iZhan CMS 制作手册

一.安装说明.....4
1.系统及其软件要求.....4
2.Apache 设置要求.....4
3.PHP 其他必要要求.....4
4.硬件要求.....5
5.安装注意事项.....5
二.开发手册.....6
1.目录结构说明.....6
2.命名规范.....9
3.数据库结构说明.....11
4.二次开发流程.....12
5.入口程序.....13
6.配置文件调用.....14
7.数据库操作说明.....14
8.系统扩展.....21
9.控制器扩展技巧.....22
10.函数库调用.....22
get_config.....22
halt.....23

set_cache 23

get_cache 24

delete_cache 24

flush_cache 25

C 25

M 25

D 26

11.升级说明 27

12.系统静态化 27

13.性能测试 28

三.模板制作指南 37

1.基本规则 37

2.常用变量 37

3.常用函数 37

4.定义变量 38

5.实例说明 39

1）文章标签 39

2）文章列表标签 40

3）栏目标签 40

4）评论标签 41

5）内容标签 42

6）内容列表标签 43

7）友情链接标签 44

8）商品信息标签 44

9）商品相册标签 45

10）商品品牌标签 46

11）商品分类标签 46

12）商品类型标签 47

13）热搜词标签 48

14）留言标签 48

15）导航标签 49

16）推荐位标签 49

17）新文章列表标签 50

18）新商品分类标签 51

6.标签数据字典 51

1）article 文章标签 52

2）Content 内容标签 53

3）contentlist 内容列表标签 54

4）articlelist 文章列表标签 56

5）comment 评论标签 57

6）goods 商品标签 58

7）Category 栏目标签 59

8）Navigation 导航标签 61

9）Position 推荐位标签 62

10）Message 留言标签 62

一.安装说明

1.系统及其软件要求

操作系统：Windows、Linux、Unix/BSD 等各种平台

（推荐 Red Hat Enterprise Linux6.2以上）

HTTP 服务器：Apache,Nginx 推荐 Apache/2.2.22 以上版本

数据库：MySQL5.1+ 推荐 MySQL5.5 以上版本

2.Apache 设置要求

必须将 RewriteEngine 模式打开

Apache 2.x 的用户请检查 conf/httpd.conf 中是否存在如下一段代码：

LoadModule Rewrite_module modules/mod_Rewrite.so

session.auto_start = 0

改成

session.auto_start = 1

3.PHP 其他必要要求

版本要求 PHP5.4

GD 库 支持 （版本要求：2.0.28+）

Session 支持 （Session Support enabled）

Zend Guard Loader 6.0.0/5.5

4.通过 FTP 上传到服务要求

必须采用二进制方式传输

必须要以上要求，否者无法安装。

程序运行最基本要求不包含数据要求（客户数据建议根据自己需求购买）：WEB 空间：100MB

数据库空间：50MB

4.硬件要求

2U机架式服务器(主流双路2U服务器)

处理器: 英特尔至强处理器双核

内存: 2GB以上 建议8 GB

存储：20GB以上

5.安装注意事项

1）解压缩包到网站根目录。

2）复制upload目录下的所有文件到网站根目录。

3）在浏览器输入您的域名，然后回车，按照提示安装即可。

4）安装完成，为了安装起见请务必删除整个install目录。

二.开发手册

1.目录结构说明

IZHANCMS

├─admin CMS 后台应用名

| ├─api API 接口

| ├─application 应用目录

| | ├─admin 应用模块

| | | ├─controllers 控制器目录

| | | ├─models 模型目录

| | ├─config 配置文件目录

| ├─data 数据缓存目录

| | └─cache

| └─template 模板文件目录

├─application CMS 前台应用目录

| ├─category 模块名

```
|   |   ├─controllers 控制器目录
|   |   └─models 模型目录
|   ├─config 配置文件目录
|─data 前台数据缓存目录
|   └─cache
├─doc 说明文档目录
├─html 生成静态文件目录
├─imc 用户认证中心
|   ├─api API 接口
|   ├─application 应用目录
|   |   ├─app 模块名
|   |   |   ├─controllers 控制器目录
|   |   |   └─models 模型目录
|   |   ├─config 配置文件目录
|   ├─data 数据缓存目录
|   |   └─cache
|   └─template 模板文件目录
├─imc_client IMC 客户端
├─install 安装程序目录
├─library 基础类库目录
|   └─iZhan
|       ├─base 基类
```

```
|       ├─cache 缓存
|       ├─classes 公用工具类
|       ├─db 数据库
|       ├─field 字段设置
|       ├─functions 公用函数库
|       ├─sql 模型建表语句
|       ├─taglib 标签库
|       |   ├─config
|       |   └─help
|       └─views 系统公用页面
|           └─html
|               ├─css
|               └─images
├─static 公用脚本文件和上传文件
|   ├─js
|   |   ├─artDialog4.1.6
|   |   ├─ckeditor
|   └─uploadfile
├─template CMS前台模板
|   ├─default
|   ├─green
└─vendor 存放第三方类库文件
```

├─hprose

└─smarty

2.命名规范

PHP 命名规范(帮助记忆版本)

==========

类名：类的文件名及类名保持一致（包括大小写），采用驼峰命名法，首字母大写。

属性名：用驼峰法命名，并且首字母小写。如：tableName

类的方法：用驼峰法命名，并且首字母小写。如：getUserInfo（ ）

工具函数：函数的命名使用小写字母和下划线的方式，例如 get_client_ip

变量名：变量名应该只包含小写字母，用下划线分隔 如：$table_name

常量名：以大写字母和下划线命名，例如 HAS_ONE 和 MANY_TO_MANY

数据表名：用小写字母加下划线方式命名

表字段名：用小写字母加下划线方式命名，例如 user_name

目录命名: 全部小写 例如：admin cache config language library

接口定义：以大写字母 I 开头，采用驼峰命名法， 例如：IProduct.php

实现接口：以 Impl 结尾，采用驼峰命名法，首字母大写 例如：ProductImpl.php

温馨提示：

1）对于只包含有 PHP 代码的文件，结束标志（"?>"）是不允许存在的，PHP 自身不需要（"?>"），这样做，可以防止它的末尾被恶意注入信息。

2）缩进由四个空格组成，禁止使用“制表符 TAB ”。

3）常量必须通过 "const" 定义为类的成员，不建议使用 "define" 定义全局常量。

4）以双下划线“__”开头的函数或方法作为魔法方法，例如 __call 和 __autoload 字符串文字 。

5）当字符串是文字(不包含变量)，应当用单引号（ apostrophe ）来括起来：

$a = 'Example String';

