varchar类型，根据需要可设置1~16000的长度

- 其他说明、参考text 单行文本

images图片(多)

images 图片(多)

多图

添加 — ✕ ×

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

最大数量:

规则:

提示信息:

生成效果

多图:

长度

对应数据库varchar类型，根据需要可设置1~16000的长度

最大数量

设置最多可上传的图片数量，注意一个文件占的长度根据你配置的保存格式来计算，比如一个文件长度50，设置能上传5文件，那长度设置不得少于250

模板获取多图

images图片(多)

可参考，(图集数组等)循环标签

- 格式示例

```
{hkcms:volist name="Info.images"}  
<div class="carousel-item {$currentstyle}">  
    
</div>  
{/hkcms:volist}
```

- 其他说明、参考text 单行文本

number数字

number 数字

添加 — ✕ ×

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

小数位:

规则:

提示信息:

生成效果

数字:

长度

如果没有小数位，对应mysql int类型，写了小数点位数，对应decimal类型

- 其他说明、参考text 单行文本

date日期

date日期

日期类型，不包含时间

添加 — ✕ ×

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

规则:

提示信息:

错误信息:

生成效果

日期:

« < 2021年 12月 > »

日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

date日期

长度

这里并不需要设置长度，数据库类型为date

- 其他说明、参考text 单行文本

datetime日期和时间

datetime日期和时间

添加 — ✕ ×

字段类型:

分组:

主表字段: 主表 副表

字段:

标题:

长度:

默认值:

规则:

提示信息:

错误信息:

生成效果，支持选择时间

«	<	2021年 12月	>	»		
日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8
选择时间		清空	现在	确定		

日期时间:

长度

这里并不需要设置长度，数据库类型为datetime

- 其他说明、参考text 单行文本

downfile文件上传

downfile文件上传

单文件上传，文件上传后缀，目前是在站点配置里面，配置全局允许上传的文件类型。

- 其他说明、参考image图片

downfiles文件上传(多)

downfiles文件上传(多)

多文件上传，文件上传后缀，目前是在站点配置里面，配置全局允许上传的文件类型。

- 其他说明、参考images图片(多)

array 数组(键值对)

array 数组(键值对)

添加 — ✕ ×

字段类型: 数组(键值对) ▼

分组: 常规 ✕ ▼

主表字段: 主表 副表

字段: arr1

标题: 数组

长度: 长度

默认值: 默认值

键名名称: 键名

键值名称: 键值

规则: ▼

生成效果

数组: 键名 键值

长度

长度在这里并不需要理会，对应mysql 类型为 text

键名名称 与 键值名称

如下图所示

array 数组(键值对)

数组: 键名 键值

键名名称 键值名称

模板获取值

例如填了以下值

数组: 键名 键值

ceshi abc

假设变量是\$Info

1. 循环输出, 字段名称是你增加的字段名

```
{hkcms:volist name="$Info['字段名称']" id="item"}  
<p>{$key} - {$item}</p>  
{/hkcms:volist}
```

2. 给定元素输出

```
{$Info['字段名称'][$键名称]}
```

如上图, 键名为ceshi

















```
{$Info['字段名称']['ceshi']} 输出 abc
```

内容管理

内容管理

根据栏目管理创建的栏目，在内容管理这里添加，录入文章。

The screenshot displays a web-based content management system (CMS) interface. On the left is a dark sidebar menu with the title '后台管理' (Backend Management). The '内容管理' (Content Management) option is highlighted in green. Below it are other menu items: '栏目管理' (Category Management), '站点模块' (Site Modules), '设置' (Settings), '站点配置' (Site Configuration), '模型管理' (Model Management), '附件管理' (Attachment Management), '权限管理' (Permission Management), and '应用中心' (Application Center). The main content area is titled '内容管理' and features a breadcrumb trail: '控制台 > 站点模块 > 内容管理'. At the top of the main area, there are search and filter controls: 'ID' (with a search input), '标题' (with a search input), '发布人' (with a dropdown), and '发布时间' (with a date range selector). Below these are buttons for '+ 添加' (Add), '删除' (Delete), '设置' (Settings), and '回收站' (Recycle Bin). A '刷新栏目' (Refresh Categories) button is also present, with a red arrow pointing to it. The main area contains a table with 10 columns: '排序' (Sort), 'ID', '标题' (Title), '所属栏目' (Category), '点击量' (Clicks), '发布人' (Author), '发布时间' (Publish Time), '状态' (Status), and '操作' (Actions). The table lists 8 records, all with a status of '正常' (Normal). At the bottom of the table, it says '显示第 1 到第 8 条记录，总共 8 条记录' (Showing 1 to 8 records, total 8 records). On the right side of the table, there are additional controls: a search icon, a refresh icon, a '显示搜索' (Show Search) button, and a '显示其他字段' (Show Other Fields) button with a dropdown arrow, also indicated by a red arrow.

排序	ID	标题	所属栏目	点击量	发布人	发布时间	状态	操作
0	9	演示数据	公司荣誉	2	admin	2021-03-22 09:24:45	正常	 
0	8	标题	公司荣誉	0	admin	2021-03-22 09:24:24	正常	 
0	7	荣誉证书D	公司荣誉	0	admin	2021-03-22 09:24:12	正常	 
0	6	荣誉证书C	公司荣誉	0	admin	2021-03-22 09:23:45	正常	 
0	5	荣誉证书B	公司荣誉	0	admin	2021-03-22 09:23:23	正常	 
0	4	荣誉证书A	公司荣誉	0	admin	2021-03-22 09:23:09	正常	 
0	3	荣誉证书	公司荣誉	0	admin	2021-03-22 09:22:54	正常	 
0	2	荣誉证书	公司荣誉	0	admin	2021-03-22 09:22:26	正常	 

自定义URL

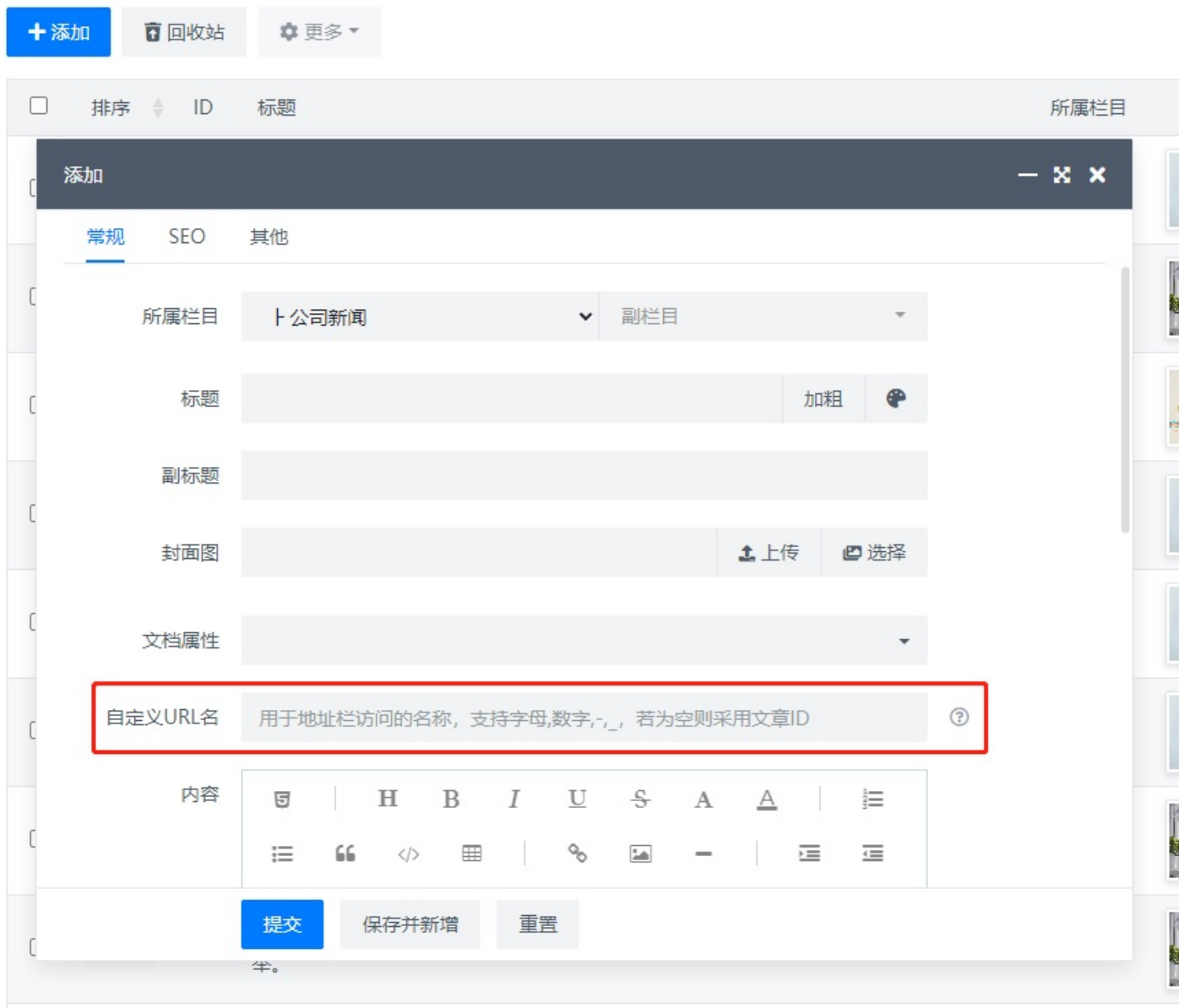
自定义URL

自定义URL后，必须开启伪静态和静态页面才生效。

如何开启伪静态和静态页面。

前往配置中心->SEO配置->URL模式。

具体说明：[伪静态](#) / [静态页面](#)



多语言

[界面多语言](#)

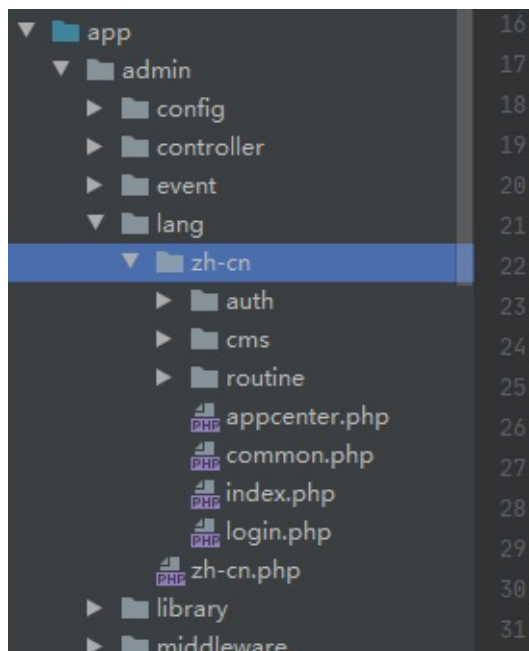
[内容多语言](#)

界面多语言

界面多语言

在每个模块下会有lang文件夹，在里面定义语言文件

- 后台路径如下：



目录说明：zh-cn.php是全局通用的语言包，在zh-cn里面的语言文件是针对控制器的。

- 格式：

简体中文对应

```
return [  
    'Submit'    => '提交',  
    'Reset'     => '重置',  
    'User ID'   => '用户ID'  
]
```

- 使用方式:

```
//如果当前语言是英文，则显示Submit，如果中文，就显示提交，如果是繁体或其他语言，根据定义的语言文件显示出来。  
lang("Submit");
```

模板多语言

参考模板开发章节

6. 模板界面多语言

内容多语言

内容多语言注意事项

内容多语言一旦开启，将以语言标识显示对应的数据，因此在开启后添加的数据跟关闭添加的数据并不相同，一定要留意

2.1.220220版本起，已实现关闭多语言后，使用默认的语言数据，并且新版本已经把内容语言与界面分开，这里可根据需要调整。

开启内容多语言

在安装的时候没有勾选演示数据默认内容多语言是关闭的。

1. 开启多语言

基础配置 邮件配置 附件配置 **多语言** 高级配置 分组配置 +

内容多语言: 开启 关闭 {site.content_lang_on}

内容语言列表:

多语言	标题	默认
zh-cn	中文简体	<input checked="" type="radio"/> 默认
en	English	<input type="radio"/> 默认

添加

后台界面语言:

多语言	标题	默认
zh-cn	中文简体	<input checked="" type="radio"/> 默认
en	English	<input type="radio"/> 默认

添加

前台界面语言:

多语言	标题	默认
zh-cn	中文简体	<input checked="" type="radio"/> 默认
en	English	<input type="radio"/> 默认

添加

提交 重置

帮助

标签调用

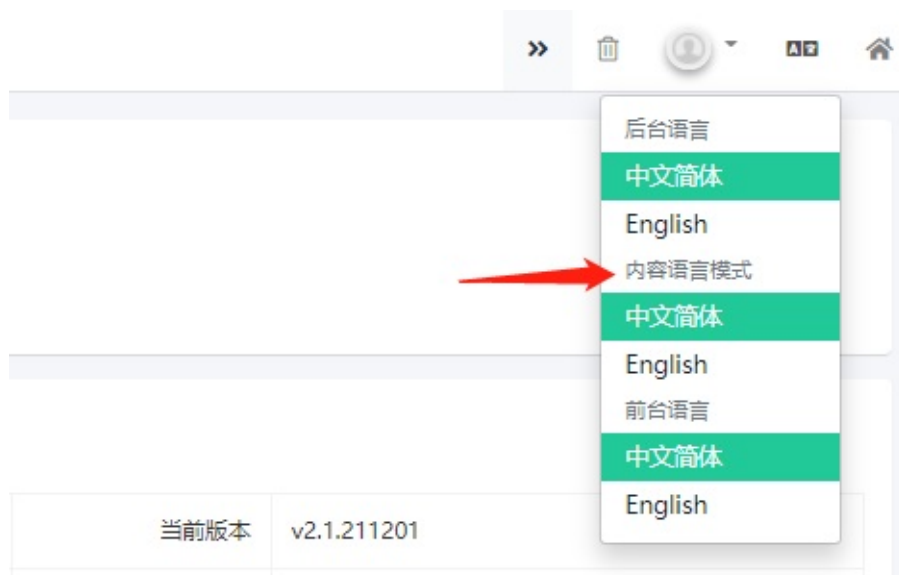
多语言列表，默认只有中文简体、英文，自行增加其他语言。

注意：这里增加后界面也需要增加对应的语言包，才能实现界面同步切换到对应的语言

默认选项是用户第一次进来时显示的语言，如果用户有切换过其他语言，再次进来则显示之前切换过的语言。

2. 开启后的效果

右上角多语言图标点击后可以看到当前内容编辑模式。



列表数据如下显示

C + 添加 删除 设置 栏目权限 扩展字段 回收站									
<input type="checkbox"/>	排序	ID	标题	栏目类型	所属模型	导航显示	状态	仅用于	操作
<input type="checkbox"/>	0	1	首页	其他链接	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(当前)中文简体	+子栏目 编辑 删除
<input type="checkbox"/>	0	3	关于我们	其他链接	单页模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(当前)中文简体	+子栏目 编辑 删除
<input type="checkbox"/>	2	7	公司简介	其他链接	单页模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English	+子栏目 编辑 删除
<input type="checkbox"/>	1	5	公司荣誉	列表栏目	文章模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(当前)中文简体	+子栏目 编辑 删除
<input type="checkbox"/>	0	31	服务	其他链接	单页模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English	+子栏目 编辑 删除
<input type="checkbox"/>	0	9	新闻中心	栏目首页	文章模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(当前)中文简体	+子栏目 编辑 删除
<input type="checkbox"/>	0	11	公司新闻	列表栏目	文章模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English	+子栏目 编辑 删除
<input type="checkbox"/>	0	13	行业资讯	列表栏目	文章模型	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	(当前)中文简体	+子栏目 编辑 删除

3. 新增数据

新增一条数据，根据当前内容模式，自动同步新增两条记录，一条中文，一条英文，英文的数据需要自己修改成对应的英文。

4. 删除一条数据，会同步删除，例如删除一条中文，则会删除对应英文记录。

5. 模板使用时，会自动根据当前语言切换数据。

模板显示多语言标签

内容多语言

参考 [获取语言列表标签](#)

多语言URL地址

多语言地址

目前支持的多语言格式为:

```
http://网址/en/
```

设置方法如下

URL 配置

URL 模式 动态 伪静态 静态页面

注意: 勾选伪静态后您的服务器必须配置URL重写才能正常访问

Url 格式 当前目录

多语言/当前目录

父目录/当前目录

多语言/父目录/当前目录

模型/栏目

自定义, [点我前往](#)

栏目标题格式 栏目名称 - 网站名称

栏目名称 - 第N页 - 网站名称

栏目名称_第N页

栏目名称_第N页_网站名称

如果想自定义可以参考

[前台URL规则配置](#)

附件管理

附件配置

附件配置功能，可以管理系统中的文件，设置水印和缩略图。



水印

根据图片表单设置即可。

水印

水印功能 开启 关闭


水印类型 图片 文字

图片宽度 ⓘ

图片高度 ⓘ

水印透明度 ⓘ

水印图片



水印位置 顶部居左 顶部居中 顶部居右
 中部居左 中部居中 中部居右
 底部居左 底部居中 底部居右

缩略图

缩略图配置根据表单设置即可。

前台调用：`{hkcms:thumb name='图片路径' /}`

[标签具体说明](#)

缩略图 — ✕ ×

缩略图: 开启 关闭

生成方式: 等比例缩放 缩放填充 居中裁剪

左上角裁剪 右下角裁剪 固定尺寸缩放

缩略图宽度: ⓘ

缩略图高度: ⓘ

应用中心

应用中心

1. 在这里分为本地应用，和在线应用。支持在线下载、安装插件与模板。
2. 本地应用

本地安装，注意在模板时就安装模板应用，在插件页面本地上传的就是插件应用

(1) 模板

模板文件



静态文件

编辑CSS，JS，语言包等等



模板语言

可以直接编辑语言包

默认前台模板

模板文件 静态文件 **模板语言** 主题配置

一键同步 语言包: zh-cn.json 保存

键名	值	模板调用	操作
Know Us	了解我们	复制	删除
About Us	关于我们	复制	删除
Enter a keyword and press Enter to search	输入关键字, 按回车搜索	复制	删除
Read More	查看更多	复制	删除
More	更多	复制	删除
All	全部	复制	删除

主题配置

编辑说明：[前往查看](#)

主题配置
— ✕ ×

QQ	前台侧边栏QQ展示	?
手机号/电话	前台侧边栏手机号	?
邮箱	前台侧边栏邮箱	?
微信	<input type="text" value="http://holuo.cn-gd.ufileos.com/hkcms/demo_qr"/> ↑ 上传 📁 选择	
	 🗑️	
公众号图片	<input type="text"/> ↑ 上传 📁 选择	

提交
重置

(2) 插件

☰ 控制台 本地
🗑️ 缓存
🗣️ 语言
🏠 首页
👤 admin
⚙️

模板 插件
🔄 导入插件 → 本地安装插件

标题	封面	介绍	作者	价格	版本	状态	操作
simditor编辑器		simditor编辑器	HkCms	免费	1.0.1	<input checked="" type="checkbox"/>	🗑️ 卸载 📦 打包
地图位置选取插件		支持百度、高德地图位置选取。	HkCms	免费	1.0.0	<input checked="" type="checkbox"/>	⚙️ 配置 🗑️ 卸载 📦 打包
前台用户中心		用户中心插件，提供基本的登录、注册、密码找回、个人中心、会员组、会员权限等功能。	HkCms	免费	1.0.4 升级	<input checked="" type="checkbox"/>	⚙️ 配置 🗑️ 卸载 📦 打包
数据库备份		数据库备份、优化、修复。	HkCms	免费	1.0.0 升级	<input checked="" type="checkbox"/>	⚙️ 配置 🗑️ 卸载 📦 打包

某些插件需要填写相关信息
打包插件

- 在线应用里面，首先需要注册HkCms官网的账号，在到站点配置或者直接在应用中心输入账号密码登录访问。

注册会员

(1) 在线模板

模板 插件

特性筛选 共 24 套模板

搜索

响应式广告策划企业简洁模板
免费 那些日子
安装 演示 211 2022-01-07

涂料油漆模板
免费 那些日子
安装 演示 30 2022-01-05

茶叶模板
免费 那些日子
安装 演示 31 2022-01-05

响应式多语言企业公司简单通...
免费 HkCms
安装 195 2022-01-03

直接安装

(2) 在线插件

模板 插件

特性筛选 共 15 个插件

搜索

序号	标题	封面	描述	作者	价格	版本	更新	状态	操作
1	一键生成应用		一键生成应用的基础文件，目前只支持模板生成。	HkCms	免费	未安装	详情	-	安装
2	sitemap地图生成		SEO的sitemap生成插件，生成后的xml地址为： http://域名/sitemap.xml ，注意：后台SEO设置url模式改为伪静态	奈落。	免费	未安装	详情	-	安装
3	友情链接插件			xiaowu	免费	未安装	详情	-	安装
3	(页面优化) 网站变灰色		一般在清明节，全国哀悼日，大地震的日子，以及一些影响力很大的伟人逝世或纪念日的时候，身为站长的我们都会让自己的网站的全部网页变成灰色(黑白色)，以表示我们对逝者的悼念	奈落。	免费	未安装	详情	-	安装
4	(网站美化) 雪花飘飘		网站出现雪花动态效果,没效果检查底部是否加了{hook("indexFooter")} 标签	奈落。	免费	未安装	详情	-	安装
5	Excel导入导出		方便excel表格数据的批量导入和导出功能，支	Inspire	免费	未安装	详情	-	安装

查看更新记录

直接安装最新版

选择版本安装

标签语法

- [模板变量输出语法](#)
- [SEO标签](#)
- [栏目标签](#)
- [站点模块标签](#)
- [面包屑导航标签](#)
- [内容列表标签](#)
- [内容列表分页标签](#)
- [格式化编辑器文本](#)
- [上一篇标签](#)
- [下一篇标签](#)
- [日期格式化标签](#)
- [\(图集数组等\)循环标签](#)
- [留言模型\(表单\)标签](#)
- [包含文件include标签](#)
- [导入标签](#)

标签使用必读

模板变量格式说明

变量格式：中括号包裹，例如：`{ $title }`。 `{` 和 `$` 之间不能有任何的空格，否则标签无效

1. 输出站点配置变量

```
// 获取站点标题  
{ $site.title }等价于{ $site['title'] }
```

注意：在一些标签参数使用中，使用`$site['title']`方式赋值。具体看标签说明。

模板函数的调用

模板调用系统函数、或自定义的函数方法：

1. 例如调用md5函数

```
方式一  
{ :md5($data.name) }
```

默认值

使用 `default` 或`(??)`两个问号方法来对变量为空时给定默认的值

1. 例如图片默认值

```
设置封面图没有上传时，给定默认值。  
{ $item.thumb | default="默认图片地址" }
```

2. 例子2，设置字段值为空时的默认值

```
{ $item.remark | default="暂无说明" }
```

模板全局变量

- `$Cate` 模板变量（数组）

栏目信息，存在于栏目分类页、列表页、单页、内容页。调用方式如下。

```
// 获取栏目标题
```

```
<h1 class="mb-3 mt-5 bread">{$Cate.title}</h1>
```

- `$Info` 模板变量 (数组)

内容详情信息, 存在单页、内容页。更多字段可前往后台找到对应模型字段查看。

```
<h2 class="mb-3">{$Info.title} // 文章标题</h2>
```

- `$Tp1` 模板配置

根据模板配置config.json里面获取。只有模板有config.json文件的, 前台才能调取。

列表标签

`content` 内容列表标签

`contentpage` 内容列表分页

`wfpage` 瀑布流分页

`filter` 筛选标签

`order` 排序标签

content 内容列表标签

章节导航

- [内容列表标签](#)
- [格式示例](#)
- [参数](#)
- [参数说明](#)
- [常用示例](#)
- [字段调用](#)

内容列表标签

查询模型数据列表、结合分页标签可实现分页、例如获取文章列表等可用该标签。

格式示例

```
// 获取模型ID为1的3条数据。
{hkcms:content model="1" num="3"}
<a href="{${item.url}}">标题 : {${item.title}</a>
{/hkcms:content}
```

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
catid	栏目ID或变量,多个栏目使用英文逗号分隔,必须是同模型 v2.2.3.220 920+	-	在列表页可以不填,自动获取。	-	默认包含下级文章的内容
model	-	-	-	-	模型ID
aid	-	-	-	v2.2.3.220 920+	指定单个文档ID
aids	文档ID:1,2,3	-	-	v2.2.3.220 920+	指定多个文档ID,英文逗号分

					隔
field	-	*	-	v2.2.3.221 215+	指定显示的字段，默认是*表示所有，注意：系统已包含(id,lang,style,views,url,category_id,status)无需重复指定
order		weigh asc,id desc	-	-	排序，格式为 order="字段名称 desc" 其中desc是倒序，asc是升序
num	格式为数字、变量、5,10	10	-	-	限制结果数量； 5,10：第五条开始，获取10条数据
where	-	-	-	-	数据库查询条件
page	1	-	- -		开启分页，配合 {hkcms:contentpage /} 分页标签
insub	1-包含下级,0-不包含下级	1	-	-	需要与catid同时使用，默认包含下级文章
					针对单页，填写

inlist	1-列表模式,0-单页内容	0	-	-	catid时, 获取catid下级的所有单页, 支持分页
more	1	-	-	v2.2.3.220 920+不在使用	获取副表的数据
flag	-	-	-	-	文档属性值
id	-	\$item	-	-	指定循环里的变量名, 例如 id='obj', 模板调用如: { \$item.title } 变成 { \$obj.title }
empty	-	-	-	-	为空的时候输出提示, 支持传入html
cache	-	3600秒	-	-	缓存时间, 默认3600秒, 对非分页数据有效
mod	-	-	-	v2.2.1.202 20603+	控制隔N行输出的内容等, 更多查看下面的示例
key	-	-	-	v2.2.1.202 20603+	指定循环变量, 如果没有指定key属性的话, 默认使用循环变量\$i
					默认是开

filter	1/0	1	-	v2.3.1.231 108+	启筛选的 (1开启, 0关闭), 关闭后不 受filter筛 选标签影 响
--------	-----	---	---	--------------------	---

参数说明

`catid` 栏目ID或变量, 在列表页时可不填, 自动获取列表页的内容数据

1. 获取属于栏目ID为7, 数量为6条的文档

```
{hkcms:content catid="7" num="6"}
// style字段是后台内容管理标题加粗、标题颜色, 如果需要那么按以下写法获取
<a href="{ $item.url}" style="{ $item.style}">标题: { $item.title}</a>
封面图:  // 更多字段可前往后台字段管理里面或手册数据字典里面查看

{/hkcms:content}
```

2. 以变量形式获取, 例如在列表页有当前栏目变量(\$Cate), `catid="$Cate.id"` \$Cate.id是栏目ID, 下面示例为获取当前栏目6条文档并开启分页

```
// page="1" 是开启分页
{hkcms:content catid="$Cate.id" num="6" page="1"}
<h5><a href="{ $item.url}">标题: { $item.title}</a></h5>
{/hkcms:content}
```

3. 在列表页, 获取当前栏目的文档, 可以不传 `catid` 参数, 如下按每页6条分页展示

```
{hkcms:content num="6" page="1"}
<h5><a href="{ $item.url}">{ $item.title}</a></h5>
{/hkcms:content}
```

4. 同时获取多个栏目ID的文档数据, `catid="栏目ID1,栏目id2"` 多个栏目ID使用英文逗号分隔开

```
{hkcms:content catid="11,12" num="6"}
<a href="{ $item.url}" style="{ $item.style}">标题: { $item.title}</a>
{/hkcms:content}
```

`aid` 获取单篇文章, 如下获取文档ID为14的记录

```
{hkcms:content aid="14"}  
<a href="{${item.url}}" style="{${item.style}}">标题 : {${item.title}</a>  
{/hkcms:content}
```

aids 获取多条文章，多个文档使用英文逗号分隔开

```
{hkcms:content aids="14,15"}  
<a href="{${item.url}}" style="{${item.style}}">标题 : {${item.title}</a>  
{/hkcms:content}
```

model 模型ID参数表示按模型id来获取记录，模型ID在模型管理里面可以看到

```
// 获取模型id为1的记录，num控制显示数量  
{hkcms:content model="1" num="5"}  
文档标题 : {${item.title}  
文档描述 : {${item.description}  
{/hkcms:content}
```

order 用于字段排序，格式为order="字段名 asc,字段名2 desc"，其中 **asc** 表示升序 **desc** 表示倒序，多个排序字段使用逗号分隔开

```
// 列表页，按最新发布的靠前排序  
{hkcms:content order="publish_time desc"}  
文档标题 : {${item.title}  
文档描述 : {${item.description}  
{/hkcms:content}
```

num 用于控制记录条数， **num="5"** 表示只获取5条记录。也可以控制从第几条开始，5,10表示第5~15条记录。注意5,10格式不能用于分页上

```
注意这种限制范围方式不得用于分页  
{hkcms:content model="1" num="5,10"}  
标题 : {${item.title}  
{/hkcms:content}
```

1. 动态控制每页展示文档条数可以在后台栏目管理里面配置（栏目管理-》添加-》更多-》分页大小）

分页大小:

提示: 适用于列表页，需要模板支持

动态文档条数写法示例，num="\$Cate.num"，\$Cate.num是当前栏目的分页大小

```
{hkcms:content num="$Cate.num" page="1"}  
// style字段是后台可设置加粗与颜色  
<a href="{ $item.url}" style="{ $item.style}">标题 : { $item.title}</a>  
{/hkcms:content}
```

where 数据库查询条件，语法跟SQL where基本相同。格式：where="字段名 = 1"，支持传入变量

```
{// 获取模型id为1的记录, where条件意思是只显示阅读量大于1的记录}  
{hkcms:content model="1" where="views > 1"}  
标题 : { $item.title}  
描述 : { $item.description}  
{/hkcms:content}
```

sql语法可参考：https://www.w3school.com.cn/sql/sql_where.asp

更多查询表达式参考：https://www.kancloud.cn/manual/thinkphp6_0/1037537

page 开启分页功能，默认是没有分页的，开启后配合 **contentpage** 分页标签 来实现生成底部下的页码

```
{// page="1" 开启分页}  
{hkcms:content model="1" page="1"}  
标题 : { $item.title}  
{/hkcms:content}
```

insub 控制是否显示下级，该属性需要结合 **catid** 来使用，默认自动包含该栏目下的所有文档数据包含子级，值为1-包含下级，0-不包含下级。

```
// 不包括子级下的文档, 注意获取的下级是跟当前栏目模型一致  
{hkcms:content catid="1" insub="0"}  
标题 : { $item.title}  
{/hkcms:content}
```

注意：获取下级栏目的文档是跟catid里面的栏目ID模型一致

inlist 列表模式用于获取多条单页模型的文章。例如catid="1"栏目ID为1是单页栏目，那么默认情况下获取的是一条单页内容，获取该单页下所有单页栏目里的文档则可以使用该参数。示例如下：

```
开启列表模式, 获取当前栏目下5条单页列表  
{hkcms:content catid="1" inlist="1" num="5"}  
<div>  
    {hkcms:raw name="$item['content']" /}  
</div>
```

```
</div>
{/hkcms:content}
```

more 用于开启列表页获取副表字段。模型管理里面字段管理可以看到字段是否是主表还是副表，后续版本默认有副表字段无需手动开启

```
开启副表访问
{hkcms:content model="1" more="1" num="5"}
副表内容：{$item.content}
{/hkcms:content}
```

flag 用于输出特定的文档属性，例如在文档属性里面新增了hot属性，那么写法如下

```
输出模型ID为1且文章带有hot属性的记录
{hkcms:content flag="hot" model="1" num="5"}
标题：{$item.title}
{/hkcms:content}
```

flag属性里面填写的是下面图片中的属性值

<input type="checkbox"/>	排序	标题	属性值
<input checked="" type="checkbox"/>	1	最新	new
<input type="checkbox"/>	1	热门推荐	hot
<input type="checkbox"/>	1	置顶	top

1. flag属性多个使用：and 或 or 关键字，例如获取同时拥有 **最新(new)** 和 **热门推荐(hot)** 的属性文档

```
// and 是同时满足，注意两边的空格
{hkcms:content catid="9" flag="new and hot"}
标题：{$item.title}<br/>
{/hkcms:content}
```

2. 获取拥有 **最新(new)** 或者有 **热门推荐(hot)** 的属性文档

```
// or 只需要包含其中一个属性即可，注意两边的空格，多个写法：flag="new or top or top" 以此类推
{hkcms:content catid="9" flag="new or top"}
```

```
标题：{$item.title}<br/>
{/hkcms:content}
```

`id` 用于改变循环变量默认是 `$item`，例如改成`$vo`则写成`id="vo"`无需加`$`。常用于多个标签嵌套时使用循环变量而不受影响

```
{hkcms:content catid="9" id="vo"}
标题：{$vo.title}<br />
{/hkcms:content}
```

