RGCMS V2使用手册

龙哥



目　录

- 系统概述
- 开始安装
- 管理后台（必懂部分）
  - 配置设置
  - 模型管理
  - 字段管理
  - 系统升级
  - 关键词库
  - TKD模板
  - 链接&Robots
  - 站点列表
  - 站点信息&全局变量
  - 栏目管理
  - 单页管理
  - 栏目内容
  - 功能管理
  - 表单管理
  - 后台总结（必看）
- 标签变量
  - 页面变量
    - 全局变量
    - 当前栏目
    - 单页内容
    - 列表页
    - 详情页
  - 标签调用
    - 站点列表
    - 栏目调用(读数据库)
    - 栏目调用(读缓存)
    - 内容列表
    - 指定模型内容
    - 自定义分页
    - 面包屑
    - 功能调用
    - 附属表内容
    - 单页内容列表
    - 多图字段
    - 表单内容列表
    - 字符串分割

搜索页列表
地区调用
关键词列表
属性列表
内容统计
比较判断
常用比较
常用判断
模板开发
模板目录结构
路径标签
模板包含
模板命名
语言包
搜索表单
自定义表单
系统函数
默认值
无图默认值
生成缩略图
获取多图的某张
时间格式化
栏目字段继承
字符截取
字符隐藏
分割获取
二维码
转纯文本
运算
避免混淆
其他常用
更新日志
问题排解

系统概述

睿谷(RGCMS)内容管理系统 V2.0

http://www.rgxx.com

:-:



该版本是在V1.0的基础上重构扩展而成，继承了V1.0版本的优点，并根据在实际开发制作中遇到的各种情况，进行了流程的优化、标签的优化与加强、后台功能的提升。

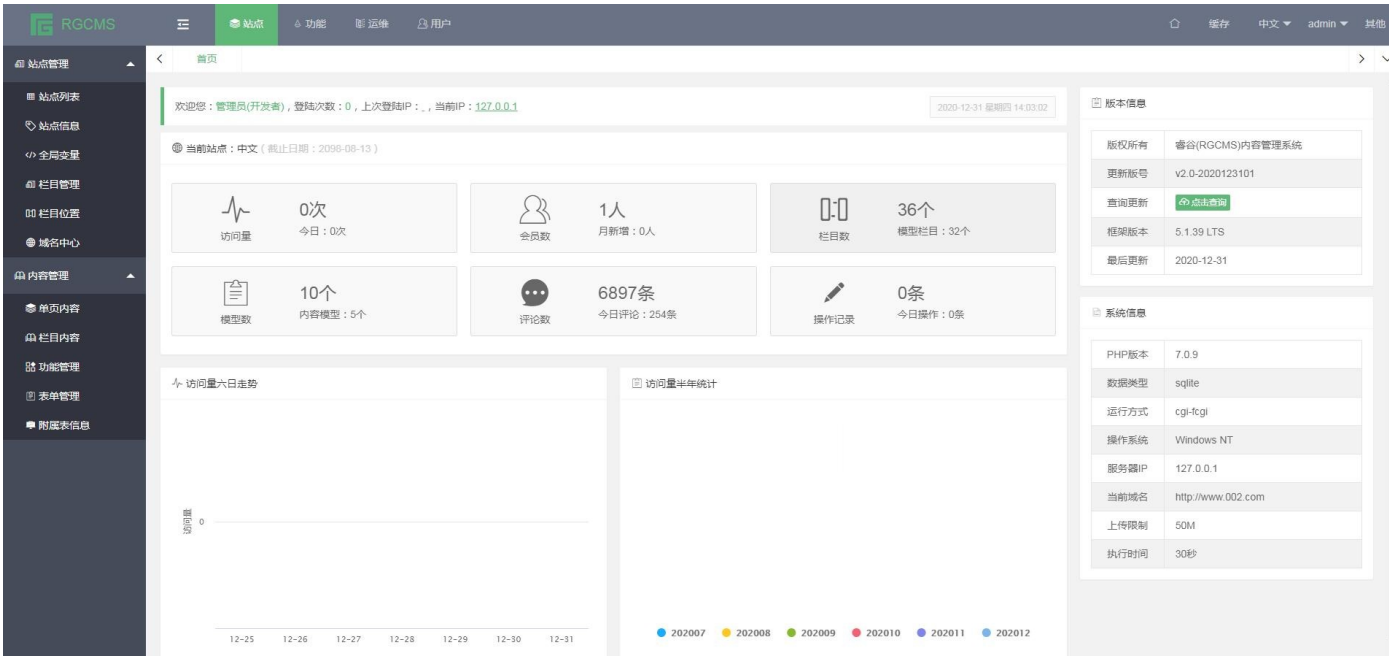
RGCMS2.0运行环境为PHP+Mysql/Sqlite
两种数据库版本自由选择且操作一模一样，无任何区别。
建议：大数据量站点推荐选择Mysql数据库

系统运行要求：

- PHP5.6+
- Mysql5.6+/Sqlite3
- PDO PHP Extension
- MBstring PHP Extension

RGCMS

开发框架：Thinkphp5.1+，Layui2.5.6



[warning]RGCMS2.0旨在让开发者在创作过程中获得更多的开发乐趣与开发成就。系统开源并遵循alpha2.0协议，您在使用过程中如有建议或合作请与15108148@qq.com联系，或进群进行交流。Q群：180506934

开始安装

> RGCMS2.0支持Mysql数据库及Sqlite，因此系统安装时会有选择数据库类型选项。
>
> 安装后，默认管理地址admin.php 默认用户名admin 密码123456
>
> 请及时修改，默认管理地址、用户名和密码

安装界面



数据库选择


数据库类型
首页
数据库类型
选择数据库类型
选择数据库类型
Mysql
Sqlite
系统信息
PDO
扩展：pdo_sqlite
已开启
扩展：curl
已开启
扩展：mbstring
已开启
函数：file_get_contents
已开启
权限：/runtime文件夹
可写
权限：/upload文件夹
可写
权限：/data文件夹
可写
权限：/public文件夹
可写
权限：/config/database.php文件
可写
权限：/app/common/model文件夹
可写
加入QQ群

Mysql数据库



Sqlite数据库(本地数据库)

> 安装后数据库位于/data/文件夹下

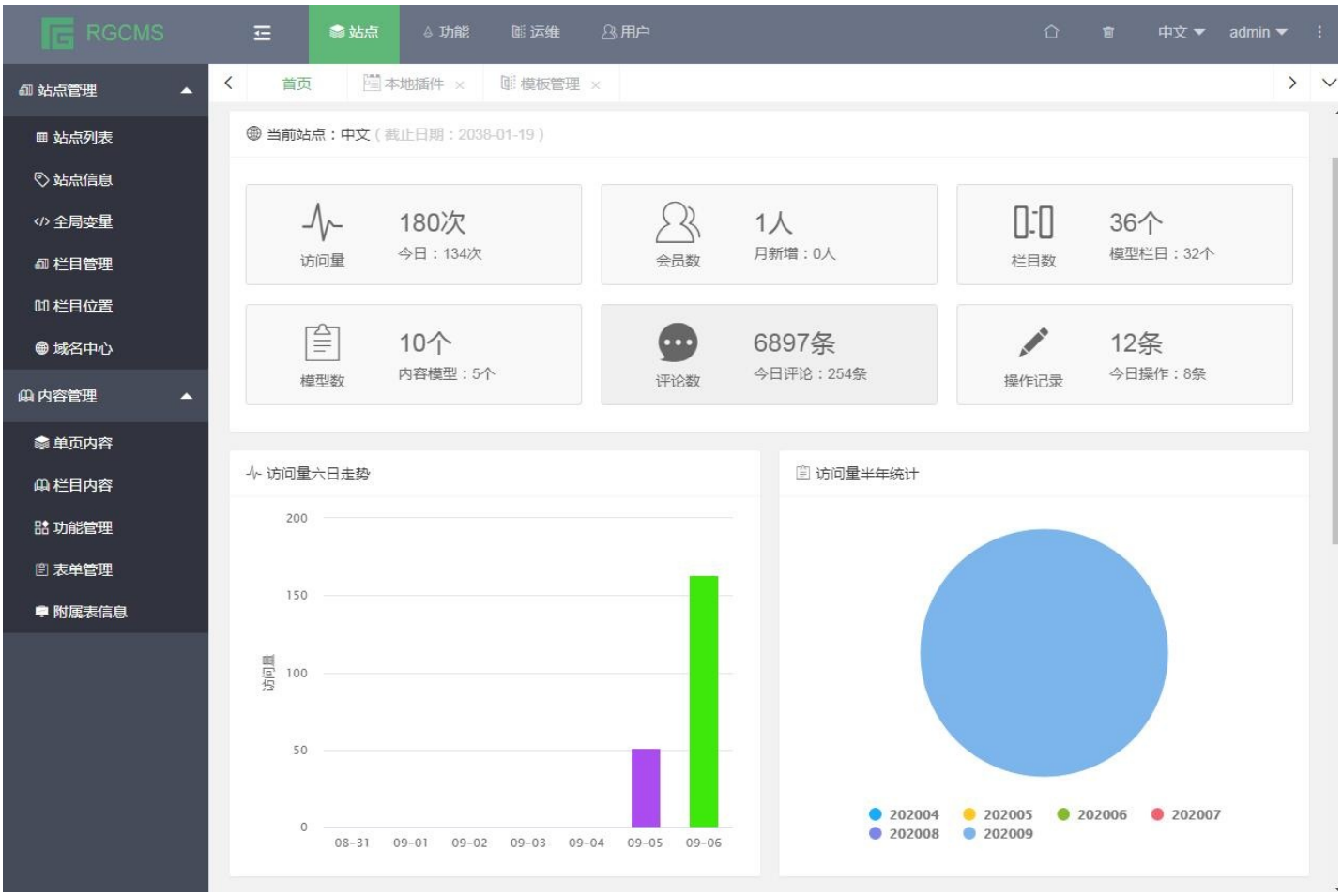

数据库类型
首页
数据库类型
Sqlite
数据库信息
数据库名称
wo25LgSi8jhz3l9
.db
默认信息
默认管理登陆/admin.php（可自行修改），默认用户名admin密码123456，请登录后台自行修改
确认安装
加入QQ群

管理后台（必懂部分）

> 麻雀虽小五脏俱全，一册在手，开发无忧

使用RGCMS2.0 版本，必须了解本章节内容，本章节涵盖了RGCMS系统的精髓，以及开发制作流程。熟悉后台操作，才能在开发制作过程中得心应手。

> 必懂部分，您必须要懂得部分。凡是本章节未提及的后台功能，基本上都是你能看的懂的。



配置设置

> 菜单位置：运维 -- 系统管理 -- 配置设置

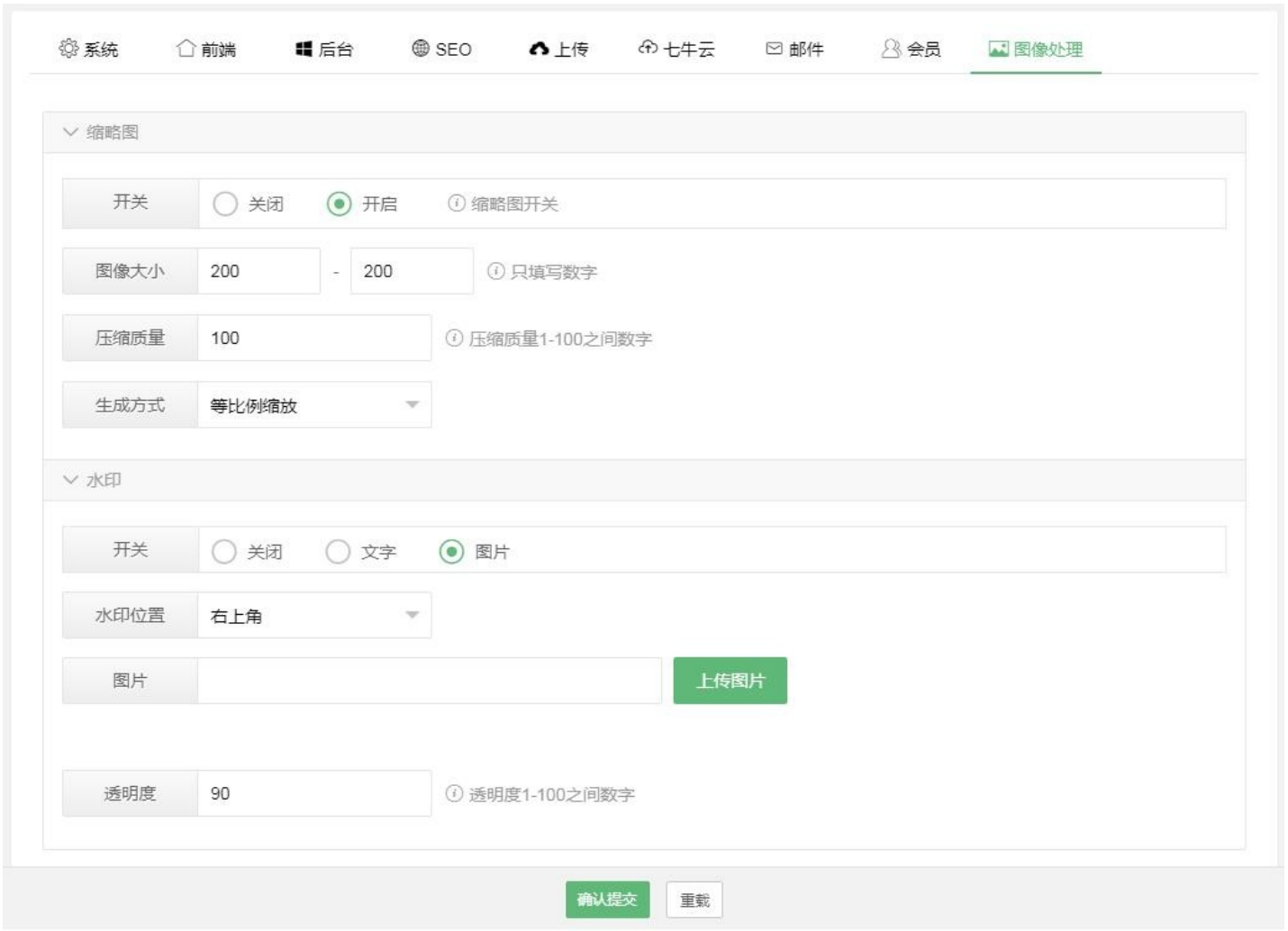
一、配置设置是系统核心参数的设置

- 系统：调试模式、Trace开关、模板目录
- 前端：缓存开关、首页命名、静态首页、敏感词设定等
- 后台：列表页每页数量、管理员登录设定
- SEO：百度验证、推送设定、准入秘钥、内链管理
- 上传：上传开关、上传文件设定及大小设定
- 七牛云：七牛云设置
- 邮件：邮件发送系统设置
- 会员：会员开关、分组等设置
- 图像处理：缩略图、水印开关和设置

> 每项设置都会有信息提示，以及相关的功能说明

二、界面预览




系统
前端
后台
SEO
上传
七牛云
邮件
会员
图像处理
缩略图
开关
关闭
开启
缩略图开关
图像大小
200
-
200
只填写数字
压缩质量
100
压缩质量1-100之间数字
生成方式
等比例缩放
水印
开关
关闭
文字
图片
水印位置
右上角
图片
上传图片
透明度
90
透明度1-100之间数字
确认提交
重载

模型管理

> 菜单位置：运维 -- 模型管理 -- 系统模型

一、模型管理是系统运行的核心功能，用于自定义各种模型类型，目前系统设定的类型有：

- 内容模型：用于创建栏目时调用的模型，是站点展示信息内容的主要模型，如：文章、产品、图片、案例、招聘等等。用于前台列表页展示以及内容展示。
- 功能模型：用于创建页面功能块的模型，例如：幻灯片、友情链接、页面块的设定与显示。
- 表单模型：定制表单，用于创建站点表单模型。也可用于创建会员表单模型。例如：在线留言、订购表单以及其他信息采集等。
- 业务模型：预留功能
- 附属表模型：附属表模型用于针对内容模型和功能模型创建附属表。当内容模型或者功能模型关联附属表后，内容模型和功能模型中的每条记录都会有自己的内容列表，如果一篇小说有不同的章节。

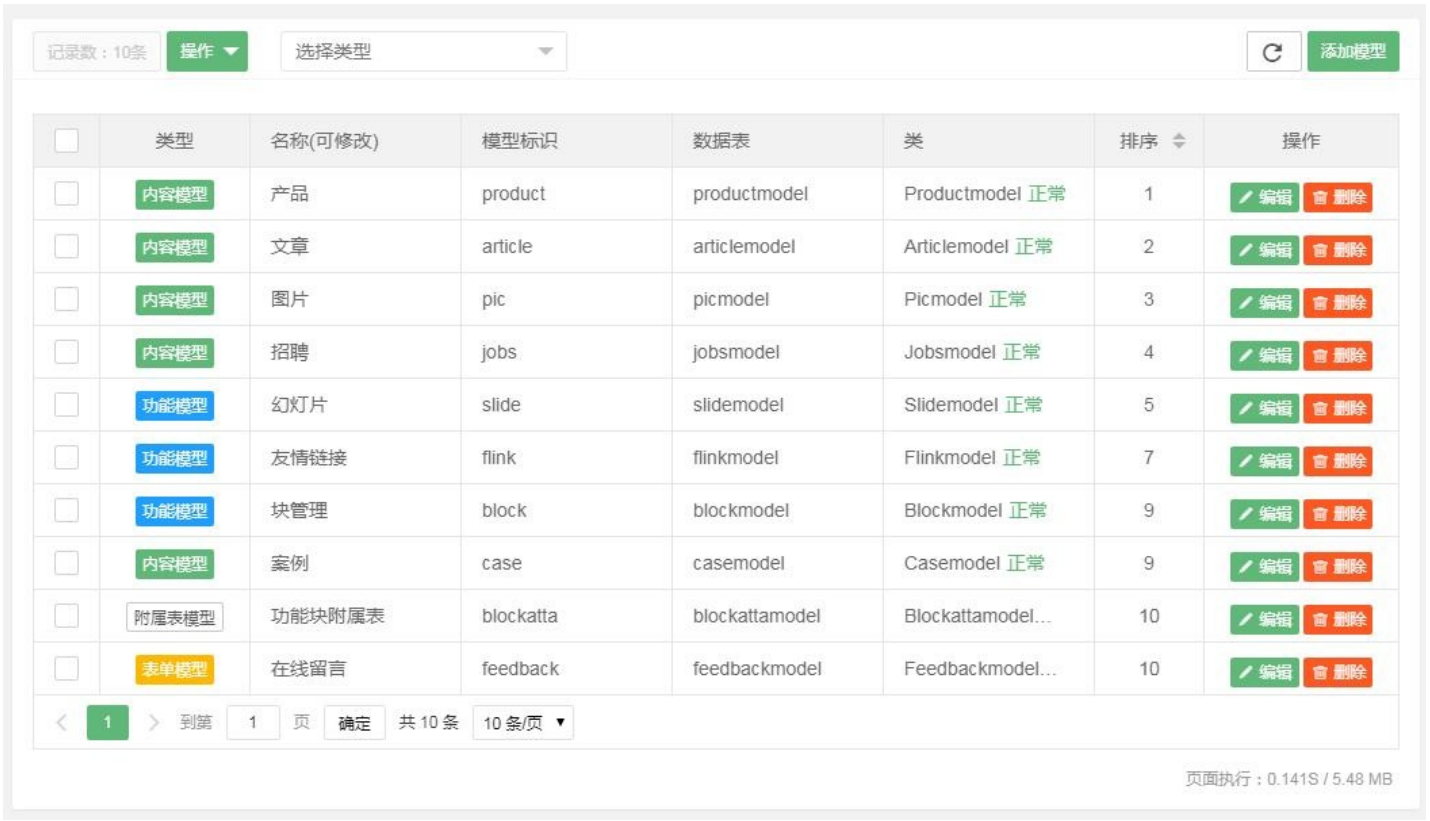
> 利用模型的附属表功能，可以再更多的应用场景中实现强大的预期效果。

二、模型操作

1. 删除模型：删除模型将删除该模型表所有数据及表，且数据不可恢复，同时也会删除该模型所建立的字段信息且不可恢复。
2. 清除数据：清除数据不会删除模型自定义字段，只会清空该模型表的所有内容。
3. 复制模型：复制模型功能是复制源模型的信息并建立新的模型数据表以及源模型所有的自定义字段，但是不会源模型表内容，只会建空表。