6）包含单引号（'）的字符串文字

当文字字符串包含单引号（apostrophe ）就用双引号括起来，特别是在 SQL 语句中，例如：$sql = "SELECT `id`, `name` from `people` WHERE `name`='Fred' OR `name`='Susan'";在转义单引号时，上述语法是首选，便于阅读。

7）禁止 SQL 语句采用 select * from abc ，要查询的字段必须带上，其他操作比如：插入等。

8）变量替换有以下几种形式：

$greeting = "Hello $name, welcome back!";

$greeting = "Hello {$name}, welcome back!";

注释模板

```
/**
 * iZhanCMS 铭万开源CMS内容管理系统 (http://cms.b2b.cn)
 *
 * 文件用途说明
 *
 *
 *
 * 文件修改记录：
 * <br>周立峰 ${date} ${time} 创建此文件
 *
 * @author 周立峰 <zhoulifeng@iZhan.cn> ${date} ${time}
 * @filename ${file} ${encoding}
```

```
* @copyright   Copyright (c) 2004-${year} iZhan Technologies Inc. (http://www.b2b.cn)
* @license     http://cms.b2b.cn/license/    iZhanCMS 1.0
* @version     SVN: $$Id$$
* @link        http://cms.b2b.cn
* @link        http://www.b2b.cn
* @package     ${class_container}
* @since       1.0.0
*/
```

3.数据库结构说明

数据库表	编码	说明
accessory	utf8_general_ci	附件管理表
admin	utf8_general_ci	后台管理员表
admin_log	utf8_general_ci	管理员操作日志表
admin_role	utf8_general_ci	后台管理员角色表
admin_shortcut	utf8_general_ci	管理员常用操作表
adposition	utf8_general_ci	广告位管理
adtype	utf8_general_ci	广告类型管理
advert	utf8_general_ci	广告管理
area	utf8_general_ci	区域表
article	utf8_general_ci	文章表
attribute	utf8_general_ci	留言类别模型属性表
category	utf8_general_ci	栏目分类表
comment	utf8_general_ci	评论表
company	utf8_general_ci	企业表
faillogin	utf8_general_ci	登录失败
field	utf8_general_ci	模型字段表
goods	utf8_general_ci	商品表
goods_album	utf8_general_ci	商品相册表
goods_attr	utf8_general_ci	商品属性表
goods_attr_value	utf8_general_ci	商品属性值表
goods_brand	utf8_general_ci	商品品牌表
goods_link_goods	utf8_general_ci	商品关联表
goods_sort	utf8_general_ci	商品分类表
goods_type	utf8_general_ci	商品类型
guide	utf8_general_ci	标签向导列表
imc_app	utf8_general_ci	应用设置
imc_users	utf8_general_ci	IMCenter 用户表
link	utf8_general_ci	友情链接
linkage	utf8_general_ci	关联表
linkage_bill	utf8_general_ci	联动菜单表

mailstate	utf8_general_ci	邮件状态
mailtemplate	utf8_general_ci	邮件模板
maintable	utf8_general_ci	主表
manager_cate_per	utf8_general_ci	管理员栏目操作权限表
member	utf8_general_ci	会员信息表
member_1	utf8_general_ci	个人表单
member_2	utf8_general_ci	企业表单
member_cate_per	utf8_general_ci	会员栏目操作权限表
member_group	utf8_general_ci	会员分组表
member_level	utf8_general_ci	会员级别表
message_manage	utf8_general_ci	留言管理
mix_model	utf8_general_ci	会员及留言模型
model	utf8_general_ci	模型表
msg_message	utf8_general_ci	站内信表
msg_relation	utf8_general_ci	站内信关系表
payment	utf8_general_ci	支付功能相关
permission	utf8_general_ci	权限菜单表
person	utf8_general_ci	个人表
position	utf8_general_ci	推荐位
position_info	utf8_general_ci	推荐位数据表
registdeal	utf8_general_ci	注册协议
remind	utf8_general_ci	提醒设置表
role_permission	utf8_general_ci	角色权限对照表
scores_rule	utf8_general_ci	积分规则表
search	utf8_general_ci	搜索表
site_log	utf8_general_ci	网站关联系统日志
synchronize	utf8_general_ci	同步登录配置表
upgrade_log	utf8_general_ci	升级日志表
web_config	utf8_general_ci	网站各基本参数配置

4.二次开发流程

1）创建数据库表结构和初始数据

2）创建应用目录或模块目录

3）开发模块控制器

4）创建数据库 model 类

5）开发模块需要工具类与模块函数（可选）

6）创建视图（模板）文件

7）运行和调试

8）制作安装与卸载

5.入口程序

iZhanCMS 是采用 MVC 设计模式开发,基于应用模块和操作的方式进行访问，采用单一入口模式进行项目部署和访问，无论访问任何一个模块或功能，只有一个统一的入口。入口程序是在前期处理用户请求的引导程序。它是唯一一个可以被最终用户可以直接请求运行的。

iZhanCMS 入口程序包含如下几行：

index.php

```
define('IN_iZhan', true);
define('APPNAME', 'home');
define('SYS_REWRITE', true);
define('DIR_ROOT', dirname(__FILE__) . DIRECTORY_SEPARATOR);
require_once DIR_BF_ROOT . 'Application.php';
app::run();
```