`empty` 为空(没有数据)提示标签，不支持传入HTML，但可以传入变量形式使用html

1. 示例如下：栏目id为111，里面没有文档时页面上显示 文档内容不存在

```
{hkcms:content catid="111" empty="文档内容不存在"}
标题：{$item.title}<br />
{/hkcms:content}
```

2. 传入变量的形式插入html标签

```
{php}$empty = '<p align="center">- 没有更多数据了 -</p>';{/php}
{hkcms:content catid="111" empty="$empty"}
标题：{$item.title}<br />
{/hkcms:content}
```

`cache` 设置缓存时间，默认是3600秒

```
// 设置7200秒
{hkcms:content catid="9" cache="7200"}
标题：{$item.title}<br />
{/hkcms:content}
```

`mod` 属性用于每隔N行输出的内容，例如控制N条记录的换行、或增加hr等，例如下面的例子每输出5个栏目增加一个下划线标签

```
{hkcms:content catid="9" mod="5"}
标题：{$item.title}<br/>
{if $mod==5} // 启用mod后, 会有$mod变量
<hr>
{/if}
{/hkcms:content}
```

1. 输出奇数偶数记录，例如：mod="2"，标签内会有\$mod变量，里面的值每次计算到2的时候重新计数例如
(1, 2 | 1, 2 | 1, 2)

```
{hkcms:content catid="9" mod="2"}
{if $mod==2}
偶数：{$item.title} <br>
{else /}
奇数：{$item.title} <br>
{/if}
{/hkcms:content}
```

key 参数用于指定循环 \$i 的变量，只有标签嵌套时可能需要重新命名\$key

```
{hkcms:content catid="9"}
// $i 是从1开始的。
<p>序号：第{$i}篇文档 </p>
<p>标题：{$item.title} </p>
{/hkcms:content}
```

改变\$i的名称

```
{hkcms:content catid="9" key="ii"}
<p>序号：第{$ii}篇文档 </p>
<p>标题：{$item.title} </p>
{/hkcms:content}
```

常用示例

- 获取栏目ID为1的列表

```
{hkcms:content catid="1"}
<p>计数变量：$i, 从1开始：第{$i}篇</p>
<p>数组索引：{$key}</p>
<p>标题：{$item.title}</p>
<p>封面：{$item.thumb}</p>
<p>URL：{$item.url}</p>
{/hkcms:content}

// 传入变量的形式
{hkcms:content catid="$Cate.id"}
<p>标题：{$item.title}</p>
<p>封面：{$item.thumb}</p>
<p>URL：{$item.url}</p>
{/hkcms:content}
```

- 根据模型ID获取

```
{hkcms:content model="1"}  
<p>循环变量, 从1开始: {$i}</p>  
<p>数组索引值: {$key}</p>  
<p>标题: {$item.title}</p>  
<p>封面: {$item.thumb}</p>  
<p>URL: {$item.url}</p>  
<p>作者: {$item.author}</p>  
<p>关键字: {$item.keywords}</p>  
{/hkcms:content}
```

- 只获取6条数据

```
{hkcms:content model="1" num="6"}  
<p>循环变量, 从1开始: {$i}</p>  
<p>数组索引值: {$key}</p>  
<p>标题: {$item.title}</p>  
<p>封面: {$item.thumb}</p>  
<p>URL: {$item.url}</p>  
<p>作者: {$item.author}</p>  
<p>关键字: {$item.keywords}</p>  
{/hkcms:content}
```

获取文章的分类信息与模型信息

```
{hkcms:content model="1"}  
<p>标题: {$item.title}</p>  
<p>封面: {$item.thumb}</p>  
<p>URL: {$item.url}</p>  
<div>栏目名称: {$item.category.title}</div>  
<div>栏目英文: {$item.category.name}</div>  
<div>模型名称: {$item.model.name}</div>  
<div>模型类型: {$item.model.type}</div>  
{/hkcms:content}
```

- 获取副表信息、后续版本默认包含

```
{hkcms:content catid="1" more="1"}  
<p>标题: {$item.title}</p>  
<p>封面: {$item.thumb}</p>  
<p>浏览量: {$item.views}</p>  
{/hkcms:content}
```


- 获取包含栏目子级的列表数据

```
注：默认是包含的，如果不想包含 insub="0" 即可
{hkcms:content catid="1" insub="1"}
<p>标题：{$item.title}</p>
{/hkcms:content}
```

- 数据分页

开启分页，每页只显示2条记录

```
{hkcms:content catid="1" page="1" num="2"}
<p>标题：{$item.title}</p>
{/hkcms:content}
<div>
  //配合分页标签
  {hkcms:contentpage item="home,pre,pageno,next,last,info,jump" /}
</div>
```

- 文档属性

flag="top and new"表示只查出同时满足top和new属性值，flags="top or new"表示只要查出有top或new属性值，注意and和or两边的空格

```
{hkcms:content num="6" page="1" flag="top or ceshi"}
  标题：{$item.title}
  循环文档属性(包含所有属性字段)：
  {hkcms:volist name="item.flags" id="vo"}
  <span>{$vo['title']}</span>
  {/hkcms:volist}
  原始值：{$item.flags_raw}
{/hkcms:content}
```

字段调用

更多字段可在模型管理》字段管理里面找到字段的名字，以下仅包含主要的字段

注意：其中 **\$item** 变量根据content标签的id参数影响，默认是：\$item。\$Info 是详情页固定调用方式

字段名	说明	调用方式	详情页
id	文档ID	{ \$item.id }	{ \$Info.id }
category_id	文档所属栏目ID	{ \$item.category_id }	{ \$Info.category_id }
category_ids	文档所属父栏目ID，例如：1,2,3	{ \$item.category_ids }	{ \$Info.category_ids }

	逗号分隔开	ids}	ds}
model_id	文档所属模型ID	{item.model_id}	{Info.model_id}
flags	文档属性，数组，使用volist标签循环	{item.flags}	{Info.flags}
- title	使用volist循环后，里面写{item.title}，文档属性标题	-	-
- name	文档属性标题	-	-
flags_raw	文档属性，多个属性逗号分隔，例如：top,new	{item.flags_raw}	{Info.flags_raw}
lang	语言标识	{item.lang}	{Info.lang}
publish_time	发布时间，格式是秒，使用date日期格式化标签，或使用下方的字段	{item.publish_time}	{Info.publish_time}
publish_time_text	发布时间，格式是年月日时分秒	{item.publish_time_text}	{Info.publish_time_text}
fullurl	包含完整域名的URL	{item.fullurl}	{Info.fullurl}

contentpage 内容列表分页

内容列表分页标签

生成页码的分页标签，需要配合内容列表标签使用

2.1.220220版本起，新增：可用于内容页，对内容分页

格式示例

```
{hkcms:contentpage item="home,pre,pageno,next,last,info,jump" /}
```

参数

参数名称	可选值	默认值	是否必填	说明
item	home-首页, pre-上一页, pageno-页码, next-下一页, last-尾页, info-数量信息, jump-跳转页码	-	是	分页格式
mobile_item	同上	-	-	手机端时显示的页码格式，该参数有值时，item则失效。注意：只有在移动端时才有效
size	-	5	-	显示页码数量。
home	-	首页	-	首页按钮显示的文字。
pre	-	上一页	-	上一页按钮显示的文字。
next	-	下一页	-	下一页按钮显示的文字。

last	-	尾页	-	尾页按钮显示的文字。
info	-	共 %s 页 / %s 条数据	-	更改默认数量显示信息，第一个%s是总页数，第二个%s是总条数
emptxt	-	共 %s 页 / %s 条数据	-	不够分页时显示提示信息
hasemp	true/false	true	-	当列表不够分页时会显示提示信息(跟item的info有区别)，true-显示提示信息，false-不显示

参数说明

`item` 属性用于控制分页显示项，如下图所示



1. 只显示上一页下一页按钮。

```
{hkcms:contentpage item="pre,next" /}
```

`mobile_item` 属性与 `item` 一致，区别在于仅控制移动端的显示

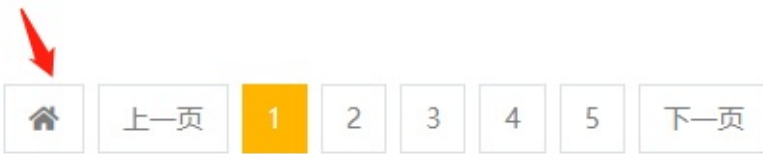
```
// 移动端只显示上一页、页码与下一页
{hkcms:contentpage item="pre,next,last,info,jump" mobile_item="pre,pageno,next" /}
```

`size` 控制页码数量，至少五个页码。

```
{hkcms:contentpage item="pre,next,last,info,jump,pageno" size="6" /}
```

`home` 控制首页按钮文字，例如首页文字改成字体图标

```
// 注意：home="" 里面不能出现双引号、'fa fa-home' 是fontawesome图标库  
{hkcms:contentpage item="home,pre,next,last,pageno" home="<i class='fa fa-home'>></i>" /}
```



`pre` 控制上一页按钮文字、`next` 控制下一页按钮文字、`last` 控制尾页按钮文字、设置方法同上一致

`info` 属性是控制分页信息。默认格式为：共 %s 页 / %s 条数据，第一个%s是总页数，第二个%s是总条数，例如上面图片中的共 2 页 11 条数据

例如：分页信息改为：共 2 页,总共 11 个产品

```
{hkcms:contentpage item="pre,next,pageno,info" info="共 %s 页,总共 %s 个产品" /}
```

`emptxt` 属性是对不够分页显示的提示信息，如下

共 1 页 2 条数据

修改方式同上

`hasemp` 属性是对不够分页的提示信息控制显示隐藏，true-显示提示信息，false-不显示

```
// 默认显示，hasemp="false" 为隐藏  
{hkcms:contentpage item="pre,next,pageno,info" hasemp="false" /}
```

示例

- 完整示例
每页显示10条记录

```
// 结合内容列表标签  
{hkcms:content catid="1" page="1" num="10"}
```

```
<p>标题：{$item.title}</p>
{/hkcms:content}
<div>
  {hkcms:contentpage item="home,pre,pageno,next,last,info,jump" /}
</div>
```

结果图，生成的html有带class，可自行写class样式定义分页风格。根据自己的需要，可自定义分页显示的项。



- 移动端覆盖默认的格式

如下代码所示，在移动端是，mobile_item覆盖item格式。移动端只显示上一页，页码，下一页。

```
{hkcms:contentpage item="home,pre,pageno,next,last,info" mobile_item="pre,pageno,next" pre="&laquo;" next="&raquo;" /}
```

- 修改不够分页提示信息

默认显示共1页，1条数据，使用emptxt属性，可修改

```
{hkcms:contentpage item="pre,pageno,next" mobile_item="pre,pageno,next" emptxt="共%s页" pre="&laquo;" next="&raquo;" /}
```

- 隐藏不够分页的提示信息。

使用hasemp参数

```
{hkcms:contentpage item="pre,pageno,next" mobile_item="pre,pageno,next" hasemp="false" pre="&laquo;" next="&raquo;" /}
```

wfpage 瀑布流分页

瀑布流分页

瀑布流ajax分页，结合content标签对列表分页。

参数

参数名称	可选值	默认值	是否必填	说明
auto	1/0	0	-	1=自动加载，0=手动触发
num	-	默认同content标签的num属性保持一致	-	每次加载条数
icon	-	加载中...	-	加载中的提示文字，也可以是图片
empty	-	暂无更多数据	-	最后一页的提示信息

参数说明

auto 是否自动加载属性，1=滚动到位置时，开始加载下一页。默认是0=手动点击加载数据，写法如下：调用标签后会有 `{ $item }` 变量，将该变量赋值到标签上即可。

```
{hkcms:wfpage auto="1"}
<div class="text-center">
  <a href="javascript:void(0);" class="btn btn-primary" { $item.click }>点击加载更多</a>
</div>
{/hkcms:wfpage}
```

num 每页记录数属性，默认同content标签的num一致。

```
{hkcms:wfpage num="2"}
<div class="text-center">
  <a href="javascript:void(0);" class="btn btn-primary" { $item.click }>点击加载更多</a>
</div>
{/hkcms:wfpage}
```

icon 加载显示的内容，默认是 加载中... ，除了文字，也支持html，html写法如下：

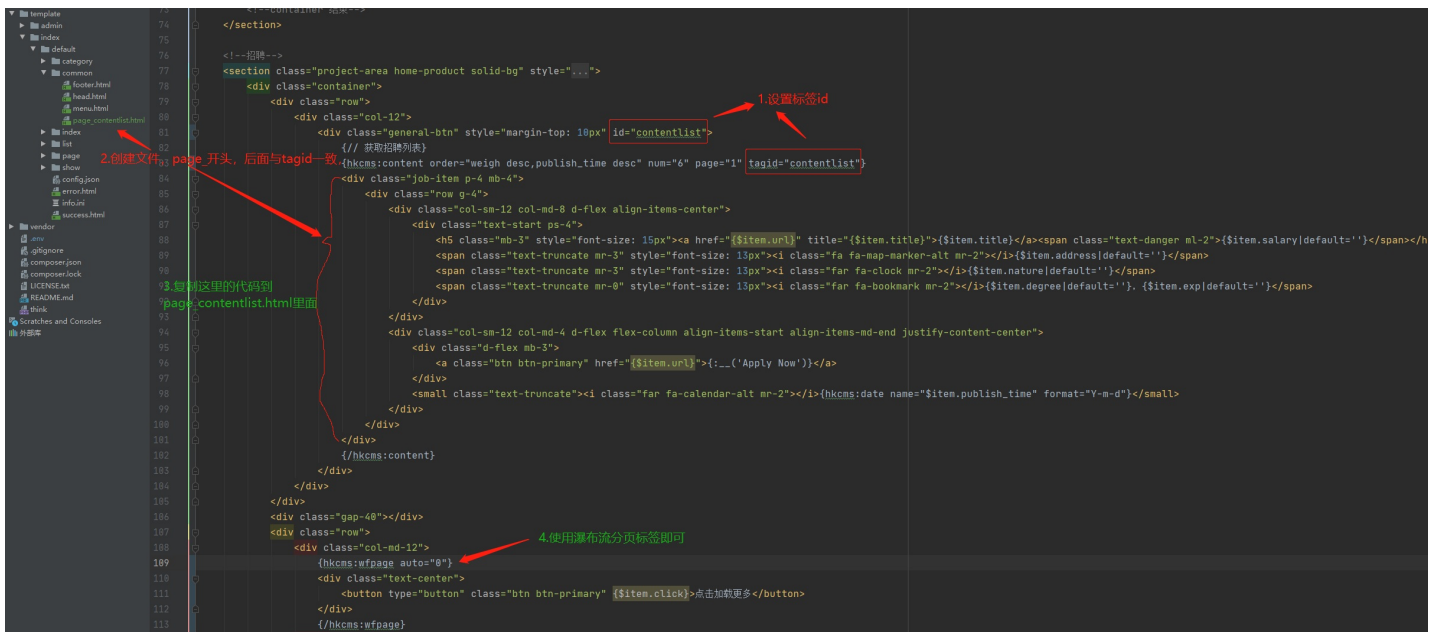
```
{php}$icon = '';{/php}
{hkcms:wfpage num="2" icon="$icon"}
<div class="text-center">
  <a href="javascript:void(0);" class="btn btn-primary" {$item.click}>点击加载
  更多</a>
</div>
{/hkcms:wfpage}
```

使用说明

详细步骤如下，示例根据默认模板人才招聘演示：

1. 设置 content 内容列表标签的上级html标签 id="contentlist"，这个名字可以随意取，设置后跟tagid属性保持一致
2. 在你的模板目录新增 common/page_contentlist.html 文件，里面的内容就放 content 标签包裹的内容。
page_contentlist文件名称格式为：page_(tagid属性里填写的名称).html
3. 然后就可以调用wfpage瀑布流分页标签

效果图，看不清的话鼠标到图片右键保存或者新窗口打开：



成功后会有个按钮显示。



简单代码示例：


```
// 设置ID、与下面的tagid一致
<div id="contentlist">
  {hkcms:content catid="$Cate['id']" num="6" page="1" tagid="contentlist"}
  <p><a href="{ $item.url }">标题 : { $item.title }</a></p>
  {/hkcms:content}
</div>
```

引用瀑布流分页标签

```
{hkcms:wfpage }
<div class="text-center">
  <button type="button" class="btn btn-primary" { $item.click }>点击加载更多</but
ton>
</div>
{/hkcms:wfpage }
```

4. common/page_contentlist.html 文件内容就是hkcms:content 标签包裹的内容

```
<p><a href="{ $item.url }">标题 : { $item.title }</a></p>
```

若想自定义分页处理函数，可在页面上定义get_page_list方法：

```
<script>
  function get_page_list(obj, key) {
  }
</script>
```

filter 筛选标签

- [filter 筛选标签](#)
- [参数](#)
- [使用注意](#)
- [参数说明](#)
- [字段调用](#)

filter 筛选标签

获取后台模型设定的筛选字段。常用于列表筛选

参数

参数名称	可选值	默认值	是否必填	说明
id	-	\$item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
currentstyle	-	active	-	指定选中的 class 样式
multiple	1/0	0	-	是否允许多选，1-开启多选，0-关闭
field	开启筛选的字段	-	-	限制显示筛选字段
alltxt	-	全部	-	第一个选项显示的文字，例如： alltxt="不限"，等于off 则不显示

使用注意

要使用筛选标签，请先在后台指定筛选的字段

filter 筛选标签

控制台 模型管理 文章模型字段

+添加 删除 设置

排序	字段	标题	字段类型	长度	默认值	主表字段	分组	允许投稿	是否排序	筛选	状态	操作
1	seotitle	seo标题	text	200		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
0	keywords	关键词	text	200		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
0	description	描述	textarea	250		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
10	author	作者	text	250	小编	主表	其他	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
0	option	选项	select	250		副表	其他	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
19	thumb	封面图	image	100		主表	常规	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
0	ceshi	测试	number	11	10	副表	常规	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
0	option2	选项2	select	250		副表	常规	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑 删除
3	views	浏览量	number	11	0	主表	其他	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	编辑
2	publish_time	发布时间	datetime	0		主表	其他	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	编辑
1	show_tpl	内容模板	text	100	show.html	主表	其他	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	编辑
20	title	标题	text	200		主表	常规	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	编辑
6	content	内容	editor	0		副表	常规	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	设置 编辑
1	weigh	排序	number	11	0	主表	常规	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	编辑
0	status	状态	radio	250	normal	主表	常规	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	编辑

点击切换

如果你的字段类型非下拉列表、单选、复选，你需要指定选项类别。如下图

添加
— ✕ ✕

提示信息:

错误信息:

附加属性:

排序:

是否筛选: 否 是

筛选列表:

为空使用选项列表, 没有选项列表这里必须填写

是否排序: 否 是

状态: 正常 隐藏

提交
重置

效果图:

选项: 全部 标题1 标题2选项2: 全部 内容 内容2

参数说明

`id` 参数, 用于指定循环的变量名, 默认是 `$item`, 设置值 `id="obj"`, 变量名即为`$obj`。

```

{hkcms:filter id="obj"}
  选项名字: {$obj.title}:

  // 选项列表, 使用volist循环标签循环输出。
  {hkcms:volist name="obj.option" id="vo"}
    <a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a>
  {/hkcms:volist}
{/hkcms:filter}

```

`currentstyle` 指定选择项的class，默认class名称为 `active`

```
{hkcms:filter id="obj" currentstyle="newclass"}
  选项名字: {$obj.title}
  {hkcms:volist name="obj.option" id="vo"}
  // currentstyle="newclass", 当该选项选中时, $vo.currentstyle的值为newclass, 没
  选择的为空。
  <a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a>
{/hkcms:volist}
{/hkcms:filter}
```

`multiple` 参数开启多选, 1=开启, 0=关闭

```
{hkcms:filter multiple="1" }
<div class="pt-2 pb-2 d-flex mb-1 mt-1">
  <div class="">{$item.title}: </div>
  <div class="filter">
    <ul class="clearfix mb-0">
      {hkcms:volist name="item.option" id="vo"}
      <li><a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a></
li>
      {/hkcms:volist}
    </ul>
  </div>
</div>
{/hkcms:filter}
```

`field` 限定字段, 当有时候不想显示所有筛选字段, 在这里可以指定筛选的字段。多个字段使用英文逗号分隔。如下:

```
// 只显示option和option2的字段
{hkcms:filter field="option,option2" }
<div class="pt-2 pb-2 d-flex mb-1 mt-1">
  <div class="">{$item.title}: </div>
  <div class="filter">
    <ul class="clearfix mb-0">
      {hkcms:volist name="item.option" id="vo"}
      <li><a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a></
li>
      {/hkcms:volist}
    </ul>
  </div>
</div>
{/hkcms:filter}
```

`alltxt` 指定第一个重置选项的名称，例如通常有：全部、不限等，默认显示 `全部`，如果不想显示等于 `off` 即可关闭

```
{hkcms:filter alltxt="不限" }
<div class="pt-2 pb-2 d-flex mb-1 mt-1">
  <div class="">{$item.title}: </div>
  <div class="filter">
    <ul class="clearfix mb-0">
      {hkcms:volist name="item.option" id="vo"}
      <li><a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a></li>
    </ul>
  </div>
</div>
{/hkcms:filter}
```

效果

选项: **不限** 标题1 标题2

选项2: **不限** 内容 内容2

筛选文字多语言化，注意前提写好多语言。多语言配置可前往多语言章节了解。

```
{hkcms:filter alltxt="lang('All')" }
<div class="pt-2 pb-2 d-flex mb-1 mt-1">
  // 多语言标题，格式为：{:__(这里写变量)}
  <div class="">{:__($item.title)}: </div>
  <div class="filter">
    <ul class="clearfix mb-0">
      {hkcms:volist name="item.option" id="vo"}
      <li><a href="{ $vo.url}" class="{ $vo.currentstyle}">{:__($vo.title)}</a></li>
    </ul>
  </div>
</div>
{/hkcms:filter}
```

关闭第一的重置选项

```
{hkcms:filter alltxt="off" }
<div class="pt-2 pb-2 d-flex mb-1 mt-1">
```

```

<div class="">{$item.title}: </div>
<div class="filter">
  <ul class="clearfix mb-0">
    {hkcms:volist name="item.option" id="vo"}
    <li><a href="{ $vo.url}" class="{ $vo.currentstyle}">{$vo.title}</a></
li>
    {/hkcms:volist}
  </ul>
</div>
</div>
{/hkcms:filter}

```

效果

选项: **标题1** 标题2选项2: **内容** 内容2

字段调用

更多字段可在《模型管理》字段管理里面找到字段的名字，以下仅包含主要的字段

注意：其中 `$item` 变量根据filter标签的id参数影响，默认是：`$item`。

字段名	说明	调用方式
title	字段标题	{ \$item.title }
name	字段名	{ \$item.name }
option	数组，里面是筛选的选项,使用volist循环标签	{hkcms:volist name="item.option" id="vo"} {/hkcms:volist}

option 内容，id="vo"

字段名	说明	调用方式
title	标题	{ \$vo.title }
name	选项值	{ \$vo.name }
url	链接地址	{ \$vo.url }
currentstyle	高亮，选中的选项有class，用于css定义颜色高亮	{ \$vo.currentstyle }

order 排序标签

order 排序标签

用于获取后台模型字段设置的排序字段

参数

参数名称	可选值	默认值	是否必填	说明
id	-	\$item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
currentstyle	-	active	-	指定选中的 class 样式

示例

1. 第一步先指定排序字段，如下

<input type="checkbox"/>	排序	字段	标题	字段类型	长度	默认值	主表字段	分组	允许投稿	是否排序	筛选	状态	操作
<input type="checkbox"/>	1	seotitle	seo标题	text	200		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	0	keywords	关键词	text	200		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	0	description	描述	textarea	250		主表	SEO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	10	author	作者	text	250	小编	主表	其他	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	19	thumb	封面图	image	100		主表	常规	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	0	ceshi	测试	number	11	10	副表	常规	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	3	views	浏览量	number	11	0	主表	其他	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	
<input type="checkbox"/>	2	publish_time	发布时间	datetime	0		主表	其他	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	
<input type="checkbox"/>	1	show_tpl	内容模板	text	100	show.html	主表	其他	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	

2. 在列表页，输出

currentstyle 值为 active，自动给当前排序加上 active

```
{hkcms:order currentstyle="active"}
<a href="{ $item.url }" class="{ $item.currentstyle }">{ $item.title }</a>
{/hkcms:order}
```


order 排序标签

效果

默认 浏览量 发布时间 测试

字段调用

注意：其中 `$item` 变量根据id参数影响，默认是：`$item`

字段名	说明	调用方式
title	字段标题	<code>{ \$item.title }</code>
field	字段名称	<code>{ \$item.field }</code>
order	asc(升序)/desc(降序)	<code>{ \$item.order }</code>
url	链接地址	<code>{ \$item.url }</code>
currentstyle	用于高亮，只有当前选中的才有该class	<code>{ \$item.currentstyle }</code>

内容标签

guestbook 留言(表单)标签 (废弃)

form 表单标签

pre 上一篇标签

next 下一篇标签

prenext 上一篇下一篇(高级)标签

fileinfo 获取附件信息(文件下载)

内容分页

raw 格式化编辑器文本(即将废弃)

guestbook 留言(表单)标签 (废弃)

留言模型(表单)标签

用于获取在后台创建的留言表单输出到前台

v2.2.1.20220603 版本起不在使用该标签

参数

参数名称	可选值	默认值	是否必填	说明
id	-	item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.action } 变成 { \$obj.action }
catid	-	-	是	栏目ID，也可传入变量
captcha	1	-	-	验证码，1-开启验证码
cache	-	3600秒	-	缓存时间，默认3600秒

参数说明

- id，指定循环里的变量名，id="obj"，如下

```
{hkcms:guestbook catid="34" id="obj"}
// 默认是$item, field的值来自于您创建的模型字段参考下图
<input type="text" class="form-control" name="{ $obj.field.name|default=''}" />
{/hkcms:guestbook}
```

☰ << 控制台 内容管理 模型管理 留言板字段

🔄 + 添加 删除 设置 ▾

<input type="checkbox"/>	排序	字段	标题	字段类型	长度
<input type="checkbox"/>	0	name	姓名	text	100
<input type="checkbox"/>	0	email	邮箱	text	250
<input type="checkbox"/>	0	phone	手机	text	20
<input type="checkbox"/>	0	content	内容	textarea	250

- catid, 留言栏目ID, 必填
- captcha, 验证码, 值为1就是开启验证码, 其中的catid可以以传入变量, 例如每个栏目页会有\$Cate栏目变量, 写法: `catid="$Cate['id']"` 或 `catid="$Cate.id"`

```
{hkcms:guestbook catid="34" captcha="1"}
<form action="{ $item.action}" method="post">
  <div class="row g-2 mb-3">
    <div class="col-md-4">
      <input class="form-control" id="captcha" name="{ $item.captcha.field}" />
      <label for="content">{:lang('Verify code')}</label>
    </div>
    <div class="col-md-2">
      <a href="javascript:changeVer();">
        
      </a>
    </div>
  </div>
  <button type="submit" class="btn btn-primary">{:lang("Submit")}</button>
</form>
{/hkcms:guestbook}
```

完整示例

```
<div class="col-xl-12 my-3">
```

```

{ //captcha=1, 开启验证码之后在下面自定义验证码表单}
{hkcms:guestbook catid="34" captcha="1"}
{ // 获取栏目名称}
<h5 class="my-3">{$item['category']['title']??' '}</h5>
{ // 请求地址}
<form action="{ $item.action}" method="post">
  <div class="row g-2 mb-3">
    <div class="col-md">
      <div class="form-floating">
        <input type="text" class="form-control" name="{ $item.field.
name|default=' '}" id="name">
        <label for="name">{:lang('Name')}</label>
      </div>
    </div>
    <div class="col-md">
      <div class="form-floating">
        <input type="text" class="form-control" name="{ $item.field.
phone|default=' '}" id="phone">
        <label for="phone">{:lang('Telephone')}</label>
      </div>
    </div>
  </div>
  <div class="row g-2 mb-3">
    <div class="col-md">
      <div class="form-floating">
        <textarea class="form-control" id="content" name="{ $item.fi
eld.content|default=' '}"></textarea>
        <label for="content">{:lang('Content')}</label>
      </div>
    </div>
  </div>
  <div class="row g-2 mb-3">
    <div class="col-md-4">
      <div class="form-floating">
        <input class="form-control" id="captcha" name="{ $item.captc
ha.field}" />
        <label for="content">{:lang('Verify code')}</label>
      </div>
    </div>
    <div class="col-md-2">
      <a href="javascript:changeVer();"></a>
    </div>
  </div>
  <button type="submit" class="btn btn-primary">{:lang("Submit")}</button>
</form>
{/hkcms:guestbook}
</div>

```



数据字典

键名	值	说明
action		请求地址
field	数组	放入表单name里面，请求的参数名
all	数组	所有留言板字段
category	数组	栏目信息
captcha	数组，包含url,field	验证码

form 表单标签

form 表单标签

用于获取在后台创建的留言表单输出到前台，注意该标签不是自动生成表单

支持 v2.2.1.20220603 及以上版本

参数

参数名称	可选值	默认值	是否必填	说明
id	-	item	-	指定循环里的变量名，例如 id='obj'，模板调用如： {item.action} 变成 {obj.action}
catid	-	-	是	栏目ID，也可传入变量
attr	-	-	-	form标签属性，注意attr里面使用单引号
cache	-	3600秒	-	缓存时间，默认3600秒

示例

在任意页面调用表单，例如在联系我们调用在线留言表单

第一步：后台模型管理，新建留言表单，如果安装的是演示数据，那么已经存在留言表单了，创建好后点击字段管理进入，如下图所示

后台管理

控制台 在线 本地 模型管理 留言表单字段

+添加 删除 设置

排序	字段	标题	字段类型	长度	默认值	主表
0	name	姓名	text	100		主
0	email	邮箱	text	250		主
0	phone	手机	text	20		主
0	content	内容	textarea	250		主

这是系统演示数据默认的字 段 在这里添加或删除你的字段

第二部：绑定到栏目

第三步：来到模板页面，在你要调用的页面上调用form表单，catid是栏目ID，参考默认模板 `page_contact`

页面里面的调用

详情看里面的注释

```
// attr属性里面的字符内容，原样输出到form标签上，下面给了个class, id, 以及enctype属性，默认已经有action、method属性了
{hkcms:form catid="34" attr="class='frm-data' id='frm' enctype='multipart/form-data'"}
// $item.category.title, 获取当前留言表单的栏目信息，$item.category里面是栏目字段
<h3 class="column-title">{$item.category.title}</h3>
<div class="form-group">
  // 语言包，不需要语言包就直接写中文即可
  <label>{:__( 'Name' )}</label>
  // $item.field 里面就是提交的字段，看上图，获取name 姓名，就是$item.field.name
  <input class="form-control form-control-name" name="{ $item.field.name | default=''}" id="name" placeholder="" type="text" required="">
</div>
<div class="form-group">
  <label>{:__( 'Mail' )}</label>
  <input class="form-control form-control-email" name="{ $item.field.email | default=''}" id="email" placeholder="" type="email" required="">
</div>
<div class="form-group">
  <label>{:__( 'Telephone' )}</label>
  <input class="form-control form-control-subject" name="{ $item.field.phone | default=''}" id="subject" placeholder="" required="">
</div>
```



```

<div class="form-group">
  <label>{:__('Content')}</label>
  <textarea class="form-control form-control-message" name="{ $item.field.content|default=''}" id="message" placeholder="" rows="10" required=""></textarea>
</div>
<div class="form-group">
  <label>{:__('Verify code')}</label>
  <div class="d-flex align-items-center">
    { // 输入框, 可以自己写个输入框来实现自定义, 样式的调整: 里面有两个class分别是: form-control(bootstrap class) form-captcha, 按这两个class调整即可}
    <div class="mr-3">{$item.captcha.input}</div>
    { // 验证码btn, 图文验证码时是图片, 其他的则是发送按钮}
    <div class="captcha" style="width: 138px">{$item.captcha.btn}</div>
  </div>
</div>
<div class="form-group">
  <button class="btn btn-primary blank" type="submit">{:__("Submit")}</button>
</div>
{/hkcms:form}