> 【重点】 所有模型的操作步骤都是先添加，然后点击修改，只有进行修改的时候，才能看到该模型的更多设置。比如：地区关联功能、附属表功能等等。

三、界面预览


记录数：10条
操作
选择类型
添加模型
类型
名称(可修改)
模型标识
数据表
类
排序
操作
内容模型 产品 product productmodel Productmodel 正常 1
内容模型 文章 article articlemodel Articlemodel 正常 2
内容模型 图片 pic picmodel Picmodel 正常 3
内容模型 招聘 jobs jobsmodel Jobsmodel 正常 4
功能模型 幻灯片 slide slidemodel Slidemodel 正常 5
功能模型 友情链接 flink flinkmodel Flinkmodel 正常 7
功能模型 块管理 block blockmodel Blockmodel 正常 9
内容模型 案例 case casemodel Casemodel 正常 9
附属表模型 功能块附属表 blockatta blockattamodel Blockattamodel... 10
表单模型 在线留言 feedback feedbackmodel Feedbackmodel... 10
编辑 删除
1 到第 1 页 确定 共 10 条 10 条/页
页面执行：0.141S / 5.48 MB


编辑
模型名称 产品
模型标识 product
模型类型 内容模型【前台栏目用，如：文章模型】
关联附属表 功能块附属表
关联地区 开
排序 1
备注：
1.创建模型自动创建公共模型类文件、数据表
2.不同模型初始字段不同
3.模型创建完毕后，设置对应字段
页面执行：0.109S / 5.63 MB
确认提交
重载
关闭

字段管理

> 菜单位置：运维 -- 模型管理 -- 字段管理

一、字段管理是对自定义模型字段的自定义

> 【重点】对字段的操作也是先增加后修改的原则，先增加字段选择字段类型，然后在修改该字段，即可看到该字段更多设置。比如：默认值、是否必填、是否是列表列、是否开启搜索、列表排序以及字段的验证规则等等。

字段类型

1. 文本框：单行文本内容，使用频率大，修改字段可设置验证规则。
   修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `列表列` 、 `列宽` 、 `列表排序` 、 `精确搜索` 、 `模糊搜索`
2. 文本域：多行文本内容，一般用于无格式要求的文本域。
   修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `精确搜索` 、 `模糊搜索`
3. 整数：整数类型
   修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `列表列` 、 `列宽` 、 `列表排序` 、 `精确搜索` 、 `模糊搜索`
4. 小数：小数类型，最多精确2位小数
   修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `列表列` 、 `列宽` 、 `列表排序` 、 `精确搜索` 、 `模糊搜索`
5. 下拉列表：下拉单选，用于数据不太多的下拉选项。
   修改可设置： `数据源` 、 `显示的字段` 、 `选中后的字段值` 、 `默认值` 、 `必填` 、 `允许修改` 、 `精确搜索` 、 `模糊搜索`
6. 单选：同下拉列表
7. 多选：同下拉列表，多选字段的数据存储格式为a,b,c,d,e,f多值以英文逗号分隔存入数据库表字段。
8. 开关：开关的值为0或1，一般用于对某个状态的判断。
   修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `列表列` 、 `列宽` 、 `列表排序` 、 `精确搜索`
9. 穿梭框：同多选
10. 颜色选择：颜色选择的值为十六进制颜色代码及 `#FFFFFF` 格式
    修改可设置： `默认值` 、 `必填` 、 `允许修改`
11. 单图上传：单张图片上传
    修改可设置： `默认值` 、 `必填` 、 `允许修改` 、 `列表列` 、 `列宽` 、 `缩略图开关` 、 `水印开关`
12. 单视频上传：单个视频上传

修改可设置： 默认值 、 必填 、 允许修改

13. 单文件上传：单个文件上传

    修改可设置： 默认值 、 必填 、 允许修改 、 列表列 、 列宽

14. 多图上传：批量上传多张图片

    修改可设置： 必填 、 允许修改 、 缩略图开关 、 水印开关

15. 多文件上传：批量上传多个文件

    修改可设置： 必填 、 允许修改

16. 日期时间：下拉日期时间选择

    修改可设置： 默认值 、 必填 、 允许修改 、 列表列 、 列宽 、 列表排序 、 精确搜索 、 模糊搜索 、 日期格式自定义

17. 单UEditor：单ueditor编辑器

    修改可设置： 默认值 、 必填 、 允许修改 、 缩略图开关 、 水印开关

18. Layedit：一个layedit编辑器无上传功能

    修改可设置： 默认值 、 必填 、 允许修改

19. 隐藏域：待定

20. 滑块：滑块是layui的内置模块，一般用于数值在一定范围内的选择，存储类型为数字。

    修改可设置： 默认值 、 必填 、 允许修改

> 每项设置都会有信息提示，以及相关的功能说明

二、字段的数据源

> 当字段类型为：下拉列表、单选、多选、穿梭框时需要设定数据源。

数据源类别有：

1. 属性数据： 属性在后台模型管理--属性维护里进行设定。如颜色分黑白红蓝紫等，档次分高档中档低档等。都是自定义设置。
2. 模型数据： 所有自定义模型中的数据
3. 其他数据： 需要指定一个表进行数据调用
4. 单页列表： 所有当前站点的单页内容
5. 站点列表： 站点列表
6. 栏目列表： 当前站点所有栏目列表，当字段类型为下拉时则栏目为分层级显示

显示的字段：是指在该字段中，所显示的内容的字段名。

选中值字段：是指在该字段中，选中当前选项后所存入数据库的字段的值。

> 例如：数据源为栏目列表：显示的字段写name，则该字段在栏目下拉列表中显示栏目名称，选中值字段写

id，则存入数据库中为该栏目的id值。



三、界面预览


模型选择
内容模型
产品
文章
图片
招聘
案例
功能模型
幻灯片
友情链接
块管理
表单模型
在线留言
业务模型
附属表模型
功能块附属
字段数：23条
字段管理
字段名称 字段名 字段类型 所属模型 必填 可修改 列表列 列宽 列表排序 操作
标题 title 文本框 幻灯片 0 -- 编辑
推荐 tuijian 开关 产品 0 编辑
标题 title 文本框 友情链接 0 -- 编辑
块描述 title 文本框 块管理 0 -- 编辑
标题 title 文本框 功能块... 0 -- 编辑
姓名 name 文本框 在线留言 0 -- 编辑
图片 img 单图上传 幻灯片 0 -- 编辑
链接 link 文本框 友情链接 0 -- 编辑
图片 img 单图上传 功能块... 0 -- 编辑
手机 mobile 文本框 在线留言 0 -- 编辑
调用标识 alias 文本框 块管理 0 -- 编辑
链接 link 文本框 幻灯片 0 -- 编辑
图片 img 单图上传 友情链接 0 -- 编辑
块主标题 title1 文本框 块管理 0 -- 编辑
块副标题 title2 文本框 块管理 0 -- 编辑

系统升级

> 菜单位置：运维 -- 系统管理 -- 系统升级

一、RGCMS2.0版本开始支持后台在线更新

在线更新运用了远程验证，所以使用该功能必须保持链接外网正常。

该功能也可查阅以往更新记录及更新内容。



关键词库

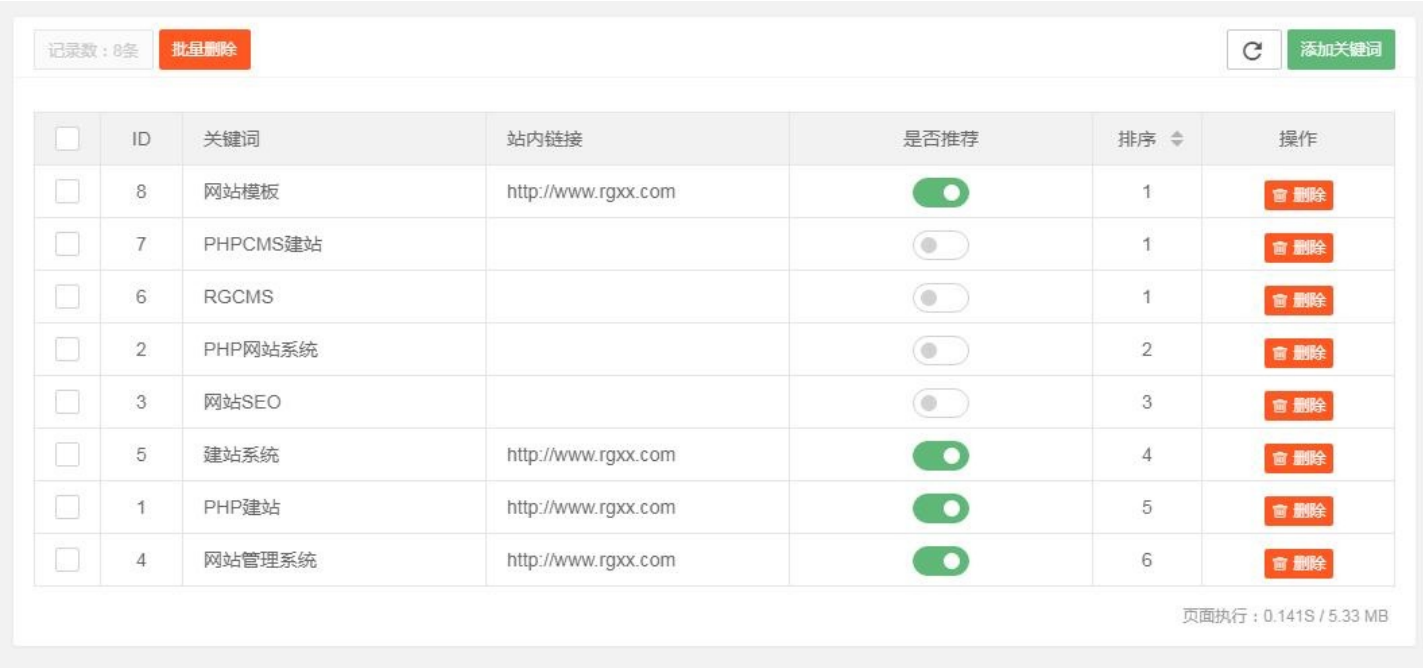
> 菜单位置：功能 -- SEO设置 -- 关键词库

一、关键词库是当前站点所有关键词的仓库

可对每个关键词进行链接设定，并在前端可通过RGCMS标签进行调用，调用方法请浏览标签部分。

同时只有在设定了关键词链接的，系统才会选择这些词进行内链转换。

二、界面预览



TKD模板

> 菜单位置：功能 -- SEO设置 -- TKD模板

一、含义

TKD：Title标题，Keys关键词，Desc描述

用法：通过系统标签嵌入到TKD模板中，可更方便快捷的定义站点的每个页面的TKD。自定义分类、自定义内容，然后在【站点】- 【站点信息】中进行相关设定。



方便快捷，确保每个页面TKD都不留空。

> 每个栏目及页面的TKD另外也可自定义，凡是存在自定义的TKD，将覆盖全局设定的TKD。

二、界面预览

记录数：12条 | 全部 | 首页标题 | 描述 | 栏目页标题 | 内容页标题 | 添加模板

	分类	模板内容	排序	操作
1	首页标题	{$site_keys},{$site_company}	1	删除
2	首页标题	{$site_keys}厂家,{$site_name}	2	删除
3	首页标题	{$site_keys}供应商,{$site_name}	3	删除
4	首页标题	中国{$site_keys}生产厂家,{$site_name}	4	删除
5	描述	{$site_name}多年来致力于为您提供各种{$site_keys}。价格优惠，质量保证，欢迎垂询。	5	删除
6	描述	{$site_name}专业生产{$site_keys}。公司以先进的生产技术和优质工艺成为行业知名供应商。欢...	6	删除
7	描述	{$site_name}可以为您提供专业的{$site_keys}系列产品，质量上乘，欢迎新老客户咨询。	7	删除
8	描述	{$site_name}是国内专业生产{$site_keys}的厂家。公司所生产的{$site_keys}质优价廉，远销多个...	8	删除
9	栏目页标题	{$nav_name},{$site_name}	9	删除
10	栏目页标题	{$nav_name},{$site_keys},{$site_name}	10	删除
11	内容页标题	{$info_title},{$nav_name},{$site_name}	11	删除
12	内容页标题	{$info_title},{$site_keys}厂家,{$site_name}	12	删除

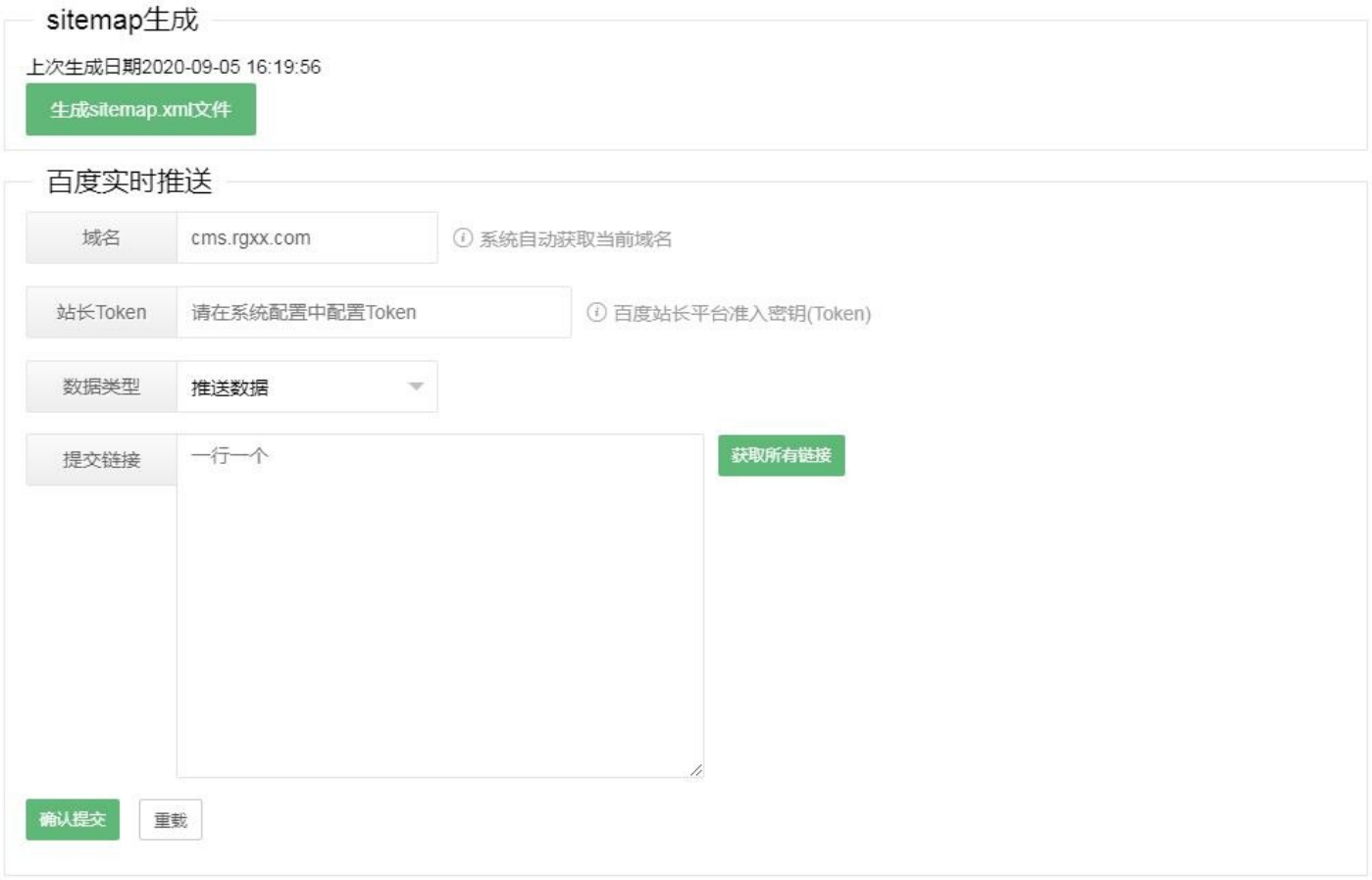
链接&Robots

> 菜单位置：功能 -- SEO设置 -- 链接提交
> 菜单位置：功能 -- SEO设置 -- Robots

一、链接提交

链接提交是指将站点URL信息提交到百度。让百度更好的去收录和抓取页面信息。

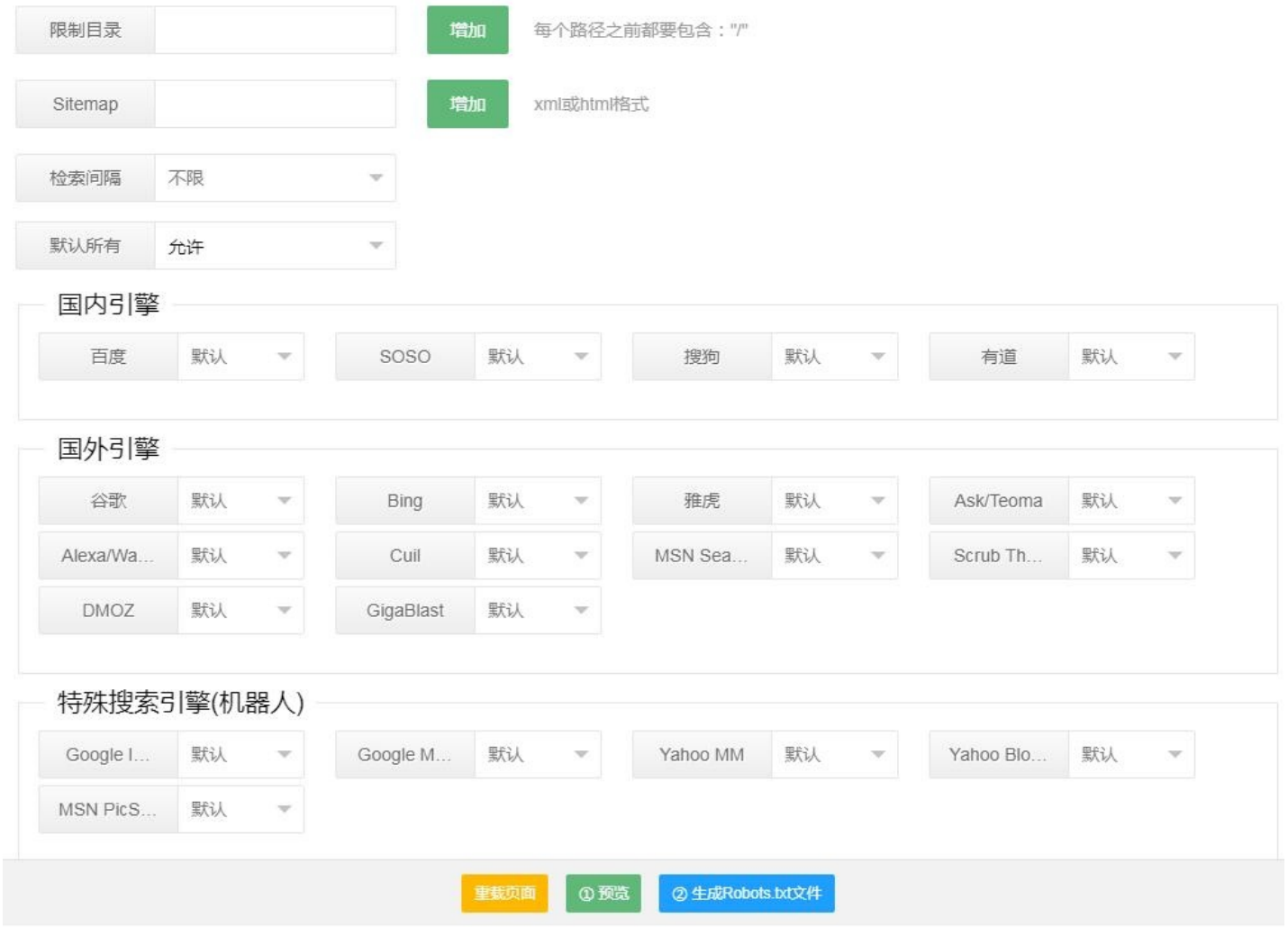
该页面同时还提供sitemap.xml在线生成功能。



> 使用该功能时，必须要获得百度站长平台准入密钥(Token)