这段代码首先加载了 iZhanCMS 框架的引导文件 Application.php ,然后它根据指定的配置文件建立了一个 Web 应用实例并运行。

6.配置文件调用

配置文件存放在每个应用中（前后台）的 application/config

用 C（$filename,$item='',$app=''）或者 get_config($filename,$item='',$app='')方法调用，使用方法见系统类库与函数库调用。

7.数据库操作说明

数据库操作 CRUD

create 增加 （在数据表中新增一行数据）	
用法	create($data='')
参数	data（可选）：要新增数据，数组形式，数组的键是数据表中的字段名，键对应的值是需要新增的数据
返回值	如果数据非法或者查询错误则返回 false 如果是自增主键，则返回主键值

```
$User = M("User"); // 实例化 User 对象
$data['name'] = 'admin';
$data['email'] = 'admin@gmail.com';
$User->create($data);
```

update 更新数据，该函数将根据参数中设置的条件而更新表中数据	
用法	update($conditions, $row)
参数	conditions：数组或字符串形式，查找条件

	row：要更新的数据，数组形式，数组的键是数据表中的字段名，键对应的值是需要更新的数据
返回值	如果数据非法或者查询错误则返回 false 如果执行成功则返回 true

```
$message = M("Message"); // 实例化 Message 对象
$param = array
        (
            'title' => $this->getParams('title'),
            'content' => $this->getParams('content'),
            'author' => $this->getParams('author'),
        );
$condition = array('id' => $this->getParams('id'));
$message ->update($condition, $param);
```

delete 删除数据，该函数将根据参数中设置的条件而删除表中数据	
用法	delete($conditions)
参数	conditions：数组或字符串形式，查找条件
返回值	如果数据非法或者查询错误则返回 false 如果执行成功则返回 true

```
$message = M("Message"); // 实例化 Message 对象
```

$message ->delete(**array**('id'=>$id));

查询操作

getone 读取符合条件的一条记录	
用法	getOne($options=array())
参数	options（可选）: 数组或字符串形式，查找条件
返回值	如果查询错误返回 false 如果查询结果为空返回 null 如果查询成功返回查询的结果

$User = M("User"); // 实例化 User 对象

// 查找 status 值为1name 值为 think 的用户数据

$User->where('status=1 AND name="think"')->find();

即使满足条件的数据不止一条，find 方法也只会返回第一条记录。

select 读取符合条件的结果集	
用法	select($options=array())
参数	options（可选）: 为数组时表示操作表达式，通常由连贯操作完成。默认为空数组。
返回值	如果查询错误返回 false 如果查询结果为空返回 null 如果查询成功返回查询结果（二维索引数组）
相关方法	通常配合连贯操作 where、field、order、limit、join 等一起使用

```
$User = M("User"); // 实例化 User 对象
// 查找 status 值为1的用户数据，以创建时间排序，返回10条数据
$list = $User->where('status=1')->order('create_time')->limit(10)->select();
```

findCount 得到符合条件的记录数	
用法	findCount($condition=null)
参数	condition（可选）: 为数组时表示操作表达式，通常由连贯操作完成。默认为空数组。
返回值	如果查询错误返回 false 如果查询结果为空返回 null 如果查询成功返回查询的结果（二维索引数组）
相关方法	通常配合连贯操作 where、field、order、limit、join 等一起使用

连贯操作

where

where 用于查询或者更新条件的定义	
用法	where($where)
参数	where（必须）: 查询或者操作条件，支持字符串、数组
返回值	当前模型实例

```
$message->where($where)->select();
```

$message->where($where)->getOne();

Field

field 用于定义要查询的字段	
用法	field($field)
参数	field（必须）：字段名，支持字符串和数组类型；如果为 true 则表示显示数据表的所有字段。
返回值	当前模型实例
备注	如果不调用 field 方法，则默认返回所有字段，和 field（'*'）等效

$message->field('id,nickname as name')->select();

$message->field(array('id','nickname'=>'name'))->select();

order

order 用于对操作结果排序	
用法	order($order)
参数	order（必须）：排序的字段名，支持字符串和数组，支持多个字段排序
返回值	当前模型实例
备注	如果不调用 order 方法，按照数据库的默认规则

order('id desc')

排序方法支持对多个字段的排序

order('status desc,id asc')

order 方法的参数支持字符串和数组，数组用法如下：

order(array('status'=>'desc','id'))

Limit

limit 用于定义要查询的结果限制（支持所有数据库类型）	
用法	limit($limit)
参数	limit（必须）：限制数量，支持字符串
返回值	当前模型实例
备注	如果不调用 limit 方法，则表示没有限制

limit('1,10')

如果使用 limit('10') 等效于 limit('0,10')

Group

group 用于数据库 group 查询支持	
用法	group($group)
参数	group（必须）：group 字段名，支持字符串
返回值	当前模型实例
备注	无

使用示例：group('user_id')

Group 方法的参数只支持字符串

Having

having 用于数据库 having 查询支持	
用法	having($having)
参数	having（必须）：having，支持字符串

返回值	当前模型实例
备注	无

使用示例：

having('user_id>0')

having 方法的参数只支持字符串

Join

join 用于数据库 join 查询支持	
用法	join($join)
参数	join（必须）: join 操作，支持字符串和数组
返回值	当前模型实例
备注	join 方法支持多次调用

使用示例：

$Model->join(' work ON artist.id = work.artist_id')->join('card ON artist.card_id = card.id')->select();

默认采用 LEFT JOIN 方式，如果需要用其他的 JOIN 方式，可以改成

$Model->join('RIGHT JOIN work ON artist.id = work.artist_id')->select();

如果 join 方法的参数要用数组表示，只能使用一次 join 方法，并且不能和字符串方式混合使用。

例如：join(array(' work ON artist.id = work.artist_id','card ON artist.card_id = card.id'))

Distinct

distinct 查询数据的时候进行唯一过滤

用法	distinct($distinct)
参数	distinct（必须）：是否采用 distinct，支持布尔值
返回值	当前模型实例

使用示例：

$Model->Distinct(true)->field('name')->select();