```

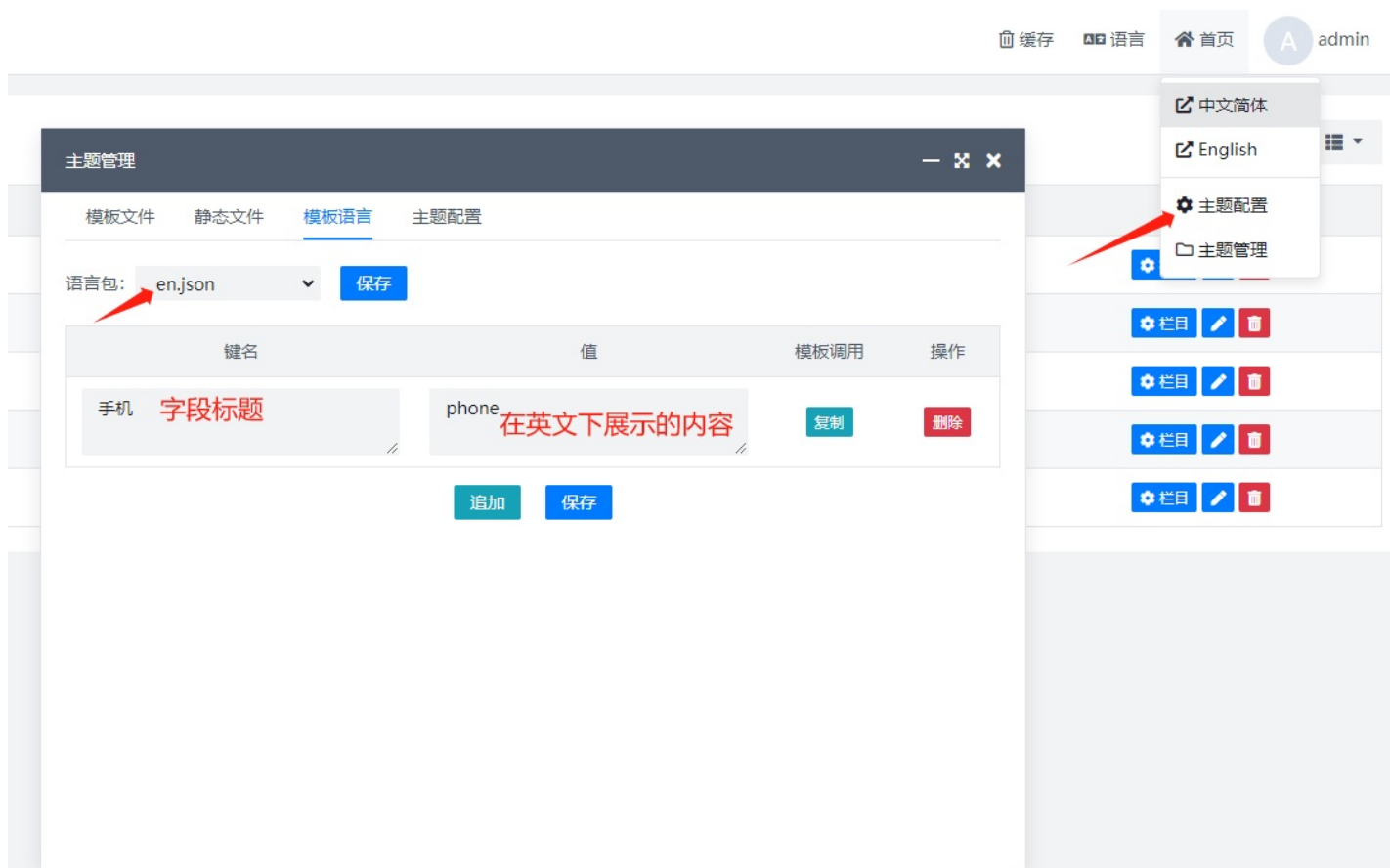
提示信息多语言化

字段如下图所示

<input type="checkbox"/>	排序	字段	标题	字段类型
<input type="checkbox"/>	1	title	申请职位	text
<input type="checkbox"/>	2	name	姓名	text
<input type="checkbox"/>	3	email	邮箱	text
<input type="checkbox"/>	4	phone	手机	text
<input type="checkbox"/>	5	content	内容	textarea

当你的字段标题是中文的情况下, 例如你有中英双语, 在中文预览下你无需操作, 在英文下你需要配置英文预览下显示的文字信息, 步骤如下

1. 在后台右上角进入主题管理如下图配置



配置完成后即可实现不同语言下提示不同的信息内容

验证码、表单配置

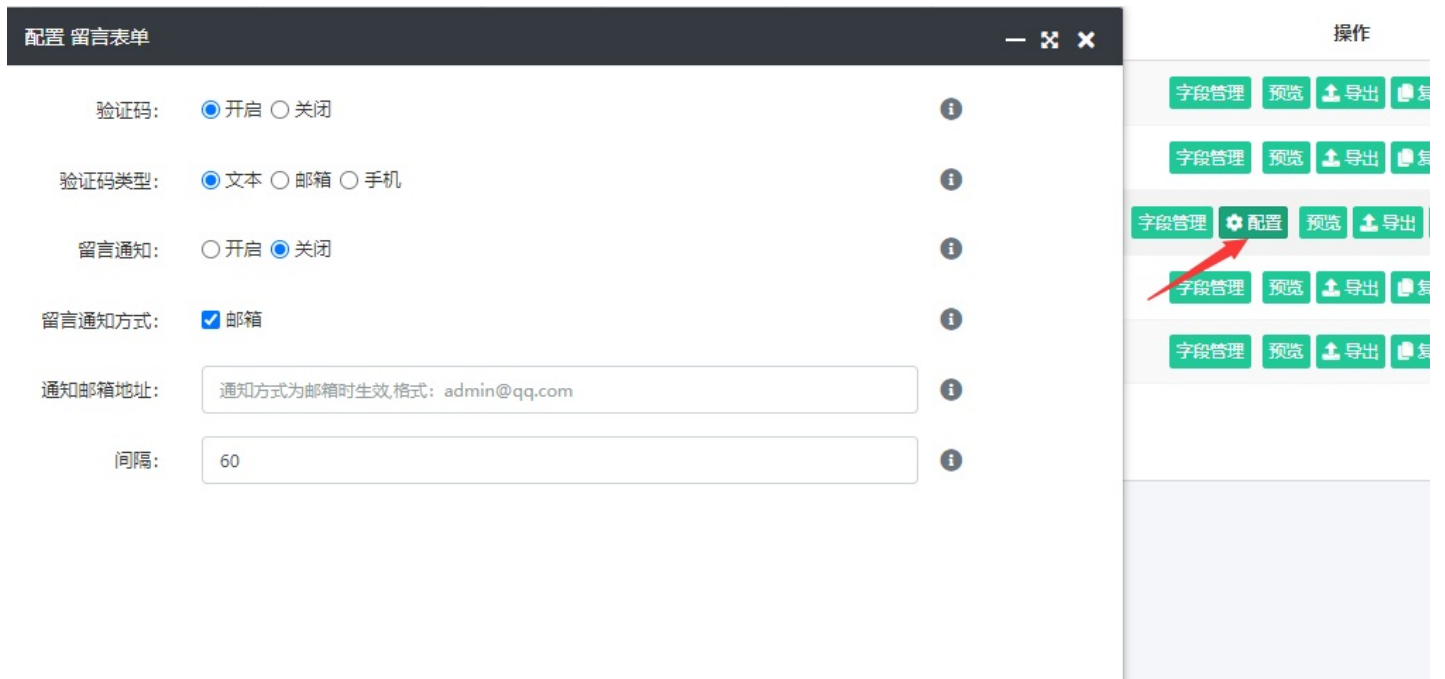
如上面完整示例

使用 `{ $item.captcha.input }` 可以获取到验证码，根据后台模型配置，会自动显示响应的输入框提示

使用 `{ $item.captcha.btn }` 自动根据后台模型配置，显示按钮或图文

确保邮箱验证码有email字段，手机验证码有phone字段

若要自定义可自行增加验证码控件。



邮箱、手机验证码控件生成后并不能实现点击，官方并没有默认集成JS代码，下面提供了演示代码，需要JQ的支持

如果你的模板使用JQ以及错误提示框（layer），那么复制下面的代码放到页面底部就可以直接实现验证码发送功能。

```
<script>
// 手机、邮箱验证码发送演示。需要jq、弹出框提示用的是layer
$(' [data-toggle="captcha_send"]').click(function (e) {
    var obj = this;
    if ($(obj).is('.disabled')) {
        return false;
    }

    // 发送类型, 手机/邮箱
    var type = $(obj).data('type');
    var frm = $(obj).parents('form'); // 获取当前form表单
    var txt = ''; // 提示信息
    var inter = 0; // 定时器

    if (type=='email') {
        txt = frm.find('input[name="row[email]"]').val(); // 获取邮箱, 确保里面
        有email字段
    } else if (type=='mobile') {
        txt = frm.find('input[name="row[phone]"]').val(); // 获取手机号
    }

    $(obj).addClass('disabled').text('发送中...');

    // 发送邮件
    $.ajax({
        type:"post",
```

```

url:$(obj).data('url'), // 请求地址
data:{type:$(obj).data('type'),obj:txt},
success: function (res) {
    if (res.code==200) {
        var seconds = 60;
        inter = setInterval(function () {
            seconds--;
            if (seconds<=0) {
                clearInterval(inter);
                $(obj).removeClass('disabled').text('发送验证码');
            } else {
                $(obj).text('重新发送 '+seconds+' s');
            }
        }, 1000)
        layer.msg('发送成功');
    } else {
        // 发送错误提示错误信息
        layer.msg(res.msg);
        $(obj).removeClass('disabled').text('发送验证码');
    }
},
error: function (res) {
    layer.msg('error');
    $(obj).removeClass('disabled').text('发送验证码');
}
});
})
</script>

```

数据字典

下面的字段使用方法，默认是{\$item.action} {\$item.field.name} 这样来调用，也可以直接使用<?php dump(\$item);?>输出所有字段到页面上查看详细信息。

一级字段列表

字段名	说明
action	请求地址
field	字段名称多行
captcha	验证码
all	字段多行，详细信息，见下方字段列表
category	栏目信息

`all` 字段列表，多行，有几个字段就有几行

字段名	说明

id	字段ID
model_id	所属模型ID
field_name	字段名
field_title	字段标题
form_type	字段类型
field_group	字段分组名，例如：常规，对应新增字段时的分组
length	字段长度
default_value	默认值
data_list	数量列表
max_number	最多数，数字文本框时有效
decimals	小数点
rules	规则
tips	提示
error_tips	错误提示
extend	扩展

pre 上一篇标签

上一篇标签

用于内容详情页，显示当前文章的上一篇内容，调用此标签会自动生成a标签。

格式

```
{hkcms:pre target="_blank" /}
```

参数

2.1.220220版本起，catid，id参数非必填项

参数名称	可选值	默认值	是否必填	说明
catid	栏目ID或变量	-	-	栏目ID，可以传入变量或指定ID
id	-	-	-	文章ID
target	_blank,_self,_parent,_top	-	-	a 标签链接打开方式，target参数值原样生成到a标签上
msg	-	-	-	当上一篇没有的情况下，显示的信息
field	-	title	-	默认显示文章标题，你可以自定义要显示的字段
len	-	-	-	限制标题长度
dot	-	-	-	超出的标题尾部拼接字符

示例

- 显示上一篇文章，如果没有了，就提示：已经没有了

```
{hkcms:pre msg="已经没有了" /}
```

html结果

```
<a href="/gnxw/rb/4.html" title="叙永七旬夫妇，“砍”出两公里盐马古道">叙永七旬夫妇，“砍”出两公里盐马古道</a>
```

- 限制标题长度为10，并加上省略号

```
{hkcms:pre len="10" dot="...." /}
```

- 显示其他字段

```
{hkcms:pre msg="已经没有了" field="description" /}
```

next 下一篇标签

下一篇标签

用于内容详情页，显示当前文章的下一篇内容。

格式

```
{hkcms:next target="_blank" msg="已经没有了" /}
```

参数

参数名称	可选值	默认值	是否必填	说明
catid	栏目ID或变量	-	-	栏目ID，可以传入变量或指定ID
id	-	-	-	文章ID
target	_blank,_self,_parent,_top	-	-	a 标签链接打开方式，target参数值原样生成到a 标签上
msg	-	-	-	当上一篇没有的情况下，显示的信息
field	-	title	-	默认显示文章标题，你可以自定义要显示的字段
len	-	-	-	限制标题长度
dot	-	-	-	超出的标题尾部拼接字符

使用方法同上一篇标签一致

prenext 上一篇下一篇(高级)标签

上下篇(数组)标签

在内容页，根据当前文档调取上一篇、下一篇文档。

示例

获取当前文档上一篇

```
{hkcms:prenext type="pre" len="10" dot="...."}
<div class="text-primary"><a href="{ $item.url}">{ $item.title}</a></div>
{else /}
<div>没有上一篇了</div>
{/hkcms:prenext}
```

参数

参数名称	可选值	默认值	是否必填	说明
type	pre/next	-	是	pre=调取上一篇，next调取下一篇
id	-	item	-	指定变量名，例如 id='obj'，模板调用如：{ \$item.action} 变成 { \$obj.action}
field	-	title	-	这里指定后结合 len、dot 参数生成
len	-	-	-	限制标题长度
dot	-	-	-	超出的标题尾部拼接字符
num	-	-	-	设置获取条数，默认是单条

参数说明

- type, 只能填写pre(上一篇)、next(下一篇)

获取上一篇数据

```
{hkcms:prenext type="pre"}  
<a href="{${item.url}}">${item.title}</a>  
{/hkcms:prenext}
```

- id, 指定变量, 默认是\$item, 如下改成obj后写法

```
{hkcms:prenext type="pre" id="obj"}  
<a href="{${obj.url}}">${obj.title}</a>  
{/hkcms:prenext}
```

- field, 这里结合len、dot参数, 如下: 设置标题长度为10超出的加上 "....."

```
{hkcms:prenext type="pre" len="10" dot="....."}  
<div>${item.title}</div>  
{/hkcms:prenext}
```

效果

演示数据演示数据数据.....

num 数量属性, 可获取多条上一篇或下一篇数据

```
{hkcms:prenext type="pre" num="10" len="10" dot="....."}  
<a href="{${item['url']}}">${item['title']}</a>  
{else /}  
<div>没有上一篇了</div>  
{/hkcms:prenext}
```

另外可以增加{else /}, 来判断在没有上一篇时的处理, 如下

```
{hkcms:prenext type="pre"}  
<div class="text-primary"><a href="{${item['url']}}">${item['title']}</a></div>  
{else /}  
<div>没有上一篇了</div>  
{/hkcms:prenext}
```

fileinfo 获取附件信息(文件下载)

fileinfo 获取附件信息

用于获取附件详细信息，同时支持文件下载，支持图片、图片(多)、文件上传、文件上传(多)组件。

参数

参数名称	可选值	默认值	是否必填	说明
id	-	\$item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
name	-	-	是	传入支持的组件字段变量
field	-	-	-	统计文件下载次数字段
aid	-	-	-	文档ID，在内容页(详情页)无需填写
model	-	-	-	模型ID，与 aid 同时使用，在内容页(详情页)无需填写
cache	-	3600秒	-	缓存时间，默认3600秒，对非分页数据有效

参数说明

`name` 属性，支持数组、字符串变量。例如：在模型增加files字段，类型是多文件上传。

```
// 假设你附件字段files, $Info是详情页都有的变量，是当前文档的信息
{hkcms:fileinfo name="$Info.files" id="vo"}
附件标题：{$vo.title}
下载地址：{$vo.url}
{/hkcms:fileinfo}
```

`field` 统计文件下载次数 的字段，该属性非必填项，例如：在模型字段新增 `download` 字段，用于记录该文档所有文件下载次数。

```
{hkcms:fileinfo name="$Info.files" id="vo" field="download"}
<a href="{ $vo.url}">本地下载</a>
{/hkcms:fileinfo}
```

`aid与model` 文档ID与模型id属性，在指定统计 `field` 属性时，通常在内容页无需填写，如果用在其他地方这里需要填写文档ID与模型ID，支持变量。

```
{hkcms:fileinfo name="$Info.files" id="vo" field="download" aid="1" model="2"}
<a href="{ $vo.url}">{ $vo.title}</a>
{/hkcms:fileinfo}
```

- 视频列表

```
{hkcms:fileinfo name="$Info.video" id="video" field="views"}
<video controls style="width: 200px" preload="auto">
  <source src="{ $video.path}" type="video/mp4">
</video>
{/hkcms:fileinfo}
```

- 图片列表

```
{hkcms:fileinfo name="$Info.thumb" id="vo" field="views"}

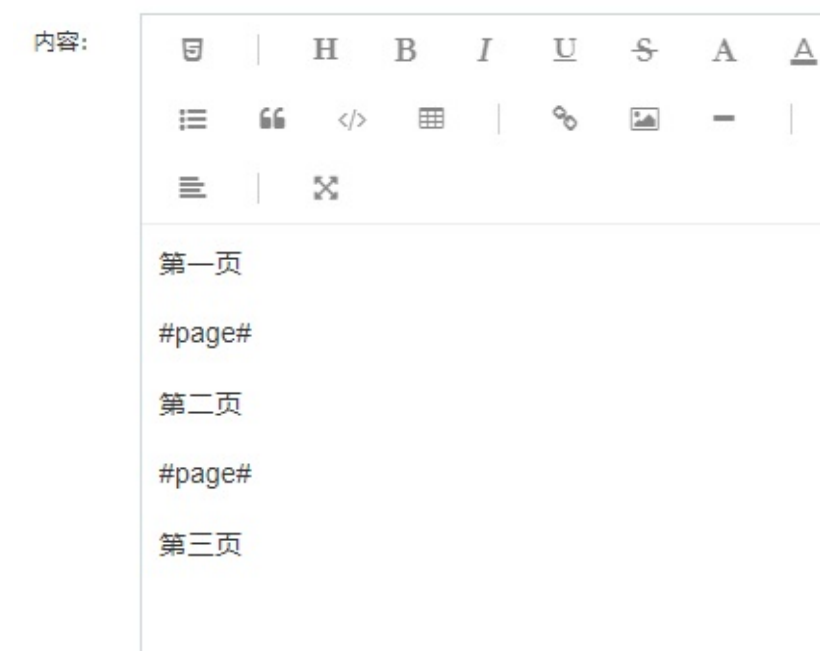
{/hkcms:fileinfo}
```

内容分页

内容分页

在详情内容增加：`#page#` 标识实现简单的内容分页

效果图



页面展示

发布日期: 2022-01-07

第一页



调用分页标签

参考下面的标签使用

[内容列表分页标签](#)

raw 格式化编辑器文本(即将废弃)

格式化编辑器文本

该标签针对编辑器内容正常输出到网页。适用于任何以编辑器类型字段的内容，或者是带有HTML标签的内容。

v2.2.1.20220603版本起，已取消限制，不再需要该标签转义，直接`{Info.content}`即可

格式示例

```
// 获取副表内容
{hkcms:raw name="$item.content"}

// 详情页
{hkcms:raw name="$Info.content"}
```

参数

参数名称	可选值	默认值	是否必填	说明
name	-	-	是	字段

示例

- 列表页获取副表content字段，content字段是编辑器里面的内容。

```
{hkcms:content catid="$Cate.id" num="8" page="1" more="1"}
  <div class="col-x1-3 p-2">
    <p class="card-text p-2"><a href="{ $item.url }">{ $item.title }</a></p>
  </div>
  // 对编辑器内容格式化，使能网页上正常显示编辑器内容
  {hkcms:raw name="$item.content"}
{/hkcms:content}
```

- 内容页，对副表content字段处理

```
<div class="content">{hkcms:raw name="$Info.content"}</div>
```

除了默认的content字段，您也可以对其他以编辑器编辑的文本或内容含有html，都可以使用该标签。

全局标签

seo 标签

channel 栏目标签

arcone 获取单条文档

breadcrumb 面包屑导航标签

adv 站点模块标签

date 日期格式化标签

substr字符(标题)截取标签

volist 循环标签

language 获取语言列表标签

lang 特定语言下显示内容标签

query 查询数据库内容

include 导入模板文件标签

原生PHP与php标签

map生成位置地图

thumb 生成缩略图

addons 插件检测

seo 标签

SEO标签

用于生成页面标题、关键字、描述标签

演示

调用方法

```
{hkcms:seo /}
```

结果

```
<title>标题</title>  
<meta name="keywords" content="关键字">  
<meta name="description" content="描述">
```

自定义方式：

全局站点配置，可以在后台站点配置修改

```
$site["home_title"]  
$site["keyword"]  
$site["description"]
```

列表页(list, category)、内容页(show)都会有的变量，可自行自定义

```
$seo_title  
$seo_keywords  
$seo_desc
```


channel 栏目标签

- [栏目标签](#)
- [格式示例](#)
- [参数](#)
- [参数说明](#)
- [示例](#)
- [数据字典](#)

栏目标签

用于调用栏目管理的数据

格式示例

```
// 获取顶级菜单栏目
{hkcms:channel type="top" where="ismenu=1"}
标题: {$item.title}
{/hkcms:channel}
```

2.1.220504_202205201800版本起，新增ismenu属性，如下

```
// 获取顶级菜单栏目、显示导航栏目菜单
{hkcms:channel type="top" ismenu="1"}
标题: {$item.title}
{/hkcms:channel}
```

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
name	栏目ID或变量	-	type值为peer或son时必填	-	栏目ID，可以传入变量或指定ID，2.2.3.220920版本起支持多个id，详情见下面参数说明

type	son peer top selftop selfson	-	-	-	指定获取的栏目层级 son-获取下级栏目 peer-表示同级栏目 top-表示顶级栏目 selftop获取顶级栏目的下的第一级全部栏目 selfson获取当前栏目的下级栏目，若下级栏目不存在依然返回当前同级栏目
num	例子： 10 或者 10,12	-	-	-	限制结果数量 例如： 10：表示获取10条 10,12：表从第10开始的12条记录
model	模型表ID	-	-	-	指定模型ID
where	-	-	-	-	数据库查询条件
order	-	weigh desc	-	-	排序方式，格式为 order="字段名称 desc" 其中desc是倒

					序，asc是升序
id	-	\$item	-	-	指定循环里的变量名，例如id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
empty	-	-	-	-	为空的时候输出提示，支持传入html
currentstyle	-	active	-	-	css class 类名，用于定义当前选中状态样式
cache	-	3600秒	-	-	缓存时间，默认3600秒
istotal	1	-	-	-	统计文档数量
ismenu	1	-	-	>=v2.1.220504	显示导航栏目菜单
lang	-	-	-	>=v2.2.1.20220603	多语言情况下获取指定语言的栏目
mod	-	-	-	>=v2.2.1.20220603	调用： { \$mod } 用于控制偶数输出、换行等，更多查看下面的示例
					指定循环值变量

key	-	-	-	>=v2.2.1. 20220603	名，默认为\$i，从1开始计数每循环一次+1
-----	---	---	---	-----------------------	------------------------

参数说明

- `name` 参数通常用于查单条特定的栏目数据。如果type值为peer或son时该值必填，支持变量或数字id 获取栏目ID为3的记录

```
{hkcms:channel name="3"}
标题：{$item.title} <br>
{/hkcms:channel}
```

根据变量来获取栏目，假设在内容详情页有 `$Info` 变量，`$Info.category.id` 是当前文章所属ID，那么我们可以通过这个id来获取栏目同级的栏目代码如下：

```
// 同级
{hkcms:channel name="$Info.category.id" type="peer"}
标题：{$item.title} <br>
{/hkcms:channel}
```

或者使用变量可以获取当前栏目的上级栏目信息

```
// $Cate 是当前 栏目的信息该变量在栏目首页、列表页、详情页、单页都存在
{hkcms:channel name="$Cate.parent_id"}
栏目标题：{$item.title}
{/hkcms:channel}
```

2.2.3.220920版本起支持多个id获取栏目。如下，多个栏目ID逗号分隔开

```
{hkcms:channel name="1,3,7"}
标题：{$item.title} <br>
{/hkcms:channel}
```

- `type` 属性用于获取的栏目层级，如下

1. `son` 获取下级栏目，需要结合name使用

```
// 获取栏目ID为1的所有下级菜单
{hkcms:channel name="1" type="son" }
```

```
标题：{$item.title} <br>
{/hkcms:channel}
```

2. peer 表示同级栏目，需要结合name使用

```
// 获取栏目ID为1的所有同级菜单
{hkcms:channel name="1" type="peer" }
标题：{$item.title} <br>
{/hkcms:channel}
```

3. top 表示顶级栏目

```
// 获取顶级菜单
{hkcms:channel type="top" ismenu="1" }
标题：{$item.title} <br>
{/hkcms:channel}
```

- `num` 限制结果数量。例如获取10条栏目那么num="10"，当然也支持从第几条开始，例如num="4,2"表示从第4条获取两条那就是第4条、第五条。

```
// 获取顶级栏目从第二条开始获取两条
{hkcms:channel type="top" ismenu="1" num="2,3"}
标题：{$item.title} <br>
{/hkcms:channel}
```

- `model` 模型id，模型ID在模型管理里面可以看到

```
// 获取模型id=1的所有栏目
{hkcms:channel model="1"}
标题：{$item.title} <br>
{/hkcms:channel}
```

```
// 使用变量的形式获取顶级栏目
{hkcms:channel model="$Info.model_id" type="top"}
标题：{$item.title} <br>
{/hkcms:channel}
```

- `where` 数据库查询条件，语法跟SQL where基本相同。格式：where="字段名 = 1"，支持传入变量

```
{// 获取模型id为1的记录，where条件意思是只显示阅读量大于1的记录}
{hkcms:content model="1" where="views > 1"}
标题：{$item.title}
```

```
描述：{$item.description}  
{/hkcms:content}
```

sql语法可参考：https://www.w3school.com.cn/sql/sql_where.asp

更多查询表达式参考：https://www.kancloud.cn/manual/thinkphp6_0/1037537

- `order` 用于字段排序例如上面的代码，格式为`order="字段名 asc,字段名2 desc"`，其中 `asc` 表示正序 `desc` 表示倒序，多个排序字段使用逗号分隔开

```
// 多个字段排序使用逗号分隔开  
{hkcms:channel name="21" type="son" order="weigh asc" }  
标题：{$item.title} <br>  
{/hkcms:channel}
```

- `id` 用于改变循环变量默认是`$item`，例如改成`$vo`则写成`id="vo"`无需加`$`。常用于多个标签嵌套时使用循环变量而不受影响

```
{hkcms:channel name="21" type="son" id="vo" }  
标题：{$vo.title} <br>  
{/hkcms:channel}
```

- `empty` 为空提示标签，不支持传入HTML，但可以传入变量

```
{hkcms:channel name="111" type="son" empty="没有更多数据了" }  
标题：{$item.title} <br>  
{/hkcms:channel}
```

传入变量的形式使用html

```
{php}$empty = '<p align="center">- 没有更多数据了 -</p>';{/php}  
{hkcms:channel name="111" type="son" empty="$empty"}  
{php}dump($item){/php}  
{/hkcms:channel}
```

- `currentstyle` 当前选中栏目的class样式用于字体高亮等自定义样式、值默认是`active`

```
// currentstyle 默认是active, 你可以在这指定其他的class  
{hkcms:channel type="top" ismenu="1" currentstyle="active"}  
// 例如：如果当前是新闻中心栏目那么只有新闻中心栏目这条记录有 active 类  
<li class="nav-item dropdown {$item.currentstyle}">  
    <a class="nav-link dropdown-toggle" href="{ $item.url}" target="{ $item.t
```

```
arget}">{$item.title} <i class="fa fa-angle-down"></i></a>
  </li>
{/hkcms:channel}
```

- `cache` 设置缓存时间单位是秒，系统默认缓存是3600秒，设置成一天：`cache="86400"`
- `istotal` 开启文档数量统计，`istotal="1"`开启

```
{hkcms:channel name="9" type="son" istotal="1" }
标题：{$item.title} <br>
数量：{$item.total} <br>
{/hkcms:channel}
```

- `ismenu` 显示导航菜单，值为`ismenu="1"`。在后台栏目管理页面可以控制导航菜单的显示。

```
// 获取顶级菜单栏目
{hkcms:channel type="top" ismenu="1"}
标题：{$item.title}
{/hkcms:channel}
```

- `lang` 指定获取特定语言的栏目，需要开启内容多语言才有效

```
// lang="en" 只获取英文不管语言怎么切换都是英文
{hkcms:channel name="9" lang="en" }
标题：{$item.title} <br>
{/hkcms:channel}
```

- `mod` 属性用于控制一定记录的换行、或增加hr或奇数偶数等标签处理，例如下面的例子每输出5个栏目增加个hr标签

```
{hkcms:channel type="top" mod="5"}

<p>标题：{$item.title} <br></p>
// $mod的值是从1开始的
{if $mod==5}
<hr>
{/if}
{/hkcms:channel}
```

mod输出奇数偶数记录，设`mod="2"`，标签内会有`$mod`变量，里面的值每次计算到2的时候重新计数例如 (1, 2, 1, 2, 1, 2)

```
{hkcms:channel type="top" mod="2"}
```

```
{if $mod==2}
偶数：{$item.title} <br>
{else /}
奇数：{$item.title} <br>
{/if}
{/hkcms:channel}
```

- `key` 参数用于指定循环*i*的变量，只有标签嵌套时可能需要重新命名*\$key*

```
{hkcms:channel type="top"}
// $i 是从1开始的。
<p>序号{$i}： <br></p>
<p>标题：{$item.title} <br></p>
{/hkcms:channel}
```

更改循环变量示例

```
{hkcms:channel type="top" key="ii"}
// key="j" 表示这个标签的循环值是{$j}，这样就可以避免嵌套的标签依然可以获取外层的标签$ii变量。
{hkcms:channel name="$item['id']" id="son" type="son" key="j" }
  <p>外层的循环值：{$ii} <br></p>
  <p>里面这个标签的$j值：{$j} <br></p>
  <p>标题：{$son.title} <br></p>
{/hkcms:channel}
{/hkcms:channel}
```

示例

- 多级嵌套获取导航栏的栏目数据

```
// type = top, 获取的是顶级栏目。ismenu=1获取导航，在栏目管理里面可以控制导航显示
{hkcms:channel type="top" ismenu="1"}
// $item 默认的循环变量名，{if}{/if}这是if语句，下面是判断是不是有下级，有下级就显示下拉列表。
<li class="{ $item.currentstyle} nav-item {if $item.has_child}dropdown{/if}">
  <a href="{ $item.url}" target="{ $item.target}" class="nav-link" title="{ $item.title}">{$item.title}</a>
  {if $item.has_child}
  <ul class="dropdown-menu mt-0">
    // 获取下级栏目，id="son"表示里面的循环变量由默认的$item变成$son，在标签嵌套中这样使用不会影响上级的$item变量
    {hkcms:channel name="$item['id']" id="son" type="son" ismenu="1"}
    <a href="{ $son.url}" class="dropdown-item { $son.currentstyle} " title="{ $son.title}">{$son.title}</a>
  {/hkcms:channel}
  {/if}
</li>
```



```
</ul>
  {/if}
</li>
{/hkcms:channel}
```

- 获取同级栏目示例

```
// $Cate是当前栏目的数据, 该变量在列表页、内容页、栏目首页模板存在
{hkcms:channel name="$Cate.id" type="peer" ismenu="1" currentstyle="btn-primary"
}
<a href="{ $item.url}" class="btn { $item.currentstyle}">{ $item.title}</a>
{/hkcms:channel}
```

- 获取指定栏目、以及不存在时显示：暂无数据

```
{/获取单条栏目信息}
{hkcms:channel name="9" empty="暂无数据"}
<h1 class="title">栏目标题：{ $item.title}</h1>
<p class="lead">栏目描述：{ $item.seo_desc}</p>
{/hkcms:channel}
```

- currentstyle示例：获取当前栏目高亮

效果如下

✓ **新闻中心** ▾ 团队展示 客户索



表示当前栏目是行业资讯，注意：你需要写好定义好class样式。代码示例

```
// 显示高亮的class, 默认是active
{hkcms:channel type="top" ismenu="1" currentstyle="active" }
// 只需要在class里面写上{ $item.currentstyle}, 如果是当前栏目会有class="active"
  <a class="{ $item.currentstyle}" >{ $item.title}</a>
{/hkcms:channel}
```

- 排序与统计文档数量

```
// 排序 desc-倒序, asc-升序。istotal=1 开启统计文档数量
{hkcms:channel name="10" type="son" cache="7200" order="update_time desc" istotal="1"}
<a href="{ $item.url}" class="btn { $item.currentstyle}">{ $item.title} 文档数: { $item.total}</a>
{/hkcms:channel}
```