二、Robots生成管理

在线更新和生成robots.txt文件


限制目录
增加
每个路径之前都要包含："/"
Sitemap
增加
xml或html格式
检索间隔
不限
默认所有
允许
国内引擎
百度
默认
SOSO
默认
搜狗
默认
有道
默认
国外引擎
谷歌
默认
Bing
默认
雅虎
默认
Ask/Teoma
默认
Alexa/Wa...
默认
Cuil
默认
MSN Sea...
默认
Scrub Th...
默认
DMOZ
默认
GigaBlast
默认
特殊搜索引擎(机器人)
Google I...
默认
Google M...
默认
Yahoo MM
默认
Yahoo Blo...
默认
MSN PicS...
默认
重载页面
① 预览
② 生成Robots.txt文件

站点列表

> 菜单位置：站点 -- 站点管理 -- 站点列表

一、站点列表管理所有站点



- 主站：一套系统只有一个主站，主站就是网站访问时的默认站点
- PC开关：如关闭PC，则显示设定的提升内容
- WAP开关：如果模板为自适应模板，请关闭。如果是独立模板，请开启，否则打开手机站为PC模板
- 创建时间：站点的建立时间
- 过期时间：站点设定过期时间，过期后站点自动关闭并提示过期提醒

> 新增站点请注意每个字段的命名规则，别名请用英文或拼音

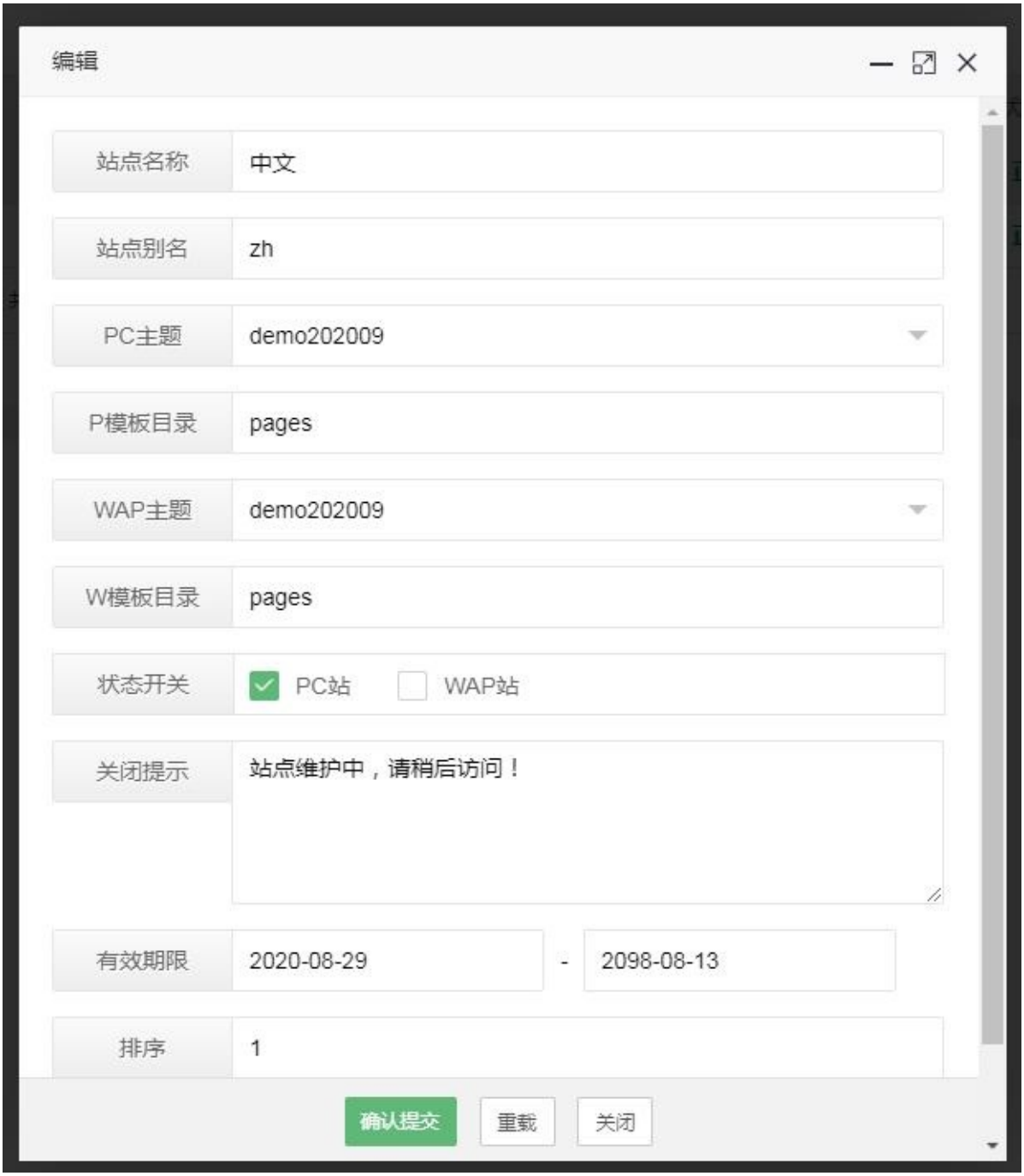
二、新增站点



>

- PC主题、WAP主题为下拉选项，系统自动识别 模板文件夹 下面的文件夹包， 模板文件夹 【系统配置】-- 【系统】中进行设定，默认为template，即路径为/public/template文件夹。
- 模板目录：为 模板文件夹 下的存放HTML模板的目录，如html或pages等自定义
- 站点别名：为站点的识别名，不可重复，一般为英文或拼音

例：如图设定



则模板文件夹路径为：\public\template\demo202009\

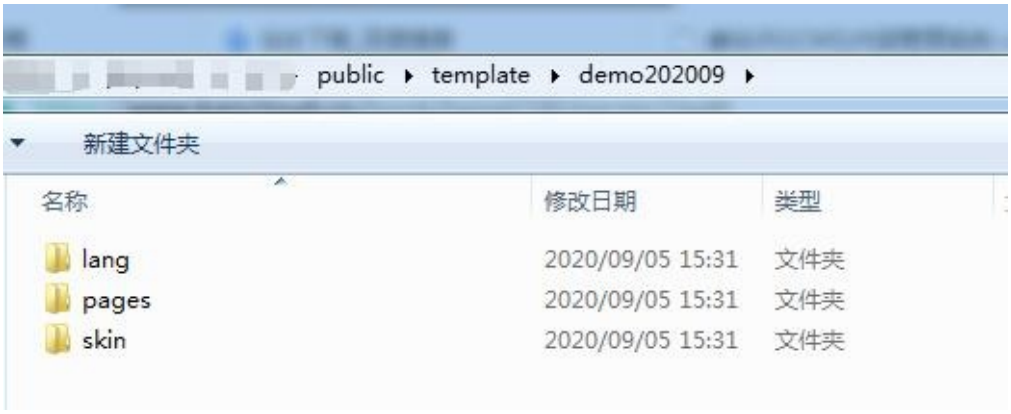
HTML模板目录为：\public\template\demo202009\pages\

模板语言包路径为：\public\template\demo202009\lang\

静态资源请放于自命名的独立文件夹内，如：skin

则静态资源路径为：\public\template\demo202009\skin\

如下图实例：

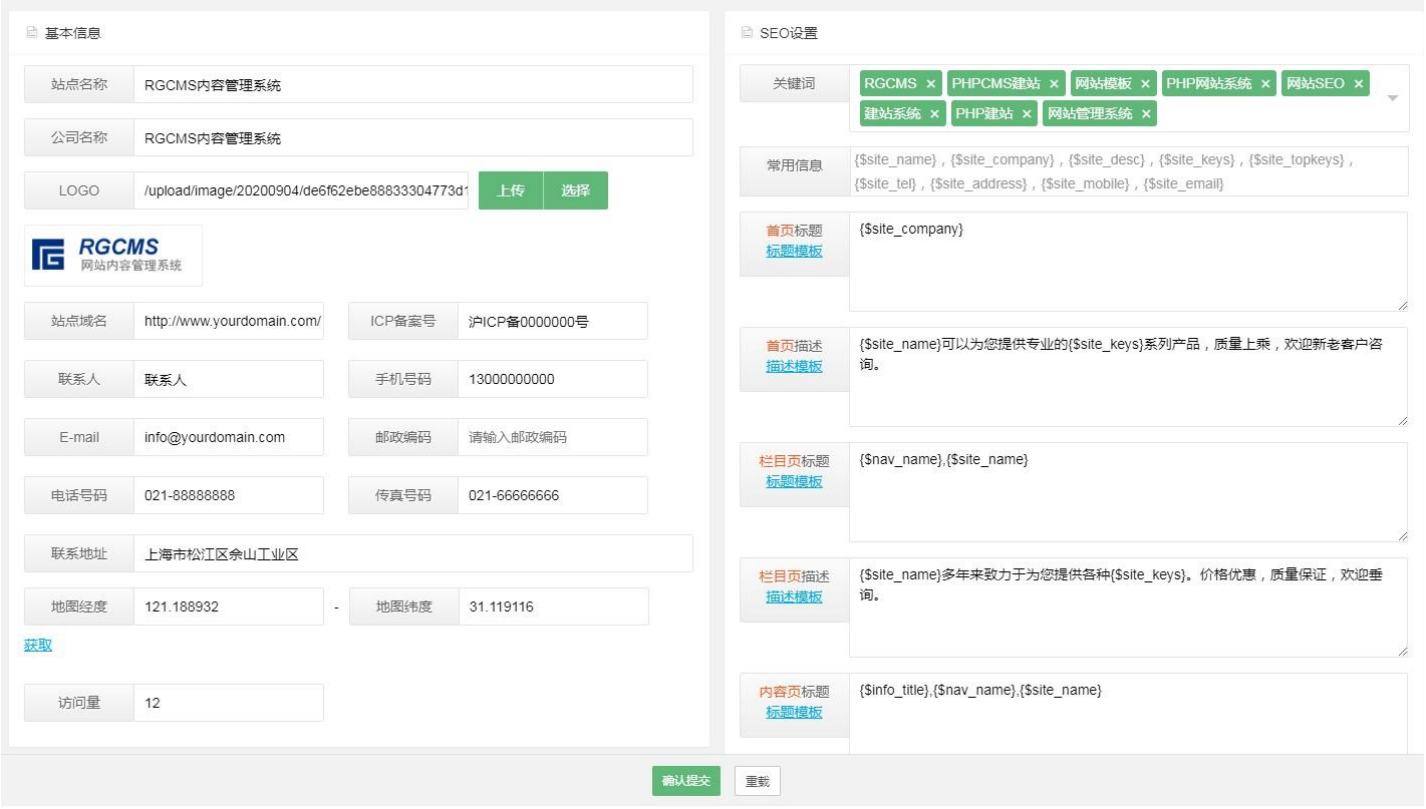


> 自命名规则的模板文件夹及HTML模板目录，能有效防止模板被直接下载。

站点信息&全局变量

> 菜单位置：站点 -- 站点管理 -- 站点信息
> 菜单位置：站点 -- 站点管理 -- 全局变量

一、站点列表管理所有站点

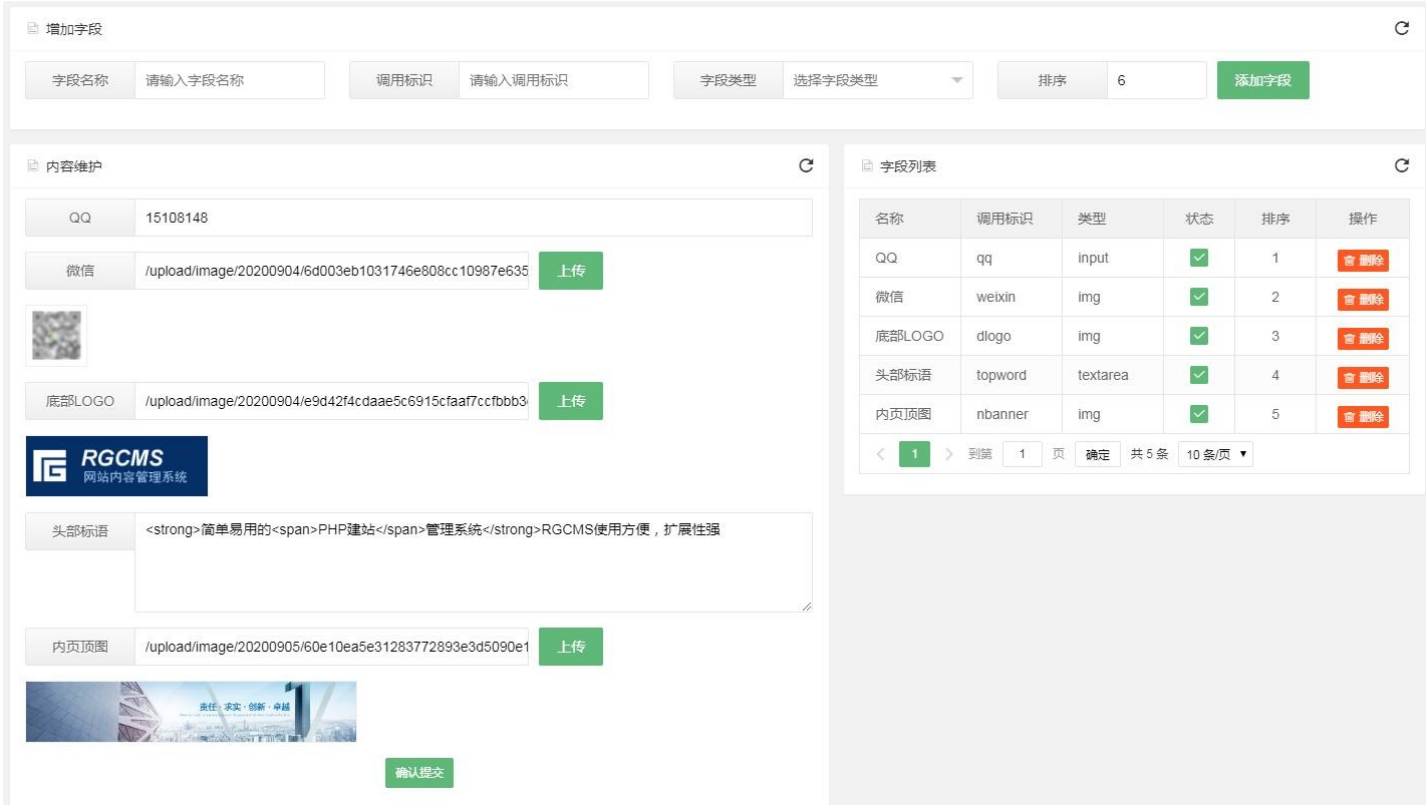


> 站点信息设置站点常规信息内容，包括站点名称，企业名称，Logo，域名，备案信息，电话，地址，邮箱，经纬度位置等。同时也设置站点的基础SEO信息，例如站点关键词，页面标题，页面描述等。
> 设置合理的页面标题、关键词、以及描述，及TKD设置，让站点所有页面的TKD都不会留空，同时TKD信息还可以根据栏目、页面等的相关自定义而不同。

二、全局变量【又名动态字段】

> 当站点信息内容无法满足需求时就需要扩展，这就叫全局变量

例：如图所示



界面包括：

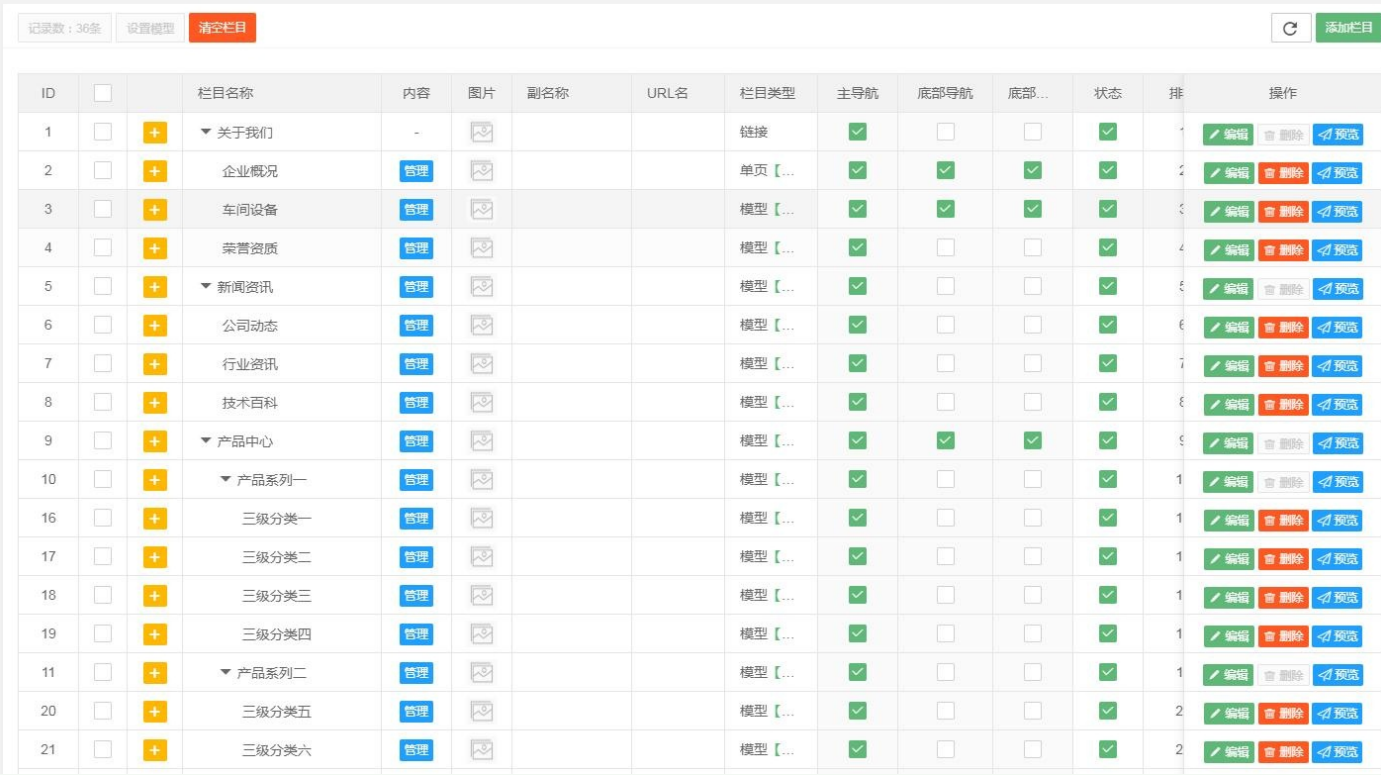
1. 增加字段
2. 字段内容维护
3. 字段列表

栏目管理

> 菜单位置：站点 -- 站点管理 -- 栏目管理

一、栏目管理是站点的核心操作，也是站点的前端框架结构

任何一个站点，都会有栏目架构，RGCMS系统的栏目管理简单明了，操作方便，采用Layui树形组件构建。



RGCMS2.0栏目系统支持N级栏目，当然你要是有N级的话。栏目列表可快速修改栏目名称、副名称、Url美化名称、开关以及排序等，还可以对栏目内容直接进行管理，链接类型除外。