8.系统扩展

1）工具类扩展和函数扩展

系统命名规范

系统类库位于系统的 iZhanCMS/library/iZhan/classes 目录下面，函数库文件目录 iZhanCMS/library/iZhan/functions，其中 common.php 基础函数库（快捷操作） functions.php 扩展函数库，该文件中包含常用工具函数。为框架中默认加载，里面的函数可直接使用。tagfun.php 为标签函数。

2）第三方库扩展

第三方库扩展通常放在 iZhanCMS\vendor 目录下面，方便管理和维护。目前主要引入了 hprose 和 smarty。为了开发方便和扩展性，还可以引入其他第三方库。

3）数据库扩展

丰富的数据库驱动，未来计划支持 MYSQL、POSTGRESQL、ORACLE、SQLITE、MSSQL、MONGODB、REDIS 等数据库。目前只支持 MYSQL。

4）缓存扩展

缓存扩展存放目录/iZhanCMS/library/iZhan/cache 下。目前 CMS 对缓存的支持主要

是：文件缓存和 Memcached 的支持。 将来还会扩展其他缓存。

9.控制器扩展技巧

如果要对已存在的控制器进行二次开发 ,为了方便升级和维护不建议直接对核心文件直接修改，您可以通过继承的形式进行二次开发。

例如：您要修改 iZhanCMS/library/iZhan/base/HomeController.php 进行二次开发。您可以按照命名规范建立您所需的程序文件。比如：ContentController。

例如 ContentController 代码如下：

```
class ContentController extends HomeController {
    public   $ContentModel;
    public function init()
    {
        $this->ContentModel = D('Content');
        parent::init();
    }
 }
```

10.函数库调用

函数库调用

get_config

get_config($filename,$item='',$app='')

作用：获取配置文件信息。

说明：$filename 是配置文件名。

$item　配置项（针对二维数组的配置文件）

$app　应用名（针对调用其他应用的配置文件，调用本应用则不用传参数）

配置文件存放在每个应用中（前后台）的 application/config

Eg:get_config('cache','cache');

halt

halt($message, $level = 'Error')

作用：显示错误信息，若调试模式关闭时（即:SYS_DEBUG 为 false 时），则将错误信息写入日志

说明：$message 消息内容

$level　日志类型. 默认为 Error. 参数：Warning, Error, Notice

set_cache

set_cache($name,$data,$filepath='',$timeout=0,$app='',$type='file',$config='file',$datatype='')

作用：写入缓存。

说明：$name　缓存名称

$data　缓存数据

$filepath　数据路径（模块名称） caches/cache_$filepath/

$type　缓存类型[file,memcache,apc]

$datatype　缓存数据类型 caches/cache_$filepath/caches_$datatype 默认

是 data

$config 配置名称

$timeout 过期时间

$app 应用名

get_cache

get_cache($name, $filepath='',$app='', $type='file', $config='file',$datatype='')

作用：读取缓存

说明： $name 缓存名称

$filepath 数据路径（模块名称） caches/cache_$filepath/

$app 应用名 admin 用于不同应用间的数据共享

$type 缓存类型[file,memcache]

$config 配置名称

$datatype 缓存数据类型 caches/cache_$filepath/caches_$datatype 默认是 data

delete_cache

delete_cache($name, $filepath='', $type='file', $config='',$datatype='')

作用：删除缓存

说明： $name 缓存名称

$filepath 数据路径（模块名称） caches/cache_$filepath/

$type 缓存类型[file,memcache]

$config 配置名称

$datatype 缓存数据类型 caches/cache_$filepath/caches_$datatype 默认是 data

flush_cache

flush_cache($type='file', $config='file')

作用：清除缓存

说明： $type 缓存类型[file,memcache]

$config 配置名称

C

C($filename,$item='',$app="")

作用：获取配置文件信息。同 getconfig（ ）

说明：$filename 配置文件名。

$item 配置项（针对二维数组的配置文件）

$app 应用名（针对调用其他应用的配置文件，调用本应用则不用传参数）

配置文件存放在 每个应用中（前后台）的 application/config

Eg:getconfig('cache','cache');

M

M($name,$prefix='',$dbsetting='',$dbconfig='')

作用：实例化一个没有模型文件的 Model

说明： $name 表名（不包含前缀）

$prefix 表的前缀

$dbconfig 数据库配置文件名 database

$dbsetting 数据库配置名 default/link1

D

D($name,$module='',$app='')

作用：实例化自定义的模型类

说明： $name 文件名

$module 模块名（不同模块间调用）

$app 项目名（home,admin)

在 model 文件里可以指定

```
//表主键
public $pk;
// 数据表前缀
public $tablePrefix = '';
//表名称
public $tableName = null;
//表全名
public $trueTableName = null;
//数据库配置
```

```
public $dbconfig = array();
//调用数据库的配置项
public $dbsetting = 'default';
```

11.升级说明

升级说明：

如果官方发布新升级包，您会在两个地方看查看到升级提示：

1）登录进入后台首页，导航下方会有升级提示，点击“现在更新”，进入更新列表，选择要更新的补丁即可；

2）登录后台，位置：网站管理->系统扩展->在线升级进入页面，点击新版本检测，检测官方发布的补丁包，如果有可用的补丁包，选择升级即可。要注意按照补丁包时要按照发布的先后顺序进行升级。

12.系统静态化

静态网页是实际存在的，无需经过服务器的编译，直接加载到客户浏览器上显示出来，不需要执行 asp,php,jsp,.net 等程序生成客户端的代码网页。常见的静态页面举例：.html 扩展名的、htm 扩展名的，至于页面静态化的作用和意义在此就不做赘述，相信各位站长肯定已经心知肚明，话不多说，马上转入正题！本站点采用了动、静态相结合的形式展示，是否静态化是可选项，具体操作是后台发布文章（商品）可选择是否静态化，如果选择了此项，静态文件全部存放在站点根目录的 html 目录下，结构为/栏目名/年月/日期/content_文章 ID.html（goods_商品 ID.html），详见官网。