数据字典

注意：其中 `$item` 变量根据channel标签id参数影响，默认是：`$item`

字段名	说明	调用方式
id	栏目ID	{ \$item.id }
model_id	模型ID	{ \$item.model_id }
parent_id	所属上级栏目ID	{ \$item.parent_id }
type	栏目类型, link: 外部链接, list: 列表栏目, category: 栏目首页	{ \$item.type }
name	栏目英文名称	{ \$item.name }
title	栏目中文名称	{ \$item.title }
url	链接地址	{ \$item.url }
fullurl	带域名的URL地址	{ \$item.fullurl }
image	栏目缩略图	{ \$item.image }
seo_title	seo标题	{ \$item.seo_title }
seo_keywords	seo关键字	{ \$item.seo_keywords }
seo_desc	seo描述	{ \$item.seo_desc }
ismenu	是否是导航菜单, 1-导航菜单, 0-不是	{ \$item.ismenu }
target	a标签跳转方式, 用于target="_blank"属性, 有_blank, _self等	{ \$item.target }
weigh	排序	{ \$item.weigh }
has_child	true 有下级, false 没有下级	{ \$item.has_child }
son_child	数组。拥有的下级栏目ID	{ \$item.son_child }
currentstyle	当前选中class, 用于CSS文字高亮	{ \$item.currentstyle }
parent_url	上级栏目的URL地址	{ \$item.parent_url }

channel 栏目标签

lang	语言标识	{item.lang}
num	列表分页大小	{item.num}

arcone 获取单条文档

arcone 获取单条文档

获取指定的文档，支持文章类模型。

参数

参数名称	可选值	默认值	是否必填	说明
id	-	\$item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
aid	-	-	是	文章ID
model	-	-	是	模型ID
more	1	-	-	获取副表的数据
empty	-	-	-	为空的时候输出提示，支持传入html
cache	-	3600秒	-	缓存时间，默认3600秒

参数说明

`aid` 文章id，可传入数字、变量，需同 `model` 模型ID参数一起使用。

```
{hkcms:arcone aid="22" model="$model"}
标题：{ $item.title }
{/hkcms:arcone}
```

`more` 获取副表数据

```
{hkcms:arcone aid="22" model="1" more="1"}
<p>{ $item.title }</p>
<p>副表内容：{ $item.content }</p>
{/hkcms:arcone}
```

breadcrumb 面包屑导航标签

面包屑导航标签

面包屑导航标签是显示网站层次信息，即当前页面在导航层次结构内的位置

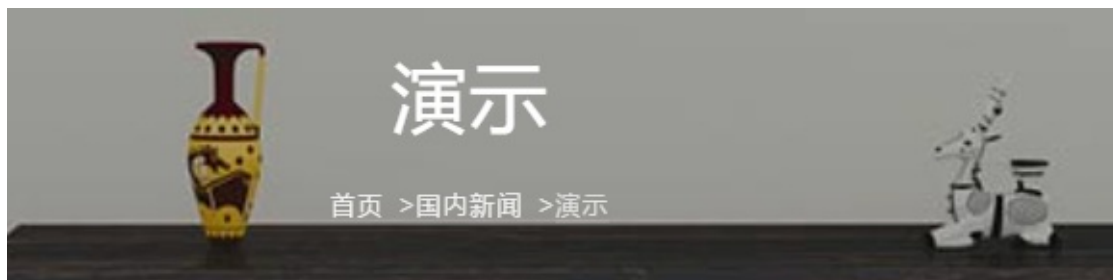
格式示例

```
<h1 class="mb-3 mt-5 bread">{$Cate.title}</h1>
<p class="breadcrumbs">{hkcms:breadcrumb class="active_a" symbol=">" /}</p>
```

html结果

```
<p class="breadcrumbs">
<a href="/" class="active_a">首页</a>&gt;
<a href="/gnxw" class="active_a">国内新闻</a>&gt;
<span>演示</span>
</p>
```

结果图



参数

参数名称	可选值	默认值	是否必填	说明
catid	栏目ID或变量	-	-	当前位置的栏目ID，可以传入变量或指定ID
symbol	-	>	-	指定分隔符
class		-	-	html class 属性，填写后除了当前位置的标签，其他标签都会加上 class 属性

currentstyle	-	active	-	css class 类名，用于定义当前位置的 class
isclick	-	0	-	css class 类名，用于定义当前位置是否可点击，1-生成a标签可点击，0=生成span标签不能点击

参数说明

`catid` 栏目ID属性，用于指定当前位置栏目，一般在内容页、列表页无需填写

```
<p class="lead fw-normal">{hkcms:breadcrumb catid="$Info.category_id" /}</p>
```

html结果

```
<p class="lead fw-normal">
  <a href="/" class="">首页</a> &gt; <a href="/news/?lang=zh-cn" class="">新闻中心</a> &gt; <span class="active">公司新闻</span>
</p>
```

`symbol` 分隔符属性，默认是 `>` 分隔符

改为 `/` 分隔符

```
{hkcms:breadcrumb symbol=" / " /}
```

`class` 类属性。例如：`class` 填写成 `item`，如下

```
{hkcms:breadcrumb class="item" /}
```

html结果

```
<a href="/" class="item">首页</a> &gt;
<a href="/news/?lang=zh-cn" class="item">新闻中心</a> &gt;
<span class="active">公司新闻</span>
```

`currentstyle` 当前位置的class名称，如上，默认是 `active` ，你可以在这指定class

`isclick` 是否可点击属性。如上，默认采用span标签， `isclick=1` 则采用a标签。

写法

```
{hkcms:breadcrumb isclick="1" /}
```

html 结果

```
<a href="/" class="">首页</a> &gt;  
<a href="/news/?lang=zh-cn" class="">新闻中心</a> &gt;  
<a href="/cd/?lang=zh-cn" class="active">公司新闻</a>
```

示例

- 指定当前位置

```
// 根据catid 属性即可指定当前位置  
<?php $temp = 2;?>  
{hkcms:breadcrumb class="active_a" symbol="-" catid="$temp" /}  
// 直接给值的方式  
{hkcms:breadcrumb class="active_a" symbol="-" catid="2" /}
```

结果

首页>国内新闻>测试  ID=2的栏目名称

adv 站点模块标签

- [站点模块标签](#)
- [简单示例](#)
- [参数](#)
- [获取图集](#)
- [获取多媒体](#)
- [获取HTML](#)
- [获取内容数据](#)
- [获取文字链接](#)

站点模块标签

站点模块标签调用的是后台站点模块数据，支持图集，多媒体，HTML内容，内容数据模块。

简单示例

```
{hkcms:adv name="home-banner"}
<div style="background-image:url({$item.image});">
数组的索引：{$key}
循环变量：{$i}，第几次循环，从1开始。
</div>
{/hkcms:adv}
```

参数

HkCms_v2.1.210909之后版本itemid获取的是排序序号。以添加图集时的顺序，从1开始

参数名称	可选值	默认值	是否必填	说明
name	-	-	是	模块标识，即后台站点模块里面的标识，每一个模块都有标识
itemid	-	-	-	模块子项序号，用于显示特定图集、视频
cache	-	3600秒	-	缓存时间，默认3600秒

id	-	\$item	-	指定循环里的变量名，例如id='obj'，模板调用如： { \$item.title} 变成 { \$obj.title}
empty	-	-	-	为空的时候输出提示，支持传入html
currentstyle	-	active	-	指定选中的图片，class名称具体看下面示例
current	-	1	-	指定第几个选中，默认active class给到第一个
num	格式为数字，或传入变量	-	-	限制数量

- `name`，模块唯一标识，值为站点模块添加时所填写的标识，如下图所示



<input type="checkbox"/>	排序	序号	标识
<input type="checkbox"/>	0	1	home_banner
<input type="checkbox"/>	0	2	links
<input type="checkbox"/>	0	3	sub

- `currentstyle`，选中的class，例如class等于active，假设第二张图片为加载页面后首先看到的，那么current="2"，第二张图集currentstyle里面就有值为active。控制哪个图片有active就首先显示出来。

```
{hkcms:adv name="home_banner" currentstyle="active" current="2"}
```

```
// 只有第二张才有currentstyle才有值，这里是文字演示，具体的要根据你的轮播图插件来。
```

```
<a href="{ $item.url}" class="{ $item.currentstyle}">{ $item.title}</a>
```

```
{/hkcms:adv}
```

获取图集

假设获取默认演示数据的 `home_banner` 标识的图集

```
{hkcms:adv name="home_banner"}

图片：{$item.image} <br>
图集标题：{$item.title} <br>
图集描述：{$item.notes} <br>
跳转地址：{$item.url} <br>
a标签跳转方式：{$item.target} <br>
选中的class：{$item.currentstyle} <br>

{/hkcms:adv}
```

a标签跳转方式由添加图集时是否开启新窗口打开，开启后target就会等于：target=_blank。

```
{hkcms:adv name="home_banner"}
<a href="{ $item.url}" { $item.target}>{$item.title}</a>
{/hkcms:adv}
```

结果示例

```
<a href="#" target="_blank">宣传视频</a>
```

只获取第二张图片,使用itemid参数

```
{hkcms:adv name="home_banner" itemid="2"}

{/hkcms:adv}
```

常用字段说明

注意：其中 `$item` 变量根据id参数影响，默认是：`$item`

字段名	说明	调用方式
type	类型，1=图集，2=多媒体，3=html，4=内容数据，5=文字链接	{ \$item.type }
title	图集里面的标题	{ \$item.title }
image	图片地址	{ \$item.image }
url	跳转地址	{ \$item.url }

notes	描述	{ \$item.notes }
weigh	排序	{ \$item.weigh }
currentstyle	class值，选中的图片才有值	{ \$item.currentstyle }
target	标签跳转方式	{ \$item.target }
recommend	一维数组，图集组的信息	-
name	图集组的标识名称	{ \$item.recommend.name }
remark	图集组标题	{ \$item.recommend.remark }

获取多媒体

基本与图集使用一致，唯一的区别是资源地址是content字段

```
// 假设多媒体标识名叫video, 那么如下获取
{hkcms:adv name="video"}
资源地址：{ $item.content } <br>
标题：{ $item.title } <br>
描述：{ $item.notes } <br>
跳转地址：{ $item.url } <br>
a标签跳转方式：{ $item.target } <br>
选中的class：{ $item.currentstyle } <br>
{/hkcms:adv}
```

常用字段说明

注意：其中 `$item` 变量根据id参数影响，默认是：\$item

字段名	说明	调用方式
type	类型，1=图集，2=多媒体，3=html，4=内容数据，5=文字链接	{ \$item.type }
title	多媒体类型标题	{ \$item.title }
content	资源地址	{ \$item.image }
url	跳转地址	{ \$item.url }
notes	描述	{ \$item.notes }
weigh	排序	{ \$item.weigh }
currentstyle	class值，选中的图片才有值	{ \$item.currentstyle }
target	标签跳转方式	{ \$item.target }

recommend	一维数组，组的信息	-
name	组的标识名称	{ \$item.recommend.name }
remark	组的标题	{ \$item.recommend.remark }

获取HTML

假设HTML标识名叫 `links` ，那么如下获取

```
{hkcms:adv name="links"}
{$item.content} // 输出HTML
{/hkcms:adv}
```

常用字段说明

注意：其中 `$item` 变量根据id参数影响，默认是：`$item`

字段名	说明	调用方式
content	资源地址	{ \$item.image }
recommend	一维数组，组的信息	-
name	组的标识名称	{ \$item.recommend.name }
remark	组的标题	{ \$item.recommend.remark }

获取内容数据

字段是根据选择的模型而定的，字段名可前往《模型管理》字段管理里面设置。

获取内容数据示例：

```
{hkcms:adv name="ceshi"}
文档标题：{$item.title}
{/hkcms:adv}
```

获取文字链接

```
{hkcms:adv name="link"}
标题：{$item.title} <br>
描述：{$item.notes} <br>
跳转地址：{$item.url} <br>
a标签跳转方式：{$item.target} <br>
选中的class：{$item.currentstyle} <br>
```

```
{/hkcms:adv}
```

常用字段说明

注意：其中 `$item` 变量根据id参数影响，默认是：`$item`

字段名	说明	调用方式
title	标题	{ <code>\$item.title</code> }
url	跳转地址	{ <code>\$item.url</code> }
notes	描述	{ <code>\$item.notes</code> }
weigh	排序	{ <code>\$item.weigh</code> }
currentstyle	class值，选中的图片才有值	{ <code>\$item.currentstyle</code> }
target	标签跳转方式	{ <code>\$item.target</code> }
recommend	一维数组，图集组的信息	-
name	图集组的标识名称	{ <code>\$item.recommend.name</code> }
remark	图集组标题	{ <code>\$item.recommend.remark</code> }

date 日期格式化标签

日期格式化标签

格式化日期时间

格式

`$Info['publish_time']` 文章发布时间

```
{hkcms:date name="$Info['publish_time']" format="Y-m-d H:i:s"}
```

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
name	-	-	是	-	待转换的值，传入变量或指定日期
format	-	Y-m-d	-	-	格式
api	-	-	-	>=v2.2.1. 20220603	详情看下面
lt	-	-	-	>=v2.2.1. 20220603	小于多少秒

format 日期与时间格式参数说明

- d - 表示月里的某天 (01-31)
- m - 表示月 (01-12)
- Y - 表示年 (四位数)
- l - 表示周里的某天
- h - 带有首位零的 12 小时小时格式
- H - 24小时格式
- i - 带有首位零的分钟
- s - 带有首位零的秒 (00 -59)
- a - 小写的午前和午后 (am 或 pm)

示例

- 内容标签里面对内容发布日期格式化

```
{hkcms:content catid="10" order="weigh desc,views desc" num="$Cate['num']}  
<span class="date">{hkcms:date name="$item.publish_time" format="Y-m-d"}</span>  
  
{/hkcms:content}
```

- 语义化日期，例如：3小时前，1个月前，56秒前，刚刚。（>=v2.2.1.20220603）

api="human" 将采用日期语义化显示

```
//, lt="2592000" 单位秒，表示超过一个月的按format里面的格式输出，不需要那直接只写api  
="human"即可  
<span class="date">{hkcms:date name="$item.publish_time" format="Y-m-d" api="hu  
man" lt="2592000"}</span>
```

全部语义化

```
<span class="date">{hkcms:date name="$item.publish_time" api="human"}</span>
```

- 时间戳(1601528401)转化成格式2020-10-01 13:00:01

```
// name 可以传入变量  
{hkcms:date name="$item.create_time" format="Y-m-d H:i:s"}
```

- 将日期2020-10-01 13:00:01，转换成2020年10月01日

```
<?php $d = '2020-10-01 13:00:01';?>  
// 传入变量  
{hkcms:date name="$d" format="Y年m月d日"}
```

substr字符(标题)截取标签

字符截取标签

可以对标题、描述等字段进行截取。

格式

截取文章标题，保留10个字，超出的...显示

```
{hkcms:substr name="$item.title" len="10" dot="..." }
```

截取描述，保留10个字，超出的...显示

```
{hkcms:substr name="$item.description" len="10" dot="...." }
```

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
name	-	-	是	无限制	传入待截取的变量，可以在列表循环、详情页使用
len	-	-	是	无限制	长度、保留字数
dot	-	...	否	无限制	超出的标题尾部拼接字符
lang	-	-	否	v2.2.4.221 230	特定语言下有效，可用于只对中文、或英文设置不同长度的字符截取

示例

name 待截取的字符变量，如下获取内容列表时，截取标题。

```
{hkcms:content model="6" num="4"}  
<div class="card-body">  
  <h5 class="text-truncate">  
    截取标题：保留6个字，后面拼接省略号...  
    {hkcms:substr name="$item.title" len="6" dot="..."}  
  </h5>  
</div>  
{/hkcms:content}
```

lang 属性，用于限制指定的语言下载取的字符长度。如下，对英文截取12个字符，中文截取8个字符，多语言情况下是配合一起使用，每个语言需要写一个字符截取。

```
<a >{hkcms:substr name="$item.title" len="12" dot="..." lang="en"}</a>  
<a >{hkcms:substr name="$item.title" len="8" dot="..." lang="zh-cn"}</a>
```

volist 循环标签

循环标签

用于循环输出数组或者对象的属性，例如使用多文件、多图片、复选、键值对的组件时，在内容页调用该标签循环输出内容。

格式示例

```
{hkcms:volist name="Info.images"}
<div class="carousel-item {$currentstyle}">
  
</div>
{/hkcms:volist}
```

参数

参数名称	可选值	默认值	是否必填	说明
currentstyle	-	active	-	指定选中的class样式具体看下面示例
current	-	1	-	指定第几个选中，默认active class给到第一个
id	-	\$item	-	指定循环里的变量名，例如id='obj'，模板调用如： { \$item.title} 变成 { \$obj.title}
name	-	-	-	传入数组、对象
offset	-	-	-	从第几条开始循环
length	-	-	-	循环次数，例如输出其中的第5~15条记录， offset="5"

				length='10'
mod	-	-	-	取余运算
empty	-	-	-	为空的时候输出提示
key	-	i	-	指定循环变量，如果没有指定key属性的话，默认使用循环变量i

参数说明

`currentstyle` 属性，用于图集数组在轮播图时进来页面看到的第一张图片，`active` 没有定义样式，需要自己定义。结合 `current` 属性，`current=1`，在第一次循环时`currentstyle`等于`active`，其他循环为空。如下：

```
// current=2, 只有循环到第二次时$currentstyle变量才有active值
{hkcms:volist name="$Info.more.files" current="2"}
<p class="{currentstyle}">{$item}</p>
{/hkcms:volist}
```

`id` 属性，用于改变循环变量，默认是`$item`，例如改成`$vo`，无需加`$`

```
{hkcms:volist name="$Info.more.files" id="vo" }
<p>{$vo}</p>
{/hkcms:volist}
```

`name` 属性，传入要输出的变量名称，如上面，取的是自己添加的字段`files`（多文件数组类型）。除了变量，还支持调用方法如下：

```
{hkcms:volist name=":ceshi()" id="vo"}
<p>{php}dump($vo);{/php}</p>
{/hkcms:volist}
```

`offset` 属性，用于限制从第几条开始循环，例如如下：结合 `length` 属性输出其中的第5~15条记录

```
{hkcms:volist name="$Info.more.files" offset="5" length="10"}
<p>{php}dump($item);{/php}</p>
{/hkcms:volist}
```

`mod` 取余运算属性，使用该属性后，在循环模板里面得到`{mod}`变量，里面的值等于循环变量 `i%mod`，`i` 是从1开始的，每循环一次`i+1`。

volist 循环标签

例如：mod="2"，判断偶数奇数。用于每一行图片左右摆放样式不一样时可以使用。更多使用方式根据实际情况设置。

```
{hkcms:volist name="$Info.more.files" mod="2"}
{$i} - {$mod}
{if $mod==1}偶数{else /}奇数{/if}
{/hkcms:volist}
```

`empty` 为空提示标签，不支持传入HTML，但可以传入变量

```
{hkcms:volist name="$Info.more.files" empty="没有更多数据了"}
{php}dump($item){/php}
{/hkcms:volist}
```

传入变量

```
{php}$empty = '<p align="center">- 没有更多数据了 -</p>';{/php}
{hkcms:volist name="$Info.more.files" empty="$empty"}
{php}dump($item){/php}
{/hkcms:volist}
```

`key` 循环变量属性，默认是i，从1开始，每循环一次增加1。

输出循环变量

```
{hkcms:volist name="$Info.more.files" }
{php}dump($i){/php}
{/hkcms:volist}
```

改变循环变量

```
{hkcms:volist name="$Info.more.files" key="k" }
{php}dump($k){/php}
{/hkcms:volist}
```

循环变量不是指数组索引值，数组索引值可以使用{\$key}。

```
{hkcms:volist name="$Info.more.files" }
{$key}
{/hkcms:volist}
```

示例

- 设置第二张选中，选中的class为：ac

有时候，我们的图集轮播图，需要默认显示第几张显示，可按一下设置参数，设置好后，将{\$currentstyle}放入你想放入的class标签上即可。

```
// Info.images 为内容里的图集，name不用加：$, 可参考模板开发下的可用变量。
{hkcms:volist name="Info.images" currentstyle="ac" current="2"}
<div class="carousel-item {$currentstyle}">
    
</div>
{/hkcms:volist}
```

language 获取语言列表标签

获取语言列表标签

用于获取前台支持的多语言列表

格式

```
{hkcms:language currentstyle="active"}
<a href="{ $item.url}" class="{ $currentstyle}">{ $item.value}, 语言标识 : { $key}新版
可以使用 : { $item.key}</a>
{/hkcms:language}
```

参数

参数名称	可选值	默认值	是否必填	说明
id	-	\$item	-	指定循环里的变量名, 例如 id='obj', 模板调用如 : { \$item.title} 变成 { \$obj.title}
currentstyle	-	active	-	指定选中的 class 样式

v2.3.0.230709 版本字段说明

```
{hkcms:language currentstyle="active"}
{// target_html表示是否新窗口打开}
<li class="{ $currentstyle}"><a href="{ $item.url}" { $item.target_html} title="{ $
item.key}">{ $item.value}</a></li>
{/hkcms:language}
```

字段名	说明	调用方式
mark	标志	{ \$item.key} 或者 { \$item.mark}
title	标题	{ \$item.value} 或者 { \$item.title}
subtitle	别名	{ \$item.subtitle}

image	图片	{item.image}
icon	图标	{item.icon}

lang 特定语言下显示内容标签

说明

用于特定语言下显示内容，

v2.3.0.230709版本及以上版本支持

格式

```
{hkcms:lang value="zh-cn"}
仅在中文下显示的内容
{/hkcms:lang}
```

参数

参数名称	可选值	默认值	是否必填	说明
value	-	-	是	填写语言标识，标识可以在后台站点配置多语言列表里面查看，多个标识请用英文逗号分隔

示例

多个语言，英文逗号隔开

```
{hkcms:lang value="zh-cn,en"}
中文和英文下显示
{/hkcms:lang}
```

使用 `{else /}` 方式

```
{hkcms:lang value="zh-cn"}
中文下显示
{else /}
其他语言下显示的内容
{/hkcms:lang}
```


query 查询数据库内容

query 查询数据库内容

用于直接写sql语句查询数据库数据

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
id	-	\$item	-	-	指定循环里的变量名，例如id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }
sql	-	-	是	-	sql语句
empty	-	-	-	-	为空的时候输出提示，支持传入html
cache	-	3600秒	-	-	缓存时间，默认3600秒
mod	-	-	-	>=v2.2.1. 20220603	用于控制偶数输出、换行等，更多查看下面的示例
key	-	-	-	>=v2.2.1. 20220603	指定循环值变量名，默认为\$i，从1开始计数每循环一次+1
	格式1: 10，表示				

num	获取10条 格式2: 10,12, 表 从第10开 始的12条 记录	-	-	>=v2.2.1. 20220603	限制结果 数量
table	-	-	-	>=v2.2.1. 20220603	表名, 不 用写前缀 不能跟sql 参数一起 用
field	-	*	-	>=v2.2.1. 20220603	限制显示 的字段
alias	-	-	-	>=v2.2.1. 20220603	设置当前 数据表的 别名, 用 于连表
join	-	-	-	>=v2.2.1. 20220603	连表
where	-	-	-	>=v2.2.1. 20220603	sql条件
order	-	-	-	>=v2.2.1. 20220603	排序
tableid	-	-	-	>=v2.2.1. 20220603	直接按主 键ID获取 记录

参数说明

- `sql` 参数, 这里写sql查询语句。在写表名时需要注意的是表前缀, 使用 `__PREFIX__` 代替, 系统会自动替换成正确的表前缀, 当然你也可以直接写你当前使用的表前缀。

```
{hkcms:query sql="select * from __PREFIX__archives"}
{php}dump($item);{/php} // 输出所有字段
{/hkcms:query}
```

include 导入模板文件标签

导入模板文件标签

用于把通用的页面分离出去，在主文件里面使用

```
{include file="根目录基于当前模板主题位置"}
```

示例

例如默认模板：头部文件放在主题目录下的common/head.html 下。按如下引用

```
{include file="common/head"}
```

原生PHP与php标签

原生PHP标签

php标签是tp框架自带的标签，用法如下：

使用php标签

例如：

```
{php}echo 'Hello,world!';{/php}
```

建议使用PHP代码的时候尽量采用php标签，因为原生的PHP语法可能会被配置禁用而导致解析错误。

使用原生php代码

```
<?php echo 'Hello,world!'; ?>
```

map生成位置地图

map生成位置地图

与内置的地图位置选取插件生成百度地图、高德地图。

2.1.220504版本起

示例

默认模板示例，\$Info 为内容

```
{hkcms:map htmlid="dituContent" attr="class='diyclass' style='width:1800px; height:400px'" title="$Info['company']" address="$Info['address']" scale="19" point="$Info['point']"}
```

参数说明

参数名称	可选值	默认值	是否必填	说明
htmlid	-	dituContent	-	div标签 id属性，你可以使用该ID写样式
attr	-	-	-	div标签属性，注意attr里面使用单引号
title	-	-	-	地图面板标题
address	-	-	-	地图面板地址
scale	-	19	-	放大级别
point	-	-	-	经纬度信息，格式为json，例如： {"lng":116.404,"lat":39.915}

完整流程：

1. 增加模型字段，附加属性增加如下配置，data-point 用于选择位置后填充到哪个输入框，data-address 用于选择位置后返回的详细地址填充到哪个输入框

```
data-toggle="address",data-point="frm-point",data-address="frm-address"
```

69	point	经纬度	text	250
68	address	详细地址	text	250

修改

字段类型: 单行文本

分组: 常规

字段: point

标题: 经纬度

长度: 250

默认值: 默认值

规则: 必须

提示信息: 提示信息

错误信息: 错误信息

附加属性: data-toggle="address",data-point="frm-point",data-address="frm-address"

排序: 69

是否筛选: 否 是

提交 重置

2. 选择位置

经纬度: [{"lng":116.404,"lat":39.915}] 位置选取

详细地址: 北京市东城区中华路甲10号

联系方式: 400-123-1234,13013013011

3. 模板调用:

```
{hkcms:map htmlid="dituContent" title="$Info['company']" address="$Info['address']" scale="19" point="$Info['point']"}
```



thumb 生成缩略图

thumb 生成缩略图

用于生成缩略图，返回生成缩略图的地址。该标签不受后台缩略图开启/关闭配置影响

格式

```

```

参数

参数名称	可选值	默认值	是否必填	支持版本	说明
name	图集变量	-	必填	v2.2.2.202 20725	
width	-	-	-	v2.2.2.202 20725	缩略图宽度，不填则使用附件里面的缩略图配置
height	-	-	-	v2.2.2.202 20725	缩略图高度，不填则使用附件里面的缩略图配置
type	参考下面	-	-	v2.2.2.202 20725	缩略图生成方式，默认使用附件里面的缩略图配置
auto	1/0	1	-	v2.2.2.202 20725	1=表示缩略图不存在时自动生成缩略图
					1=表示生成图片标

ishtml	1/0	0	-	v2.2.2.202 20725	签, 0=仅 返回缩略 图图片地 址
--------	-----	---	---	---------------------	-----------------------------

参数说明

- `name` 传入变量, 例如在content标签列表用

```
{hkcms:content catid="10"}

{/hkcms:content}
```

- `width` 与 `height` 分别是缩略图宽度与缩略图高度, 这里不填写则使用的是后台缩略图配置



指定宽高, 如果是按比例缩放宽为250像素时停止缩放

```
{hkcms:content catid="10"}

{/hkcms:content}
```

- `type` 缩略图生成方式。

type可选值如下:

- 1: 等比例缩放
- 2: 缩放填充
- 3: 居中裁剪

thumb 生成缩略图

4 : 左上角裁剪

5 : 右下角裁剪

6 : 固定尺寸缩放

居中裁剪示例, type等于3

```
{hkcms:content catid="10"}  
  
{/hkcms:content}
```

- `auto` 自动生成缩略图, 当传入的图片没有缩略图系统将自动帮该图生成缩略图, 默认是自动生成
- `ishtml` 返回带有图片标签的缩略图

```
{hkcms:content catid="10"}  
{hkcms:thumb name="$item.thumb" ishtml="1" /}  
{/hkcms:content}
```

效果 :

```

```

addons 插件检测

插件检测标签

用于检测插件是否安装

HkCms_v2.2.1.20220603版本起

参数

参数名称	可选值	默认值	是否必填	说明
name	-	-	是	插件标识，可在插件详情页，或安装插件后看到标识
id	-	\$item	-	指定循环里的变量名，例如 id='obj'，模板调用如： { \$item.title } 变成 { \$obj.title }

简单示例

检测标签插件是否安装，例如标签插件，安装后才执行里面的标签列表获取

```
{hkcms:addons name="tags"}
  <!--// 需要安装tags插件，上面检测到安装了tags标签，这里才会被执行。-->
  {hkcms:taglist arcid="$Info['id']" model="$Info['model_id']"}
  <a href="{ $item.url }">{ $item.title }</a>
  {/hkcms:taglist}
{/hkcms:addons}
```

标签管理

`taglist` 标签(获取标签列表)

`tagarclist` 根据标签获取文章内容列表

taglist标签(获取标签列表)

taglist标签

用于获取标签列表

HkCms_2.2.4.230206版本起

参数

参数名称	可选值	默认值	是否必填	说明
tid	标签ID或0	-	-	0-表示获取所有
arcid	文章ID	-	tid与arcid选一个	获取指定文章包含的标签
model	模型ID	-	acrid填写时, 该值不能为空	指定标签属于的模型
catid	栏目ID	-	-	指定标签属于的栏目
order	-	views desc (点击量降序)	-	排序
num	-	-	-	限制结果数量
where	-	-	-	sql筛选条件
page	1	-	-	开启分页, 配合 {hkcms:contentpage /} 分页标签
id	-	\$item	-	指定循环里的变量名, 例如 id='obj', 模板调用如: {\$item.title} 变成 {\$obj.title}
empty	-	-	否	为空的时候输出提示, 支持传入html

cache	-	3600秒	-	缓存时间，默认3600秒
currentstyle	-	active	-	css class 类名，用于定义当前选择状态样式

简单示例

1. 获取所有标签

```
{hkcms:taglist tid="0"}
{php}dump($item);{/php} // 输出所有字段
{/hkcms:taglist}
```

2. 获取指定标签

```
{hkcms:taglist tid="3"}
{php}dump($item);{/php} // 输出所有字段
{/hkcms:taglist}
```

3. 文章详情页，获取当前文章所有标签，也可用于列表页

详情页：

```
{hkcms:taglist arcid="$Info.id" model="$Info.model_id"}
<a href="{ $item.url }">{ $item.title }</a>
{/hkcms:taglist}
```

4. 只获取特定模型的标签

```
{hkcms:taglist model="1"}
{php}dump($item);{/php} // 输出所有字段
{/hkcms:taglist}
```

5. 只获取特定栏目的标签 (1.0.3版本)

```
{hkcms:taglist catid="1"}
{php}dump($item);{/php} // 输出所有字段
{/hkcms:taglist}
```

5. 分页,page=1:开启分页，num=10，每页显示10条记录

taglist标签(获取标签列表)

```
{hkcms:taglist tid="0" page="1" num="10" order="views desc"}
<div class="col-xs-3">
  <a href="{ $item.url}" target="_blank">{$item.title}</a>
</div>
{/hkcms:taglist}

// 显示页码
{hkcms:contentpage item="home,pre,pageno,next,last,info" mobile_item="pre,pageno,next" pre="&laquo;" next="&raquo;" /}
```

tagarclist根据标签获取文章内容列表

tagarclist根据标签获取文章内容列表

用于根据标签获取文章列表

HkCms_2.2.4.230206版本起

参数

参数名称	可选值	默认值	是否必填	说明
tid	标签ID或0	-	-	0-表示获取所有
arcid	文章ID	-	tid与arcid选一个	获取指定文章包含的标签
model	模型ID	-	acrid填写时, 该值不能为空	指定标签属于的模型
catid	栏目ID	-	-	指定标签属于的栏目
order	-	views desc (点击量降序)	-	排序
num	-	-	-	限制结果数量
where	-	-	-	sql筛选条件
page	1	-	-	开启分页, 配合 {hkcms:contentpage /} 分页标签
id	-	\$item	-	指定循环里的变量名, 例如 id='obj', 模板调用如: {\$item.title} 变成 {\$obj.title}
empty	-	-	否	为空的时候输出提示, 支持传入html

cache	-	3600秒	-	缓存时间，默认3600秒
currentstyle	-	active	-	css class 类名，用于定义当前选择状态样式

简单示例.