> 列表顶部的清空栏目按钮功能：一般来讲，如果当前栏目存在子栏目，则是无法进行删除的，这样有效保护了内容，防止误删。【清空栏目】按钮，用于一次性删除当前站点所有栏目、内容模型的内容、附属表内容、动态字段内容，不包括其他模型、模型字段以及附属表字段等。可以理解为初始化当前站点内容，但只限于内容模型内容。

二、新增(编辑)栏目

新增编辑栏目即对当前站点的栏目进行新增和编辑操作。

界面如下：



特色：

1. 批量添加栏目，批量设置栏目名称。

2. 栏目类别选择：RGCMS2.0栏目系统支持单页系统、系统模型、链接等类型。当选择单页内容时，可直接新建单页也可单页列表里进选择；当选择系统模型时，将从系统模型中的内容模型里进行选择。注意，栏目类型的系统模型，只能选择系统模型中建立的内容模型。如：产品、文章、图片、招聘等模型用于站点列表展示的。

3. 模板：设定内容模型展现时，所运用的模板以及详情页模板。

4. 关键词：可对不同栏目设立不同关键词及描述，当栏目中有自定义关键词和描述时，将自动替换系统所设置的全局TKD信息。

5. URL美化：自定义美化url名称。系统默认栏目名称为 `/n-栏目id/` ，美化后为 `/美化名称/` 。

6. 设置栏目位置：栏目位置为自定义位置，功能位于【站点】-- 【站点管理】 -- 【栏目位置】，栏目位置的概念类似于栏目分组，模板中可对不同分组的栏目进行不同的调用。强大而灵活。

单页管理

> 菜单位置：站点 -- 内容管理 -- 单页内容

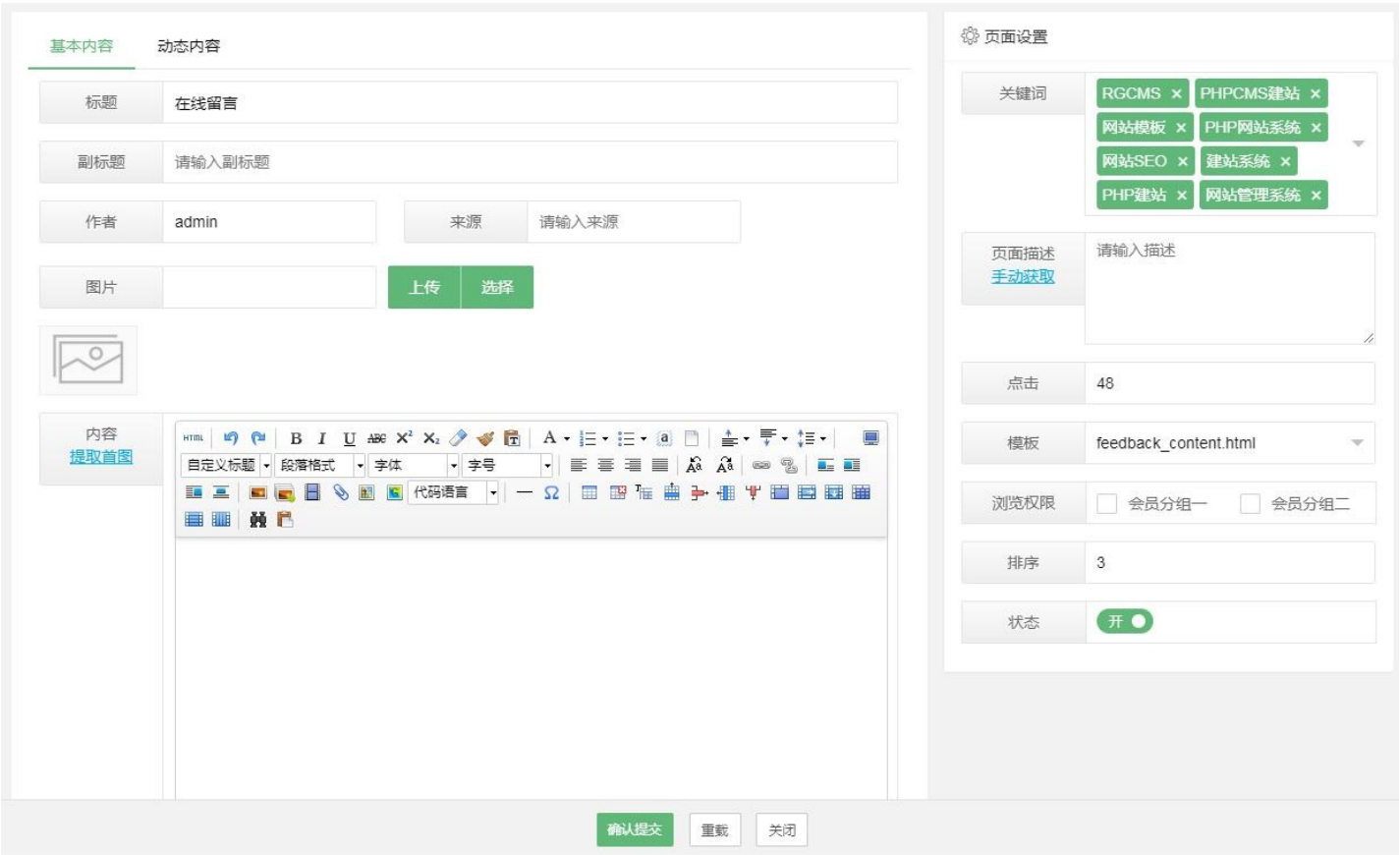
一、单页用于栏目调用或者非栏目内容页面的独立显示

单页属于独立模型，不同于自建模型，单页只有一个展示页面，内容一般为概括、介绍、联系方式、表单页面等。



二、新增(编辑)单页

界面如下：



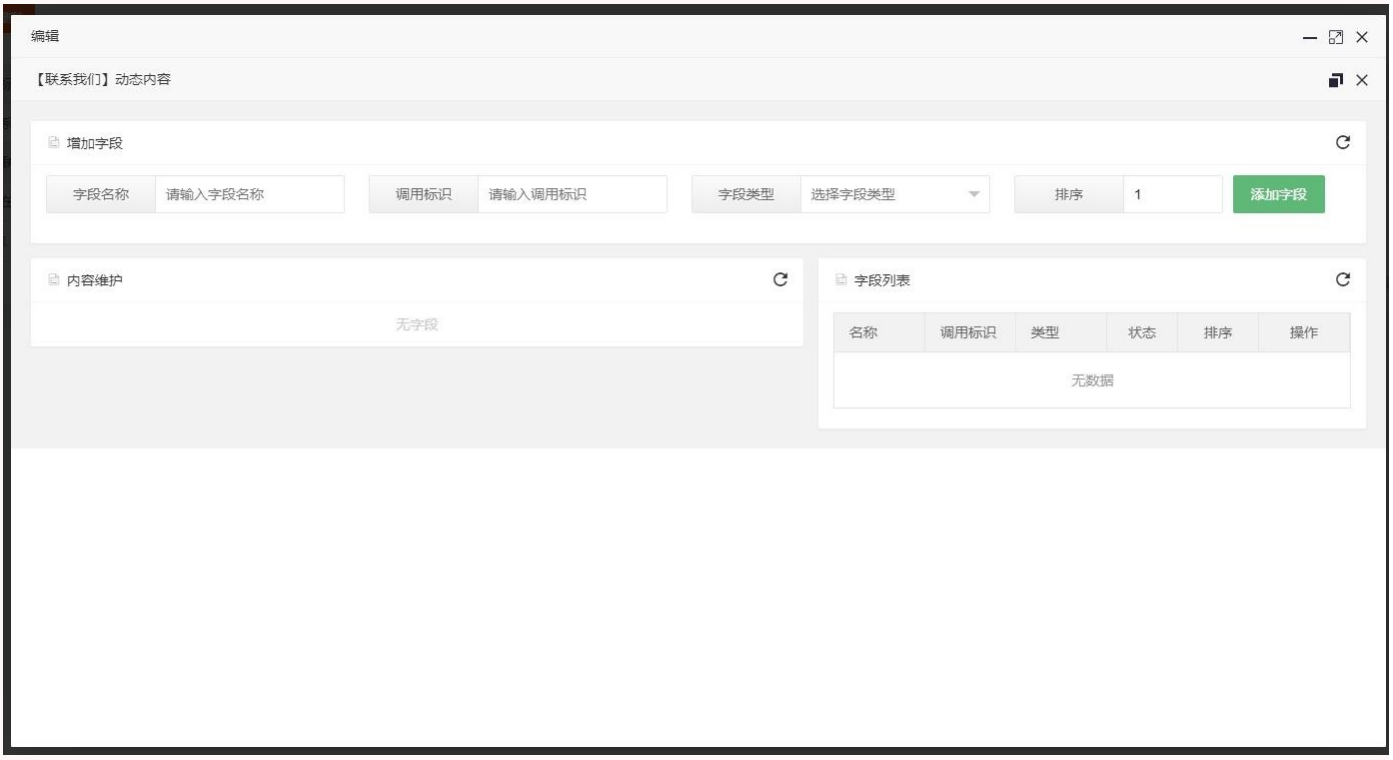
> 1. 单页标题：必填，例如企业概况。

2. 副标题：非必填，一般为主标题的英文等。
3. 作者、来源：非必填。
4. 图片：默认提供一个单图片上传入口。
5. 内容：内容默认为ueditor编辑器，内容为该页面所展示的主体内容。
6. 页面关键词、描述：设置页面关键词和描述后，将自动覆盖掉系统设定的TKD。
7. 模板：可为当前单页设置不同的单页模板。

三、单页的动态字段内容

在修改单页时，才会出现顶部的动态内容选项。

动态内容一般用于扩展当前单页字段，如默认字段不能满足当前单页内容时，可扩展动态字段内容。但是，动态字段内容仅限于该条单页内容，其他单页内容无效。



动态字段操作同全局变量操作

栏目内容

> 菜单位置：站点 -- 内容管理 -- 栏目内容

一、栏目内容管理 内容模型 的所有内容

首页

栏目内容管理菜单为 自定义的内容模型 的名称，点击不同的模型进入对应的内容管理。

首页显示有当前模型的当前站点的记录总数、回收站内情况以及是否有附属表。

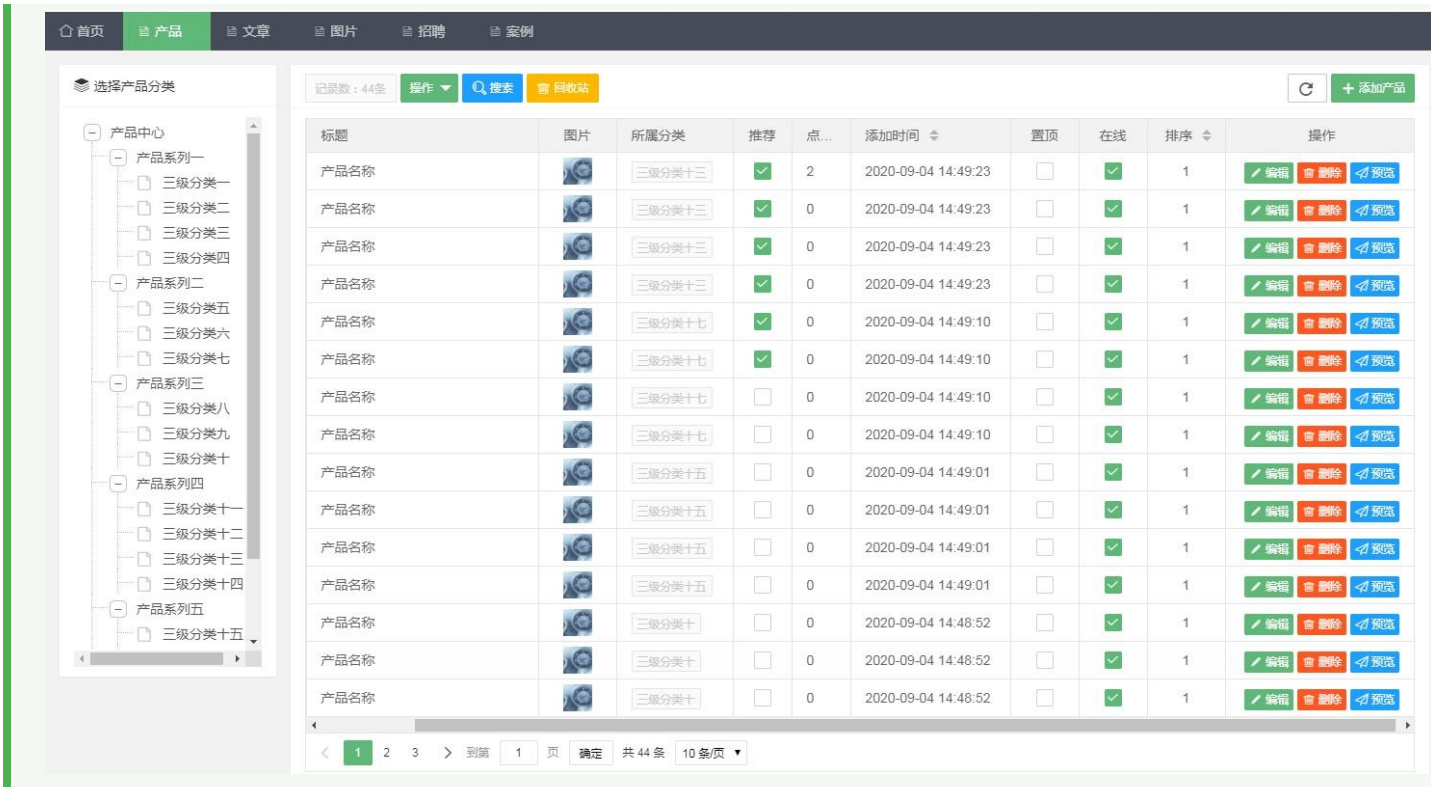


内容列表

由于模型为内容模型，因此内容列表左侧为当前内容模型的所有栏目分类。

点击不同分类查看不同分类内容。

列表中显示的自定义字段，可在模型字段中进行设置，并可设置列表列的宽度以及是否允许修改等。



当该内容模型关联了附属表功能，列表中会有附属表内容链接，打开即可进行附属表内容的管理。

二、新增(编辑)内容

界面如下：


基本内容
标题
请输入标题
栏目
三级分类一
作者
管理员
来源
请输入来源
缩略图
上传
选择
多图
上传
选择
开关
推荐
内容
提取首图
自定义标题
段落格式
字体
字号
代码语言
元素路径:
字数统计
其他设置
文件名(拼)
输入自定义文件名
外链
如有外链则填写
关键词
请选择关键词
页面描述
手动获取
请输入描述
模板
默认为栏目指定模板
Target
打开方式
点击
0
发布时间
2020-09-07 13:12:39
排序
1
状态
在线
置顶
页面执行：0.093S / 4.34 MB
确认提交
重载
关闭

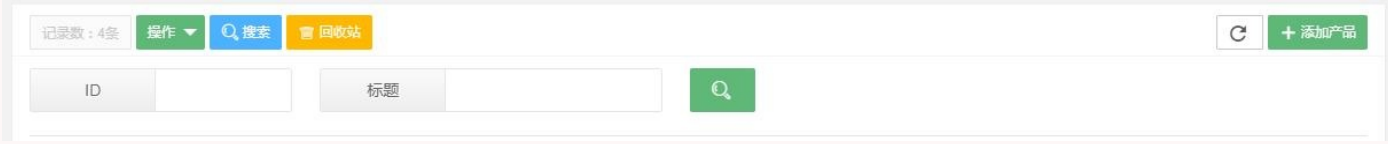
1. 自定义固定字段
2. 如内容模型开启了地区，则可显示地区选择
3. 自定义url美化
4. 设置关键词、页面描述
5. 单独设置内容模板
6. 设置打开方式以及点击量
7. 可设定多图的某图为缩略图
8. 可提取内容首图为缩略图
9. 可自动获取内容文字为页面描述
10. 编辑内容时，可对当前内容进行动态字段扩展