13.性能测试

本次测试结果作为参考：

1.铭万 CMS

E:\wamp\bin\apache\apache2.2.22\bin>ab-n1000-c50

http://www.cms.com/admin/admin/index/init?menuid=3

This is ApacheBench, Version 2.3 <$Revision: 655654 $>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.cms.com (be patient)

Completed 100 requests

Completed 200 requests

Completed 300 requests

Completed 400 requests

Completed 500 requests

Completed 600 requests

Completed 700 requests

Completed 800 requests

Completed 900 requests

Completed 1000 requests

Finished 1000 requests

Server Software: Apache/2.2.22

Server Hostname: www.cms.com

Server Port: 80

Document Path: /admin/admin/index/init?menuid=3

Document Length: 3208 bytes

Concurrency Level: 50

Time taken for tests: 9.912 seconds

Complete requests: 1000

Failed requests: 0

Write errors: 0

Non-2xx responses: 1000

Total transferred: 3645000 bytes

HTML transferred: 3208000 bytes

Requests per second: 100.89 [#/sec] (mean)

Time per request: 495.600 [ms] (mean)

Time per request: 9.912 [ms] (mean, across all concurrent requests)

Transfer rate: 359.12 [Kbytes/sec] received

Connection Times (ms)

min mean[+/-sd] median max

Connect: 0 0 0.3 0 1

Processing: 23 484 62.4 491 597

Waiting: 23 483 62.4 490 597

Total: 23 484 62.4 491 597

Percentage of the requests served within a certain time (ms)

50% 491

66% 498

75% 505

80% 508

90% 520

95% 532

98% 548

99% 553

100% 597 (longest request)

E:\wamp\bin\apache\apache2.2.22\bin>

以上结果指出，在并发 50 个请求的情况下，完成 1000 次的访问请求，共花了 9.912 秒，这个程序每秒可处理 100.89 个请求。

2. php cms

E:\wamp\bin\apache\apache2.2.22\bin>ab-n1000-c50 http://phpcms.b2b.iZhan.c

n/index.php?m=member&c=member&a=manage&menuid=72&pc_hash=alUkF4

This is ApacheBench, Version 2.3 <$Revision: 655654 $>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking phpcms.b2b.iZhan.cn (be patient)

Completed 100 requests

Completed 200 requests

Completed 300 requests

Completed 400 requests

Completed 500 requests

Completed 600 requests

Completed 700 requests

Completed 800 requests

Completed 900 requests

Completed 1000 requests

Finished 1000 requests

Server Software: Apache/2.2.22

Server Hostname: phpcms.b2b.iZhan.cn

Server Port: 80

Document Path: /index.php?m=member

Document Length: 2546 bytes

Concurrency Level: 50

Time taken for tests: 12.960 seconds

Complete requests: 1000

Failed requests: 0

Write errors: 0

Total transferred: 2788000 bytes

HTML transferred: 2546000 bytes

Requests per second: 77.16 [#/sec] (mean)

Time per request: 648.000 [ms] (mean)

Time per request: 12.960 [ms] (mean, across all concurrent requests)

Transfer rate: 210.08 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.5	0	2
Processing:	91	643 211.2	592	1806
Waiting:	90	636 207.9	589	1796
Total:	92	644 211.2	592	1806

Percentage of the requests served within a certain time (ms)

50% 592

66% 646

75% 712

80% 762

90% 885

95% 1036

98% 1290

99% 1504

100% 1806 (longest request)

E:\wamp\bin\apache\apache2.2.22\bin>

以上结果指出，在并发 50 个请求的情况下，完成 1000 次的访问请求，共花了 12.960 秒，这个程序每秒可处理 77.16 个请求。

3.DEDE CMS

E:\wamp\bin\apache\apache2.2.22\bin>ab -n 1000 -c 50 http://dede.b2b.iZhan dede/content_list.php

This is ApacheBench, Version 2.3 <$Revision: 655654 $>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking dede.b2b.iZhan.cn (be patient)

Completed 100 requests

Completed 200 requests

Completed 300 requests

Completed 400 requests

Completed 500 requests

Completed 600 requests

Completed 700 requests

Completed 800 requests

Completed 900 requests

Completed 1000 requests

Finished 1000 requests

Server Software: Apache/2.2.22

Server Hostname: dede.b2b.iZhan.cn

Server Port: 80

Document Path: /dede/content_list.php

Document Length: 0 bytes

Concurrency Level: 50

Time taken for tests: 10.085 seconds

Complete requests: 1000

Failed requests: 0

Write errors: 0

Non-2xx responses: 1000

Total transferred: 448000 bytes

HTML transferred: 0 bytes

Requests per second: 99.16 [#/sec] (mean)

Time per request: 504.250 [ms] (mean)

Time per request: 10.085 [ms] (mean, across all concurrent requests)

Transfer rate: 43.38 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.5	0	1
Processing:	35	498 163.4	454	1496
Waiting:	35	494 158.9	452	1496
Total:	35	498 163.4	455	1496

Percentage of the requests served within a certain time (ms)

50% 455

66% 490

75% 529

80% 561

90% 688

95% 847

98% 980

99% 1093

100% 1496 (longest request)