在标签页 `tags_list` 页面，获取文章列表

```
// 在标签列表页有$Tags变量，即当前标签，这里代码意思是：获取当前标签下的文章。里面的字段是文章的字段
{hkcms:tagarclist tid="$Tags['id']" num="6" page="1"}

    <a href="{ $item.url}" title="{ $item.title}"></a>

    <h5><a href="{ $item.url}" title="{ $item.title}">{ $item.title}</a></h5>

    <h6>{hkcms:date name="$item['publish_time']" /}</h6>
    <p class="card-text"><a href="{ $item.url}">{ $item.description}</a></p>
</div>
{/hkcms:tagarclist}
配合分页

{hkcms:contentpage item="home,pre,pageno,next,last,info" mobile_item="pre,pageno,next" pre="&laquo;" next="&raquo;" /}
```

在其他位置获取，例如你的标签ID是10，那么获取文章里面有10的标签列表

```
// 在标签列表页有$Tags变量，即当前标签，这里代码意思是：获取当前标签下的文章。里面的字段是文章的字段
{hkcms:tagarclist tid="10" num="6" }

    <a href="{ $item.url}" title="{ $item.title}"></a>

    <h5><a href="{ $item.url}" title="{ $item.title}">{ $item.title}</a></h5>

    <h6>{hkcms:date name="$item['publish_time']" /}</h6>
    <p class="card-text"><a href="{ $item.url}">{ $item.description}</a></p>
</div>
```

tagarclist根据标签获取文章内容列表

```
{/hkcms:tagarclist}
```



导入标签

导入标签

在安装一些插件时，插件提供一些标签库，我们可以使用taglib标签引用这些标签库到模板。

标签的命名空间，导入后即可使用demo标签库

```
{taglib name="\addons\demo\librarys\Demo" /}
```

例如标签库里面有seo标签，那么使用方式如下

```
{demo:seo /}
```

模板开发

- [获取站点配置](#)
- [可用的变量](#)
- [模板目录结构说明](#)
- [模板描述文件\(重要\)](#)
- [模板静态目录及调用静态文件](#)
- [模板界面多语言](#)
- [模板配置JSON](#)
- [模板搜索页](#)
- [编写模板](#)
- [模板文件打包](#)

1. 创建模板

创建模板

首先安装“一键生成应用”插件，来帮助我们快速的创建插件。

操作说明

1. 安装



2. 进入生成界面，填写模板信息，点击提交

模板

* 模板类型: 响应式 非响应式 (PC和手机分离)

* 模板标题: 演示模板

* 模板简介: 演示模板

* 应用标识: ucyglprhz

随机生成

* 作者: demo

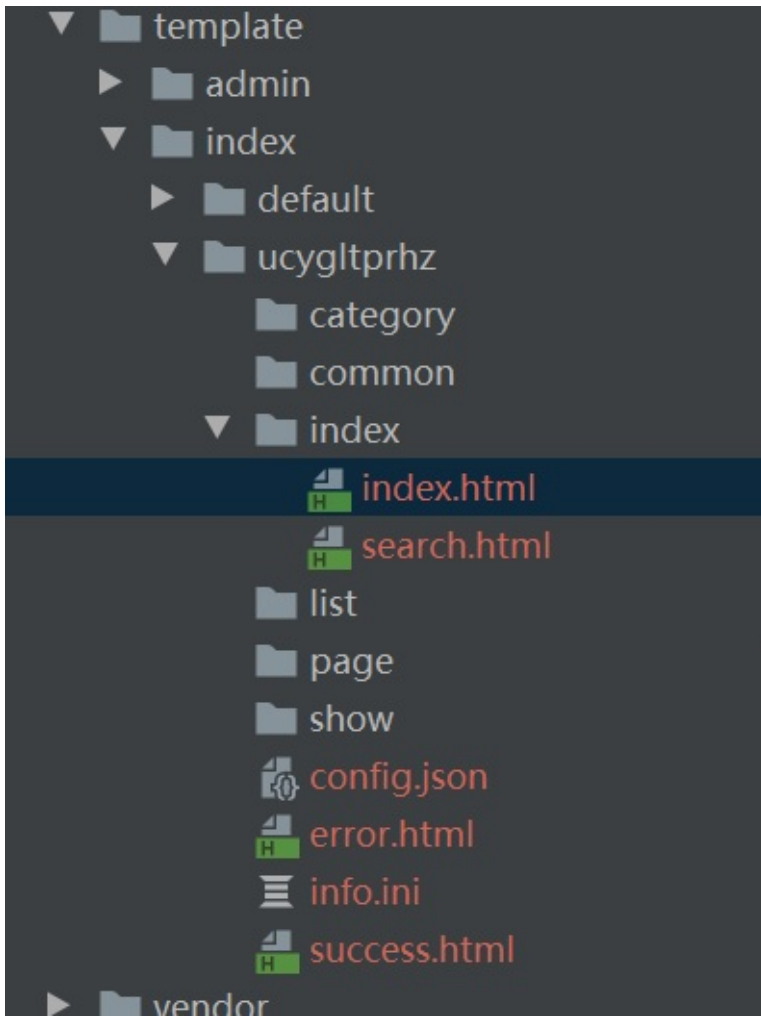
* 版本号: 1.0.0

更多配置

提交

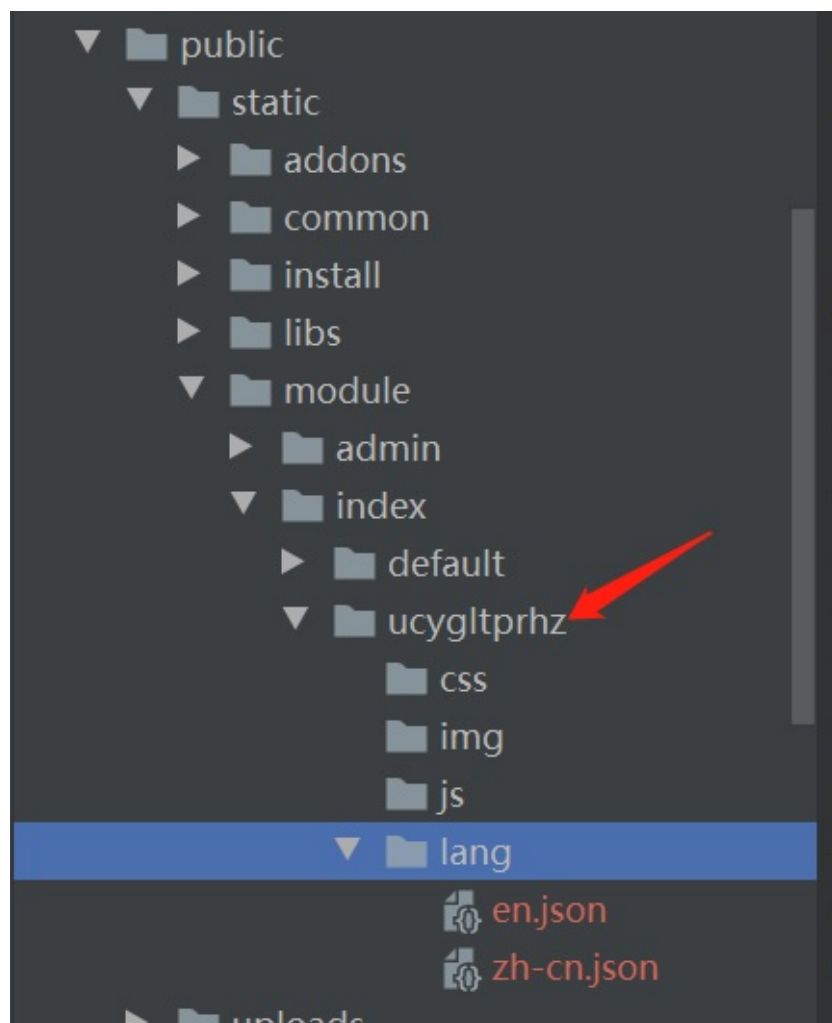
3. 生成效果

1. 创建模板



静态资源目录

1. 创建模板



当然你也可以手动创建

2. 模板目录结构

模板目录

- 响应式主题

template 模板安装目录

├─admin 后台模板目录

├─index 前台模板目录

│ ──主题名称

│ │ ──category文件夹 栏目分类模板

│ │ ──list文件夹 列表页模板

│ │ ──show文件夹 详情页模板

│ │ ──page文件夹 单页模板

│ │ ──index文件夹 首页

│ │ ──error.html 前台错误页

│ │ ──success.html 操作成功页

│ │ ──config.json 模板配置

│ │ ──info.ini 主题描述文件(重要)

- 非响应式

主题目录下需要先创建pc、mobile文件夹，然后再是模板

│ ──主题名称

│ │ ──pc文件夹 电脑访问

│ │ │ ──目录结构同上一致

│ │ ──mobile文件夹 手机端访问

│ │ │ ──目录结构同上一致

│ │ ──info.ini 主题描述文件，同pc，mobile同级(重要)

栏目分类模板

在栏目管理-》添加栏目-》，选择栏目分类，这个时候会出现栏目首页的选项，这里的模板选项数据就来自于栏目分类文件夹里的模板。

添加 - [] X

常规 SEO 更多

栏目类型: 栏目分类 列表栏目 其他链接

所属模型:

上级栏目:

栏目名称|英文:

链接:

栏目首页:

排序:

导航显示: 显示 隐藏

列表页模板

列表页模板文件夹用于栏目管理里的列表模板选项。

2. 模板目录结构

栏目类型: 栏目分类 列表栏目 其他链接

所属模型:

上级栏目: 作为一级菜单

栏目名称|英文: 例如:
国内新闻|china
国际新闻|world

链接: 指定栏目地址 “其他链接” 类型则是跳转的地址
list.html

列表模板:

内容模板:

排序: 0

详情页模板

列表页模板文件夹用于栏目管理里的列表模板选项，内容模板选项。

栏目类型: 栏目分类 列表栏目 其他链接

所属模型:

上级栏目: 作为一级菜单

栏目名称|英文: 例如:
国内新闻|china
国际新闻|world

链接: 指定栏目地址 “其他链接” 类型则是跳转的地址
list.html

列表模板:

内容模板:

排序: 0

单页模板

2. 模板目录结构

这里可以存放列表类的模型，即只有一个页面时。栏目类型选为其他链接，注：其他链接要应用到模型，需要选所属模型。

添加 — □ ×

常规 SEO 更多

栏目类型: 栏目分类 列表栏目 其他链接

所属模型:

上级栏目: 作为一级菜单

栏目名称|英文: 例如:
国内新闻|china
国际新闻|world

链接: 指定栏目地址，“其他链接”类型则是跳转的地址

页面模板:

排序: 0

导航显示: 显示 隐藏

3. 模板描述文件(重要) info.ini

info.ini 文件

模板主题描述文件

```
type = "template" # 表示模板 (必填)
module = 'index' # 所属应用模块, 默认写index即可。(必填)
name = 'default' # 模板主题名字, 如果模板主题上传到应用中心, 那这个名字切记跟应用中心的标识保持一致, 以保证可以更新下载。(必填)
title = '默认模板' # 模板主题标题 (必填)
description = '' # 模板主题描述 (选填)
author = 'hkcms' # 模板主题作者 (必填)
website = '' # 模板主题作者网站 (选填)
version = '1.0.0' # 版本号, 注: 保持与应用中心上传的版本保持一致 (必填)

# 数据库表, 多个表英文逗号分隔, 不用写表前缀, 用于安装时检测表是否存在, 如果存在则提示用户, 防止数据覆盖 (选填)
database = 'table2,table1,table3'

# 文件目录检测, 多个目录英文逗号分开, 以项目根目录开始, 用于安装时检测是否可写 (选填)
dir = 'app/admin/,文件夹1,文件夹2'

# 依赖的插件, 左边写插件标识名(name), 右边写标识 (选填)
[addon]
simditor = '1.0.0'
user = '1.0.0'
```

name (必填)

模板名称, 以字母开头。注意发布到应用市场时该名称必须唯一, 参考 [插件名称检测](#)

type (必填)

模板就写 `type="template"` 即可

title (必填)

插件标题

author (必填)

插件作者, 必填字段

version (必填)

3. 模板描述文件(重要) info.ini

版本号，格式为：xx.xx.xx。

database

数据库，不需要写表前缀，用于安装时检测数据库表是否存在，如果已存在则提示用户。

dir

目录，用于目录权限检测，如果检测不通过，提示用户需要有写入的权限。

addon

插件依赖，左边为插件名称(插件唯一标识)，右边是依赖的最低插件版本号，在这里填写后，如果用户未安装依赖的插件则会提示用户需要先安装某某插件。

提示：依赖的插件名称获取方式：在安装所需的插件后找到 `info.ini` 文件里面的name，这个就是插件名称。

4. 变量输出格式

模板变量

变量格式：中括号包裹，例如：`{ $title }`。 `{` 和 `$` 之间不能有任何的空格，否则标签无效

1. 例如输出站点配置变量

```
// 获取站点标题  
{ $site.title }等价于{ $site['title'] }
```

注意：在一些标签参数使用中，使用`$site['title']`方式赋值。具体看标签说明。

模板函数的调用

模板调用系统函数、或自定义的函数方法：

1. 例如调用md5函数

```
方式一  
{ $data.name | md5 }  
方式二  
{ :md5( $data.name ) }
```

默认值

使用 `default` 或`(?)`两个问号方法来对变量为空时给定默认的值

1. 例如图片默认值

设置封面图没有上传时，给定默认值。
{ \$item.thumb | default="默认图片地址" }

2. 例子2，设置字段值为空时的默认值

```
{ $item.remark | default="暂无说明" }
```

4. 模板变量

模板变量

- `$Cate` 模板变量 (数组)

栏目信息，存在于栏目分类页、列表页、单页、内容页。调用方式如下。

```
// 获取栏目标题
<h1 class="mb-3 mt-5 bread">{$Cate.title}</h1>
<p><?php dump($Cate); //查看所有字段 ?></p>
```

- `$Info` 模板变量 (数组)

内容详情信息，存在单页、内容页。更多字段可前往后台找到对应模型字段查看。

```
<h2 class="mb-3">{$Info.title}</h2>
<p><?php dump($Info); //查看所有字段 ?></p>
```

- `$Tp1` 模板配置

根据模板配置config.json里面获取。只有模板有config.json文件的，前台才能调取。

5. 获取站点配置

获取站点配置

在后台【站点配置】页面，里面的数据可以给到前台调用。调用方式如下

```
<p>{$site.cdn}</p>
```

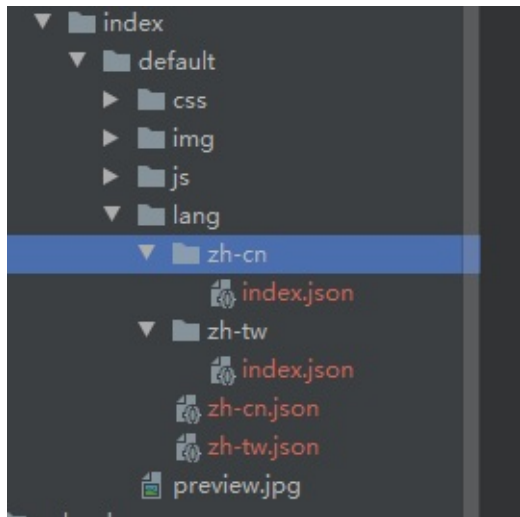
按下图的格式即可调用对应的配置值。

配置项	配置值	调用格式
网站名称	HkCms开源内容管理系统	{\$site.title}
CDN	CDN、填写域名后静态资源也将改变，无需"/"结尾	{\$site.cdn}
首页标题	站点名称/品牌名 - 网站首页	{\$site.home_title}
首页关键字	HkCms开源内容管理系统	{\$site.keyword}
首页描述	HkCms开源内容管理系统,是一款完全开源免费的PHP+MYSQL系统,核心采用了Thinkphp6.0框架等众多开源软件,同时核心功能也作为开源软件发布	{\$site.description}
备案号	粤ICP备10000000号-1	{\$site.icp}
版本号	1.0.0	{\$site.version}
网站LOGO	/static/common/image/logo.png	{\$site.logo}

6. 模板界面多语言

模板界面多语言

静态文件目录下创建lang文件夹，如下创建了简体、繁体，默认英文对应。



zh-cn.json，模板全局可使用，写在zh-cn文件夹下的index.json，只在index控制器下有效。

格式

zh-cn文件

```
// 简体
{
  "The title is right at the top of the pag": "标题就在页面的顶端"
}
```

zh-tw文件

```
// 繁体
{
  "The title is right at the top of the pag": "標題就在頁面的頂端"
}
```

在模板html写上：

```
{:lang('The title is right at the top of the pag')} // 系统会根据当前语言，输出对应的语言包。如果没找到，就是默认英文。
```

语言切换方法

将下面的代码复制到合适的位置，点击链接即可完成切换。

```
<li class="nav-item">
  <a href="?lang=zh-cn">中文</a>
  <a href="?lang=zh-tw">繁体</a>
  <a href="?lang=en">英文</a>
</li>
```

7. 模板静态目录及调用静态文件

模板静态目录

public

├─static // 静态文件目录

| └─module

| | └─index // 前台模板主题目录

| | | └─模板主题名称

| | | | └─CSS、html相关文件

| | | | └─lang 语言包文件夹

| | | | └─preview.jpg // 封面图、预览图

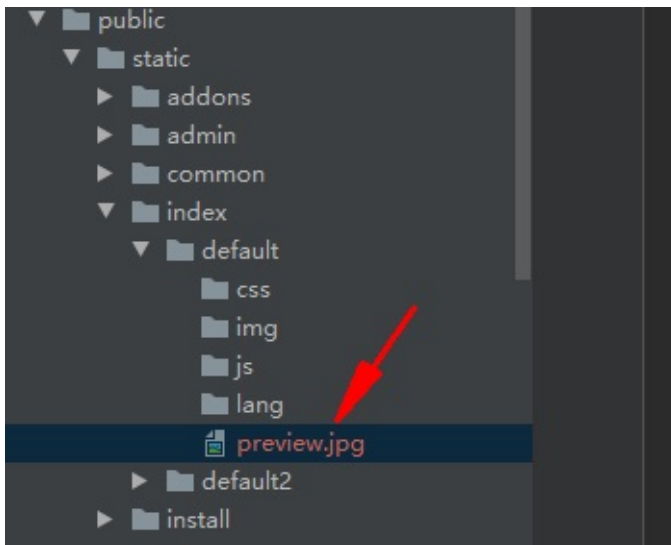
| | | | └─...

模板引用方法

```
// __static__ 模板变量, 生成的路径为 : /static/index/主题名称  
<link rel="stylesheet" href="__static__/css/animate.css">
```

```
// __libs__ 模板变量, 生成的路径是 : /static/libs  
<!-- 使用已有的字体库 -->  
<link rel="stylesheet" href="__libs__/fontawesome-free/css/all.css" />
```

预览图



设置，在后台就能看到预览图效果，如下图所示

7. 模板静态目录及调用静态文件



8. 模板定义插件事件

模板定义插件事件

用于模板支持安装更多的插件，可参考默认模板。

- 放于模板head标签里面，通常插件会在这加载设置所需的样式、预处理等

```
{:hook("index_head")}
```

- 放于模板底部

```
{:hook("index_footer")}
```

9. 模板搜索页

搜索页

HkCms有提供基础搜索页，在默认模板下的index目录下search.html文件有演示案例。

使用方法

1. 调用方式

复制以下表单放置任意位置即可，定义 `model_id` ，以查询不同的副表模型数据。

```
<form class="d-flex" action="{:url('/search/index')}}" method="get">
  <!-- model_id的value值为空时，无法获取副表的字段。 -->
  <input type="hidden" name="model_id" value="">
  <input class="form-control" type="search" name="keyword" placeholder="关键字"
  aria-label="Search">
  <button class="btn btn-primary">搜索</button>
</form>
```

2. 搜索结果页

结果页放到index目录下，并命名search.html

- 结果页变量

```
$_param__ // 查询(搜索)参数
$modelInfo__ // 所有模型信息
$__SDATA__ // 搜索到的列表结果
$__SPAGE__ // 分页数据
```

变量说明

- `$_param__`

搜索参数数组，里面包含：`keyword`(用户搜索的关键字)、`model_id`(模型ID)、`category_id`(模型ID)、`orderby`(排序字段)、`ordermode`(排序类型，`asc`=升序，`desc`=降序)

```
{$_param__[ 'keyword' ]} // 搜索的关键字
{$_param__[ 'category_id' ]} // 筛选的栏目ID
{$_param__[ 'model_id' ]} // 筛选的模型ID
```

- `$_modelInfo__`

模型信息，使用`volist`输出

```
{hkcms:volist name="__modelInfo__"}  
模型别名：{$item.alias}  
{/hkcms:volist}
```

- `$_SDATA__`

结果列表

```
{hkcms:volist name="__SDATA__"}  
标题：{$item.title}、更多字段前往后台字段管理里面查看  
{hkcms:color name="$item.title"} // 搜索结果文字高亮  
{/hkcms:volist}
```

搜索文字高亮使用(v2.2.3.220920+)下面的标签：

`{hkcms:color name="$item.title"}`，默认是红色，如果想改成蓝色并加粗如下：

```
{hkcms:color name="$item.title" style="color:blue;font-weight:bold"}
```

高亮效果图

[演示数据] 公司全新学习衍生式设计的基础知识

 小编  2021-11-15  3 浏览

公司全新学习衍生式设计的基础知识，坚持一个行业前行。

[查看更多 →](#)

- `$_SPAGE__`

分页对象，使用contentpage分页标签，了解更多前往内容列表分页标签查看

```
{hkcms:contentpage name="__SPAGE__" item="home,pre,pageno,next,last,info" mobil  
e_item="pre,pageno,next" pre="&laquo;" next="&raquo;" /}
```

10. 模板配置JSON

模板配置JSON

放在模板的根目录下，用于用户自定义一些配置、前台调取。

支持text、textarea、radio、checkbox、select、image、images、date、datetime控件

组件可前往这里了解：[字段类型与组件](#)

使用说明

一级写法 说明

```

{
  1 "field": {
    "title"      : "标题",
    "type"       : "text",
    2 "tips"      : "提示",
    "rules"      : "required;length(3~16)",
    "error_tips" : "值必须",
    "value"      : ""
  }
}

```

1可以更改
2不可以更改

key	更改key值	value值	是否必填	说明
field	可更改	-	-	field改为king 前台调用方式 {\$Tpl.king}
title	否	随便填	否	-
type	否	text textarea radio checkbox select image images date datetime	---	具体说明 字段类型与组件
tips	否	随便填	否	后台模板配置 时候提示效果
				验证规则 如果多个验证

rules	否	-	否	请用 ; 隔开 查看规则
error_tips	否	随便填	否	自定义验证规则错误信息。填写了，会覆盖规则的错误提示消息。
value	否	-	否	默认值

多级写法 说明

```

{
  1 'base': {
    2 "title": "基础配置",
    "item": {
      "text" : {"title": "标题"...},
      "textarea" : {"title": "描述"...},
      3 "radio" : {"title": "是否显示"...},
      "checkbox" : {"title": "多项选择"...},
      "select" : {"title": "下拉选择"...},
      "selects" : {"title": "下拉多项选择"...}
    }
  },
  1 "more": {
    "title": "更多选项",
    2 "item": {
      "image" : {"title": "图片"...},
      "images" : {"title": "图片"...},
      3 "date" : {"title": "日期"...},
      "datetime" : {"title": "日期时间"...}
    }
  }
}

```

1 可更改，不能重复

2 不可更改

3 跟一级写法一样，可以更改，同一个item不能重复

key	更改key值	value值	是否必填	说明
base/more	可更改	-	-	base改为 king 前台调用方式 {Tpl.king.item中的key值}
title	否	随便填	否	-
item	否	-	是	多个一级写法 组成 key值不能重

使用示例

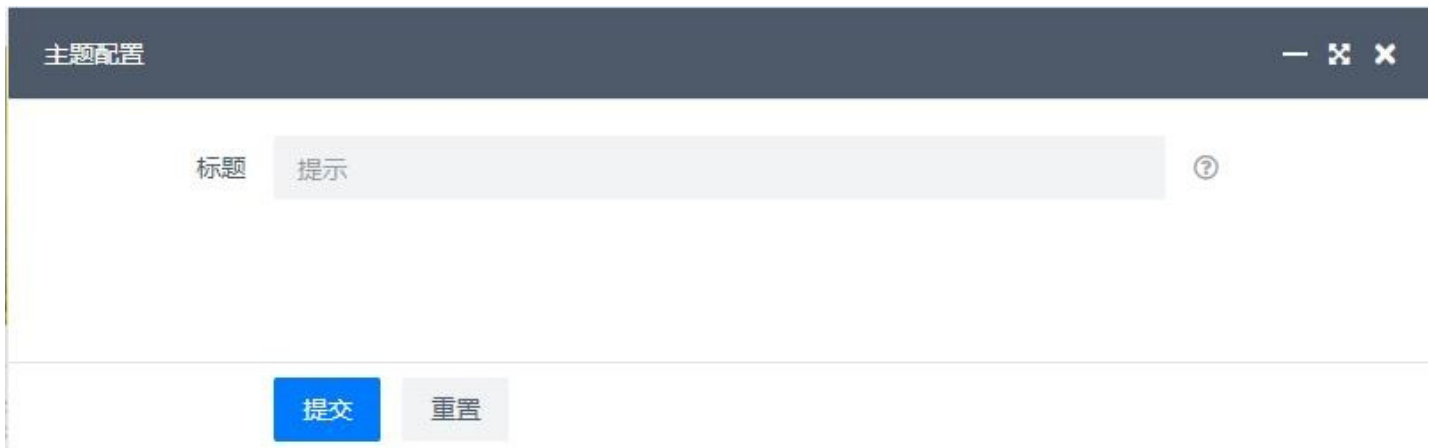
简单写法(一级)

前台调用方式，如下配置文件。调取标题字段的值。

`{$Tpl.field}` 或者 `{$Tpl['field']}`

```
{
  "field": {
    "title": "标题",
    "type": "text",
    "tips": "提示",
    "rules": "required;length(3~16)",
    "error_tips": "值必须",
    "value": ""
  }
}
```

效果图



配置分组写法 (多级)

调取标题

`{$Tpl.base.text}` 或者 `{$Tpl['base']['text']}`

示例代码

```
{
  "base": {
    "title": "基础配置",
    "item": {
      "text": {
        "title": "标题",
        "type": "text",

```

```
    "tips": "提示",
    "rules": "required;length(3~16)",
    "error_tips": "值必须",
    "value": ""
  },
  "textarea": {
    "title": "描述",
    "type": "textarea",
    "tips": "",
    "rules": "",
    "error_tips": "",
    "value": ""
  },
  "radio": {
    "title": "是否显示",
    "type": "radio",
    "tips": "",
    "rules": "",
    "error_tips": "",
    "options": {
      "1": "显示",
      "0": "不显示"
    },
    "value": "1"
  },
  "checkbox": {
    "title": "多项选择",
    "type": "checkbox",
    "tips": "",
    "rules": "",
    "error_tips": "",
    "options": {
      "1": "显示1",
      "2": "显示3",
      "0": "不显示"
    },
    "value": "1,2"
  },
  "select": {
    "title": "下拉选择",
    "type": "select",
    "tips": "我是提示",
    "rules": "",
    "error_tips": "",
    "options": {
      "1": "测试1",
      "2": "测试2",
      "3": "测试3"
    },
    "value": "2"
  }
```

```

    },
    "value": "2"
  },
  "selects": {
    "title": "下拉多项选择",
    "type": "selects",
    "tips": "我是提示",
    "rules": "",
    "error_tips": "",
    "options": {
      "1": "测试1",
      "2": "测试2",
      "3": "测试3"
    },
    "value": "2,3"
  }
},
"more": {
  "title": "更多选项",
  "item": {
    "image": {
      "title": "图片",
      "type": "image",
      "tips": "请上传图片",
      "rules": "required",
      "error_tips": "请上传图片",
      "value": ""
    },
    "images": {
      "title": "图片",
      "type": "images",
      "tips": "请上传图片",
      "rules": "required",
      "error_tips": "请上传图片",
      "max_number": 5,
      "value": ""
    },
    "date": {
      "title": "日期",
      "type": "date",
      "tips": "",
      "rules": "required",
      "error_tips": "日期选择",
      "value": ""
    },
    "datetime": {
      "title": "日期时间",
      "type": "datetime",
      "tips": "日期时间",

```

10. 模板配置JSON

```
    "rules": "",
    "error_tips": "日期时间",
    "value": ""
  }
}
```

分组后的效果图

基础配置 更多选项

标题: ⓘ

描述:

是否显示: 显示 不显示

多项选择: 显示1 显示3 不显示

下拉选择: ▼ ⓘ

下拉多项选择: ⓘ

测试1

测试2

测试3

提交 重置

11. 导入演示数据

导入演示数据

在模板主题根目录命名为：demodata.sql，在安装的时候用户勾选演示数据，即可导入演示数据。演示数据不是必须的，根据自己意愿来提供。

数据库里面的表前缀使用 @prefix@ 代替，如下

```
insert into `@prefix@archives`(`id`,`category_id`,`model_id`,`admin_id`,`title`,`thumb`,`seotitle`,`keywords`,`description`,`tags`,`show_tpl`,`publish_time`,`create_time`,`update_time`,`delete_time`,`weigh`,`author`,`url`,`views`,`lang`,`status`,`flags`)  
  
values (null,6,1,1,'Honor certificate','/static/module/index/default/img/honor.jpg','','','','','show.html',1636988345,1636988345,1636988345,NULL,0,'小编','','0','en','normal','');
```

效果

默认前台模板2
— ✕ ✕

数据库表检测

数据库表	检测结果
hkcms_agc	✔ 通过
hkcms_archives	⚠ 已存在

目录文件权限

目录文件	检测结果
D:/hkcms/HkCms-develop/public/static/	✔ 通过
D:/hkcms/HkCms-develop/public/admin.php	✔ 通过
D:\hkcms\HkCms-develop\public\static\module\index\	✔ 通过

演示数据

导入演示数据

确认安装
刷新
取消

12. 卸载删除演示数据

卸载删除演示数据

用于删除导入的演示数据，如果开发者提供了演示数据就必须提供一份删除演示数据的sql文件

在模板主题根目录命名为：undemodata.sql

数据库里面的表前缀使用 @prefix@ 代替，如下

```
insert into `@prefix@archives`(`id`,`category_id`,`model_id`,`admin_id`,`title`  
,`thumb`,`seotitle`,`keywords`,`description`,`tags`,`show_tpl`,`publish_time`,`  
create_time`,`update_time`,`delete_time`,`weigh`,`author`,`url`,`views`,`lang`,`  
`status`,`flags`)  
  
values (null,6,1,1,'Honor certificate','/static/module/index/default/img/honor.  
jpg','','','','','show.html',1636988345,1636988345,1636988345,NULL,0,'小编','','0  
, 'en', 'normal', '');
```

将在卸载模板时执行undemodata.sql文件

13. 模板文件打包

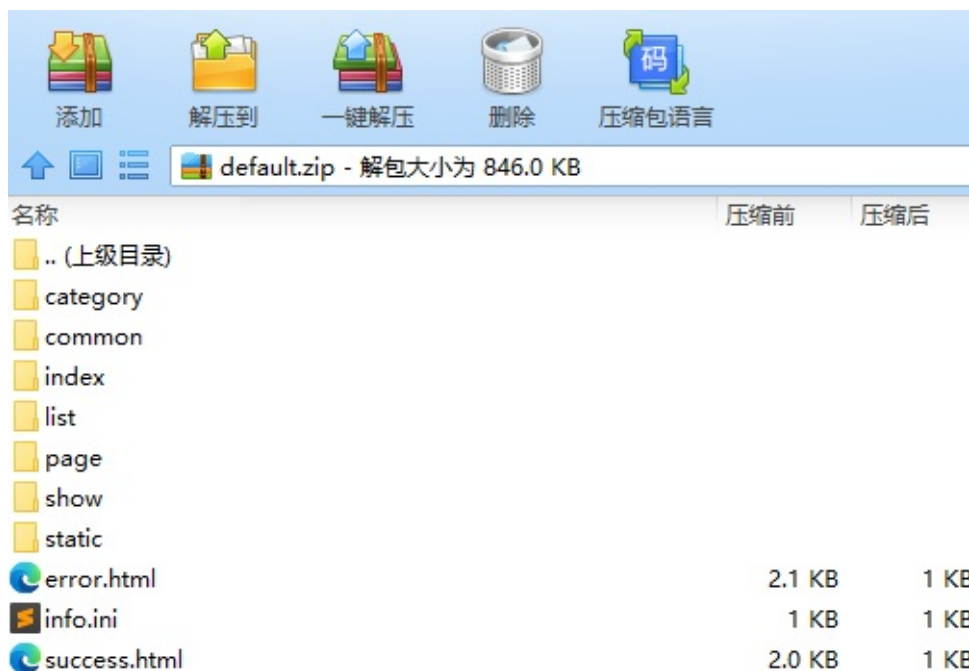
模板文件打包

模板编写完成后，复制模板名称文件下的根目录，同时将静态文件复制进来，命名为“static”。如下图

category	2021/4/15 19:46
common	2021/4/15 19:46
index	2021/4/15 19:46
list	2021/4/15 19:46
page	2021/4/15 19:46
show	2021/4/15 19:46
static	2021/4/16 9:22
error.html	2021/4/10 10:34
info.ini	2021/4/15 19:50
success.html	2021/4/10 10:34