动态内容用于扩展当前内容字段，如默认字段不能满足当前内容时，可扩展动态字段内容。但是，动态字段内容仅限于该条内容，其他内容无效。

三、列表页搜索功能

列表页顶部提供搜索功能。

默认提供ID和标题搜索，更多搜索内容请设置自定义字段搜索开关，并设置为精确搜索或模糊搜索



功能管理

> 菜单位置：站点 -- 内容管理 -- 功能管理

一、功能管理是管理 功能模型 的所有内容

首页

功能管理菜单为 自定义的功能模型 的名称，点击不同的模型进入对应的功能管理。

首页显示有当前模型的当前站点的记录总数以及是否有附属表。



内容列表

由于模型为功能模型，因此列表左侧无栏目分类。

功能模型一般为幻灯片、友情链接、站点块管理以及其他非栏目内容的展示。

列表中显示的自定义字段，可在模型字段中进行设置，并可设置列表列的宽度以及是否允许修改等。

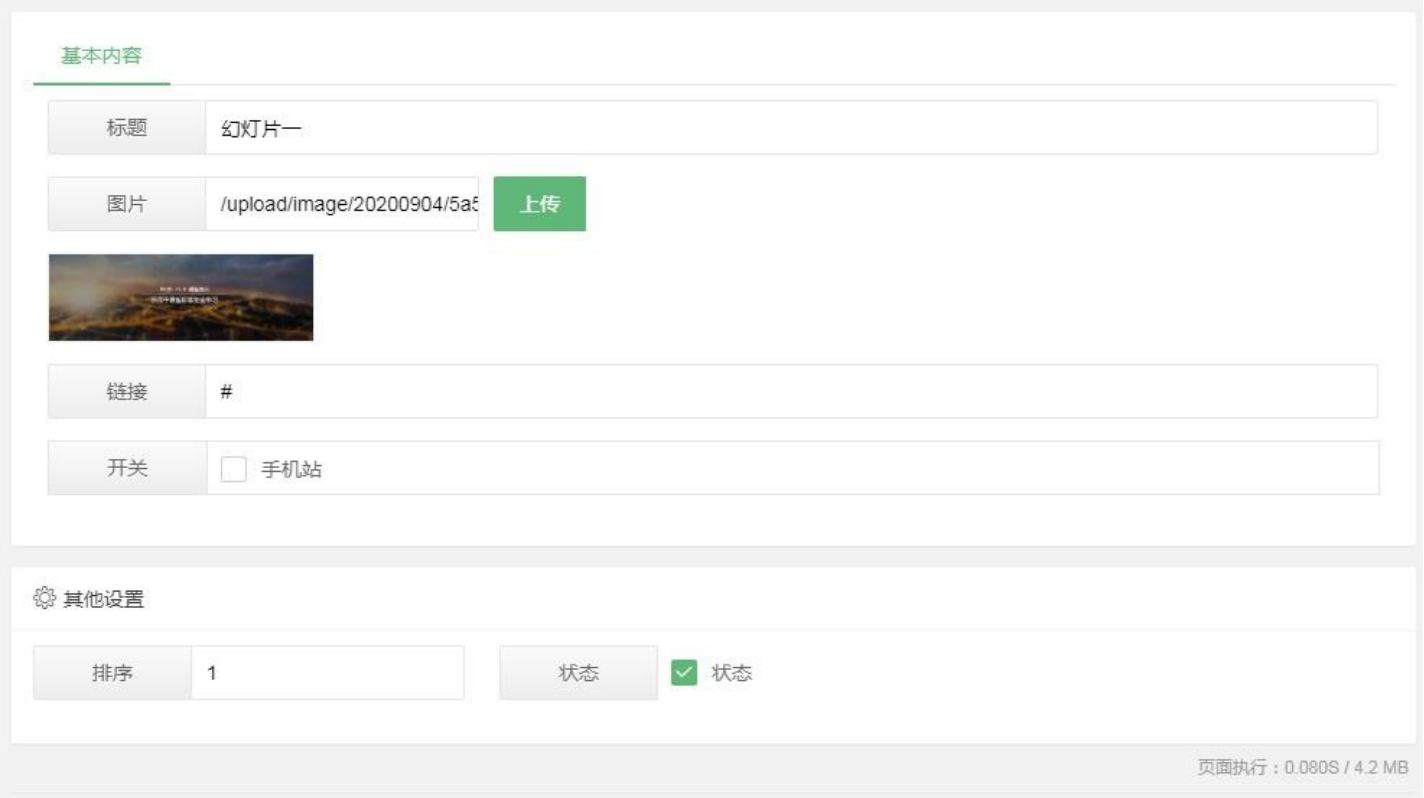


当该功能模型关联了附属表功能，列表中会有附属表内容链接，打开即可进行附属表内容的管理。



二、新增(编辑)内容

界面如下：

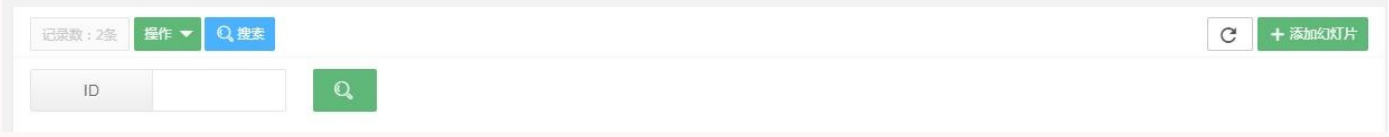


1. 自定义固定字段
2. 功能模型无地区管理功能
3. 设置排序
4. 设置内容状态
5. 由于功能模型字段为全自定义，因此无动态字段功能

三、列表页搜索功能

列表页顶部提供搜索功能。

默认提供ID搜索，更多搜索内容请设置自定义字段搜索开关，并设置为精确搜索或模糊搜索



表单管理

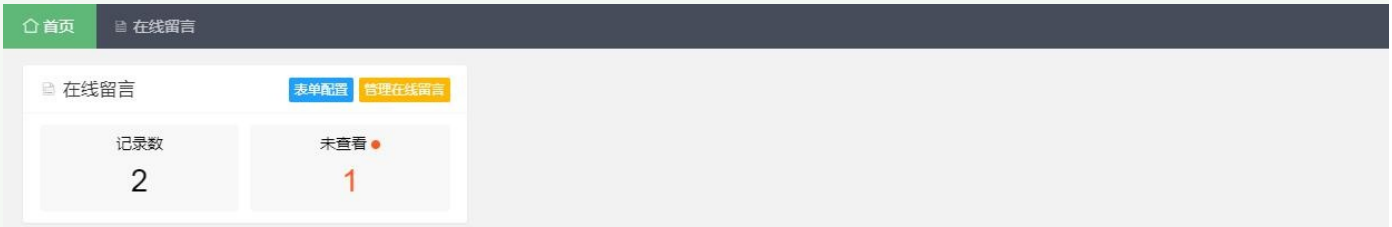
> 菜单位置：站点 -- 内容管理 -- 表单管理

一、表单管理是管理 表单模型 从站点提交的内容的管理，如在线留言

首页

表单管理菜单为 自定义的表单模型 的名称，点击不同的模型进入对应的表单管理。

首页显示有当前模型的当前站点的记录总数、未查看的表单记录。



内容列表

由于模型为表单模型，因此列表左侧无栏目分类。

表单模型一般为留言、订单、信息采集等通过前端提交数据至后台。

列表中显示的自定义字段，可在模型字段中进行设置，并可设置列表列的宽度以及是否允许修改等。



> 表单模型无附属表功能

二、查看(编辑)数据

界面如下：

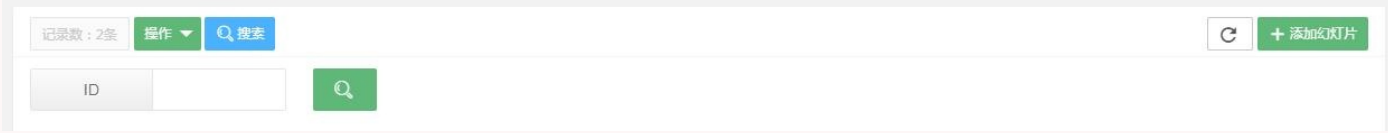


1. 左侧为表单自定义固定字段
2. 表单模型无地区管理功能
3. 设置状态
4. 设置内容回复
5. 表单模型无动态字段功能

三、列表页搜索功能

列表页顶部提供搜索功能。

默认提供ID搜索，更多搜索内容请设置自定义字段搜索开关，并设置为精确搜索或模糊搜索



后台总结（必看）

模型和字段

1. 模型自定义，字段自定义
2. 模型分：内容模型、功能模型、表单模型、业务模型、附属表模型
3. 模型和字段都是先新增然后后修改的设置步骤
4. 字段可设置规则、必填、列表里显示、列表宽度、是否可修改、是否可搜索
5. 同一类型字段可重复定义
6. 只有设置了可搜索的字段，前台和后台管理里的搜索才起作用
7. 当字段类型为：下拉列表、单选、多选、穿梭框时需要设定数据源
8. 设置数据源时，显示的字段是指在该字段中，所显示的内容的字段名。选中值字段是指在该字段中，选中当前选项后所存入数据库的字段的值
9. 创建模型都会自动创建模型数据表以及该模型的类文件
10. 内容模型、功能模型可关联附属表
11. 内容模型可关联地区

系统配置

1. 查看具体错误，请打开系统配置中的调试模式和trace
2. 模板存放目录配置
3. 开发时请关闭前台缓存，网站上线运营请打开缓存提高效率
4. 敏感词为系统内置，会自动将敏感词过滤成*号
5. 实现链接推送百度，必须设置百度站长的准入秘钥

站点内容

1. 掌握站点模板存放路径与命名
2. 新增站点后，刷新后台可在右上角进行站点管理切换
3. 全局变量是针对站点信息字段不够用时所建立
4. 栏目可设置自定义分组，可一次性清空所有栏目以及内容模型内容、附属表内容、动态字段内容
5. 栏目内容管理，根据栏目分类
6. 除内容模型外其他模型无栏目分类
7. 单页、栏目内容可动态扩展该条内容字段
8. 功能内容可实现页面的块管理
9. 表单内容可设置提醒，有查看与未查看提醒功能

> 其他未提及功能，请至后台进行测试查看，如果疑问请入Q群交流：180506934

标签变量

RGCMS2.0 基于Thinkphp5.1+开发，因此模板中适用所有TP5.1语法与标签。

TP5.1模板链接：https://www.kancloud.cn/manual/thinkphp5_1/354069

> RGCMS2.0 标签是TP5.1内置标签的扩展，让开发使用者能简单的使用标签达到更理想的效果，只需简单设置标签属性，即可实现预期的结果。
>
> RGCMS2.0 根据传统模板开发经验，取其精华去其糟粕，简化了标签的使用，简化了标签的形态特征。

> RGCMS2.0 标签全部以rg作为标签开始，循环标签以/rg 作为标签结束

如：顶层栏目标签

```
{rg:nav}
<a href="{$nav.url}">{$nav.name}</a>
{/rg:nav}
```

顶层栏目以及二级栏目

```
{rg:nav} // 顶层
<li><a href="{$nav.url}">{$nav.name}</a></li>
    {rg:nav pid="$nav['id']"} // 二级
    <a href="{$nav.url}">{$nav.name}</a>
    {/rg:nav}
{/rg:nav}
```

页面变量

全局变量
当前栏目
单页内容
列表页
详情页

全局变量

> RGCMS2.0的全局变量，可以在模板中任意调用

一、站点信息

站点信息格式 `{$site_*}` *代表字段名

> 适用页面：任何包括首页

变量	说明
{$site_id}	站点ID
{$site_title}	站点名称，如：中文
{$site_name}	站点标题
{$site_company}	公司名称
{$site_icp}	网站备案号
{$site_logo}	站点LOGO地址
{$site_tel}	电话
{$site_fax}	传真
{$site_address}	地址
{$site_mobile}	手机
{$site_code}	邮编
{$site_contact}	联系人
{$site_email}	电子邮箱
{$site_lng}，{$site_lat}	地图进度和纬度
{$site_views}	站点访问次数
{$site_keys}	首页关键词
{$site_desc}	站点描述
{$site_alias}	站点别名，如：ch,en
{$site_isdefault}	是否为主站点，值为0,1
{$site_domain}	站点域名
{$site_expired}	站点是否过期，值为0,1
{$site_url}	站点首页的链接，如：/en

二、模板路径变量

变量	说明
{$tempdir}	站点模板路径

```
<link rel="stylesheet" type="text/css" href="{$tempdir}/static/css/style.css"/>
```

三、页面标题、关键词、描述标签【TKD标签】

> RGCMS2.0的TKD标签所有页面可统一标签

变量	说明
{$seo_title}	当前页面标题 ， `<title>{$seo_title}</title>`
{$seo_keys}	当前页面关键词信息， `<meta name="keywords" content`
{$seo_desc}	当前页面描述信息， `<meta name="description" cont`

四、表单提交变量

> RGCMS2.0的表单提交变量，一般用于前台页面的表单提交以及内容搜索表单提交或ajax地址

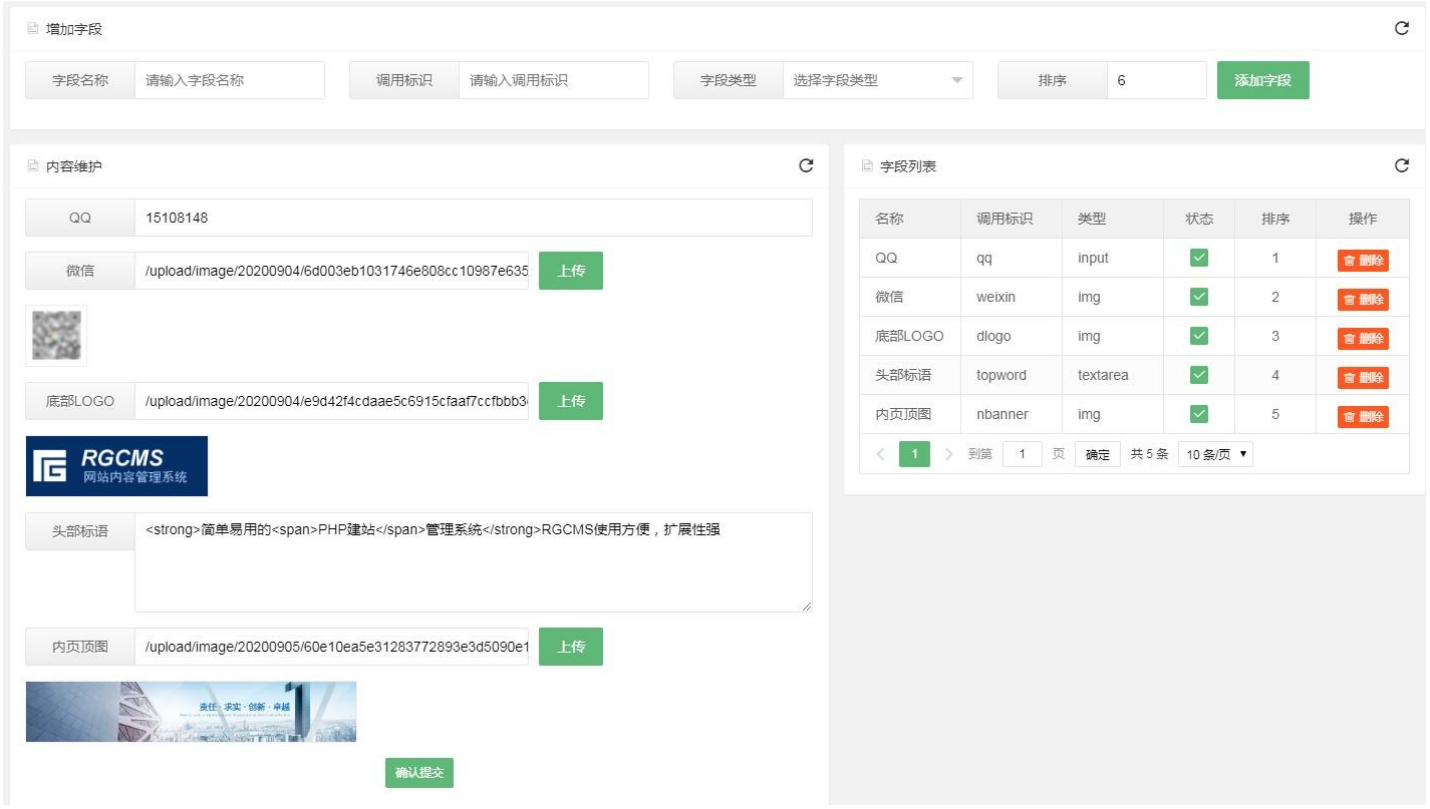
变量	说明
{$url_search}	搜索提交地址
{$url_form}	表单提交地址

五、全局扩展变量

> RGCMS2.0的全局扩展变量是对站点信息的扩展，当站点信息无法满足需求时，可用全局变量

变量	说明
{$site_ext_调用标识}	如：{$site_ext_weixin}

全局标签添加位置：【站点】--【站点管理】--【全局变量】


增加字段
字段名称
请输入字段名称
调用标识
请输入调用标识
字段类型
选择字段类型
排序
6
添加字段
内容维护
QQ
15108148
微信
/upload/image/20200904/6d003eb1031746e808cc10987e635
上传
底部LOGO
/upload/image/20200904/e9d42f4cdaae5c6915cfaaf7ccfbbb3
上传
RGCMS
网站内容管理系统
头部标语
简单易用的PHP建站管理系统RGCMS使用方便，扩展性强
内页顶图
/upload/image/20200905/60e10ea5e31283772893e3d5090e1
上传
确认提交
字段列表
名称
调用标识
类型
状态
排序
操作
QQ
qq
input
1
删除
微信
weixin
img
2
删除
底部LOGO
dlogo
img
3
删除
头部标语
topword
textarea
4
删除
内页顶图
nbanner
img
5
删除
1
到第
1
页
确定
共 5 条
10 条/页

当前栏目

> 所有栏目页可用固定变量：全局变量、当前栏目变量

栏目页变量

当前栏目变量格式 `{$nav_*}` *代表字段名

当前栏目的顶级栏目变量格式 `{$topnav_*}` *代表字段名

> 适用页面：所有栏目页以及内容详情页

当前栏目信息

变量	说明
{$nav_id}	栏目ID
{$nav_name}	栏目名称
{$nav_ename}	栏目副名称
{$nav_alias}	栏目别名
{$nav_pid}	栏目上级ID，顶级栏目为0
{$nav_type}	栏目类型：about，model，link
{$nav_img}	栏目图片
{$nav_keys}	关键词
{$nav_desc}	描述
{$nav_torder}	栏目排序
{$nav_target}	栏目打开方式
{$nav_url}	栏目url
{$nav_hasChild}	子栏目个数
{$nav_fids}	当前栏目的所有父栏目ID包含本身ID，如1,2,3,4
{$nav_ext_*}	栏目扩展字段，*代表扩展字段调用标识

当前栏目的顶级栏目信息

变量	说明
{$topnav_id}	顶层栏目ID
{$topnav_name}	顶层栏目名称

{$topnav_ename}	顶层栏目副名称
{$topnav_alias}	顶层栏目别名
{$topnav_pid}	顶层栏目上级ID，为0
{$topnav_type}	顶层栏目类型：about，model，link
{$topnav_img}	顶层栏目图片
{$topnav_keys}	顶层关键词
{$topnav_desc}	顶层描述
{$topnav_torder}	顶层栏目排序
{$topnav_target}	顶层栏目打开方式
{$topnav_url}	顶层栏目url
{$topnav_hasChild}	顶层栏目的子栏目个数
{$topnav_ext_*}	顶层栏目扩展字段，*代表扩展字段调用标识

单页内容

> 单页可用固定变量：全局变量、当前栏目变量、单页变量

当前单页内容变量

当前单页内容变量格式 `{$info_*}` *代表字段名

> 适用页面：栏目类型为单页

变量	说明
{$info_id}	单页ID
{$info_title}	单页标题
{$info_etitle}	副标题
{$info_author}	作者
{$info_source}	来源
{$info_img}	单页单图
{$info_keys}	关键词
{$info_desc}	描述
{$info_hits}	点击量
{$info_content|raw}	单页主内容
{$info_create_time}	添加时间
{$info_update_time}	修改时间
{$info_ext_*}	单页动态字段，*代表动态字段调用标识

列表页

> 列表页可用固定变量：全局变量、当前栏目变量、列表页变量

列表页变量

> 适用页面：栏目类型为模型

> 列表页可以使用全局变量、栏目变量，另外列表页还附加了当前的模型信息

变量	说明
{$modelinfo_id}	当前栏目的模型ID
{$modelinfo_title}	当前栏目的模型名称
{$modelinfo_name}	当前栏目的模型标识
{$modelinfo_tablename}	当前栏目的模型表
{$modelinfo_classname}	当前栏目的模型类名
{$modelinfo_attamodel_id}	当前栏目的模型所关联的附属表模型ID

详情页

> 详情页可用固定变量：全局变量、当前栏目变量、列表页变量、详情页变量

当前内容详情变量

当前内容详情变量变量格式 `{$info_*}` *代表字段名

上一篇变量 `{$prev}`

下一篇变量 `{$next}`

> 适用页面：栏目类型为模型的详情页

变量	说明
{$info_id}	内容ID
{$info_title}	内容标题
{$info_navlist_id}	内容所属的栏目ID
{$info_cmsmodels_id}	内容所属的模型ID
{$info_author}	作者
{$info_source}	来源
{$info_img}	内容单图
{$info_imgs}	内容多图
{$info_keys}	关键词
{$info_desc}	描述
{$info_hits}	点击量
{$info_istop}	是否置顶：0|1
{$info_target}	内容打开方式
{$info_content|raw}	内容主内容
{$info_create_time}	添加时间
{$info_update_time}	修改时间
{$info_*}	内容自定义字段，*代表字段调用名
{$info_ext_*}	内容动态字段，*代表动态字段调用标识

上一篇下一篇的常规使用：

> 使用举例

```
<ul class="content-exp">
<li><a href="{$prev['url']}">上一篇：{$prev['title']|default="没有了"}</a></li>
<li><a href="{$next['url']}">下一篇：{$next['title']|default="没有了"}</a></li>
</ul>
```

标签调用

站点列表
栏目调用(读数据库)
栏目调用(读缓存)
内容列表
指定模型内容
自定义分页
面包屑
功能调用
附属表内容
单页内容列表
多图字段
表单内容列表
字符串分割
搜索页列表
地区调用
关键词列表
属性列表
内容统计

站点列表

> 站点列表用于列出所有站点的名称链接及其他相关信息，如：中文版 | ENGLISH

站点列表

标签： `{rg:site 参数名称="参数值"}{/rg:site}`

> 适用页面：任何

> 标签属性：循环

参数列表

参数(?为可选参数)	默认值	使用示例	解释
?name	site	name="site"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示
?limit		limit="4"	输出数量