E:\wamp\bin\apache\apache2.2.22\bin>

以上结果指出，在并发 50 个请求的情况下，完成 1000 次的访问请求，共花了 10.085 秒，这个程序每秒可处理 99.16 个请求。

三.模板制作指南

1.基本规则

MOCMS 标签必须以 {mo}开头，并以{/mo}结尾

<!--{mo:article typeid="5" row="10" page="$_GET['page']"}-->

<a title='<!--{$title}-->' href="<!--{$url}-->">

<!--{/mo:标签名称}-->

<!--{mo:标签名称 （空格）参数名=“值” （空格）}-->

<!--{值}-->

<!--{/mo:标签名称}-->

2.常用变量

$_GLOBAL[‘username’] 判断用户登录返回用户名

$cid 当前页面的栏目 id

$pid 当前页面的父栏目 ID

3.常用函数

logo()获取 logo

beian()获取备案

powerby()获取版权

description()获取网站描述

cid2name(10)通过栏目 ID 获取栏目名字

getMoreUrl(2)获取某个栏目下更多表签

date(“Y-m-d,H:i:s”,time())格式化时间

csubstr($str,30,’...’,0) ‘$str’：要截取的字符串；‘30’:要截取的长度；’....’：后面省略的字符；‘0’:开始位置

maketimes($time) 获取距离当前时间几秒前，几分前，几个小时前，几天前,几个星期前，几个月前，几年前

cid2Info(10)获取某个栏目的信息

使用：{php $info=cid2Info(10);}{$info[‘catname’]}栏目名称

4.定义变量

{php $i=1;} {php $string = date('Y-m-d');}

变量：以$定义

{$page} {$nextpage} {$more}普通变量

{$array['name']} 数组变量

{$array['book']['name']} 二维数组

注：标签变量可以在 js 中使用，如：提示信息

循环

{foreach$data $n $r}

{$n[name]}

{/foreach}

if else 判断

{if $langue="zh"}

{php $typeid=1}

{elseif $langue="en"}

{php $typeid=2}

.....

{else}

{php $typeid=3}

{/if}

5.实例说明

1）文章标签

标签名称： article

标签功能: 文章标签

使用实例：

```
{mo:article row="10" cid="1" return="data"}
  {foreach $data  $key $value}
    <tr><td>标题：<a href='{$value["url"]}'>{$value["title"]}</a> </td>
    <td>时间：{date("Y-m-d",$value["time"])}  </td></tr>
  {/foreach}
{/mo:article}
```

参数说明：

row ="100" 返回数目

cid ="10" 栏目 ID

return ="data" 返回值名称：默认用 data

2）文章列表标签

标签名称： articlelist

标签功能: 文章列表标签

使用实例：

```
{mo:articlelist row="100" from="100" pagesize="100" pagenum="10" cid="1" type="son
return="data"}
  {foreach $data  $key $value} □
<tr><td>标题：<a href='{$value["url"]}'>{$value["title"]}</a> </td>
<td>时间：{date("Y-m-d",$value["time"])}  </td></tr>□
  {/foreach}□
{/mo:articlelist}
```

参数说明：

row ="100" 返回数目

from ="100" 从第几条查询

pagesize ="100" 分页数量

pagenum ="10" 每页显示的页码（不能小于 5）

pagevar ="page" 分页变量：可以避免同一个页面多个分页相互影响

cid ="10" 调取某个栏目下的文章:栏目 ID，没有 type 参数调用本身栏目下的文章列表

type ="son" 调用 cid 下面的子栏目的文章列表

type ="parent" 调用 cid 的父栏目的文章列表

type ="all" 调用 cid 的子栏目以及本身栏目的文章列表

return ="data" 返回值名称：默认用 data

3）栏目标签

标签名称： category

标签功能: 栏目标签

使用实例：

```
{mo:category row="10" cid="45" order="ordernum" type="son" return="data"}
  {foreach $data  $key $value}
<a href='{$value["url"]}'>{$value["catname"]}</a>
  {/foreach}
{/mo:category}
```

参数说明：

row ="10" 导航显示的条数:默认调取 10 条记录

cid ="45" 栏目 ID:当 type 为"all 或者 top"的时候无效

type ="son" 调用 cid 下面的子栏目

type ="parent" 调用 cid 的父栏目

type ="self" 调用 cid 的本身栏目

type ="top" 调用一级栏目

type ="all" 调用所有栏目

order="ordernum" 排序方式：默认按排序字段升序

order="id desc" 排序方式：id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

4）评论标签

标签名称： comment

标签功能: 评论标签

使用实例：

```
{mo:comment id="1" row="10" modelid="1" return="data"}
  {foreach $data  $key $value}
```

```
<tr><td>内容：{$value["comment_content"]}</a> </td>
<td>时间：{date("Y-m-d",$value["reply_time"])}</td></tr>
  {/foreach}
{/mo:comment}
```

参数说明：

row ="100" 返回数目

id ="10" 内容 ID

modelid ="1" 模型 ID

pagesize ="10" 每页显示条数

pagenum ="10" 每页显示的页码（不能小于 5）

pagevar ="page"　分页变量：可以避免同一个页面多个分页相互影响

from ="0" 内容开始位置

return ="data" 返回值名称：默认用 data

5）内容标签

标签名称： content

标签功能:　内容标签

使用实例：

```
{mo:content row="10" cid="1" return="data"}
  {foreach $data  $key $value}
    <tr><td>标题：<a href='{$value["url"]}'>{$value["title"]}</a> </td>
    <td>时间：{date("Y-m-d",$value["time"])}  </td></tr>
  {/foreach}
{/mo:content}
```

参数说明：

row ="100" 返回数目

cid ="10" 栏目 ID

modelid ="100" 模型 id，可选

return ="data" 返回值名称：默认用 data

6）内容列表标签

标签名称： contentlist

标签功能: 内容列表标签

使用实例：

```
{mo:contentlist row="10" cid="1" pagesize="100" return="data"}
  {foreach $data  $key $value}
    <tr><td>标题：<a href='{$value["url"]}'>{$value["title"]}</a> </td>
    <td>时间：{date("Y-m-d",$value["time"])}  </td></tr>
  {/foreach}
{/mo:contentlist}
```

参数说明：

row ="100" 返回数目

from ="100" 从第几条查询

model ="1" 模型 id,调用模型下面的内容

pagesize ="100" 分页数量

pagenum ="10" 每页显示的页码（不能小于 5）

pagevar ="page" 分页变量：可以避免同一个页面多个分页相互影响

cid ="10" 调取某个栏目下的文章:栏目 ID，没有 type 参数调用本身栏目下的文章列表

type ="son" 调用 cid 下面的子栏目的文章列表

type ="parent" 调用 cid 的父栏目的文章列表

type ="all" 调用 cid 的子栏目以及本身栏目的文章列表

return ="data" 返回值名称：默认用 data

7）友情链接标签

标签名称： friendlink.lib.php

标签功能: 友情链接标签

使用实例：

```
{mo:friendlink row="10" return="data"}
  {foreach $data  $key $value}
    <tr><td>名称：<a href='{$value["com_url"]}'>{$value["name"]}</a>
  {/foreach}
{/mo:friendlink}
```