全选文件，右键压缩，结构如下图所示。

注意：如果上传到应用中心，info.ini 里面的版本与应用标识跟上传到应用中心所填写信息保持一致。



- 新版支持后台一键导出

14. 整站源码打包

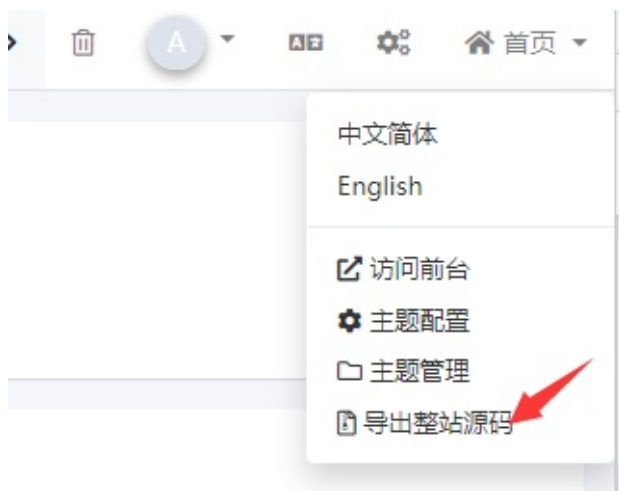
整站源码打包

除了上一章模板文件打包外，系统还提供了整站源码一键打包功能，你可以很方便的打包当前模板以及所配套的程序、数据发布到应用中心售卖。流程步骤也是跟模板创建一致

导出

在制作好你的模板后，在后台点击导出整站源码。

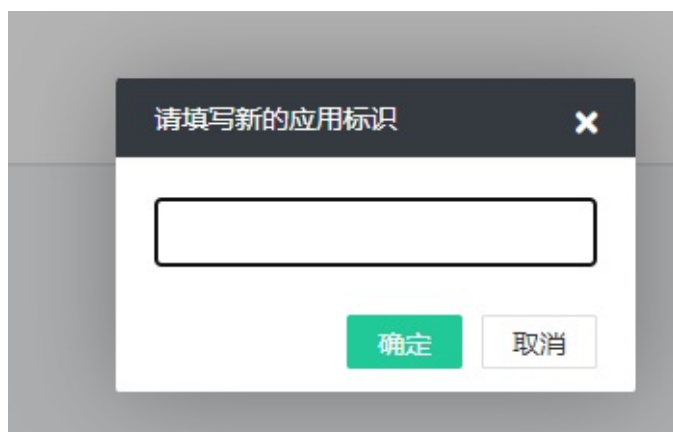
通过右上角导出当前主题



或在应用中心本地里面导出当前模板



填写模板标识，该标识用于应用中心发布时填写。



标识与之对应

* 标题:

* 应用标识:

* 分类: 整站源码

* 封面: 上传

如果标识已存在请重新生成新的标识后再试，也可通过 [插件名称检测](#)

包含的文件与数据

模板只包含当前主题的模板不涉及数据库账号密码，导出的数据库不包含管理员以及日志、官网账号等信息

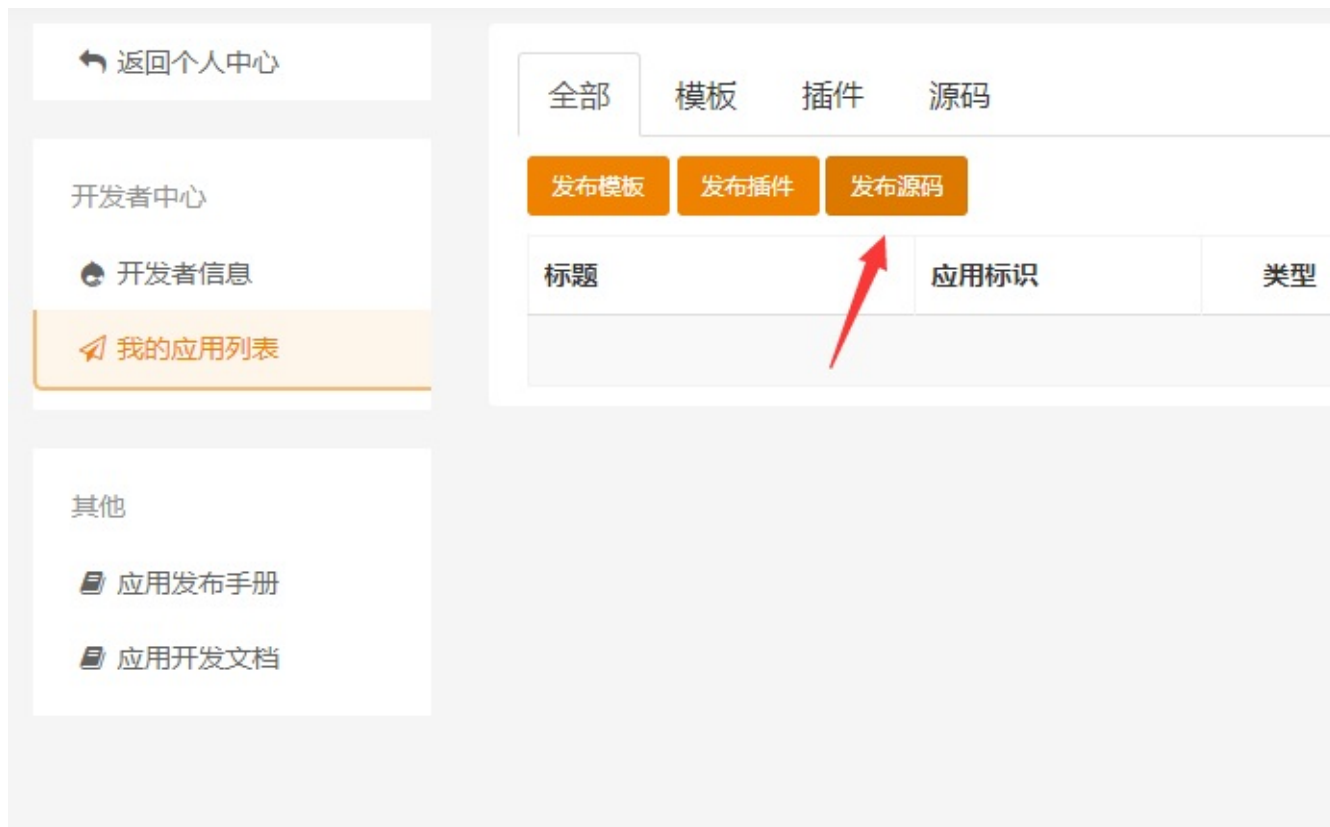
.. (上级目录)	
addons	
app	
config	
extend	
public	
runtime	
template	
vendor	
.env	1 KB
composer.json	1.4 KB
composer.lock	111.8 KB
install.sql	255.4 KB
think	1 KB

← 导出的数据库

注意：导出会包含的附件，请手动删除不需要的附件。

源码发布

1. 注册官网账号并认证开发者
2. 在侧边栏找到开发者中心进入开发者中心发布源码



在发布源码界面上传刚导出的源码包即可。

源码的安装

源码安装与HkCms安装一致

插件开发

1. 创建插件
2. 目录结构
3. 插件名称类
4. 插件信息(info.ini)
5. 插件事件
6. 配置文件
7. install 安装目录
8. 创建菜单
9. 插件控制器
10. 插件视图
- 10-1 插件主题化
11. 插件模型
12. 模型事件(已废弃)
13. 插件内置函数
- 13-1 插件函数库
14. 插件多语言
15. 插件中间件
16. 插件路由
17. 插件验证器
18. 标签库
19. 第三方类库
20. 数据库文件导入

插件打包

后台前端使用帮助

1. 创建插件

创建插件

首先安装“一键生成应用”插件，来帮助我们快速的创建插件。

操作说明

需要调整，先手动创建所需的文件夹

2. 目录结构

目录结构

以demo插件名称为例，以下包含常见的插件目录与文件，根据您的插件需要自行新建、删除目录

```

addons文件夹
├─demo 插件名称标识
│   ├─controller 插件控制器
│   ├─model 插件模型
│   ├─view 插件视图
│   ├─middleware 插件中间件
│   ├─validate 插件验证器
│   ├─lang 插件语言包，仅用于插件内的访问
│   ├─taglib 标签库
│   └─install 安装目录，与app、template、static、public目录保持一致，具体看后面的章节
│       ├─Demo.php 与插件名称保持一致，其中首字母大写（该文件必须存在）
│       ├─config.php 配置文件
│       ├─common.php 插件函数库（插件内使用）
│       ├─route.php 插件路由
│       ├─service.ini 插件服务注册
│       ├─install.sql 插件安装时自动导入数据库
│       ├─upgrade.sql 插件升级时自动执行改SQL文件
│       ├─demodata.sql 安装时用户勾选导入演示数据时自动导入
│       └─info.ini 插件信息（该文件必须存在）

```

其中 `info.ini` 文件与 `Demo.php` 文件必须存在，其中 `Demo.php` 文件名称取决于你的插件名称，主要用于安装、卸载、禁用、启用、处理事件逻辑。

插件名称标识

命名规则：小写字母，支持字母数字结合，建议使用字母

例如你的插件名称叫 `name`，那么目录结构如下

```

addons文件夹
├─name
│   ├─Name.php 首字母大写
│   └─info.ini 插件信息

```

注意：为防止插件名称与其他插件名称重名，可在HkCms后台应用中心安装“自动生成应用插件”插件来创建插件，如果你是手动创建可前往：[插件名称检测](#)

3. 插件名称类

插件名称类

与插件名称一致，其中首字母大写。例如Demo类对应demo标识。主要用于安装、卸载、禁用、启用、处理事件逻辑。该文件必须存在

目录位置

```
addons
├─demo
│   └─Demo.php ->与插件名称保持一致，其中首字母大写（该文件必须存在）
│   └─info.ini
```

文件代码示例

每个名称类，需要继承Addons类，里面有提供获取插件信息，配置文件，加载模板等方法。

```
<?php
declare (strict_types=1);

// 命名空间, addons文件夹, demo 你的插件文件夹
namespace addons\demo;
use think\Addons;
class Demo extends Addons
{
    // 插件安装时的处理(必须存在该方法)
    public function install()
    {
        return true;
    }

    // 插件卸载时的处理(必须存在该方法)
    public function uninstall()
    {
        return true;
    }

    // 插件启用时触发(可选)
    public function enable()
    {
        return true;
    }

    // 插件禁用时的处理(可选)
```

3. 插件名称类

```
public function disable()  
{  
    return true;  
}  
  
// 插件升级时的处理(可选)  
public function upgrade()  
{  
    return true;  
}  
}
```


4. 插件信息(info.ini)

插件信息

插件信息是插件必须的信息，位于插件根目录下。命名为 `info.ini` 。

```
name = "demo"
type = "addon"
title = "simditor编辑器"
description = "simditor编辑器"
author = "HkCms"
version = "1.0.0"
status = 1

database = "table2,table1,table3"
dir = "app/admin/,app/index/"

[addon]
simditor = '1.0.0'
user = '1.0.0'
```

name (必填)

插件名称，以字母开头，必须同目录大小写一致，可参考目录结构说明。

type (必填)

插件就写 `type="addon"` 即可

title (必填)

插件标题

author (必填)

插件作者，必填字段

version (必填)

版本号，格式为：xx.xx.xx。

database

数据库，不需要写表前缀，用于安装时检测数据库表是否存在，如果已存在则提示用户。

dir

目录，用于目录权限检测，如果检测不通过，提示用户需要有写入的权限。

addon

插件依赖，左边为插件名称(插件唯一标识)，右边是依赖的最低插件版本号，在这里填写后，如果用户未安装依赖的插件则会提示用户需要先安装某某插件。

提示：依赖的插件名称获取方式：在安装所需的插件后找到 `info.ini` 文件里面的name，这个就是插件名称。

5. 插件事件

事件说明

事件的定义写在 `插件名称类` 文件里面，方法命名方式以Hook结尾，字母驼峰式命名。

插件能使用的事件

2.1.220504版本起插件事件不在使用下划线调用，但依然支持旧版的下划线方式

事件名称	插件名称类使用方式	描述	参数说明
addons_init 2.1.220504版本起：addonsInit	addonsInitHook	插件初始化，这个事件能影响到前台、后台，优先级最高	-
addonMiddleware	addonMiddlewareHook	插件全局中间件，只要访问任意插件前台时触发	请求的Request对象
addonsBegin	addonsBeginHook	插件路由调度前事件，访问插件控制器时触发	请求的Request对象
addonModuleInit	addonModuleInitHook	插件控制器初始化前事件，访问插件控制器时触发	请求的Request对象
addonsActionBegin	addonsActionBeginHook	插件控制器操作方法执行前事件，访问插件控制器操作方法时触发	传入控制器对象与将要操作的方法
indexRoute	indexRouteHook	访问前台时，在路由初始化时触发，	无
adminRoute	adminRouteHook	访问后台时，在路由初始化时触发	无
apiRoute	apiRouteHook	访问api时时，在路由初始化时触发	无
configInit	configInitHook	访问前台或后台时，站点配置初始化后触发（前后台的基础类）	站点配置
adminLoginSucc	adminLoginSucc	后台登录成功后标	

ess	essHook	签位	登录信息
indexHead	indexHeadHook	前台模板头部标签位	无
indexFooter	indexFooterHook	前台模板底部标签位	无
themeChange	themeChangeHook	模板主题切换事件	传入 index_theme , 或 admin_theme
uploadAfter	uploadAfterHook	文件上传后触发	传入文件信息数组
uploadDel	uploadDelHook	文件删除后触发	传入删除的模型对象
addonConfig_save		插件保存配置时触发 (已废弃)	提交的配置表单
插件标识 ConfigSave	插件或模板配置保存时触发 (2.1.220220新增)	提交的配置表单, 例如你的插件叫: demo, 事件定义为: demoConfigSaveHook	
apiLoginMiddleware	apiLoginMiddlewareHook	API登录中间件	请求Request对象
apiAuthMiddleware	apiAuthMiddlewareHook	API权限中间件	请求Request对象
HttpRun	HttpRunHook	应用开始标签位 (TP内置)	无
HttpEnd	HttpEndHook	应用结束标签位 (TP内置)	当前响应对象实例
RouteLoaded	RouteLoadedHook	路由加载完成 (TP内置)	无
LogWrite	LogWriteHook	日志write方法标签位 (TP内置)	当前写入的日志信息

使用方式

例如插件名称叫 `demo` , 在 `Demo` 插件名称类里面, 若要使用 `indexHead` 前台模板头部事件。这里首先需要模板有放置该标签事件, 可前往默认模板找到 `head.html` 文件, 即可看到 `{:hook("indexHead")}` , 当访问前台页面执行到 `{:hook("indexHead")}` 时就会调用下面的 `indexHeadHook` 方法。

完整写法如下

```

<?php
declare (strict_types=1);
namespace addons\demo;
use think\Addons;
class Demo extends Addons
{
    public function install()
    {
        return true;
    }

    public function uninstall()
    {
        return true;
    }

    // 前台模板执行到<head></head>时触发，通常用于加载JS、CSS达到特定的需求
    public function indexHeadHook()
    {
        // 加载js
        return "<script src='/static/main.js'></script>";
        // 或直接渲染视图，可参考插件视图
        return $this->fetch('index/index');
    }
}

```

扩展

当我们在写业务逻辑时，需要自己定义一些事件标签位来给其他插件实现功能扩展，那么我们可以参考下面来实现。

1. hook方法

系统有提供 `hook` 方法来触发事件，方法说明：

```

/**
 * $event 事件名称
 * $params 传入参数
 * $once = true 表示返回一个结果，$once = false 表示返回全部结果
 * $original = true 表示 返回TP trigger原始数据，false - 返回字符串，如果原样返回数据是
 数组即会转换成字符串
 */
hook($event, $params, $once, $original)

```

2. 触发事件

例如定义触发 `showTest` 事件。

5. 插件事件

```
// 控制器触发, 传入数组
hook('showTest', ['a'=>'测试']);

// 模板触发, 传入数组
{:hook("showTest", ['a'=>'测试'])}

// 接收返回值: 输出原始数据。第四个参数给true即可
$arr = hook("showTest", ['a'=>'测试'], true, true);
```

注意：当有多个插件定义showTest事件时，返回值是数组的，并不建议这样来使用事件

上面定义了 showTest 触发事件方法，那么在其他插件就可以定义 showTest 事件来实现扩展，如下

```
<?php
declare (strict_types=1);
namespace addons\demo;
use think\Addons;
class Demo extends Addons
{
    public function install()
    {
        return true;
    }
    public function uninstall()
    {
        return true;
    }

    // 对应showTest
    public function showTestHook()
    {
        echo "测试";
    }
}
```

6. 配置文件

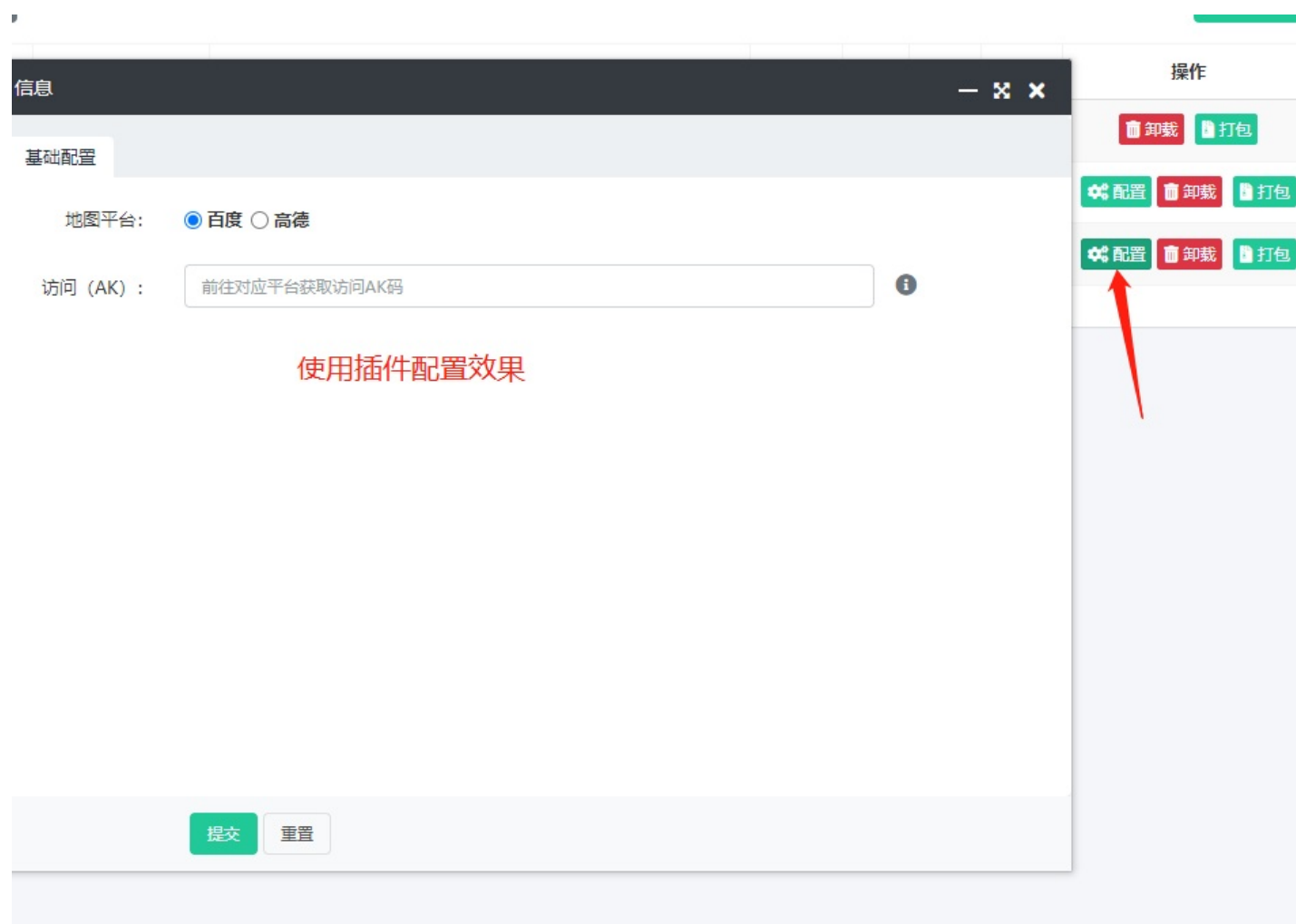
配置文件

插件配置文件用于给用户自定义一些选项，用户安装后，可根据需要灵活配置你的插件。

目前支持text、textarea、radio、checkbox、select、image、images、date、datetime、array组件。

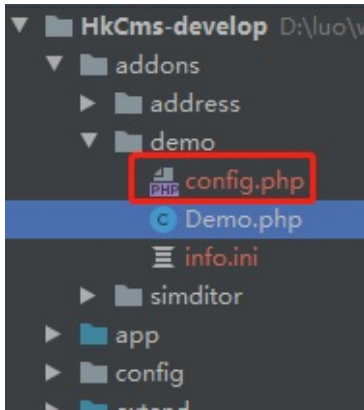
组件可前往这里了解：[字段类型与组件](#)

配置后的效果



文件路径

```
addons
├─demo
│  ├─Demo.php
│  ├─config.php    配置文件
│  └─info.ini
```



无标签页示例

```
<?php
return [
    // 名称, 保持唯一, 获取时: $tmp['value1'] 即可获取value值。
    'value1' => [
        'title' => '标题', // 标题
        'type' => 'text', // 组件类型
        'tips' => '提示', // 提示信息
        'rules' => 'required;length(3~16)', // 验证规则, 可参考字段类型与组件下的验证规则, 多个规则;隔开
        'error_tips'=>'值必须', // 错误的提示信息
        'value' => '' // 默认值
    ],
    'value2' => [
        'title' => '描述',
        'type' => 'textarea',
        'tips' => '',
        'rules' => '',
        'error_tips'=>'',
        'value' => ''
    ],
    'radio' => [
        'title' => '是否显示',
        'type' => 'radio',
        'tips' => '',
        'rules' => '',
        'error_tips'=>'',
        'options' => [ // 选项
            '1' => '显示',
            '0' => '不显示'
        ],
        'value' => '1'
    ],
    'checkbox' => [
        'title' => '多项选择',
        'type' => 'checkbox',
        'tips' => '',

```



```

    'rules' => '',
    'error_tips'=>'',
    'options' => [
        // key=>标题
        '1' => '显示1',
        '2' => '显示3',
        '0' => '不显示'
    ],
    'value' => '1,2' // 多选时英文逗号分隔, 对应选项key值
],
'select' => [
    'title' => '下拉选择',
    'type' => 'select',
    'tips' => '我是提示',
    'rules' => '',
    'error_tips'=>'',
    'options' => [
        '1' => '测试1',
        '2' => '测试2',
        '3' => '测试3'
    ],
    'value' => '2'
],
'array1' => [
    'title' => '测试array组件',
    'type' => 'array',
    'tips' => '',
    'rules' => '',
    'error_tips'=>'',
    'data_list'=>[// 键值名称
        'key'=>'标题',
        'value'=>'值'
    ],
    'value' => [
        '标题1'=>'2'// 默认值
    ]
],
'value5' => [
    'title' => '动态下拉url类型',
    'type' => 'selectpage', // 动态下拉
    'tips' => '',
    'rules' => '',
    'error_tips'=>'',
    'data_list'=>[
        'type'=>'url',//链接地址类型
        'url'=>'/cms.flags/index',//链接地址类型
        'show-field'=>'title', // 显示字段
        'key-field'=>'id',// 存储字段
        'search-field'=>'title',// 搜索字段
        'and-or'=>'',// 值为and、或者or, search-field为多个时, 值为and时表示同时
    ]
]

```

成立反之其中一个成立即可

```

        'order-by'=>'',// 排序:id desc,view asc
        'param'=>[
            'status'=>'normal' // 指定额外的条件
        ],
        'multiple'=>0,// 多选, 1=开启, 0=关闭
        'pagination'=>0,// 分页, 1=开启, 0=关闭
        'select-only'=>1,// 只读, 1开启, 0关闭
        'enable-lang'=>1,// 启用多语言区分, 1开启, 0关闭
    ],
    'value' => ''
],
'value4' => [
    'title' => '动态下拉table类型',
    'type' => 'selectpage',
    'tips' => '',
    'rules' => '',
    'error_tips'=>'',
    'data_list'=>[
        'type'=>'table',// 数据源来自数据库表
        'table'=>'flags',// flags表, 不用写表前缀
        'show-field'=>'title',
        'key-field'=>'id',
        'search-field'=>'title,name',
        'and-or'=>'or',
        'order-by'=>'id desc',
        'param'=>[
            'status'=>'normal'
        ],
        'multiple'=>1,// 多选, 1=开启, 0=关闭
        'pagination'=>0,// 分页, 1=开启, 0=关闭
        'select-only'=>0,// 只读, 1开启, 0关闭
        'enable-lang'=>1,// 启用多语言区分, 1开启, 0关闭
    ],
    'value' => ''
],
];

```

效果图

信息
— ✕ ×

标题: i

描述:

是否显示: 显示 不显示

多项选择: 显示1 显示3 不显示

下拉选择:

测试2
▼
i

如果配置项过多，可以使用下面方式，生成标签页

标签页分组示例

使用标签页布局时，获取的配置结构是不一样的。

```
return [
  // 自定义名称，获取：$tmp['base']['value1']
  'base'=>[
    // 标签页标题
    'title' => '基础配置',
    'item' => [ // 标签页包含的配置项
      'value1' => [
        'title' => '标题',
        'type' => 'text',
        'tips' => '提示',
        'rules' => 'required;length(3~16)',
        'error_tips'=>'值必须',
        'value' => ''
      ],
      'value2' => [
        'title' => '描述',
        'type' => 'textarea',
        'tips' => '',
        'rules' => '',
        'error_tips'=>'',
        'value' => ''
      ]
    ],
  ],
],
```

```
'radio' => [
  'title' => '是否显示',
  'type' => 'radio',
  'tips' => '',
  'rules' => '',
  'error_tips'=>'',
  'options' => [
    '1' => '显示',
    '0' => '不显示'
  ],
  'value' => '1'
],
'checkbox' => [
  'title' => '多项选择',
  'type' => 'checkbox',
  'tips' => '',
  'rules' => '',
  'error_tips'=>'',
  'options' => [
    '1' => '显示1',
    '2' => '显示3',
    '0' => '不显示'
  ],
  'value' => '1,2'
],
'select' => [
  'title' => '下拉选择',
  'type' => 'select',
  'tips' => '我是提示',
  'rules' => '',
  'error_tips'=>'',
  'options' => [
    '1' => '测试1',
    '2' => '测试2',
    '3' => '测试3'
  ],
  'value' => '2'
],
'selects' => [
  'title' => '下拉多项选择',
  'type' => 'selects',
  'tips' => '我是提示',
  'rules' => '',
  'error_tips'=>'',
  'options' => [
    '1' => '测试1',
    '2' => '测试2',
    '3' => '测试3'
  ],
  'value' => '2,3'
```

```

    ]
  ],
  'more'=>[
    'title'=>'更多选项',
    'item'=>[
      'image' => [
        'title' => '图片',
        'type' => 'image',
        'tips' => '请上传图片',
        'rules' => 'required',
        'error_tips'=>'请上传图片',
        'value' => ''
      ],
      'images' => [
        'title' => '图片',
        'type' => 'images',
        'tips' => '请上传图片',
        'rules' => 'required',
        'error_tips'=>'请上传图片',
        'max_number'=>5,
        'value' => ''
      ],
      'date' => [
        'title' => '日期',
        'type' => 'date',
        'tips' => '',
        'rules' => 'required',
        'error_tips'=>'日期选择',
        'value' => ''
      ],
      'datetime' => [
        'title' => '日期时间',
        'type' => 'datetime',
        'tips' => '日期时间',
        'rules' => '',
        'error_tips'=>'日期时间',
        'value' => ''
      ]
    ]
  ]
];

```

效果

基础配置
更多选项

标题: i

描述:

是否显示: 显示 不显示

多项选择: 显示1 显示3 不显示

下拉选择:

测试2
▼

i

下拉多项选择:

测试1

测试2

测试3

▲
▼

i

标签选项页

获取插件配置

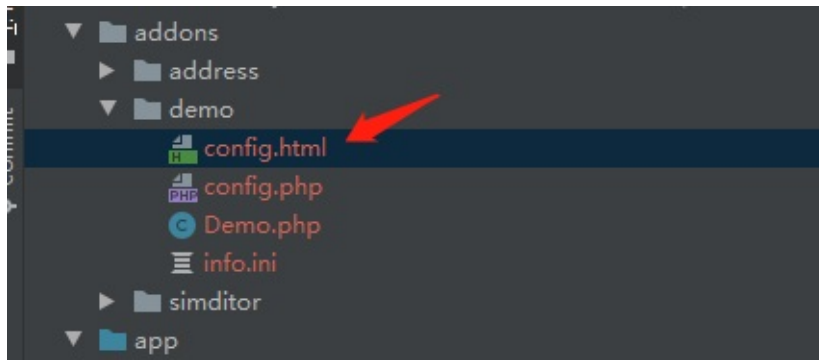
```
/**
 * $type = 'template' 表示模板应用, $type='addon' 表示插件应用
 * $name 插件标识名称
 * $module 模板应用会用到, 用于确定获取后台模板配置、或前台模板、或其他模块的模板。
 * $complete true-表示获取所有结构数组, false-表示获取配置值
 */
get_addons_config($type, $name, $module, $complete)

// 例如获取demo插件配置
get_addons_config('addon', 'demo');
```

自定义配置页

如果系统自带的配置页, 满足不了你的需求时, 您可以自定义配置页, 只需在自己的插件目录新建config.html即可。

6. 配置文件



在html页面获取的变量有：`$sdata` 里面的值结构就是你写的配置文件结构，你可参考默认的表单生成文件 `template/admin/adminlte/appcenter/field.html`。

打印输出：

```
<?php dump($sdata);?>
```

```
^ array:2 [▼
  "base" => array:2 [▼
    "title" => "基础配置"
    "item" => array:6 [▶]
  ]
  "more" => array:2 [▼
    "title" => "更多选项"
    "item" => array:4 [▼
      "image" => array:6 [▶]
      "images" => array:7 [▶]
      "date" => array:6 [▶]
      "datetime" => array:6 [▶]
    ]
  ]
]
```

提示：若对自定义配置页仍有疑问，欢迎前往Q群咨询。

7. install 安装目录

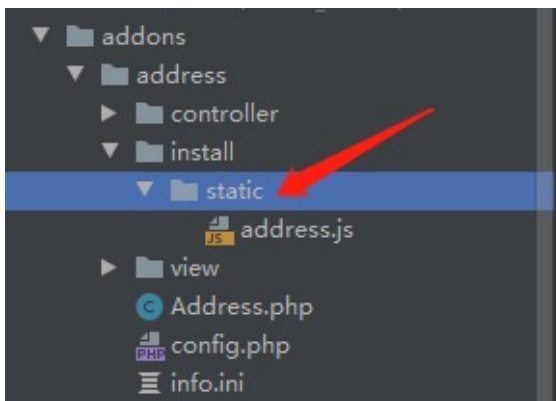
install 目录

如果你的插件是基于后台或前台增加功能，那么可以新建install目录来实现对前台或后台的扩展。

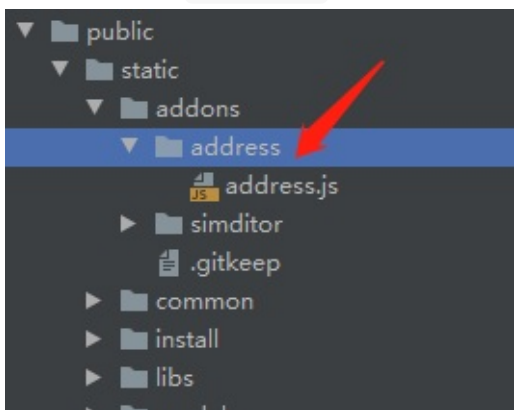
目前支持template、app、public、static文件夹的写入。

static插件静态文件

例如下图的 address 插件，在 install 目录创建 static 文件夹，里面放入js文件。（放css，图片等都可以）



安装后，或启用插件后，static 里面的文件就移动到 /public/static/addons/插件名称标识/ 目录下。例如下图 address 插件安装后的效果