常用字段调用对照表

```
["id"] => 站点id
["title"] => 站点语言名称
["alias"] => 站点标识
["isdefault"] => 是否主站点，1|0
["domain"] => 站点域名（2020123101新增：当该站点绑定域名并设置为主域名）
["name"] => 站点名称
["company"] => 公司名称
["url"] => 首页url
...
```

> 使用举例

```
<div class="lang">语言选择：
    {rg:site}
    <a href="{$site['url']}" title="{$site['title']}" style="padding:0 10px;">{
$i}.{$site['title']}</a>
    {/rg:site}
</div>
```

栏目调用(读数据库)

> 访问数据库调用站点栏目

栏目列表

标签： `{rg:nav 参数名称="参数值"}{/rg:nav}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?pid	0	pid="2"或 pid="$nav['id']"	上级栏目ID
?model		model="product"	根据模型选栏目
?id		id="1"或 id="1,2,3,4"	根据栏目ID直接调用
?eid		eid="1"或 eid="1,2,3,4"	排除的栏目ID
?palias		palias="news"	根据上级栏目别名调用2020123101新增
?alias		alias="about,product,pic"	根据栏目别名调用2020123101新增
?order	torder asc,id desc	order="torder asc,id desc"	排序
?group	G0	group="G1"，系统自动转大写	根据栏目位置分组调用
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释

?site	当前站点	site="1"	站点ID
?limit	0	limit="4"	输出数量
?name	nav	name="nav2"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用固定字段调用对照表

```
["id"] => 栏目ID
["name"] => 栏目名称
["ename"] => 栏目副名称
["site_id"] => 栏目所属站点ID
["pid"] => 栏目上级ID
["img"] => 栏目图片
["keys"] => 栏目关键词
["desc"] => 栏目描述
["G0"] => 是否是G0栏目位置，1|0
["G1"] => 是否是G1栏目位置，1|0
["G2"] => 是否是G2栏目位置，1|0
["G3"] => 是否是G3栏目位置，1|0
["G4"] => 是否是G4栏目位置，1|0
["G5"] => 是否是G5栏目位置，1|0
["G6"] => 是否是G6栏目位置，1|0
["G7"] => 是否是G7栏目位置，1|0
["G8"] => 是否是G8栏目位置，1|0
["target"] => 链接打开方式
["create_time"] => 创建时间
["update_time"] => 修改时间
["hasChild"] => 拥有子栏目的个数
["url"] => 栏目url
["iscurrent"] => 【高亮常用】是否是当前栏目，包括所有的父层判断，true|false
```

栏目扩展字段调用对照表

```
["ext"]["*"] => 扩展字段调用标识
```

> 使用举例

一个常规的栏目列表及其二级栏目

```
// 顶级导航
{rg:nav}
```

```
// 判断是否有子栏目，判断是否属当前栏目高亮
<li class="{if $nav['hasChild']}dropdown{/if} {if $nav['iscurrent']} active{/if
}">
// 如果有子栏目则显示<i>
{if $nav['hasChild']}<i class="arr"></i>{/if}
// 输出导航名称、链接以及打开方式
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
    <div class="dropdown-box">
        // 二级导航开始，调用pid参数值为$nav['id']
        {rg:nav pid="$nav['id']"}
            // 二级导航内容
            <p><a href="{$nav['url']}">{$nav['name']}</a></p>
        // 二级导航结束
        {/rg:nav}
    </div>
</li>
// 顶级导航结束
{/rg:nav}
```

站点顶级栏目

```
{rg:nav}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

根据指定id调用栏目，pid参数失效

```
{rg:nav id="1"}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

根据多个id调用栏目，pid参数失效

```
{rg:nav id="1,2,3,4,5"}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

排除id调用栏目

```
{rg:nav eid="3,5,8"}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

根据栏目位置调用，pid参数失效

```
{rg:nav group="G2"}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

根据模型名称调用

```
{rg:nav model="news"}
<a href="{$nav['url']}" target="{$nav['target']}">{$nav['name']}</a>
{/rg:nav}
```

综合调用

```
{rg:nav pid="3" eid="7,8,9" where="img not null" order="torder desc,id asc" emp
ty="无栏目" name="lanmu" key="n"}
<a href="{$lanmu['url']}" target="{$lanmu['target']}">{$lanmu['name']}</a>
这是第{$n}个栏目
{/rg:nav}
```

根据栏目别名调用2020123101新增

```
{rg:nav palias="products" order="torder desc,id asc" empty="无栏目" name="lanmu"
 key="n"}
<a href="{$lanmu['url']}" target="{$lanmu['target']}">{$lanmu['name']}</a>
这是第{$n}个栏目
{/rg:nav}
```

栏目调用(读缓存)

> 从缓存数据中调用站点栏目
> 不支持排序，默认排序根据栏目排序由小到大

栏目列表

标签： `{rg:navi 参数名称="参数值"}{/rg:navi}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?pid	0	pid="2"或 pid="$navi['id']"	上级栏目ID
?model		model="product"	根据模型选栏目
?id		id="1"或 id="1,2,3,4"	根据栏目ID直接调用
?eid		eid="1"或 eid="1,2,3,4"	排除的栏目ID
?palias		palias="news"	根据上级栏目别名调用
?alias		alias="about,product,pic"	根据栏目别名调用
?group	G0	group="G1"，系统自动转大写	根据栏目位置分组调用
?where		where="'istop','=','a'|'istuijian','=','b'"，与nav标签where方法不一样，多条件用|分割	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?site	当前站点	site="1"	站点ID
?limit	0	limit="4"	输出数量
?name	navi	name="navi2"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用固定字段调用对照表

```
["id"] => 栏目ID
["name"] => 栏目名称
["ename"] => 栏目副名称
["site_id"] => 栏目所属站点ID
["pid"] => 栏目上级ID
["img"] => 栏目图片
["keys"] => 栏目关键词
["desc"] => 栏目描述
["G0"] => 是否是G0栏目位置，1|0
["G1"] => 是否是G1栏目位置，1|0
["G2"] => 是否是G2栏目位置，1|0
["G3"] => 是否是G3栏目位置，1|0
["G4"] => 是否是G4栏目位置，1|0
["G5"] => 是否是G5栏目位置，1|0
["G6"] => 是否是G6栏目位置，1|0
["G7"] => 是否是G7栏目位置，1|0
["G8"] => 是否是G8栏目位置，1|0
["target"] => 链接打开方式
["create_time"] => 创建时间
["update_time"] => 修改时间
["hasChild"] => 拥有子栏目的个数
["url"] => 栏目url
["iscurrent"] => 【高亮常用】是否是当前栏目，包括所有的父层判断，true|false
```

栏目扩展字段调用对照表

```
["ext"]["*"] => 扩展字段调用标识
```

> 使用举例

一个常规的栏目列表及其二级栏目

```
// 顶级导航
{rg:navi}
// 判断是否有子栏目，判断是否属当前栏目高亮
<li class="{if $navi['hasChild']}dropdown{/if} {if $navi['iscurrent']} active{/
if}">
// 如果有子栏目则显示<i>
{if $navi['hasChild']}<i class="arr"></i>{/if}
// 输出导航名称、链接以及打开方式
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
    <div class="dropdown-box">
        // 二级导航开始，调用pid参数值为$nav['id']
        {rg:navi pid="$navi['id']"}
            // 二级导航内容
            <p><a href="{$navi['url']}">{$navi['name']}</a></p>
        // 二级导航结束
        {/rg:navi}
    </div>
</li>
// 顶级导航结束
{/rg:navi}
```

站点顶级栏目

```
{rg:navi}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

根据指定id调用栏目，pid参数失效

```
{rg:navi id="1"}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

根据多个id调用栏目，pid参数失效

```
{rg:navi id="1,2,3,4,5"}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

排除id调用栏目

```
{rg:navi eid="3,5,8"}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

根据栏目位置调用，pid参数失效

```
{rg:navi group="G2"}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

根据模型名称调用

```
{rg:navi model="news"}
<a href="{$navi['url']}" target="{$navi['target']}">{$navi['name']}</a>
{/rg:navi}
```

综合调用

```
{rg:navi pid="3" eid="7,8,9" empty="无栏目" name="lanmu" key="n"}
<a href="{$lanmu['url']}" target="{$lanmu['target']}">{$lanmu['name']}</a>
这是第{$n}个栏目
{/rg:navi}
```

根据栏目别名调用

```
{rg:navi palias="products" empty="无栏目" name="lanmu" key="n"}
<a href="{$lanmu['url']}" target="{$lanmu['target']}">{$lanmu['name']}</a>
这是第{$n}个栏目
{/rg:navi}
```

内容列表

> 内容列表的调用

内容列表

标签： `{rg:list 参数名称="参数值"}{/rg:list}`

> 适用页面：任何

> 标签属性：循环

> list标签只调用内容模型的内容，其他模型内容无法调用

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?model		model="product"	根据模型选数据
?nid		nid="2"或nid="$nav_id"	根据一个栏目ID列内容列表
?nids		nids="1,2,3,4"	根据多个栏目ID列内容列表
?enid		enid="1,2,3,4"	排除的栏目ID后的内容列表
?nalias		nalias="news"	根据栏目别名调用2020123101新增
?naliases		naliases="product1,product2,product3"	根据多个栏目别名调用2020123101新增
?ealias		ealias="product1"	要排除的栏目别名2020123101新增
?limit	10	limit="4"	每页数量
?total	false	total="1000"	输出总量
?order	create_time desc	order="id desc"，当order="rand"为随机	排序，默认时间从近到远

?ispage	false	ispage="true"	开启分页，一个页面只允许一个分页
?diypage	false	diypage="true"	是否启用自定义分页，需ispage=true，用法直达自定义分页
?timing	false	timing="true"	开启定时内容，默认关闭
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?name	list	name="list2"	变量名称
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
自定义的字段
["*"] => * 代表自定义的字段名

固定字段
["id"] => 内容ID
["title"] => 内容标题
["navlist_id"] => 所属栏目ID
["link"] => 内容外链
["author"] => 作者
["source"] => 来源
["img"] => 单图
["imgs"] => 多图
["keys"] => 内容关键词
["desc"] => 内容描述
["hits"] => 内容点击量
["content"] => 详细内容
["istop"] => 是否置顶，1|0
["target"] => 内容页打开方式
["site_id"] => 内容所属站点ID
["torder"] => 内容排序
```

```
["create_time"] => 添加时间
["update_time"] => 编辑时间
["delete_time"] => 删除时间
["url"] => 当前内容的url

当前内容的栏目信息
["navlist"]["id"] => 栏目ID
["navlist"]["name"] => 栏目名称
["navlist"]["ename"] => 栏目副名称
["navlist"]["pid"] => 栏目上级ID
["navlist"]["img"] => 栏目单图
["navlist"]["keys"] => 栏目关键词
["navlist"]["desc"] => 栏目描述
["navlist"]["target"] => 栏目的打开方式
["navlist"]["hasChild"] => 拥有子栏目的个数
["navlist"]["url"] => 栏目url

当前内容的模型信息
["navlist"]["cmsmodels"]["id"] => 模型ID
["navlist"]["cmsmodels"]["title"] => 模型名称，如产品
["navlist"]["cmsmodels"]["name"] => 模型标识，如product
["navlist"]["cmsmodels"]["tablename"] => 模型表名
["navlist"]["cmsmodels"]["classname"] => 模型类名
["navlist"]["cmsmodels"]["attamodel_id"] => 模型附属表ID
```

使用举例

一个常规的内容列表调用

```
// 调用6条product模型中的内容
{rg:list model="product" limit="6"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
```

输出当前栏目页的列表，当栏目为内容模型时

```
// 栏目列表页中当前栏目变量为$nav_id
{rg:list nid="$nav_id" limit="6" ispage="true"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
```

```
{/rg:list}
// 分页，注意一个页面中只能有一个分页，只能有一个ispage="true"
{$page|raw}
```

或者

```
// 当前栏目页内容列表，nid可省略
{rg:list limit="6" ispage="true"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
// 分页，注意一个页面中只能有一个分页，只能有一个ispage="true"
{$page|raw}
```

根据多个栏目ID调用内容列表，如产品

```
// 调用product模型中栏目属于3,5,6,8,9的内容，输出6条
{rg:list nids="3,5,6,8,9" model="product" limit="6"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
```

综合调用（内置分页），如产品

```
// 输出当前栏目页内容，分页，6条一页，排除4,5栏目id，定时内容开启，筛选出istop为1的内容，调用变量为prolist
{rg:list nid="$nav_id" limit="6" ispage="true" enid="4,5" timing="true" where="istop = 1" name="prolist"}
<li>
    <img src="{$prolist['img']}">
    <a href="{$prolist['url']}"  title="{$prolist['title']}">{$prolist['title']}</a>
    所属栏目：<a href="{$prolist['navlist']['url']}">{$prolist['navlist']['name']}</a>
</li>
{/rg:list}
// 分页，注意一个页面中只能有一个分页，只能有一个ispage="true"
{$page|raw}
```

综合调用（自定义分页），如产品

```
// 输出当前栏目页内容，自定义分页，6条一页，排除4,5栏目id，定时内容开启，筛选出istop为1的内容，调用变量为prolist
{rg:list nid="$nav_id" limit="6" ispage="true" diypage="true" enid="4,5" timing="true" where="istop = 1" name="prolist"}
<li>
    <img src="{$prolist['img']}">
    <a href="{$prolist['url']}"  title="{$prolist['title']}">{$prolist['title']}</a>
    所属栏目：<a href="{$prolist['navlist']['url']}">{$prolist['navlist']['name']}</a>
</li>
{/rg:list}
// 分页，只有list开启ispage=true 自定义分页才有效
<div class="pages">
    <ul>
    <li><a href="{$page['homeUrl']}">首页</a></li>
    <li><a href="{$page['prevUrl']}">上一页</a></li>
    {rg:diypage limit="2"}
    <li {if $diypage['iscurrent']}class="thisclass"{/if}><a href="{$diypage['url']}">第{$diypage['page']}页</a></li>
    {/rg:diypage}
    <li><a href="{$page['nextUrl']}">下一页</a></li>
    <li><a href="{$page['lastUrl']}">尾页</a></li>
    </ul>
</div>
```

2020123101新增

根据栏目别名调用内容列表，如产品

```
{rg:list naliases="product1,product2,product3" limit="6"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
```

```
{rg:list nalias="news" limit="6"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
```

指定模型内容

> 根据模型和内容ID指定调用内容

指定模型的内容

标签： `{rg:content 参数名称="参数值"}{/rg:content}`

> 适用页面：任何

> 标签属性：循环(无分页，无where，无limit，无url)，默认排序为torder由小到大

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
id	必选	id="4"或 id="1,2,3,4,5"	内容ID
model	必选	model="product"	模型标识

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?name	content	name="content"	变量名称
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示

> 使用举例

一个常规的指定模型内容调用

```
{rg:content id="1" model="product"}
{$content['id']}
{$content['title']}
{$content['img']}
{$content['keys']}
...
```

```
{/rg:content}
```

自定义分页

> 内容列表自定义分页

自定义分页

固定标签：

标签	解释
{$page['homeUrl']}	第一页链接
{$page['lastUrl']}	最后一页链接
{$page['prevUrl']}	上一页链接
{$page['nextUrl']}	下一页链接
{$page['total']}	所有内容总数
{$page['currentPage']}	当前页码
{$page['lastPage']}	最后一页页码

页码循环标签： `{rg:diypage 参数名称="参数值"}{/rg:diypage}`

> 适用页面：list标签外部，且list标签开启ispage=true且diypage=true

> 标签属性：循环

> 【报错提醒】一个页面只允许一个分页开启的list

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?limit	0	limit="2"	分页里显示（2*limit+1）个页码，为0或不设置时为所有页码

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?name	diypage	name="diypage"	变量名称

?key	i	key="i"	循环内计数变量
?empty		empty=""	无结果提示

字段调用对照表

```
页码循环标签内固定字段
["page"] => 页码
["url"] => 页链接
["iscurrent"] => 是否是当前页，高亮用
```

调用（内置分页），如产品

```
// 输出当前栏目页内容，内置分页，6条一页
{rg:list nid="$nav_id" limit="6" ispage="true" }
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
// 分页，注意一个页面中只能有一个分页，只能有一个ispage="true"
{$page|raw}
```

调用（自定义分页），如产品

```
// 输出当前栏目页内容，自定义分页，6条一页
{rg:list nid="$nav_id" limit="6" ispage="true" diypage="true"}
<li>
    <img src="{$list['img']}">
    <a href="{$list['url']}"  title="{$list['title']}">{$list['title']}</a>
    所属栏目：<a href="{$list['navlist']['url']}">{$list['navlist']['name']}</a>
</li>
{/rg:list}
// 分页，只有list开启ispage=true 自定义分页才有效
<div class="pages">
    <ul>
    <li><a href="{$page['homeUrl']}">首页</a></li>
    <li><a href="{$page['prevUrl']}">上一页</a></li>
    {rg:diypage limit="2"}
    <li {if $diypage['iscurrent']}class="thisclass"{/if}><a href="{$diypage['ur
l']}">第{$diypage['page']}页</a></li>
    {/rg:diypage}
    <li><a href="{$page['nextUrl']}">下一页</a></li>
    <li><a href="{$page['lastUrl']}">尾页</a></li>
    </ul>
</div>
```

面包屑

面包屑属于非循环类标签，用于显示当前页面导航信息

面包屑

标签： `{rg:location 参数名称="参数值" /}`

适用页面：栏目页、搜索页

标签属性：闭合非循环

参数列表

参数(?为可选参数)	默认值	使用示例	解释
?mark	>	mark=">"	面包屑分割标记
?home	当前模板语言包定义的Home值	home="网站首页"	首页的名称
?msg	页面信息	msg="搜索记录"	非栏目页面时显示的名称，如在搜索页中

使用举例

```
{rg:location /}
{rg:location home="网站首页" mark="-" /}
```

功能调用

> 功能内容列表的调用

功能内容列表

标签： `{rg:func 参数名称="参数值"}{/rg:func}`

> 适用页面：任何

> 标签属性：循环

> func标签只调用功能模型的内容，其他模型内容无法调用

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
name	必选	name="slide"	功能模型名称
?limit	10	limit="4"	输出数量
?order	torder asc,id desc	order="id desc"，当order="rand"为随机	排序，默认按排序从小到大
?ispage	false	ispage="true"	开启分页，一个页面只允许一个分页
?diypage	false	diypage="true"	是否启用自定义分页，需ispage=true，用法直达自定义分页
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?key	i	key="i"	循环内计数变量

?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
自定义的字段
["*"] => * 代表自定义的字段名

固定字段
["id"] => 内容ID
["torder"] => 排序数字
```

> 使用举例

一个常规的功能调用

```
如幻灯片模型标识为slide
// link为自定义的链接字段名
// img为自定义的图片字段名
{rg:func name="slide"}
<a href="{$slide['link']}" class="swiper-slide" style="background-image: url({$
slide['img']});"></div>
{/rg:func}
```

附属表内容

> 附属内容的调用

附属内容列表

标签： `{rg:atta 参数名称="参数值"}{/rg:atta}`

> 适用页面：任何

> 标签属性：循环

> 概念：附属内容是指，一条内容还有属于这条内容的附属内容。如：一个小说里面还包含很多章节。
> atta标签只调用内容模型和功能模型的内容的附属内容，其他模型内容无法调用

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?model	详情页中默认为当前内容模型的ID model为ID数字：自动查找该模型所关联的附属模型 model为附属模型标识时：直接调用附属模型内容	model="1"：调用模型ID为1所关联的附属模型 model="productatta"：直接调用附属模型标识为productatta的模型	根据模型选数据
?cid	详情页中默认为当前内容ID	cid="2"或cid="$info_id"	查找内容id为cid的内容的附属列表
?limit	10	limit="4"	输出数量
?order	create_time desc	order="id desc"，当order="rand"为随机	排序，默认时间从近到远
?ispage	false	ispage="true"	开启分页，一个页面只允许一个分页
?diypage	false	diypage="true"	是否启用自定义分页，需ispage=true，用