参数说明：

row ="100" 返回数目

order ="sort" 排序 id

return ="data" 返回值名称：默认用 data

8）商品信息标签

标签名称： goods

标签功能: 商品信息标签

使用实例：

```
{mo:goods row="10" id="45" order="brandid"  return="data"}
  {foreach $data  $key $value}
    <a href='{$value["url"]}'>{$value["brandname"]}</a>
  {/foreach}
{/mo:goods}
```

参数说明：

row ="100" 返回数目

pagesize ="10" 每页显示条数

pagenum ="10" 每页显示的页码（不能小于 5）

pagevar ="page"　分页变量：可以避免同一个页面多个分页相互影响

from ="0" 内容开始位置

id ="45" 商品 ID

cid ="1" 栏目 id

brandid ="1" 品牌 id

typeid ="1" 类型 id

sortid ="1" 类别 id

type ="son" 调用 cid 下面的子栏目的商品列表

type ="all" 调用 cid 的子栏目以及本身栏目的商品列表

order="hits desc" 按点击量排序（降序）

order="created" 排序方式：发布时间(升序)

9）商品相册标签

标签名称： goodsalbum

标签功能: 商品相册标签

使用实例：

```
{mo:goodsalbum row="10" id="45" order="goodsid"  return="data"}
  {foreach $data  $key $value}
    <img src='{UPLOAD_PATH}/goods/{$value["photo"]}'/>
  {/foreach}
{/mo:goodsalbum}
```

参数说明：

row ="10" 显示的条数:默认调取 10 条记录

id ="45" 商品 ID

order="goodsid desc" 排序方式：商品 id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

10）商品品牌标签

标签名称： goodsbrand

标签功能: 商品品牌标签

使用实例：

```
{mo:goodsbrand row="10" id="45" order="brandid"  return="data"}
  {foreach $data  $key $value}
    <a href='{$value["url"]}'>{$value["brandname"]}</a>
  {/foreach}
{/mo:goodsbrand}
```

参数说明：

row ="10" 导航显示的条数:默认调取 10 条记录

id ="45" 品牌 id:当 type 为"all 或者 top"的时候无效

order="brandid desc" 排序方式：id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

11）商品分类标签

标签名称： goodssort

标签功能: 商品分类标签

使用实例：

```
{mo:goodssort row="10" order="sortid" return="data"}
  {foreach $data  $key $value}
    <a href="#">{$value["sortname"]}</a>
  {/foreach}
```

```
{/mo:goodssort}
```

参数说明：

row ="10" 导航显示的条数:默认调取 10 条记录

id ="45" 类别 id:当 type 为"all 或者 top"的时候无效

order="sortid desc" 排序方式：列别 id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

12）商品类型标签

标签名称： goodstype

标签功能: 商品类型标签

使用实例：

```
{mo:goodstype row="10" id="45" order="typeid" return="data"}
  {foreach $data  $key $value}
    <a href="#">{$value["typename"]}</a>
  {/foreach}
{/mo:goodstype}
```

参数说明：

row ="10" 导航显示的条数:默认调取 10 条记录

id ="45" 类别 id

order="typeid desc" 排序方式：id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

13）热搜词标签

标签名称： hotkey

标签功能: 热搜词标签

使用实例：

```
{mo:hotkey row="10" modelid="1" order="nums desc"  return="data"}
  {foreach $data  $key $value}
    <a href='{$value["url"]}'>{$value["title"]}</a>
  {/foreach}
{/mo:goods}
```

参数说明：

row ="100" 返回数目

from ="0" 内容开始位置

order="nums desc" 按排序 ID（降序）

14）留言标签

标签名称： message

标签功能: 留言标签

使用实例：

```
{mo:comment modelid="1" row="10"  return="data"}
  {foreach $data  $key $value}
    <tr><td>内容：{$value["comment_content"]}</a> </td>
    <td>时间：{date("Y-m-d",$value["reply_time"])}  </td></tr>
  {/foreach}
{/mo:comment}
```

参数说明：

row ="100" 返回数目

modelid ="10" 留言类别 ID:必选

modelid ="1" 模型 ID

pagesize ="10" 每页显示条数

pagenum ="10" 每页显示的页码（不能小于 5）

pagevar ="page" 分页变量：可以避免同一个页面多个分页相互影响

from ="0" 内容开始位置

return ="data" 返回值名称：默认用 data

15）导航标签

标签名称： navigation

标签功能: 导航标签

使用实例：

```
{mo:navigation row="10" order="ordernum" return="data"}
{foreach $data  $key $value}
<a href='{$value["url"]}'>{$value["catname"]}</a>
{/foreach}
{/mo:article}
```

参数说明：

row ="10" 导航显示的条数:默认调取 10 条记录

order="ordernum" 排序方式：默认按排序字段升序

order="id desc" 排序方式：id 排序(降序)

order="created" 排序方式：发布时间(降序)

return ="data" 返回值名称：默认用 data

16）推荐位标签

标签名称： position

标签功能: 推荐位标签

使用实例：

```
{mo:position id="1" row="10"  return="data"}
  {foreach $data  $key $value}
    <tr><td>标题：{$value["title"]}</a> </td></tr>
  {/foreach}
{/mo:position}
```

参数说明：

row ="100" 返回数目

id ="1" 推荐位 ID

from ="0" 内容开始位置

return ="data" 返回值名称：默认用 data

17）新文章列表标签

标签名称： iarticlelist

标签功能: 调取文章列表

使用实例：

```
{izhan:iarticlelist }
{/izhan:iarticlelist}
```

参数说明：

```
'size'       => '10',       //每页多少条
'pagevar'    => 'page',     //page 变量标示符
'showpage' => '10',         //显示多少个页码
'cid'        => '0',        //栏目 ID
'type'       => '',         //类型 son(子集栏目)  ''(本身栏目)  all(本身加子集栏目)
'return'     => 'data',     //数据返回接收变量
```