- 后台全局JS

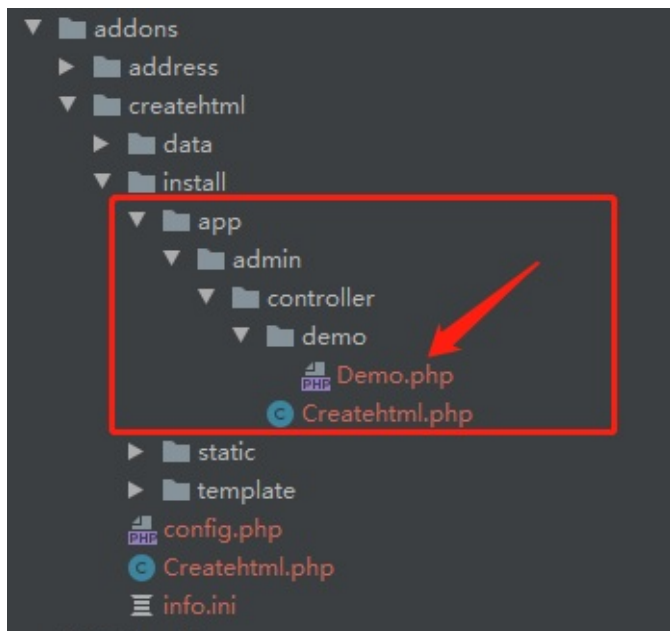
当你的 static 目录下有跟你插件同名的JS时，该文件会自动被后台加载。例如你的插件叫 demo ，安装后 static/demo/demo.js 有 demo.js ，后台就会引用 demo.js

app、public 目录示例

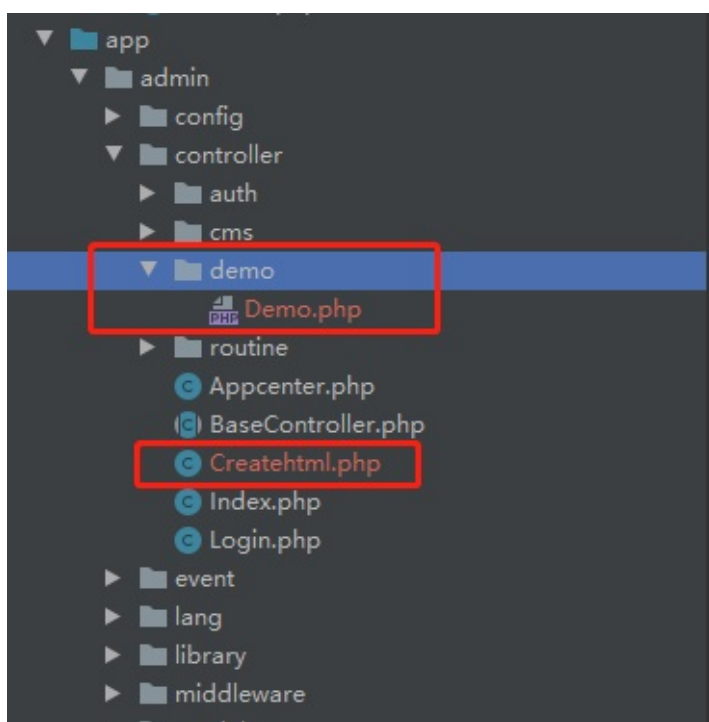
app 目录和 public 目录跟项目根目录下的 app 目录保持一致的目录结构即可。

示例图：

7. install 安装目录



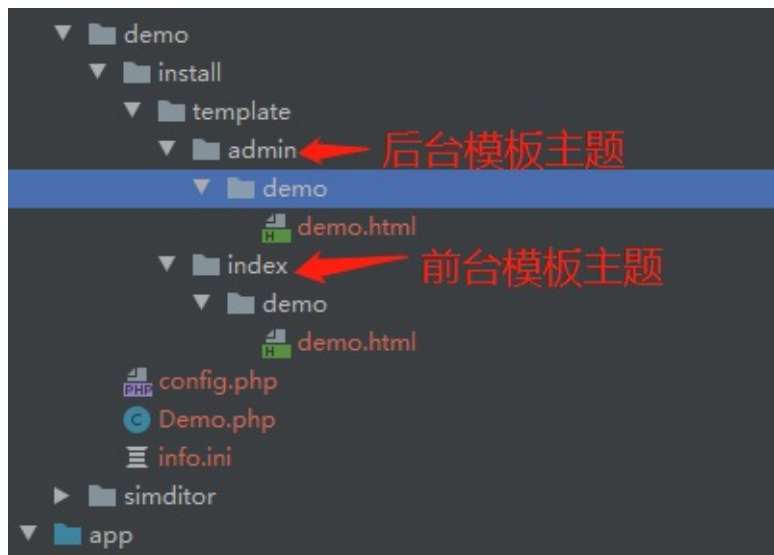
安装后根据目录结构复制到后台控制了。



template 主题目录示例

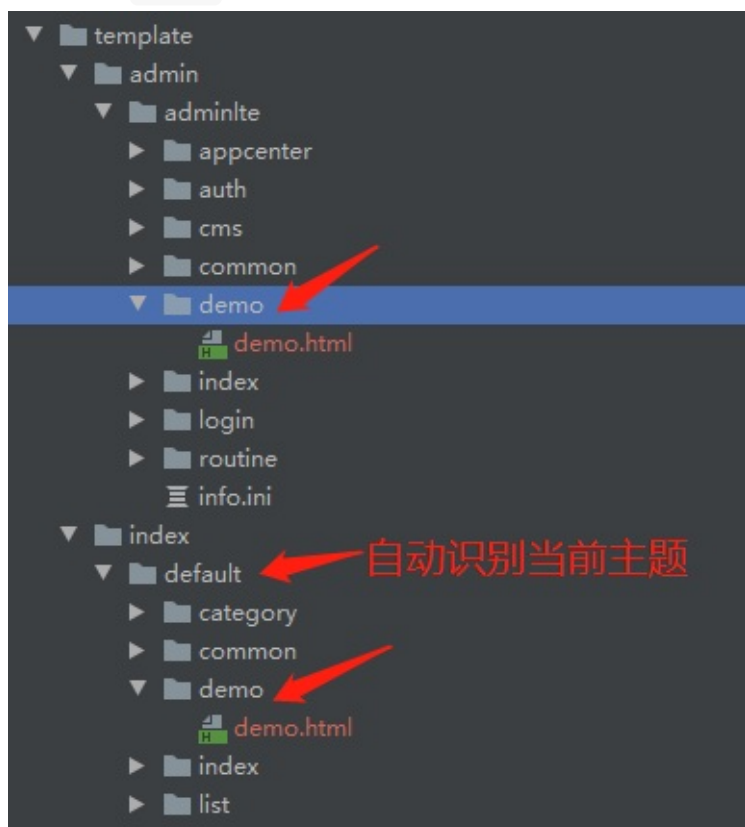
为方便安装到指定的主题目录，你只需按如下图放置，无需创建主题文件夹，在安装的时候会自动安装到当前模块主题文件夹下。

7. install 安装目录



注意：上图只有index、admin模块文件夹，如果你写了一个不存在的模块，系统会认为你的插件也想模板主题化则会新建模块主题。若想插件主题化可参考插件主题化章节。

安装后，demo 文件夹将覆盖到对应模块主题目录下，如下图所示：



注意：install 安装目录里面的文件，在禁用插件时会移除安装的文件，启用时则会覆盖

8. 创建菜单

创建后台菜单

菜单数据以数组形式定义，定义好之后可以通过 `create_menu` 函数手动创建菜单，通过 `del_menu` 函数来手动删除菜单。当然，除了以上方式还支持系统自动判断导入菜单，请参考下面的自动导入方式。

字段说明

2.1.220220版本之前，字段需要填写完整

字段名	说明	默认值	是否必填
parent_id	父菜单ID或父菜单的name，默认0，表示最顶级	0	-
title	菜单名称，菜单多语言目前需要在admin_route后台路由事件中手动导入菜单语言包	-	必填
name	权限规则	-	必填
route	路由	-	-
icon	图标	fas fa-align-justify/far fa-circle	-
remark	备注	-	-
weigh	排序	0	-
type	类型:0-权限规则,1-菜单,2-菜单头(提供的额外标识)	1	-
child	子级菜单，有子级的话按一级菜单字段填写即可	-	-

1. name，权限规则

写法：控制器/方法，二级文件夹则是文件夹名/控制器/方法，注意二级需要填写路由

2. route，路由地址

写法：文件夹名.控制器/方法，常用于有文件夹的情况，注意是“点”

8. 创建菜单

3. icon, 图标

基于 `fontawesome` , 一级菜单默认为 `fas fa-align-justify` , 二级菜单默认 `far fa-circle` 图标地址 : <https://fa5.dashgame.com/#/图标>

4. type, 菜单类型

未填写 `type` 时系统会自动判断菜单类型, 在认为有子级情况下认为是菜单。若有特殊情况, 您需要填写 `type` 字段

自动导入示例

在插件名称类里面增加公共属性 `$menu` , 在安装时检测到有这个属性就会自动导入, 另外在 `卸载` 、 `启用` 、 `禁用` 、 `更新` 等操作时系统会自动对菜单进行相应的操作, 例如插件禁用, 菜单也会被禁用。

注意: 如果菜单已存在并且这个菜单不属于该插件就不会导入, 如果已存在, 属于该插件则是更新菜单。

插件升级操作时, 同样会根据menu导入, 如果存在则是更新该菜单, 如有额外处理, 可在upgrade方法中操作

```
<?php
declare (strict_types=1);
namespace addons\demo;
use think\Addons;
class Demo extends Addons
{
    // 菜单
    public $menu = [
        [
            'title'=>'插件测试',
            'name'=>'demo',
            'child'=>[
                ['title'=>'查看', 'name'=>'demo/index'],
                ['title'=>'添加', 'name'=>'demo/add'],
                ['title'=>'修改', 'name'=>'demo/edit'],
                ['title'=>'删除', 'name'=>'demo/del'],
            ]
        ]
    ];

    public function install()
    {
        return true;
    }

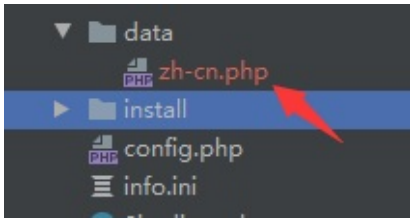
    public function uninstall()
    {
        return true;
    }
}
```

8. 创建菜单

```
}
```

菜单多语言

在插件目录，在data目录下创建zh-cn.php文件，如下图



目前后台内置的中英语言包、因此在上面讲的\$menu变量里面title全部写成英文，然后下面的写上英文跟中文的对照。

```
<?php
return [
    'Addon test' => '插件測試'
];
```

支持其他语言包写法，例如繁体中文命名为：zh-tw.php，那么代码如下：

```
<?php
return [
    'Addon test' => '插件測試'
];
```

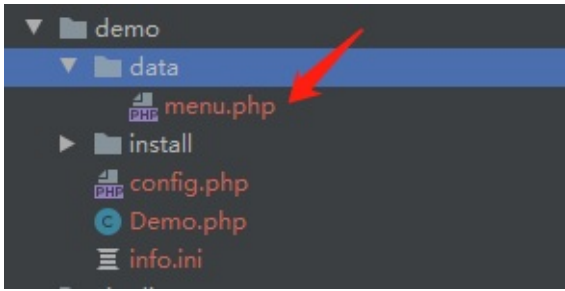
创建文件后我们需要在插件初始化时加载语言包

```
public function addonsInitHook()
{
    // 获取语言包对象
    $lang = app()->lang;
    // 加载语言包, $this->addon_path 插件目录, DIRECTORY_SEPARATOR 是 '/' 或 '\', $
    lang->getLangset() 是当前语言
    $lang->load($this->addon_path.'data'.DIRECTORY_SEPARATOR.$lang->getLangset()
    .'php');
}
```

手动导入

如下，我们可以创建PHP文件，放到插件的data目录下，如下

8. 创建菜单



```
<?php
return [
    [
        // 父菜单ID或父菜单name或0
        "parent_id" => 57,
        // 标题
        "title" => '菜单名称',
        // 权限认证规则
        "name"=>"createhtml",
        // 路由地址
        "route" => "",
        // fontawesome 图标
        "icon" => 'fas fa-print',
        // 备注
        "remark" => "",
        // 排序
        "weigh" => 10,
        // 菜单类型
        "type" => 1,
        // 子级
        "child"=>[
            [
                // 标题
                "title" => '菜单名称1',
                // 权限认证规则
                "name"=>"createhtml/index1",
                // 子级
                "child"=>[
                    ["title" => '菜单名称2', "name"=>"createhtml/index",]
                ]
            ],
            [
                // 标题
                "title" => '菜单名称3',
                // 权限认证规则
                "name"=>"createhtml/index",
            ]
        ]
    ],
];
```

8. 创建菜单

在安装的时候导入，使用 `create_menu` 方法导入。导入后无需管禁用、启用与卸载的操作了，系统会自动去操作菜单。

```
public function install()
{
    // 菜单数组文件路径
    $menu = include $this->addon_path.'data'.DIRECTORY_SEPARATOR.'menu.php';
    // 参数一给路径，参数二给当前插件的标识
    create_menu($menu,$this->getName());
    return true;
}
```

方法的使用

- `create_menu` 创建菜单

参数1为菜单数组，参数2为菜单属于的插件标识。

注意：如果菜单已存在并且这个菜单不属于该插件就不会导入，如果已存在，属于该插件则是更新菜单

```
// 在插件里面可以直接获取到插件标识，当然你也可以直接写上你的插件标识
create_menu($menu, $this->getName());
```

- `status_menu` 更改状态：启用与禁用.

参数1为：1=启用，0=禁用，参数2为插件名称标识

```
status_menu(1, $this->getName()); // 启用菜单
```

```
// 例如插件叫demo
status_menu(0, 'demo'); // 禁用demo的菜单
```

- `del_menu` 删除菜单

```
// 参数1为插件标识
del_menu($name);
```

9. 插件控制器

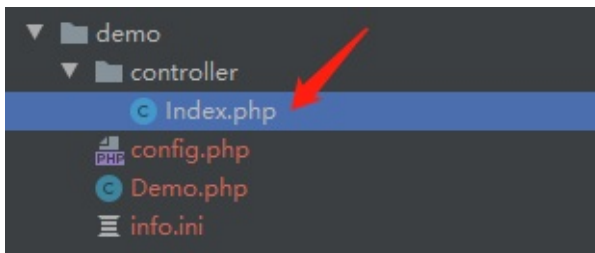
插件控制器

插件除了基于前台与后台外，还可以作为独立的模块。创建控制器后即可赋予插件前台访问功能。

创建控制器

控制器放在 `controller` 文件夹下，以 `demo` 插件为例，创建首页控制器，路径为：

`addons/demo/controller/Index.php`，如下图



Index.php 代码

```
<?php

namespace addons\demo\controller;

use think\addons\Controller;

class Index extends Controller
{
    public function index()
    {
        return '首页';
    }
}
```

浏览器访问地址格式：<http://网址/addons/插件名称/控制器名/操作方法>

如上 `demo` 插件，那么首页控制器访问如下，没写控制器那么访问的是 `Index` 控制器，操作方法是 `index`

<http://网址/addons/demo/index/index>

首页默认是 `Index`，操作方法是 `index`，那么可以省略，如下即可直接访问到

<http://网址/addons/demo/>

基类控制器

创建的控制器继承 `think\addons\Controller` 基类，文件位置在：`vendor/hkcms/cms-addons/src/addons/Controller.php`

继承之后有以下属性、方法可以调用，你可根据自己的需要重写某些属性、方法

属性名	说明	使用说明
\$error_tmpl	错误页	默认是插件视图根目录下的error.html，参考app/common/tpl/error.html
\$success_tmpl	成功页	同上，只有使用error方法或success方法时有效
\$app	容器变量	有路由、配置、路由等实例对象，更多了解： https://www.kancloud.cn/manual/thinkphp6_0/1037489
\$request	请求对象	用于获取请求的信息，更多了解： https://www.kancloud.cn/manual/thinkphp6_0/1037518
\$name	插件标识名称	
\$addon_path	当前插件根路径	
\$cache	缓存	更多了解： https://www.kancloud.cn/manual/thinkphp6_0/1037634
\$site	站点配置数组	即后台站点配置，\$this-site['title'] 获取网站名称

控制器方法

1. `error` 方法，输出错误信息，如果是ajax请求则返回的是json数据。如果不是ajax确保视图目录下有error.html文件，你可通过\$error_tmpl来指定位置，只能是视图目录下。

```
// 在控制器里面使用
public function test()
{
    $this->error('出错了');
}
// $msg 提示信息
// $url = 跳转地址
// $data = 返回的数据
// $wait = 等待几秒后跳转，默认3秒
```

9. 插件控制器

```
// $header = 头部信息  
$this->error($msg, $url, $data,$wait, $header)
```

2. `success` 方法，输出正确提示信息，同上一样。
3. `result` 方法返回指定的数据类型，例如json、xml，默认是json,常用于接口。

```
public function test()  
{  
    $this->result('操作成功',['lists'=>[]],200, 'json');  
}
```

```
/**  
 * $msg 提示信息  
 * $data 要返回的数据  
 * $code 返回的code  
 * $type 返回数据格式  
 * $header 发送的Header信息  
 */  
$this->result($msg = '', $data = [], $code = 200, $type = 'json', array $header = [])
```

4. `fetch` 加载模板页面方法

如下，参数一填写模板位置，位置是基于插件视图为基准，另外如果为空默认以当前控制器/操作方法方式找到模板位置。参数二为模板变量参数

```
/**  
 * $template 模板文件名  
 * $vars 模板输出变量  
 */  
$this->fetch($template = '', $vars = [])
```

以demo插件为例，以下是Index控制器加载模板，参数没有填写那么加载的模板路径为：

addons/demo/view/index/test.html

```
public function test()  
{  
    return $this->fetch();  
}
```

加载其他模板，路径为：addons/demo/view/test/index.html

```
public function test()
```

9. 插件控制器

```
{  
    return $this->fetch('test/index');  
}
```

5. display 渲染模板内容

与fetch方法不同的是，fetch方法时获取传入的文件位置，display方法是直接传入字符串渲染。

```
/**  
 * @param string $content 模板内容  
 * @param array $vars 模板输出变量  
 */  
$this->display($content = '', $vars = [])
```

示例

```
public function test()  
{  
    $html = '<a>{$test}</a>';  
    return $this->display($html, ['test'=>111]);  
}
```

6. assign 模板变量赋值方法

```
/**  
 * @param mixed $name 要显示的模板变量  
 * @param mixed $value 变量的值  
 */  
$this->assign($name, $value = '')
```

示例：如下，这样定义好后，在模板页面写入{\$txt}即可。

```
public function test()  
{  
    $txt = '测试';  
    $this->assign('txt', $txt);  
    return $this->fetch();  
}
```

7. getInfo 获取插件信息

```
public function test()  
{  
    $ini = $this->getInfo();  
}
```

9. 插件控制器

```
    dump($ini);  
}
```

8. getConfig 获取插件配置

true = 获取完整配置，即返回config.php里面的完整数组，false 仅获取键值。

```
/**  
 * @param bool $type 是否获取完整配置  
 */  
$this->getConfig($type = false)
```

示例

```
public function test()  
{  
    $config = $this->getConfig();  
    dump($config);  
}
```

可参考内容：

[插件内置函数](#)

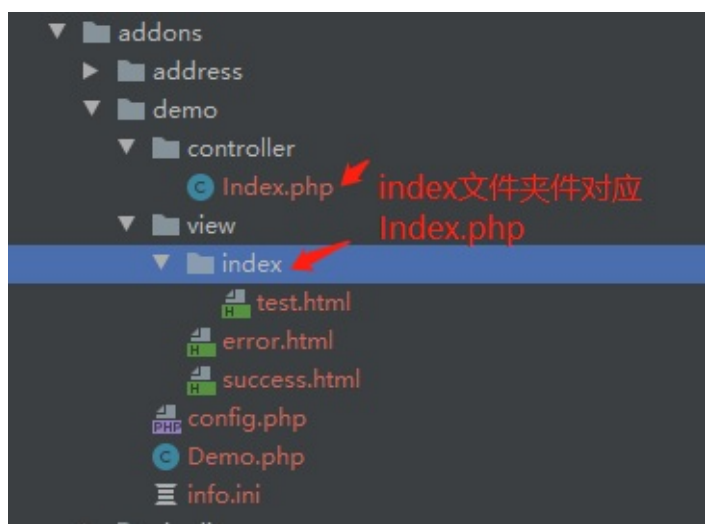
10. 插件视图

插件视图

在插件根目录新建 `view` 文件夹即可使用。里面的文件按照控制器名/操作方法的方式创建相应的html即可。

视图模板里的变量、函数的使用、模板规则等请参考：<https://www.kancloud.cn/manual/think-template/1286406>

示例图



10-1 插件主题化

插件主题化

当插件有自己的前台时，可把页面主题化。主题化之后可实现跟cms前台一样模板切换，也可从应用中心发布该插件的模板，用户通过应用中心安装针对该插件的模板。

另外插件前台控制器你需要参考cms前台来初始化你的路径、如果有标签库参考标签库章节。

目录结构

新建 `install` 目录，在目录下新建与插件名称一致的文件夹，如下图demo插件目录结构



`info.ini`文件参考模板开发章节

3. 模板描述文件(重要) `info.ini`

其中比较重要的是`module`属性，值应当为插件标识，例如demo插件那么值为demo

```
type = "template" # 表示模板 (必填)
module = 'demo' # 模板所属插件
name = 'default'
title='默认'
```

效果



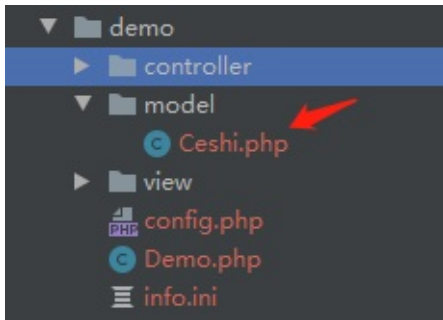
11. 插件模型

插件模型

插件的模型跟 thinkphp 的使用也是一样的，更多了解可参考：https://www.kancloud.cn/manual/thinkphp6_0/1037580

1. 创建插件模型

以demo插件为例，如下图创建Ceshi表模型



Ceshi模型代码，注意命名空间，创建好之后就可以根据手册增删改查操作。

```
<?php
// 命名空间
namespace addons\demo\model;

use think\Model;

class Ceshi extends Model
{
}
```


12. 模型事件(已废弃)

模型事件

如果我们想对后台文章模型或单页等模型新增、修改、删除时进行一些操作，那么我们可以在插件初始化事件中定义相应的模型事件。

已废弃

事件	描述
AfterRead	查询后
BeforeInsert	新增前
AfterInsert	新增后
BeforeUpdate	更新前
AfterUpdate	更新后
BeforeWrite	写入前
AfterWrite	写入后
BeforeDelete	删除前
AfterDelete	删除后
BeforeRestore	恢复前
AfterRestore	恢复后

示例

监听文章模型更新后

```
// 插件初始化事件
public function addonsInitHook()
{
    Event::listen('model.app\admin\model\cms\Archives.AfterUpdate', function ($model){
        // 获取没更新前的数据
        $origin = $model->getOrigin();
        // 获取更新后的数据
        $data = $model->getData();
    });
}
```

监听文章模型新增后事件

12. 模型事件(已废弃)

```
// model.模型类命名空间.事件  
Event::listen('model.app\admin\model\cms\Archives.AfterInsert', function ($model)  
{  
  
})
```

更多模型监听，可前往后台模型查看 [app/admin/model/](#)

13. 插件内置函数

插件内置函数

函数名称	说明
hook	触发事件，前往“5.插件事件”了解
addons_url	url地址生成
get_addons_info	获取插件信息(info.ini)
get_addons_info_all	获取所有本地插件信息(info.ini)，参数1传入template表示获取所有安装模板信息，addon表示获取所有本地插件信息
get_addons_config	获取插件配置

- addons_url url地址生成

```
/**
 * $url 在插件内只需要写：控制器/操作方法，在其他模块写：插件名称://控制器/操作方法
 * $param 参数，附加到URL，用于传参
 * @param bool|string $suffix 生成的URL后缀
 * @param bool|string $domain 域名
 * @return bool|string
 */
addons_url($url = '', $param = [], $suffix = true, $domain = false)
```

示例，在插件index控制器生成URL，跳转到test方法

```
<?php
namespace addons\demo\controller;
use think\addons\Controller;
class Index extends Controller
{
    public function index()
    {
        return '<a href="'.addons_url("index/test")."'>dsfsfsd</a>';
    }
    public function test()
    {
        $config = $this->getInfo();
        dump($config);
    }
}
```

13. 插件内置函数

其他模块生成URL需要写上插件名称，例如前台模板跳转到插件控制器test方法，如下：

```
<a href="{:addons_url('demo://index/test')}">sadasdas</a>
```

- `get_addons_info` 获取插件信息

```
/**
 * $name 插件名称
 * $type 插件类型, template/addon
 * $module 所属模块
 */
get_addons_info($name, $type='addon', $module='index')
```

示例，获取test插件信息

```
public function index()
{
    $info = get_addons_info('test');
    dump($info);
}
```

- `get_addons_config` 获取插件配置

```
/**
 * $type 类型, 可选值: template/addon
 * $name 插名称标识
 * $module 模块
 * $complete true-获取所有结构数组, false-获取配置值
 */
get_addons_config($type, $name, $module='', $complete=false)
```

示例，获取test配置信息

```
public function index()
{
    $arr = get_addons_config('addon', 'test');
    dump($arr);
}
```

前台函数

助手函数库

后台函数

[助手函数库](#)

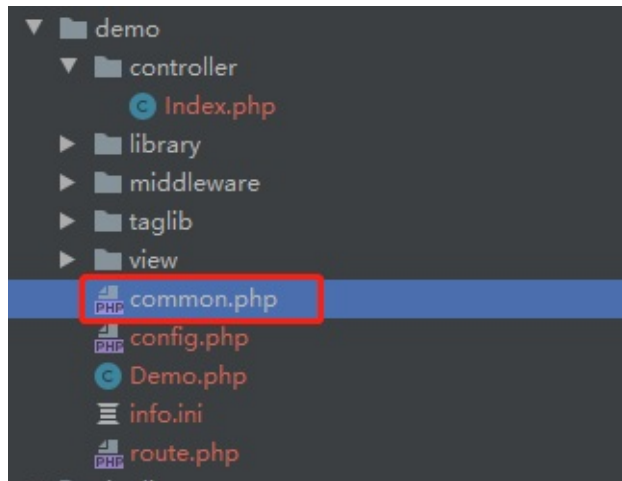
公共函数

[助手函数库](#)

13-1 插件函数库

插件函数库

插件函数库用于插件内部使用，在插件根目录创建 `common.php` 文件即可。



demo函数示例，之后在插件控制器等插件内部即可调用demo函数。

```
<?php

if (!function_exists('demo')) {
    function demo()
    {
        echo 1;
    }
}
```

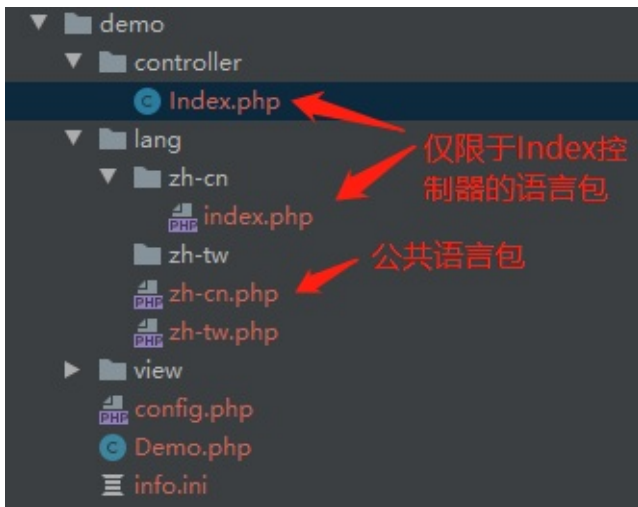
14. 插件多语言

多语言

插件多语言仅用于插件内的访问

创建语言包

在你的插件根目录下创建 `lang` 名字的文件夹，该文件夹里面放语言包即可，路径为：
addons/demo/lang/。如下图所示



其中里面还创建了 `zh-cn` 文件夹，该文件夹里的语言包不是全局的，作用范围仅限控制器，跟控制器名保持一致，小写命名。如上图 `index.php` 语言包仅作用于 `Index` 控制器

多语言的使用

使用 `lang` 函数即可调用

```
/**
 * $name 语言变量名
 * array $vars 动态变量值
 * $lang 指定语言
 */
lang(string $name, array $vars = [], string $lang = '')
```

zh-cn 语言包

```
<?php
return [
    'Demo' => '演示'
];
```

Index控制器使用

```
<?php
namespace addons\demo\controller;
use think\addons\Controller;
class Index extends Controller
{
    public function index()
    {
        echo lang('Demo'); // 当语言切换到zh-cn时会输出演示, 如果没有匹配到就输出Demo
    }
}
```

- 动态变量值

当我们需要里面某些值需要动态化时即可使用参数2传入, 如下

```
<?php
return [
    'Hello, %s!' => '您好, %s!'
];
```

```
public function index()
{
    // 中文结果, 您好, admin! 支持多个%s占位符。
    echo lang('Hello, %s!', ['admin']);
}
```


15. 插件中间件

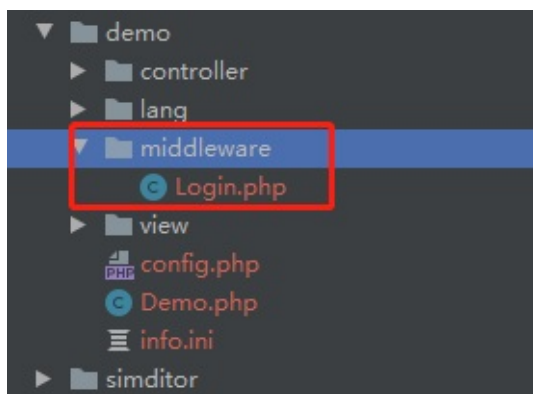
插件中间件

插件中间件用于访问插件前台、或前台、后台时对HTTP请求拦截或过滤。例如你可以使用中间件控制哪些方法需要登录才能访问。

注意：2.1.220220版本暂时不支持控制器中间件，访问插件前台可以使用的addonMiddleware事件，该事件监听所有访问插件前台的操作，详情前往插件事件了解。

创建中间件

中间件放置于：addons/demo/middleware/



示例代码：

```
<?php

declare (strict_types=1);

namespace addons\demo\middleware;

class Login
{
    // handle 为入口执行方法，必须。第一个参数是`Request`对象，第二个参数是一个闭包。
    public function handle($request, \Closure $next)
    {
        // 判断session里面有用户信息，没有说明没有登录，跳转到登录页
        if (!$request->session('User')) {
            return redirect((string)addons_url('login/index'));
        }
        return $next($request);
    }
}
```

前置中间件：写在请求具体操作前为前置中间件，即：`$next($request);`前，如上面所示

后置中间件：执行完请求后为后置中间件，即：`$next($request);` 后面

```
<?php

declare (strict_types=1);

namespace addons\demo\middleware;

class Login
{
    // handle 为入口执行方法，必须。第一个参数是`Request`对象，第二个参数是一个闭包。
    public function handle($request, \Closure $next)
    {
        $response = $next($request);
        // 需要的操作
        ...
        // 返回响应
        return $response;
    }
}
```

中间件传值到控制器方法

```
<?php

declare (strict_types=1);

namespace addons\demo\middleware;

class Login
{
    // handle 为入口执行方法，必须。第一个参数是`Request`对象，第二个参数是一个闭包。
    public function handle($request, \Closure $next)
    {
        $request->Demo = 1; // 控制器里面 request()->Demo 即可获取值
        return $next($request);
    }
}
```

使用中间件

执行顺序分别为：

插件全局中间件->控制器中间件

插件全局中间件

内置有插件全局中间件，可通过 `addonMiddleware` 事件来执行。

控制器中间件

控制器中间件在插件控制器里面注册，如下，定义后访问Index控制器的任意方法都会经过中间件。

```
<?php

namespace addons\demo\controller;

use think\addons\Controller;

class Index extends Controller
{
    // 控制器中间件
    public $middleware = [
        \addons\demo\middleware>Login::class
    ];

    public function index()
    {
        echo "登录成功";
    }

    public function login()
    {
        echo "登录页";
    }
}
```

使用except/only来设置生效的操作

only 示例

```
// 控制器中间件
public $middleware = [
    \addons\demo\middleware>Login::class=>['only'=>['index','test']] // Log
in中间仅对index、test方法拦截
];
```

except 示例

```
// 控制器中间件
public $middleware = [
    \addons\demo\middleware>Login::class=>['except'=>['test']] // 除了test
方法，其他的所有操作方法都会经过Login中间件
];
```


16. 插件路由

插件路由

默认情况下我们的插件前台访问地址是

<http://网址/addons/插件名称/控制器/操作方法>

例如访问demo插件的Index控制器，index操作方法，那么地址如下：

<http://网址/addons/demo/index/index>

自定义路由

我们可以通过自定义路由实现个性化插件访问，有自动自定义路由或手动自定义路由方式。

路由参考链接：https://www.kancloud.cn/manual/thinkphp6_0/1037495

注意：2.1.220220版本暂时只能手动写路由。

- 手动写路由

如下，定义好后浏览器访问：<http://xxxx.com/ceshi>，就能去到demo插件下的Index控制器下的index操作方法

```
/**
 * 插件初始化
 */
public function addonsInitHook()
{
    $execute = '\\think\\addons\\Route::execute';
    // url地址, 即浏览器输入的地址
    \\think\\facade\\Route::rule('/ceshi', $execute)
        ->append([
            'addon' => 'demo', // 插件名称
            'controller' => 'index', // 控制器
            'action' => 'index' // 操作方法
        ]);
}
```