			法直达自定义分页
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?name	atta	name="atta2"	变量名称
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
自定义的字段
["*"] => * 代表自定义的字段名

固定字段
["id"] => 内容ID
["hits"] => 点击量
["create_time"] => 添加时间
["url"] => 当前附属内容详情页的url
```

> 使用举例

一个常规的附属表调用

```
// 功能模型为block
// 调用该功能模型的每个记录的附属内容列表
{rg:func name="block"}
<div class="section bg1">
  <div class="container">
    <div class="section-title wow fadeInUp"><small>{$block['title1']}</small>{$
block['title2']}</div>
    <ul class="in-jjfa clearfix">
        // 这里开始调用每个记录的附属内容
         {rg:atta model="blockatta" cid="$block['id']" order="id asc"}
            <li class="wow fadeInUp"> <a href="{$atta['url']}">
            <div class="img-cover"><span style="background-image: url({$atta['i
mg']});"></span>
              <p>{$atta['title']}</p>
            </div>
```

```
            <div class="text">
              {$atta['content']|raw}
              <span class="more">{:lang('More')}</span> </div>
            </a> </li>
          {/rg:atta}

    </ul>
  </div>
</div>
{/rg:func}
```

单页内容列表

> 单页内容列表的调用和单页内容调用

单页内容列表

标签： `{rg:about 参数名称="参数值"}{/rg:about}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?id		id="1"或 id="1,2,3,4,5"	调用单页ID或多个ID
?nid		nid="1"	根据栏目ID调用，栏目必须为单页属性
?nalias		nalias="news"	根据栏目别名调用 2020123101新增
?limit	10	limit="4"	输出数量
?order	torder asc,create_time desc	order="id desc"	排序，默认按排序从小到大

> id和nid参数必须只能调用一个，id支持多个单页id，nid只调用一个栏目id

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?name	about	name="about"	变量名称
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示

?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
固定字段
["id"] => 单页ID
["title"] => 标题
["etitle"] => 副标题
["author"] => 作者
["source"] => 来源
["img"] => 图片
["keys"] => 关键词
["desc"] => 描述
["hits"] => 点击量
["content"] => 单页内容
["create_time"] => 创建时间
["update_time"] => 修改时间

自定义的动态字段
["ext"]["*"] => * 代表自定义的字段名
```

> 使用举例

一个常规的单页内容调用

```
// 调用栏目id为20且栏目类型为about的单页内容
{rg:about nid="20"}
    <img src="{$about.img}" alt="">
    <h4>{$about.title}<small>{$about.etitle}</small></h4>
    <p>{$about.content|totxt|cut_str=180}</p>
{/rg:about}
```

调用多个单页的列表

```
// 调用单页id为1,2,3,4,5
{rg:about id="1,2,3,4,5"}
    <li>{$about.title}
    <p>{$about.content|totxt|cut_str=180}</p>
    </li>
{/rg:about}
```

2020123101新增

根据栏目别名调用单页内容可输出当前栏目url

```
{rg:about nalias="news"}
    <img src="{$about.img}" alt="">
    <h4>{$about.title}<small>{$about.etitle}</small></h4>
    <p>{$about.content|totxt|cut_str=180}</p>
    <a href="{$about.url}">{$about.title}</a>
{/rg:about}
```

多图字段

> 用于解析字段类型为多图的内容，将多图字段内容循环解析

多图字段内容解析

标签： `{rg:imgs 参数名称="参数值"}{/rg:imgs}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
imgs	必选	如：imgs="$info_imgs"	多图字段内容
?default_src	0	default_src="$info_img"	如果多图字段为空时的默认图片地址
?default_title	0	default_title="图片标题"	如果多图字段为空时的默认图片名称

> 【报错提示】调用时，请对应好多图字段。详情页中一般为$info_多图字段标识，如果在循环标签内可能为$list['多图字段标识']，list随循环体内变量名称而改变

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?name	imgs	name="imgs"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示

多图字段调用对照表

```
固定字段
["src"] => 图片地址
["title"] => 图片标题
```

> 使用举例

一个常规的多图调用

```
// 当多图字段imgs内容为空时，则显示img缩略图内容
{rg:imgs imgs="$info_imgs" default_src="$info_img"}
<div class="swiper-slide img-center">
<div class="imgauto"><img src="{$imgs['src']}" alt="{$imgs['title']}" rel="{$im
gs['src']}" class="jqzoom" title="{$imgs['title']}"></div>
</div>
{/rg:imgs}
```

表单内容列表

> 表单内容列表的调用

表单内容列表

标签： `{rg:form 参数名称="参数值"}{/rg:form}`

> 适用页面：任何

> 标签属性：循环

> form标签只调用表单模型的内容，其他模型内容无法调用

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
model	必选	model="feedback"或model="20"	表单模型标识或ID
?limit	10	limit="4"	输出数量
?order	create_time desc	当order="rand"为随机	排序，默认按时间从近到远
?where		where="istop=1"，符合sql语句	附加条件
?ispage	false	ispage="true"	开启分页，一个页面只允许一个分页
?diypage	false	diypage="true"	是否启用自定义分页，需ispage=true，用法直达自定义分页

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?key	i	key="i"	循环内计数变量

?name	form	name="form2"	变量名称
?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
自定义的字段
["*"] => * 代表自定义的字段名

固定字段
["id"] => 内容id
["pid"] => 上级内容id
["user_ip"] => ip记录
["user_id"] => 会员id，默认为0
["reply"] => 管理回复内容
["create_time"] => 创建时间
["update_time"] => 回复时间或修改时间
["adminview"] => 管理员是否查看：0未1已
```

使用举例

```
调用一个表单ID为20的表单数据
<ul>
    {rg:form model="20"}
        留言者IP：{$form.user_ip}
        回复内容：{$form.reply}
        留言时间：{$form.create_time}
        留言字段信息：{$form.*} * 为表单自定义字段
    {/rg:form}
</ul>
```

如果是登陆用户，还可以调用

```
留言者用户名：{$form.user.username|default="匿名"}
留言者昵称：{$form.user.nickname}
留言者性别：{$form.user.sex}
留言者头像：{$form.user.face}
```

字符串分割

> 用标签的形式进行字符串按照指定的分隔符号分割成数组，然后循环输出

字符串分割

标签： `{rg:each 参数名称="参数值"}{/rg:each}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
str	必选	$str="1,2,3,4,5"	需要分割的字符串
?mark	|	mark="|"	分割符号
?order	left	order="right"	输出顺序，left从左开始，right从右开始

> id和nid参数必须只能调用一个，id支持多个单页id，nid只调用一个栏目id

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?key	i	key="i"	循环内计数变量
?limit		limit="2"	输出记录数
?offset		offset="2"	从第几个开始输出

> 使用举例

一个常规的字符串内容分割调用

```
假设某自定义字段为language，值为 php|asp|java|python
<ul>
{rg:each str="$info_language" mark="|" order="right"}
    <li class="show{$i}">{$each}</li>
```

```
{/rg:each}
</ul>
说明：
1. 从右往左显示分割数据
2. {$i} 从1开始计数
```

搜索页列表

> 只用在搜索页中的列表标签

搜索页结果列表

标签： `{rg:search 参数名称="参数值"}{/rg:search}`

> 适用页面：栏目内容模型的搜索页，其他模型无效

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?limit	10	limit="4"	每页输出数量
?order	create_time desc	order="id desc"	排序，默认时间从近到远
?ispage	false	ispage="true"	开启分页，一个页面只允许一个分页
?diypage	false	diypage="true"	是否启用自定义分页，需ispage=true，用法直达自定义分页
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?name	search	name="search"	变量名称
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用搜索结果字段调用对照表

```
自定义的字段
["*"] => * 代表自定义的字段名

固定字段
["id"] => 内容ID
["title"] => 内容标题
["navlist_id"] => 所属栏目ID
["link"] => 内容外链
["author"] => 作者
["source"] => 来源
["img"] => 单图
["imgs"] => 多图
["keys"] => 内容关键词
["desc"] => 内容描述
["hits"] => 内容点击量
["content"] => 详细内容
["istop"] => 是否置顶, 1|0
["target"] => 内容页打开方式
["site_id"] => 内容所属站点ID
["torder"] => 内容排序
["create_time"] => 添加时间
["update_time"] => 编辑时间
["delete_time"] => 删除时间
["url"] => 当前内容的url

当前搜索结果的栏目信息
["navlist"]["id"] => 栏目ID
["navlist"]["name"] => 栏目名称
["navlist"]["ename"] => 栏目副名称
["navlist"]["pid"] => 栏目上级ID
["navlist"]["img"] => 栏目单图
["navlist"]["keys"] => 栏目关键词
["navlist"]["desc"] => 栏目描述
["navlist"]["target"] => 栏目的打开方式
["navlist"]["hasChild"] => 拥有子栏目的个数
["navlist"]["url"] => 栏目url
```

> 使用举例

一个常规的搜索结果列表调用

```
{rg:search ispage="true" limit="6"}
    <li><a href="{$search['url']}" title="{$search['title']}">
    <div class="date"><strong>{$search['create_time']|date="d"}</strong>{$searc
h['create_time']|date="Y-m"}</div>
```

```
    <div class="text">
    <h4>{$search['title']}</h4>
    <p>{$search['navlist']['name']}</p>
    </div>
    </a></li>
{/rg:search}
// 分页，注意一个页面中只能有一个分页，只能有一个ispage="true"
{$page|raw}
```

地区调用

地区列表

地区列表

标签： `{rg:region 参数名称="参数值"}{/rg:region}`

适用页面：任何

标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?pid	0	pid="$region['id']"	地区父ID，默认由省级开始

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?name	region	name="region"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示

常用搜索结果字段调用对照表

```
固定字段
["id"] => 地区ID
["name"] => 地区名称
["parent_id"] => 上级ID
["initial"] => 地区拼音首字母
["pinyin"] => 地区拼音
["citycode"] => 城市区号
["adcode"] => 地区编号
["lng_lat"] => 经纬度
```

使用举例

列出所有省级

```
{rg:region}{$region['name']}{/rg:region}
```

关键词列表

> 站点关键词列表

关键词列表

标签： `{rg:sitekey 参数名称="参数值"}{/rg:sitekey}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?limit	0	limit="4"	每页输出数量
?order	torder asc	order="id desc"	排序，默认排序从小到大
?where		where="istop=1"，符合sql语句	附加条件

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点	siteid="1"	站点ID
?name	sitekey	name="sitekey"	变量名称
?key	i	key="i"	循环内计数变量
?empty		empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用搜索结果字段调用对照表

```
固定字段
["id"] => 关键词ID
["name"] => 关键词名称
["link"] => 关键词链接
["istop"] => 是否推荐, 0|1
["create_time"] => 创建日期
["update_time"] => 更新日期
```

使用举例

一个常规的关键词列表调用

```
{rg:sitekey where="istop=1"}<a href="{$sitekey['link']}">{$sitekey['name']}</a>
{/rg:sitekey}
```

属性列表

> 调用系统设定的属性内容

指定属性列表

标签： `{rg:attr 参数名称="参数值"}{/rg:attr}`

> 适用页面：任何

> 标签属性：循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
name	必选	name="yanse"	属性标识

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?order	torder asc,id desc	order="id desc"	排序，默认根据排序由小到大
?field		field="title"	需要调用的字段
?value		value="红色,白色"	调用的字段的值
?limit	0	limit="4"	输出数量
?key	i	key="i"	循环内计数变量
?empty	未查询到相关记录！	empty="无记录"	无结果提示
?offset		offset="2"	从第几个开始输出

常用字段调用对照表

```
["id"] => 属性内容id
["title"] => 属性内容名称
["color"] => 属性颜色代码
["img"] => 属性图片地址
["torder"] => 属性排序号
```

```
...
```

> 使用举例

一个常规的指定属性调用

```
{rg:attr name="yanse"}
{$attr['id']}
{$attr['title']}
{$attr['img']}
{$attr['color']}
...
{/rg:attr}
```

内容统计

> 计算栏目内容或模型内容数量

内容数量统计

标签：`{rg:count 参数名称="参数值" /}

> 适用页面：任何

> 标签属性：非循环

常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?nid	当前栏目ID	pid="1"或pid="$nav['id']"	统计当前栏目内容数量
?model		model="product"	统计模型内容数量

不常用参数列表

参数(?为可选参数)	默认值	使用示例	解释
?siteid	当前站点ID	siteid="1"	站点ID
?where		where="istop=1"，符合sql语句	附加条件

> 使用举例

当前栏目的内容数量

```
{rg:count /}
```

栏目列表中使用

```
{rg:count nid="$navi['id']"/}
```

统计news模型中置顶的内容数量

```
{rg:count model="news" where="istop=1"/}
```

比较判断

常用比较

常用判断

常用比较

> RGCMS2.0 内核采用THINKPHP5.1.*开发，因此也集成了TP的所有比较标签，具体了解请移步：https://www.kancloud.cn/manual/thinkphp5_1/354085

比较标签用于简单的变量比较，复杂的判断条件可以用if标签替换，比较标签是一组标签的集合，基本上用法都一致，如下：

```
{比较标签 name="变量" value="值"}
内容
{/比较标签}
```

系统支持的比较标签以及所表示的含义分别是：

标签	含义
eq或者 equal	等于
neq 或者notequal	不等于
gt	大于
egt	大于等于
lt	小于
elt	小于等于
heq	恒等于
nheq	不恒等于

他们的用法基本是一致的，区别在于判断的条件不同，并且所有的比较标签都可以和else标签一起使用。

例如，要求name变量的值等于value就输出，可以使用：

```
{eq name="name" value="value"}value{/eq}
```

或者

```
{equal name="name" value="value"}value{/equal}
```

也可以支持和else标签混合使用：

```
{eq name="name" value="value"}
相等
{else/}
```

```
不相等
{/eq}
```

当 name变量的值大于5就输出

```
{gt name="name" value="5"}value{/gt}
```

当name变量的值不小于5就输出

```
{egt name="name" value="5"}value{/egt}
```

比较标签中的变量可以支持对象的属性或者数组，甚至可以是系统变量，例如： 当vo对象的属性（或者数组，或者自动判断）等于5就输出

```
{eq name="vo.name" value="5"}
{$vo.name}
{/eq}
```

当vo对象的属性等于5就输出

```
{eq name="vo:name" value="5"}
{$vo.name}
{/eq}
```

当$vo['name']等于5就输出

```
{eq name="vo['name']" value="5"}
{$vo.name}
{/eq}
```

而且还可以支持对变量使用函数 当vo对象的属性值的字符串长度等于5就输出

```
{eq name="vo:name|strlen" value="5"}{$vo.name}{/eq}
```

变量名可以支持系统变量的方式，例如：

```
{eq name="Think.get.name" value="value"}相等{else/}不相等{/eq}
```

通常比较标签的值是一个字符串或者数字，如果需要使用变量，只需要在前面添加“$”标志： 当vo对象的属性

等于$a就输出

```
{eq name="vo:name" value="$a"}{$vo.name}{/eq}
```

所有的比较标签可以统一使用compare标签（其实所有的比较标签都是compare标签的别名），例如： 当name变量的值等于5就输出

```
{compare name="name" value="5" type="eq"}value{/compare}
```

等效于

```
{eq name="name" value="5" }value{/eq}
```

其中type属性的值就是上面列出的比较标签名称

常用判断

> RGCMS2.0 内核采用THINKPHP5.1.*开发，因此也集成了TP的所有条件判断标签，具体了解请移步
> https://www.kancloud.cn/manual/thinkphp5_1/354086

SWITCH标签

用法：

```
{switch 变量 }
    {case value1 }输出内容1{/case}
    {case value2}输出内容2{/case}
    {default /}默认情况
{/switch}
```

使用示例：

```
{switch User.level}
    {case 1}value1{/case}
    {case 2}value2{/case}
    {default /}default
{/switch}
```

可以使用函数以及系统变量，例如：

```
{switch User.level|intval }
    {case 1}admin{/case}
    {default /}default
{/switch}
```

对于case属性可以支持多个条件的判断，使用”|”进行分割，例如：

```
{switch Think.get.type}
    {case gif|png|jpg}图像格式{/case}
    {default /}其他格式
{/switch}
```

表示如果 `$_GET["type"]` 是gif、png或者jpg的话，就判断为图像格式。

也可以对case的value属性使用变量，例如：

```
{switch $User.userId}
```

```
    {case $adminId}admin{/case}
    {case $memberId}member{/case}
{/switch}
```

> 使用变量方式的情况下，不再支持 `|` 分割的多个条件判断用法。

IF标签

用法：

```
{if 表达式}value1
{elseif 表达式 /}value2
{else /}value3
{/if}
```

用法示例：

```
{if ( $name == 1) OR ( $name > 100) } value1
{elseif $name == 2 /}value2
{else /} value3
{/if}
```

范围判断

范围判断标签包括 `in` / `notin` / `between` / `notbetween` 四个标签，都用于判断变量是否中某个范围。

IN和NOTIN

用法： 假设id赋值为1：

```
$id =    1;
```

我们可以使用in标签来判断模板变量是否在某个范围内，例如：

```
{in name="id" value="1,2,3"}
id在范围内
{/in}
```

最后会输出： `id在范围内` 。

如果判断不在某个范围内，可以使用notin标签：

```
{notin name="id" value="1,2,3"}
id不在范围内
{/notin}
```

最后会输出： `id不在范围内` 。

可以把上面两个标签合并成为：

```
{in name="id" value="1,2,3"}
id在范围内
{else/}
id不在范围内
{/in}
```

value属性也可以使用变量，例如：

```
{in name="id" value="$range"}
id在范围内
{/in}
```

$range变量可以是数组，也可以是以逗号分隔的字符串。

BETWEEN 和 NOTBETWEEN

可以使用 `between` 标签来判断变量是否在某个区间范围内，可以使用：

```
{between name="id" value="1,10"}
输出内容1
{/between}
```

同样，也可以使用 `notbetween` 标签来判断变量不在某个范围内：

```
{notbetween name="id" value="1,10"}
输出内容2
{/notbetween}
```

也可以使用 `else` 标签把两个用法合并，例如：

```
{between name="id" value="1,10"}
输出内容1
{else/}
输出内容2
{/between}
```

当使用 `between` 标签的时候， `value` 只需要一个区间范围，也就是只支持两个值，后面的值无效，例如

```
{between name="id" value="1,3,10"}
输出内容1
{/between}
```

实际判断的范围区间是 `1~3` ，而不是 `1~10` ，也可以支持字符串判断，例如：

```
{between name="id" value="A,Z"}
输出内容1
{/between}
```

value属性也可以使用变量，例如：

```
{between name="id" value="$range"}
输出内容1
{/between}
```

EMPTY/NOTEMPTY 标签

empty标签用于判断某个变量是否为空，用法：

```
{empty name="name"}
name为空值
{/empty}
```

如果判断没有赋值，可以使用：

```
{notempty name="name"}
name不为空
{/notempty}
```

可以把上面两个标签合并成为：

```
{empty name="name"}
name为空
{else /}
name不为空
{/empty}
```

模板开发

模板目录结构
路径标签
模板包含
模板命名
语言包
搜索表单
自定义表单

模板目录结构

> RGCMS2.0 目录结构

模板文件夹目录

> 假设默认模板文件夹template，当前主题为demo
> /public/template/demo/ 目录下的结构为：

- html（模板HTML文件存放的文件夹，自定义）
  - index_index.html -- 首页
  - head.html -- 头部
  - foot.html -- 页尾
  - left.html -- 左侧
  - about_content.html -- 默认单页模板
  - product_list.html -- 默认product模型列表页
  - product_content.html -- 默认product详情页
  - news_list.html -- 默认news模型列表页
  - news_content.html -- 默认news详情页
  - search_index.html -- 默认搜索页
  - feedback_content.html -- 自定义单页
  - contact_content.html -- 自定义单页
  - ……
- lang（语言包）
  - ch.lang.php -- 站点标识为ch的语言包
  - en.lang.php -- 站点标识为en的语言包
  - jp.lang.php -- 站点标识为jp的语言包
  - ……
- static（静态资源文件文件夹如css，js，images，font等，非要求）
  - css （非标准）
  - js（非标准）
  - images（非标准）

路径标签

> 系统默认模板目录位于/public/下的template文件夹（可后台配置），灵活配置你的模板文件夹，能有效的避免一些不法的猜测

> 站点模板目录标签为{$tempdir}

标签使用