自动导入路由文件

在插件根目录下新建 `route.php` 文件，路径为：`addons/插件名称/route.php`

示例如下：定义好后浏览器访问：<http://xxxx.com/ceshi>，就能去到demo插件下的Index控制器下的index操作方法。

```
<?php
```

```
return [  
    // url地址=>插件名称/控制器/操作方法  
    'ceshi'=>'demo/index/index'  
];
```

- 域名形式

下面绑定了两个域名，那么访问地址是：<http://demo.xxxx.com/ceshi>，以及<http://local.xxx.com/ceshi>

```
<?php  
return [  
    [  
        'domain' => 'demo',  
        'rule' => [  
            'ceshi'=>'demo/index/index'  
        ]  
    ],  
    [  
        'domain' => 'local',  
        'rule' => [  
            'ceshi'=>'demo/index/index'  
        ]  
    ]  
];
```

若要实现动态，例如给到用户来配置路由，那么你在这个文件里面可以查询数据库，按如上数组格式返回即可

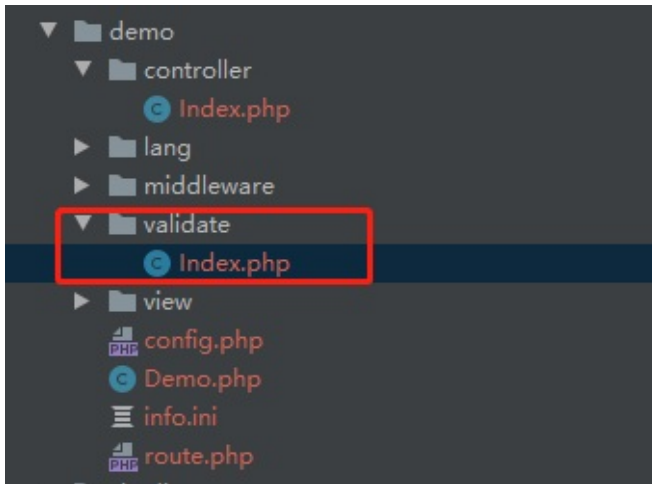
17. 插件验证器

插件验证器

验证器同TP的一致，更多的验证可参考 https://www.kancloud.cn/manual/thinkphp6_0/1037629

文件路径：例如demo插件

addons/demo/validate/Index.php



示例

```
<?php
declare (strict_types = 1);
namespace addons\demo\validate;
use think\Validate;
class Index extends Validate
{
    /**
     * 定义验证规则
     * 格式：'字段名' => ['规则1', '规则2'...]
     * @var array
     */
    protected $rule = [
        'username' => 'require|alphaDash|length:3,16|checkUser',
    ];

    /**
     * 定义错误信息
     * 格式：'字段名.规则名' => '错误信息'
     * @var array
     */
    protected $message = [
        'username.checkUser' => '用户名已存在',
    ];
}
```

17. 插件验证器

```
/**
 * 自定义方法
 * @param $value
 * @param $rule
 * @param array $data
 * @return bool|string
 */
protected function checkUser($value, $rule, $data=[])
{
    return true;
}
}
```

控制器使用

```
try {
    $data = [];
    validate(\addons\demo\validate\Index::class)->check($data);
} catch (ValidateException $e) {
    // 验证失败 输出错误信息
    dump($e->getError());
}
```


18. 标签库

标签库

标签库可以应用到插件前台使用，也可作为扩展前台HkCms标签库。

语法可参考 <https://www.kancloud.cn/manual/think-template/1286424>

也可参考前台标签库，位置：app/index/taglib/HkCms.php

手动导入

注意：2.1.220220版本及之前版本只支持手动导入。

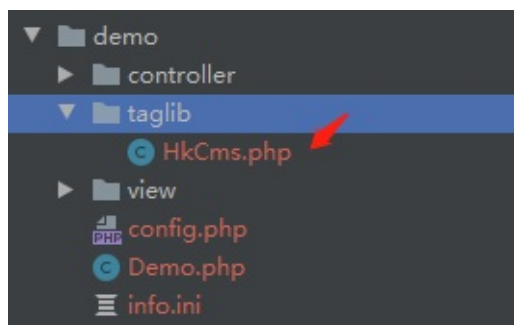
```
/**
 * 插件初始化时导入标签
 */
public function addonsInitHook()
{
    // 加载标签
    $view = $this->app->config->get('view');
    $view['taglib_pre_load'] = $view['taglib_pre_load'].'.addons\tags\library\HkCms';
    $this->app->config->set($view, 'view');
}
```

如果不预先加载标签，那么在模板就需要使用 `<导入标签 导入`

`<导入标签`

自动导入标签库

在插件根目录创建 `taglib\HkCms.php` 文件，如下demo插件演示，创建后网站前台即可调用标签库，注意只生效于前台。



HkCms.php 内容

```
<?php
```

18. 标签库

```
declare (strict_types=1);

namespace addons\demo\taglib;

use think\template\TagLib;

class HkCms extends TagLib
{
    /**
     * 定义标签列表
     */
    protected $tags = [
        'demo' => ['attr'=>'title', 'close'=>0], // 标签列表
    ];

    // tag+demo, 首字母大写, $tag为定义的属性
    public function tagDemo($tag)
    {
        return "demo:". $tag['title'];
    }
}
```

创建完后，在前台模板写上如下代码即可调用，可能需要清理缓存，建议开发插件时打开调试模式。

```
{hkcms:demo title="标题" /}
```

19. 第三方类库

引用第三方类库

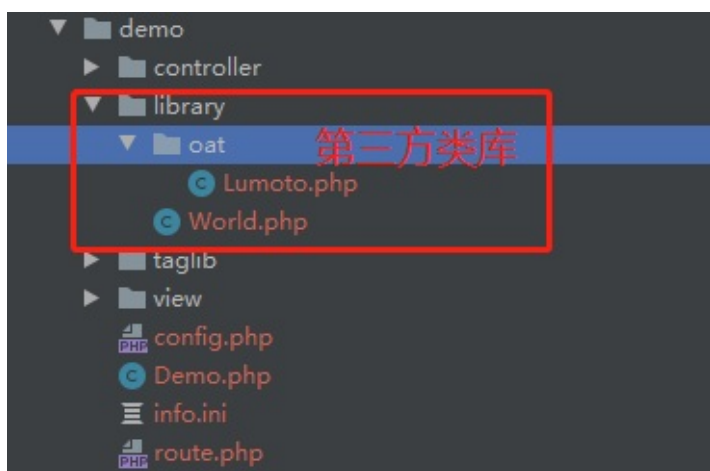
将你的第三方类库直接放到插件根目录或者新建文件夹放好。

例如 `demo` 插件下新建 `libaray` 文件夹，之后把类库放入 `library` 文件夹。

修改类库的命名空间

通过上面，类库放进来并不能调用到，我们需要增加或修改命名空间。

如下图，放入了两个类库文件，我们只需把命名空间改成：`namespace addons\插件名称\library;`，`library`里面还有文件夹就继续按目录层次写下去即可。



根据插件命名空间调整，如下所示

```
<?php
// addons\插件名称\目录
namespace addons\demo\library;

class World
{
    public function show()
    {
        echo 1;
    }
}
```

```
<?php
namespace addons\demo\library\oat;

class Lumoto
{
}
```

改好后调用

```
$w = new \addons\demo\library\World();  
$w->show();
```

通过初始化事件自动注册

如果类库文件过多或者是一个composer包，修改起来麻烦，可参考如下代码，在插件名称、插件初始化时执行

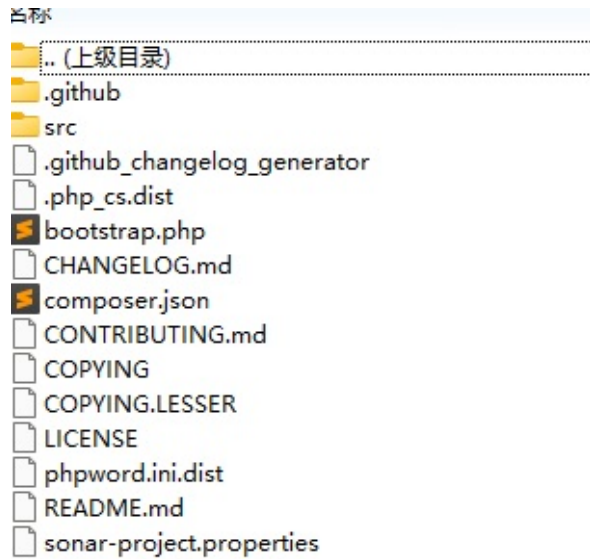
2.2.1.20220603版本起支持

如下：在 `addonsInitHook` 事件中调用了 `addNamespace` 自动注册第三方类库方法

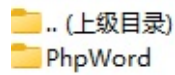
```
<?php  
  
declare (strict_types=1);  
  
namespace addons\demo;  
  
use think\Addons;  
  
class Demo extends Addons  
{  
    public function install()  
    {  
        return true;  
    }  
  
    public function uninstall()  
    {  
        return true;  
    }  
  
    public function addonsInitHook()  
    {  
        $this->addNamespace('命名空间');  
    }  
}
```

例如：以PhpWord类库为例，我们下载后的压缩包目录结构如下：

19. 第三方类库



去到src目录，有如下文件夹



把PhpWord文件夹放入到addons\demo\library\PhpOffice目录下，之所以还要创建PhpOffice目录因为他的命名空间是

```
PhpOffice\PhpWord
```

所以我们需要在创建 `PhpOffice` 文件夹，完成这一步后在 `addonsInitHook` 事件中添加命名空间

```
$this->addNamespace('PhpOffice\PhpWord');
```

在插件里面或者其他地方使用类似于PhpOffice\PhpWord空间的类库时都会在这加载。

composer 方式安装第三方类库

2.2.1.20220603版本起支持

使用composer方式之前你需要先了解什么是composer，链接地址：<https://pkg.xyz/#how-to-install-composer>

1. 分别建立vendor文件夹与composer.json文件

电脑 > 数据 (E:) > www > hkcms > addons > demo >

名称	修改日期	类型
vendor	2022/6/16 23:56	文件夹
composer.json	2022/6/16 23:56	JSON 文件
Demo.php	2022/6/16 20:45	PHP 文件
info.ini	2022/6/16 17:12	INI 文件

2. composer.json内容

```
{
  "name": "hkcms/demo",
  "description": "demo plugin",
  "type": "hkcms-addon",
  "license": "apache2.0",
  "authors": [
    {
      "name": "demo",
      "email": "demo@hkcms.cn"
    }
  ],
  "config": {
  },
  "require": {
  }
}
```

`name` composer包的名字，格式为“vendor名称/包名”

`description` 描述信息，简单描述下你的插件

`type` 固定为hkcms-addon

`license` 固定 apache2.0

`authors` 作者，按上面的格式改掉即可

..

3. 跟composer一样在当前插件目录安装、更新、删除第三方库包。

例如安装第三方微信SDK

```
composer require overtrue/wechat:~4.0 -vvv
```

安装成功后即可使用。

注意，插件必须是安装后才能生效

20. 数据库文件导入

安装数据库

在开发完成后，导出需要用到的数据表，保存插件所在文件夹根目录，命名为 `install.sql`，在安装的时候导入数据库表。

你需要把表的前缀替换成 `@prefix@`，如下所示

```
CREATE TABLE IF NOT EXISTS `@prefix@tablename` (  
    .....  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 手动导入方式

```
// 文件绝对路径  
$sql = $this->addon_path.'data'.DIRECTORY_SEPARATOR.'install.sql';  
  
// 写入数据库  
create_sql($sql);
```

升级数据库

插件所在文件夹根目录，命名为 `upgrade.sql`，在更新的时候会执行该文件。

同样你需要把表的前缀替换成 `@prefix@`

演示数据数据库文件

插件根目录命名为：`demodata.sql`，在安装的时候用户勾选演示数据，即可导入演示数据。

数据库里面的表前缀使用 `@prefix@` 代替，如下

```
insert into `@prefix@archives` values (75,20,1,2,'测试','','','','','show.htm  
l',1637382374,1637382417,1637382417,NULL,0,'default','','0','zh-cn','normal');
```

效果

默认前台模板2

数据库表检测

数据库表	检测结果
hkcms_agc	✔ 通过
hkcms_archives	⚠ 已存在

目录文件权限

目录文件	检测结果
D:/hkcms/HkCms-develop/public/static/	✔ 通过
D:/hkcms/HkCms-develop/public/admin.php	✔ 通过
D:\hkcms\HkCms-develop\public\static\module\index\	✔ 通过

演示数据

导入演示数据

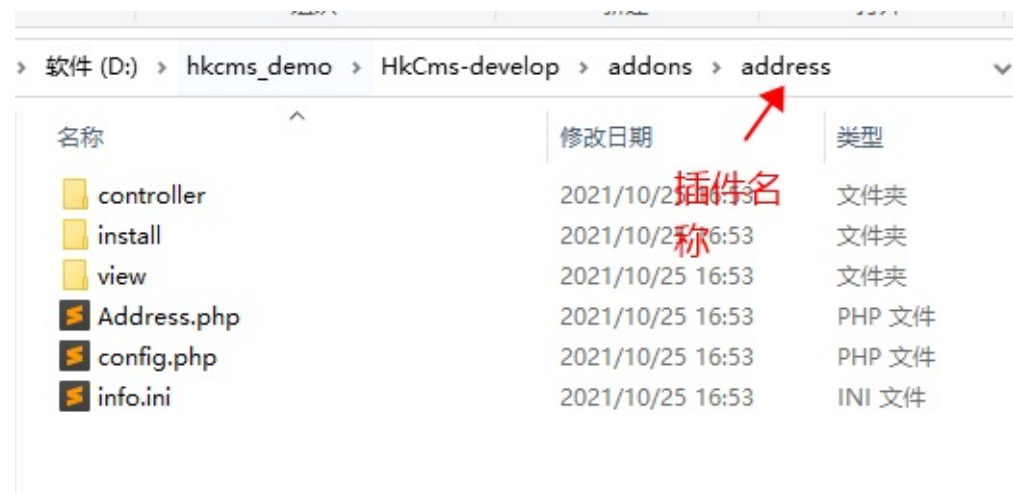
确认安装 刷新 取消

插件打包

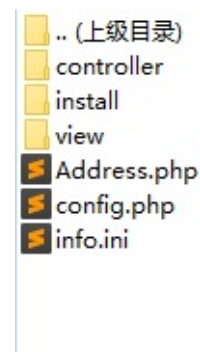
手动打包

在自己的插件目录，例如插件名字叫address，进入address目录，压缩目录，压缩成ZIP格式

目录示例



压缩包内容



自动打包

序号	名称	介绍	作者	价格	版本	状态	操作
1	simditor编辑器	simditor编辑器	HkCms	免费	1.0.1	<input checked="" type="checkbox"/>	🗑️ 卸载 📦 打包
2	地图位置选取插件	支持百度、高德地图位置选取。	HkCms	免费	1.0.0	<input checked="" type="checkbox"/>	⚙️ 配置 🗑️ 卸载 📦 打包

后台前端使用帮助

访问链接：<http://htdemo.hkcms.cn/>

二次开发帮助(更新中)

#二次开发

本目录下主要讲解HkCms系统架构、路由、控制器、视图、模型、事件、前端组件的使用，持续更新中...

后台前端开发帮助

访问链接：<http://htdemo.hkcms.cn/>

路由(链接地址、访问规则)

[URL访问](#)

[路由定义](#)

[URL生成](#)

URL访问

URL访问

没有定义路由的情况下，那么url访问方式如下，配置了路由就按路由的规则来访问。

```
http://serverName/index.php（或者其它入口文件）/控制器/操作?参数1=值
```

隐藏入口文件

注意：您需要配置URL重写规则方能正常访问，部分 `apache` 服务无需配置，如不能正常访问前往[伪静态（url重写）](#) 章节中配置，前台的 `index.php` 在后台 `站点配置->高级配置` 中可以选择隐藏 `index.php`。

1. 使用域名绑定的方式，在 `config/app.php` 文件下，可配置域名绑定应用，配置之后那么效果：

```
http://api.xxxx/控制器/操作?参数1=值
```

```
// 域名绑定（自动多应用模式有效）
'domain_bind' => [
    'api' => 'api',
    '*' => 'index',
],
```

路由定义

路由定义

详细路由配置参考https://www.kancloud.cn/manual/thinkphp6_0/1037495

应用路由

路由文件在每个应用目录下，例如 `index` 路由在 `index/route/route.php` 。

URL生成

URL 生成

用于生成URL链接地址，系统提供的助手函数为 `url`。

函数说明

```
/**
 * Url生成
 * @param string      $url      路由地址
 * @param array       $vars     数组变量
 * @param bool|string $suffix   生成的URL后缀
 * , 例如html, 默认是html
 * @param bool|string $domain   域名
 * @return UrlBuild
 */
function url(string $url = '', array $vars = [], $suffix = true, $domain = false
): UrlBuild
```

模板使用方式

```
{:url('/Index/demo')}
```

入口文件说明

url 生成跟入口文件配置挂钩，这里为了兼容不支持伪静态的默认是显示入口文件的，如果你配置了伪静态是可以选择隐藏的，这样生成的url地址不会带入口文件。

基础配置
邮件配置
附件配置
多语言
高级配置
分组配置
+

应用中心账号:

应用中心密码:

后台模板:

前台模板:

URL 模式: 伪静态 动态

前台入口文件: 显示 隐藏

URL 伪静态:

不使用路由方式生成

1. 应用内访问

带有参数形式生成，参数1的格式为：`url('控制器/操作方法')`

```
// url生成，默认是带有html后缀的，url方法返回的是对象可方便额外的操作，使用(string)转字符串
$url = (string) url('/index/demo', ['aaa'=>1]);
dd($url); // 输出结果
```

前台入口文件隐藏后效果，如果未配置隐藏入口文件，则前面会有 `/index.php/`，以下示例默认是配置了入口文件隐藏。

```
/index/demo.html?aaa=1
```

生成带有域名的链接地址，第四个参数为 `true` 表示生成的链接地址加域名

```
// 自动生成域名、加后缀
$url = (string) url('/demo/index', ['aaa'=>1], 'html', true);
// 结果
http://xxx.cms.com/demo/index.html?aaa=1
```

生成锚点示例，生成锚点注意前面不需要加 `/`，如下

```
// 锚点跟子域名同时存在时需要锚点在前, 域名在后
$url = (string)url('index/demo#ddd', ['aaa'=>1, 'b'=>'sfsfs'], '', true);

// 结果
http://xxx.cms.com:8080/index/demo?aaa=1&b=sfsfs#ddd
```

2. 访问其他应用, 域名绑定

域名绑定配置在 `config/app.php` 中, 目前有以下配置, 这里配置好后记得在域名提供商做好解析。

```
// 域名绑定
'domain_bind' => [
    'api' => 'api',
    '*' => 'index',
],
```

示例: 生成锚点并绑定到 `api` 域名, 通过生成后的链接即可访问到 `api` 应用。提示: 如果入口文件未配置隐藏, 那么入口文件还是存在的。

```
// 生成锚点, 并绑定域名
$url = (string)url('index/demo#ddd@api', ['aaa'=>1, 'b'=>'sfsfs']);

// 隐藏入口文件结果
http://api.cms.com/index/demo.html?aaa=1&b=sfsfs#ddd
// 未隐藏结果
http://api.cms.com/index.php/index/demo.html?aaa=1&b=sfsfs#ddd
```

示例2: 也可使用url方法的参数4来实现, 如下

```
// 指定子域名为api
$url = (string)url('/index/demo', ['aaa'=>1], 'html', 'api');
// 也支持完整域名, 如果不是80端口需要域名+端口
$url = (string)url('/index/demo', ['aaa'=>1], 'html', 'local.cms.com:8080');
```

3. 访问其他应用, 入口文件

如果不通过域名解析按入口文件来访问, 需要生成其他应用链接时可以通过以下写法实现

```
$url = (string)url('api:/index/demo', ['aaa'=>1, 'b'=>'sfsfs'], '', true);
```

生成结果

```
http://local.cms.com/api.php/index/demo?aaa=1&b=sfsfs
```

使用路由方式生成

了解这里前先了解路由定义，假设 `index` 应用配置了以下路由

```
Route::rule('/demo', '/index/demo');
```

根据路由生成地址写法如下，参数1的地址需要跟上面的参数2保持一致。

```
$url = (string)url('/index/demo', ['aaa'=>1]);
```

结果

```
/demo.html?aaa=1
```

使用域名

```
$url = (string)url('/index/demo', ['aaa'=>1], '', true);
```

会员管理

前台限制页面登录或权限控制

前台限制页面登录或权限控制

说明

实现登录控制与权限控制分别是下面两个中间件负责：

Login 负责登录

```
\app\index\middleware>Login::class
```

Auth 负责权限控制

```
\app\index\middleware\Auth::class
```

登录限制

在需要限制的控制器增加 `$middleware` 属性，例如下面的 `Demo` 控制器

```
<?php
namespace app\index\controller;
class Demo extends BaseController
{
    protected $middleware = [ // 增加权限控制属性，整个控制器将需要登录
        \app\index\middleware>Login::class
    ];
}
```

只限制特定方法需要登录，其他方法不需要登录可以使用下面两个参数

1. `only` 限制部分方法生效，如下示例（限制`index,show`方法需要登录）

```
<?php
namespace app\index\controller;

class Demo extends BaseController
{
    protected $middleware = [ // 限制index, show方法需要登录
        \app\index\middleware>Login::class=> ['only'=>['index', 'show']]
    ];

    public function index()
    {
        echo 1;
    }
}
```

```
}

public function show()
{
    echo 2;
}

public function lists()
{
    echo 2;
}
}
```

2. except 排除部分方法，其他方法生效，示例

```
<?php
namespace app\index\controller;
class Demo extends BaseController
{
    protected $middleware = [ // index方法不需要登录，其他的方法需要登录
        \app\index\middleware>Login::class=> ['except'=>['index']]
    ];

    public function index()
    {
        echo 1;
    }

    public function show()
    {
        echo 2;
    }

    public function lists()
    {
        echo 2;
    }
}
```

权限控制

`$middleware` 属性里面写入如下格式

```
<?php

namespace app\index\controller;
```

```
class Demo extends BaseController
{
    protected $middleware = [
        \app\index\middleware>Login::class,
        \app\index\middleware\Auth::class // 限制权限
    ];

    public function index()
    {
        echo 1;
    }

    public function show()
    {
        echo 2;
    }
}
```

权限控制也支持only、except来限制部分方法生效，使用方式同登录限制一致。

附录

[数据字典](#)

[助手函数库](#)

数据字典

数据字典

注意：数据库表名前缀是在安装时填写的前缀。以实际表名为准。

- hkcms_category (栏目表)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	栏目ID	int	10	√	√	×	NULL
2	mode_l_id	模型ID	int	10	×	×	×	0
3	parent_id	栏目父ID	int	10	×	×	×	0
4	type	类型 link-外部链接 list-列表 category-栏目分类	enum		×	×	×	link
5	app	所属应用(模块)	varchar	15	×	×	×	cms
6	name	栏目名称	varchar	50	×	×	×	
7	title	栏目标题	varchar	50	×	×	×	
8	url	指定url	varchar	250	×	×	×	''
9	image	栏目图片	varchar	250	×	×	×	
10	category_tpl	栏目分类模板	varchar	100	×	×	×	

11	list_tpl	列表模板	varchar	100	×	×	×	
12	show_tpl	内容模板	varchar	100	×	×	×	
13	seo_title	SEO标题	varchar	250	×	×	×	
14	seo_keywords	SEO关键字	varchar	250	×	×	×	
15	seo_desc	SEO描述	varchar	250	×	×	×	
16	ismenu	导航显示:1-显示,0-隐藏	tinyint	4	×	×	×	1
17	target	弹出方式	varchar	20	×	×	×	_self
18	weigh	排序	tinyint	11	×	×	×	0
19	delete_time	删除时间	int	10	×	×	√	NULL
20	update_time	更新时间	int	10	×	×	√	NULL
21	create_time	创建时间	int	10	×	×	√	NULL
22	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal

- hkcms_config (站点配置表)

序号	字段	说明	数据	长度	自增	主键	允许	默认
----	----	----	----	----	----	----	----	----

序号	字段	说明	类型	长度	自增	主键	空	值
1	id	ID	int	10	√	√	×	NULL
2	group	分组： basic s(基础) mail (邮箱) exten d(扩展)	varc h a r	30	×	×	×	
3	name	键名	varc h a r	50	×	×	×	
4	title	名称	varc h a r	100	×	×	×	
5	value	值	text		×	×	√	NULL
6	type	类型	varc h a r	30	×	×	×	
7	data_l ist	数据 列表	text		×	×	√	NULL
8	tip	提示	varc h a r	100	×	×	×	
9	style	样式	text		×	×	√	NULL

- hkcms_admin (后台管理员)

序号	字段	说明	数据 类型	长度	自增	主键	允许 空	默认 值
1	id	ID	int	10	√	√	×	NULL
2	usern ame	登录 名称	varc h a r	20	×	×	×	NULL
3	nickn ame	昵称	varc h a r	50	×	×	×	
4	email	邮箱	varc h a r	50	×	×	×	
5	pass word	密码	varc h a r	32	×	×	×	NULL

6	salt	密码盐	varchar	10	×	×	×	NULL
7	avatar	头像	varchar	255	×	×	×	
8	remark	备注	varchar	255	×	×	×	
9	logintime	登录时间	int	11	×	×	√	NULL
10	loginip	登录IP	varchar	40	×	×	×	
11	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal
12	create_time	创建时间	int	11	×	×	√	NULL
13	update_time	更新时间	int	11	×	×	√	NULL

- hkcms_admin_log (后台操作日志)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	ID	int	10	√	√	×	NULL
2	admin_id	用户id	int	10	×	×	×	NULL
3	username	用户名	varchar	30	×	×	×	
4	useragent	useragent	varchar	255	×	×	×	
5	title	标题	varchar	100	×	×	×	NULL
6	url	URL	varchar	1500	×	×	×	NULL
7	ip	IP地	varchar	20	×	×	×	NULL

7	ip	址	ar	20	×	×	×	NULL
8	content	内容	text		×	×	×	NULL
9	create_time	操作时间	int	11	×	×	×	NULL

- hkcms_admin_panel (管理后台常用菜单)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	auth_rule_id	菜单ID	int	11	×	×	×	0
2	admin_id	用户ID	int	11	×	×	×	0

- hkcms_attachment (附件)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	ID	int	10	√	√	×	NULL
2	user_id	用户ID	int	11	×	×	×	NULL
3	user_type	用户类型:1-后台管理员	tinyint	1	×	×	×	1
4	title	附件名	char	50	×	×	×	
5	path	附件路径	char	100	×	×	×	
6	size	附件大小(字节)	int	11	×	×	×	0
7	ext	附件扩展	char	10	×	×	×	
	mime	附件						

8	_type	mime Type	char	100	×	×	×	
9	md5	md5	char	32	×	×	×	
9	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal
9	update_time	更新时间	int	11	×	×	√	
9	create_time	创建时间	int	11	×	×	√	

- hkcms_auth_group (角色管理)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	ID	mediumint	8	√	√	×	NULL
2	name	名称	varchar	20	×	×	×	
3	parent_id	父级	mediumint	8	×	×	×	0
4	rules	规则	text		×	×	×	NULL
5	remark	备注	varchar	255	×	×	×	
6	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal
7	create_time	创建时间	int	11	×	×	√	NULL

8	e_time	时间	int	11	×	×	√	NULL
---	--------	----	-----	----	---	---	---	------

- hkcms_auth_group_access (权限分组表)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	admin_id	管理员ID	int	10	×	×	×	NULL
2	group_id	角色组ID	mediumint	8	×	×	×	NULL
3	create_time	创建时间	int	8	×	×	√	NULL
4	update_time	更新时间	int		×	×	√	NULL

- hkcms_auth_rule (菜单规则)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	ID	int	10	√	√	×	NULL
2	parent_id	父级	int	10	×	×	×	0
3	name	规则	char	80	×	×	×	
4	title	标题	char	20	×	×	×	
5	route	路由	char	80	×	×	×	NULL
6	icon	图标	char	50	×	×	×	
7	remark	备注	char	250	×	×	×	
8	weight	排序	int	11	×	×	×	0
9	type	类型:0-权限规则,1-菜单,2-	tinyint	2	×	×	×	1

		单,2-菜单头						
10	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal
11	is_nav	快速导航:1-是,0-否	tinyint	2	×	×	×	0
12	condition	附加条件	char	100	×	×	×	
13	create_time	创建时间	int	11	×	×	√	NULL
14	update_time	更新时间	int	11	×	×	√	NULL

- hkcms_category_priv (栏目权限)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	category_id	栏目ID	int	10	×	×	×	NULL
2	auth_group_id	管理员ID	mediumint	8	×	×	×	NULL
3	action	操作方法	char	20	×	×	×	

- hkcms_model (模型表)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值

1	id	ID	int	10	√	√	×	NULL
2	name	模型名称	char	50	×	×	×	NULL
3	table name	表名称	char	50	×	×	×	NULL
4	remark	备注说明	char	255	×	×	×	
5	controller	控制器	char	20	×	×	×	Archives
6	type	类型:sing-独立表,more-主表加副表	enum	'single','more'	×	×	×	more
7	category	栏目模板	varchar	100	×	×	×	
8	list_tpl	列表模板	varchar	100	×	×	×	
9	show_tpl	内容模板	varchar	100	×	×	×	
10	create_time	创建时间	int	11	×	×	√	NULL
11	update_time	更新时间	int	11	×	×	√	NULL
12	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal

- hkcms_model_field (模型字段表)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
----	----	----	------	----	----	----	-----	-----

2	mode_id	模型ID	int	10	×	×	×	NULL
3	field_name	字段名称	varchar	20	×	×	×	NULL
4	field_title	字段中文名	varchar	50	×	×	×	
5	form_type	字段类型	varchar	20	×	×	×	NULL
6	field_group	字段分组	varchar	50	×	×	×	常规
7	length	长度	int	11	×	×	×	250
8	default_value	默认值	text		×	×	√	NULL
9	data_list	选项列表	text		×	×	√	NULL
10	max_number	最大数量	int	11	×	×	×	0
11	decimals	小数位	tinyint	4	×	×	×	0
12	rules	规则	varchar	255	×	×	×	
13	tips	提示信息	varchar	255	×	×	×	
14	error_tips	错误提示	varchar	255	×	×	×	
15	extend	附加属性	varchar	500	×	×	×	
16	iscore	主表字段:1-是,0-不是	tinyint	1	×	×	×	0
	defau	默认字						

17	default_field	字段:1-默认,0-不是	tinyint	1	×	×	×	0
18	weigh	排序	int	11	×	×	×	0
19	status	状态:normal-正常,hidden-禁用	enum	'normal','hidden'	×	×	×	normal
20	create_time	创建时间	int	11	×	×	√	NULL
21	update_time	更新时间	int	11	×	×	√	NULL

- hkcms_model_field_bind (模型字段绑定栏目表)

序号	字段	说明	数据类型	长度	自增	主键	允许空	默认值
1	id	ID	int	10	√	√	×	NULL
2	category_id	栏目ID	int	11	×	×	×	NULL
3	model_field_id	字段ID	int	11	×	×	×	NULL
4	create_time	创建时间	int	11	×	×	√	NULL
5	update_time	更新时间	int	11	×	×	√	NULL

助手函数库

公共函数库

文件路径：/app/common.php

函数名称	说明
get_doc_tota	获取栏目文档数量
to_guid_string	根据PHP各种类型变量生成唯一标识字符
get_random_str	获取随机数
load_js	加载插件JS
is_really_writable	判断文件是否可写
index_url	前台URL地址生成
build_tp_rules	nice validator前端验证规则生成TP验证规则
get_template_path	获取当前模板主题根路径
get_category	获取栏目信息
get_category_sub	获取栏目所有下级ID
get_category_parent	获取栏目所有上级ID
get_category_top	获取栏目顶级栏目信息
controller	获取模型管理操作的控制器
create_menu	创建菜单，见“8.创建菜单”说明
status_menu	更改菜单状态
del_menu	删除所属插件的菜单
create_sql	执行SQL文件
cdn_url	获取完整上传文件的资源访问路径
site	获取站点配置里面的数据
clear_cache	缓存日志清理
get_curlang	获取当前内容语言

admin 后台

文件路径：/app/admin/common.php

函数名称	说明
has_rule	权限规则判断

lang_content_add	lang_content_add
lang_content_del	内容多语言关联删除
lang_content_get	获取关联的内容语言ID

index 前台

暂无

API接口

说明

目前已实现基础的API，持续完善中...

接口地址 <https://apifox.com/apidoc/shared-6ccd9345-3dd1-499e-94a5-ec6e0c4f2c20>