```
<link rel="stylesheet" type="text/css" href="{$tempdir}/static/css/public.css"/
>
<link rel="stylesheet" type="text/css" href="{$tempdir}/static/css/jquery.bxsli
der.css"/>
<link rel="stylesheet" type="text/css" href="{$tempdir}/static/css/style.css"/>
<script type="text/javascript" src="{$tempdir}/static/js/jquery1.11.3.min.js"><
/script>
<script type="text/javascript" src="{$tempdir}/static/js/jquery.bxslider.min.js
"></script>
<script type="text/javascript" src="{$tempdir}/static/js/index.js"></script>
```

模板包含

> RGCMS2.0的模板包含使用方便直观，使得建站快捷，修改方便

包含调用

标签： `{include file="*" /}`

* 代表要包含的文件名，注意不带后缀

> 例如：

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

表示包含了当前站点模板HTML文件目录下的head.html文件

```
{include file="left" /}
```

表示包含了当前站点模板HTML文件目录下的left.html文件

```
{include file="foot" /}
```

表示包含了当前站点模板HTML文件目录下的foot.html文件

模板命名

> RGCMS2.0的模板默认有自己的命名方式，当然你也可以使用自己的个性命名

模板命名

首页：后台配置的首页模板名称.html `默认为index_index.html`

单页：默认为：about_content.html，如：自定义命名_content.html

列表：默认为：栏目模型标识_list.html，如：news_list.html或自定义命名_list.html

详情页：默认为：栏目模型标识_content.html，如：news_content.html或自定义命名_content.html

搜索：search_模型标识.html，如不存在则默认为search_index.html，如：search_news.html，模型标识根据搜索提交的参数相关

表单：一般表单使用单页制作，只需在单页模板中嵌入表单即可，如命名为feedback_content.html制作在线留言

其他模板

如：

head.html

top.html

foot.html

nav.html

left.html

...

一般用于被包含

语言包

> RGCMS2.0模板新增语言包功能，语言包位于模板目录下lang文件夹内，系统自动识别并加载

> 语言包作用：一般用于制作多语言网站，设定页面上一些固定的词句，在不同的语言下转换为不同的语言。

语言包命名及模板文件夹目录

语言包命名：站点标识.lang.php

如当前站点为中英日三语版，对应站点标识为ch，en，jp

则语言包有三个：

ch.lang.php

en.lang.php

jp.lang.php

文件格式

```php
<?php
return [
    // 站点信息
    'Tel'                       => '电话',
    'Mobile'                    => '手机号',
    'Fax'                       => '传真',
    'Email'                     => '邮箱',
    'Address'                   => '地址',
    'Wechat'                    => '微信',
    'Website'                   => '网址',
    'Contact person'            => '联系人',

    // 标题
    'Home'                      => '首页',
    'Product'                   => '产品',
    'News'                      => '新闻',
    'About'                     => '关于',
    'Contact'                   => '联系',
    'Contact us'                => '联系我们',
    'Sitemap'                   => '网站地图',
    'Recommended'               => '产品推荐',
    'Description'               => '详细内容',
    'Related'                   => '相关内容',

    // 页面文字
    'Addtime'                   => '发布日期',
    'Source'                    => '来源',
    'Hits'                      => '点击量',
    'Prev'                      => '上一条',
    'Next'                      => '下一条',
    'Keywords'                  => '关键词',
    'Location'                  => '当前位置',
    'Query'                     => '在线咨询',
    'More'                      => '查看更多',
    'View'                      => '查看更多',
    'Links'                     => '友情链接',
    'Online service'            => '在线客服',
    'Search'                    => '搜索',
    'Submit'                    => '提交',
    'Reset'                     => '重置',
    'Hot keywords'              => '热门关键词',
    'Please enter key words'=> '请输入关键字',
    'Submission Failed'         => '提交失败：请检查信息是否完整，格式是否正确！',
    'Submitted successfully'=> '提交成功！'
];
```

模板中调用

使用{:lang('标识')}进行语言包的调用

如：

{:lang('Tel')}，在ch站点中显示为“电话”，如果在jp.lang.php中设置了

```
'Tel' => '電話'
```

则在jp标识的站点中，显示为“電話”

如果未找到对应语言包，则默认显示"Tel"

搜索表单

> RGCMS2.0 的搜索目前采用get方式提交

> RGCMS2.0的搜索，搜索模型必须为内容模型，且如果要搜索自定义字段内容，则必须在自定义字段中开启搜索，选择精确搜索或模糊搜索，否则搜索无结果。

HTML代码

```
搜索结果的模板
如果在搜索字段中定义了model = news，那么搜索结果模板默认为news_search.html
以此类推，如果model = product，那么默认搜索结果模板为product_search.html
如果当前模型默认搜索模板不存在，则最终自动选择default_search.html 作为搜索结果模板
```

```
前台html页面搜索代码
<form action="{$url_search}" method="get">
    <input type="text" name="title">
    <input type="text" name="其他字段">
    <input type="hidden" name="model" value="news">
    <input type="submit" name="" value="">
</form>
说明：
1. model字段为搜索的模型标识，如news或product等
2. 你可以在此基础上进行选择性搜索开发
```

搜索结果页标签调用

请阅读【标签变量】--【标签调用】--【搜索页列表】

点这里直达：搜索页列表

自定义表单

> RGCMS2.0的自定义表单功能使用方便，模型为自定义的表单模型，稍懂HTML的你可以构建任意表单接受数据，使用简单、安全

前台表单的使用

> 表单的前台HTML使用POST提交方式，表单系统内置十几种常用判断规则，您无需进行前台判断即可使用

```
<form action="{$url_form}" method="post">
    <input type="text" name="表单定义字段" value="" />
    其他如单选、多选、下拉等请自行扩展
    重点：必须在form体内添加以下2行代码
    <input type="hidden" name="model" value="模型标识，如：feedback" />
    {:token('__form__')}
    最后提交按钮
    <button type="submit">提交</button>
    <button type="reset">重置</button>
</form>
```

如果开启验证码验证，则表单中需添加验证码字段

```
// 验证码字段
<input type="text" name="captcha">
// 验证码图片
<img class="codeimg" src="{:captcha_src()}" alt="captcha" onclick="this.src='{:
captcha_src()}'" style="width:100%;" />
```

> 当然也可以使用ajax提交

```
**Html代码：**
<form id="myform">
    <input type="text" name="表单定义字段" value="" />
    其他如单选、多选、下拉等请自行扩展
    重点：必须在form体内添加以下2行代码
    <input type="hidden" name="model" value="模型标识，如：feedback" />
    {:token('__form__')}
    最后提交按钮
    <button class="form-submit">提交</button>
    <button type="reset">重置</button>
</form>

**Js代码：**
```

```
<script type="text/javascript">
$(".form-submit").click(function(){
  $.ajax({
    type: "POST",
    dataType: "json",
    url: "{$url_form}" ,//url
    data: $('#myform').serialize(),
    success: function (result) {
        if (result.code == '0') {
            // 这里写失败的提示
        } else {
          // 这里是成功
          window.location.reload();
        }
    },
    error : function() {
        alert("网络异常！");
    }
  });
  return false;
});
</script>
```

系统函数

默认值
无图默认值
生成缩略图
获取多图的某张
时间格式化
栏目字段继承
字符截取
字符隐藏
分割获取
二维码
转纯文本
运算
避免混淆
其他常用

默认值

> 默认值用于标签变量为空或内容不存在时显示的信息

默认值

用法： `标签变量|default="*"`

* 代表要显示的默认内容

> 所有标签变量适用，内容中禁止含有 `|` 字符

> 如：
> `{$info_title|default="标题未填写"}` 当 `{$info_title}` 无值为空时或未定义时显示”标题未填写“

无图默认值

> 无图默认值用于图片变量无值或(七牛未开启)图片不存在时。显示默认图片路径

无图默认值

用法： `标签变量|noimg`

> 请在图片变量中进行使用

如：

`{$info_img|noimg}` 当 `{$info_img}` 无值为空时或(七牛未开启)图片不存在时，值为 `/public/static/images/nopic.jpg`

生成缩略图

> 根据标签中的图片生成对应尺寸的缩略图

生成缩略图 2021012601新增

用法： `图片变量|thumb=宽,高,位置编号,图片质量,类型`

缩略图位置：/上传目录/thumb/

位置编号：

1：等比例缩放

2：缩放后填充（默认）

3：居中裁剪

4：左上角裁剪

5：右下角裁剪

6：固定尺寸缩放

图片质量：

50-100 默认80

> 宽高必填且大于0，当宽高大于当前图片宽高时，不生成缩略图，bmp格式不生成缩略图

> 如：
>
> `{$list.img|thumb=200,200}` 生成200*200的缩略图
>
> `{$list.img|thumb=200,200,1}` 等比例生成200*200的缩略图
>
> `{$list.img|thumb=200,200,3}` 居中裁剪生成200*200的缩略图
>
> `{$list.img|thumb=200,200,2,100}` 生成质量100的200*200的填充类型的缩略图
>
> `{$list.img|thumb=200,200,2,100,'jpg'}` 生成质量100的200*200的填充类型的jpg缩略图
>
> ...

获取多图的某张

> 获取多图字段中的某张图片

获取多图的某张

用法： `多图字段变量|get_img`

> 如：
>
> `{$info_imgs|get_img}` 默认获取多图字段imgs的第一张图片url
>
> 获取第二张
>
> `{$info_imgs|get_img=2}` 获取多图字段imgs的第二张图片url
>
> ...

时间格式化

date：时间格式化

```
{$info_create_time|date='Y-m-d H:i:s'} 输出：2019-07-01 13:17:39
{$info_create_time|date='Y年m月d日 H时:i分:s秒'} 输出：2019年07月01日 13时17分39秒
```

常用字母含义

- Y - 年份的四位数表示
- y - 年份的两位数表示

- m - 月份的数字表示（从 01 到 12）
- n - 月份的数字表示，不带前导零（1 到 12）
- M - 月份的短文本表示（用三个字母表示）

- d - 一个月中的第几天（从 01 到 31）
- j - 一个月中的第几天，不带前导零（1 到 31）

- H - 24 小时制，带前导零（00 到 23）
- G - 24 小时制，不带前导零（0 到 23）
- h - 12 小时制，带前导零（01 到 12）
- g - 12 小时制，不带前导零（1 到 12）

- i - 分，带前导零（00 到 59）
- s - 秒，带前导零（00 到 59）

- a - 小写形式表示：am 或 pm
- A - 大写形式表示：AM 或 PM

- D - 星期几的文本表示（用三个字母表示）
- l（'L' 的小写形式）- 星期几的完整的文本表示
- N - 星期几的 ISO-8601 数字格式表示（1表示Monday[星期一]，7表示Sunday[星期日]）
- w - 星期几的数字表示（0 表示 Sunday[星期日]，6 表示 Saturday[星期六]）
- z - 一年中的第几天（从 0 到 365）

> 更多请参考PHP内置函数date()使用方法

栏目字段继承

> RGCMS的栏目继承，意思是：当该栏目某字段为空时，则自动继承上一级栏目的该字段，直到该字段有值为止。如果该栏目所有上级的该字段都为空，则返回空值。

栏目继承

用法： `标签变量|findup="*"`

* 代表要findup的参数，可默认不写

> 当参数不写时，则为当前栏目的 `img` 字段

> 默认为 `img` 字段：
> `{$nav_img|findup}` 当，当前栏目的字段 `{$nav_img}` 无值为空时，则自动往上级栏目寻找值
> 该场景常用于，建立栏目大图后，无需设置子栏目大图，如果设置了子栏目大图，则显示子栏目的大图，否则显示上级栏目大图

> 其他字段：
> `{$nav_ename|findup="ename"}` 当，当前栏目的字段 `{$nav_ename}` 无值为空时，则自动往上级栏目寻找 `ename` 值

字符截取

> 字符截取用于当标签变量输出过长字符串需要截取时使用的方法

字符截取

用法： `标签变量|cut_str=字符长度`

> 当字符串超过限定的字符长度时，显示·...·

> 如： `{$site_title}` 的值为"欢迎使用RGCMS建站管理系统"
> `{$site_title|cut_str=9}` 得到 `欢迎使用RGCMS...`

字符隐藏

> 字符隐藏一般用于敏感信息的字符替换，如手机号：138****1234，姓名：王**等。

字符隐藏

用法： `标签变量|hidestr=开始位置,替换长度`

> 开始位置：
>
> 0：第一个字符开始
>
> 正数：指定位置开始
>
> 负数：从字符尾端开始
>
> ---
>
> 替换长度：可选，规定要隐藏的字符串长度。默认是直到字符串的结尾
>
> 正数：从 开始位置 参数所在的位置隐藏
>
> 负数：从字符串末端隐藏

> 如：
>
> `手机：{$site_mobile|hidestr=3,4}` 显示为，手机：138****1234
>
> `联系人：{$site_contact|hidestr=1}` 显示为，联系人：王**

分割获取

> 分割获取用于获取字符串中某一段字符信息，其中要有分割特征符号

分割获取

用法： `标签变量|getone=(获取第几个, 分割字符)`

> 所有标签变量适用，切勿获取超过分割数量的字符，分割字符默认为 `|` ，因此分割字符中禁止含有 `|` 字符

如： `{$site_tel}` 的值为13000000000|13111111111
`{$site_tel|getone}` 获取13000000000
`{$site_tel|getone=2}` 获取13111111111

如： `{$site_tel}` 的值为13000000000$13111111111
`{$site_tel|getone=1,'$'}` 获取13000000000
`{$site_tel|getone=2,'$'}` 获取13111111111

二维码

> 二维码生成

二维码生成

用法： `{:qrcode()}`

> 自动生成当前页面的二维码，如需指定页面url生成则添加url内容

> 如： `{:qrcode()}` 为当前页面url二维码
> `{:qrcode("http://www.rgxx.com")}` 则为http://www.rgxx.com的url二维码

转纯文本

> 转文本函数一般用户内容的截取，转文本将过滤掉所有的html标签和php标签，并清除空格

转文本

用法： `标签变量|totxt`

> 转文本将清除所有的格式和html标签

> 如： `{$info_content}` 的值为

```
<p>
	上世纪九十年代，我们还住在老伴儿单位的宿舍，邻居梅梅养了一只小狗，是亲戚送她的，捉来的时候只有两个月大。
</p>
<p>
	梅梅的女儿给这狗取名“阿卜”，据说是她女儿喜欢某部电影里的一个角色，那个角色就叫“阿卜”。
</p>
<p>
	刚捉来的时候小阿卜比巴掌大不了多少，毛绒绒的像只玩具狗。阿卜属于哈巴狗，嘴短耳大，眼睛鼓鼓的，黑亮黑亮的像两个大玻璃珠，全身狗毛黑白间杂，不算好看也不难看。把它捉在手上，它那两只玻璃珠眼睛就把你盯着，尾巴不停地摇，不时伸舌头舔舔你手掌，又可爱又可怜，很逗人喜欢。
</p>
```

> `{$info_content|totxt}` 的值为

```
上世纪九十年代，我们还住在老伴儿单位的宿舍，邻居梅梅养了一只小狗，是亲戚送她的，捉来的时候只有两个月大。梅梅的女儿给这狗取名“阿卜”，据说是她女儿喜欢某部电影里的一个角色，那个角色就叫“阿卜”。刚捉来的时候小阿卜比巴掌大不了多少，毛绒绒的像只玩具狗。阿卜属于哈巴狗，嘴短耳大，眼睛鼓鼓的，黑亮黑亮的像两个大玻璃珠，全身狗毛黑白间杂，不算好看也不难看。把它捉在手上，它那两只玻璃珠眼睛就把你盯着，尾巴不停地摇，不时伸舌头舔舔你手掌，又可爱又可怜，很逗人喜欢。
```

运算

我们可以对模板输出使用运算符，包括如下支持。

运算符	使用示例
+	{$a+$b}
-	{$a-$b}
*	{$a*$b}
/	{$a/$b}
%	{$a%$b}
++	{$a++} 或 {++$a}
--	{$a--} 或 {--$a}
综合运算	{$a+$b*10+$c}

在使用运算符的时候，不再支持前面提到的函数过滤用法，例如：

```
{$user_score+10}
{$user_score*$user_level}
```

三元运算

模板可以支持三元运算符，例如：

```
{$info_status? '正常' : '错误'}
```

还支持如下的写法：

```
{$info_title ?? '默认值'}
```

表示如果有设置 `$info_title` 则输出 `$info_title` ,否则输出 `默认值` 。

```
{$info_title?='默认值'}
```

表示$name为真时才输出默认值。

```
{$info_title ?: 'NO'}
```

表示如果$name为真则输出$name，否则输出NO。

```
{$a==$b ? 'yes' : 'no'}
```

前面的表达式为真输出yes,否则输出no， 条件可以是==、===、!=、!==、>=、<=

避免混淆

> 避免混淆是当页面中有其他语言结构主要是JS语法与标签类似或有冲突，可使用以下方法避免做到不冲突、

添加空格

用法： `在{和$之间添加空格`

例：JS语句为{$("name").val()} 等，可在{和$之间添加空格，为{ $("name").val()}。

使用不解析标签

用法： `{literal}*{/literal}`

* 代表要不解析的内容

```
{literal}
    {$("name").val()}
{/literal}
```

上面的 `{$("name").val()}` 标签被 `literal` 标签包含，因此并不会被模板引擎解析，而是保持原样输出。

所有可能和内置模板引擎的解析规则冲突的地方都可以使用 `literal` 标签处理

其他常用

> RGCMS2.0 内核采用THINKPHP5.1.*开发，因此也集成了TP的所有函数，具体了解请移步：https://www.kancloud.cn/manual/thinkphp5_1/354074

MD5

```
{$site_name|md5}
```

对变量进行MD5加密

raw：不转义输出

```
{$site_name|raw}
```

如：`{$page|raw}`，文章内容`{$info_content|raw}`

upper：转换为大写

```
{$info_email|upper} 输出：INFO@BAIDU.COM
```

lower：转换为小写

```
{$info_title|lower}
```

更新日志

> 版本号：2021050801

更新内容：

修复PHP7.2环境下抛出异常提醒的bug；

修复后台字段设置模糊或精确搜索时，前端搜索结果相反的bug；

删除404页面中rgcms标识；

后台layui更新至最新版，去除原来dropdown组件，采用layui内置下拉组件；

修复前端表单提交后，数据表未存入关键字段内容的bug；

修复批量添加栏目时，排序不递增的bug；

合并栏目管理左上角功能按钮为下拉按钮，新增栏目批量删除功能；

修复Mysql下，新增某些内容状态为关闭状态的bug；

修复添加属性时，状态为禁用的bug；

新增批量新增属性内容时自动排序；

新增后台模型内容管理中对文本域、下拉、单选、多选、开关等字段的筛选搜索；

修复后台模型管理中，选择栏目后筛选搜索为全局搜索的bug；

属性调用标签attr新增field和value调用方式；

后台栏目管理新增编辑栏目开关，关闭编辑栏目后，可使用伸缩功能；

新增文件下载路由；

增强前端文件下载功能，可保存为文件原文件名，下载链接为/down.html?file=文件地址或文件MD5值；

调整了tp内核中Collection的where方法中的两处细节；

问题排解