'order'　　=> '',　　//排序

'thumb'　　=> 'false',　//是否调取缩略图

18）新商品分类标签

标签名称：　igoodssort

标签功能:　新商品分类标签

使用实例：

```
{izhan:igoodssort}
{/izhan:igoodssort}
```

参数说明：

'id'　　=> '0', //分类 id　，id=0　或者空　表示取全部分类

'level'　=> '0', //取到第几级

'return' => 'data', //返回标志

'type'　　=> 'all', //此参数在 id!=0 时才有效　son(当 id 不等于 0 时　只取子集) all(当 id 不等于 0 时　也包含自己)

'struct' => 'relation'　　//返回结果的数据结构形式　等于 relation 就是父子级关系的数据结构，但是只能显示 3 层

6.标签数据字典

/*****************以下内容为标签数据字典*******************/

/*****************详细的请参考数据表结构*******************/

表结构地址　/doc/*.sql 文件

1）article 文章标签

参数：id=’1’ 文章 ID

值：

`id` 编号

`url` 内容链接地址

`cid` 栏目链接地址

`catname` 栏目名称

`content` 内容

`template` 模板

`readpower` 阅读权限

`allowcomment` 评论选项

`cid` 栏目 id

title` 标题

`subtitle` 副标题

`thumb` 缩略图为数组 $thumb[‘src’]引用地址 $thumb[‘savename’]源 文件名

$thumb[‘filename’]上传后的文件名 $thumb[‘size’]文件大小

`keywords` 关键字

`description` SEO 描述

`brief` 内容简介

`source` 来源

`sortnum` 排序编号

`publishtime`　　发布时间

`hits`　　点击量

`publishuser`　　发布人

`username`　　作者

`updatetime`　　修改时间

`updateuser`　　修改人

`created`　　生成时间不能修改的

2）Content 内容标签

值：

`id`　　编号

`url`　　内容链接地址

`cid`　　栏目链接地址

`catname`　　栏目名称

`content`　　内容

`template`　　模板

`readpower`　　阅读权限

`allowcomment`　　评论选项

`cid`　　栏目 id

title`　　标题

`subtitle`　　副标题

`thumb` 缩略图为数组 $thumb[‘src’]引用地址 $thumb[‘savename’]源文件名 $thumb[‘filename’] 上传后的文件名 $thumb[‘size’]文件大小 $thumb[‘isimage’]是否是图片

`keywords` 关键字

`description` SEO 描述

`brief` 内容简介

`source` 来源

`sorttype` 排序方式|--0:默认排序,1:置顶一周，2：置顶一月，3：置顶三月,4:置顶半年,5:置顶一年

`sortnum` 排序编号

`publishopt` 发布选项 1 生成静态文件,2 动态浏览

`publishtime` 发布时间

`hits` 点击量

`publishuser` 发布人

`username` 作者

`updatetime` 修改时间

`updateuser` 修改人

`created` 生成时间不能修改的

3）contentlist 内容列表标签

值：

`id` 编号

`url`　　内容链接地址

`cid`　　栏目链接地址

`catname`　　栏目名称

`cid`　　栏目 id

title`　　标题

`subtitle`　　副标题

`thumb`　　缩略图为数组 $thumb[‘src’]引用地址 $thumb[‘savename’]源文件名 $thumb[‘filename’] 上传后的文件名 $thumb[‘size’]文件大小 $thumb[‘isimage’]是否是图片

`keywords`　　关键字

`description`　　SEO 描述

`brief`　　内容简介

`source`　　来源

`sortnum`　　排序编号

`publishopt`　　发布选项 1 生成静态文件,2 动态浏览

`publishtime`　　发布时间

`hits`　　点击量

`publishuser`　　发布人

`username`　　作者

`updatetime`　　修改时间

`updateuser`　　修改人

`created`　　生成时间不能修改的

4）articlelist 文章列表标签

值：

`id`　　编号

`url`　　内容链接地址

`cid`　　栏目链接地址

`catname`　　栏目名称

`cid`　　栏目 id

title`　　标题

`subtitle`　　副标题

`thumb`　　缩略图为数组 $thumb[‘src’]引用地址 $thumb[‘savename’]源文件名 $thumb[‘filename’] 上传后的文件名 $thumb[‘size’]文件大小

`keywords`　　关键字

`description`　　SEO 描述

`brief`　　内容简介

`source`　　来源

`sortnum`　　排序编号

`publishtime`　　发布时间

`hits`　　点击量

`publishuser`　　发布人

`username`　　作者

`updatetime`　　修改时间

`updateuser`　　修改人

`created`　　生成时间不能修改的

5）comment 评论标签

值：

`comment_id`　　评论

`username`　　评论人

`replyer`　　回复人

`comment_infoid`　被评论信息的 id

`comment_modelid`模型 id，即评论对应信息的模型 id，如果是 2 则代表是对商品的评

`comment_userid`　　评论者 ID

`comment_content`　　评论内容

`comment_status`　　评论审核状态 1 通过，2 不通过，3 待审核

`comment_time`　　评论时间

`reply_isreply`　　是否回复 1 回复 2 没回复

`reply_content`　　回复内容

`reply_userid`　　回复者 ID

`reply_time`　　回复时间

6）goods 商品标签

值：

`goodsid`　　商品 ID

`goodssn`　　商品货号

`goodsname`　　商品名

`subname`　　商品副标题

`cid`　　所属栏目 ID

`sortid`　　所属分类 ID

`brandid`　　所属品牌 ID

`is_sell`　　是否上架 1 上架 2 下架

`typeid`　　所属类型 ID 添加商品时可以不选择商品类型

`userid`　　发布人 ID

`username`　　发布人

`hits`　　点击量

`isbest`　　是否精品 1 是,2 不是

`isnew`　　是否新品 1 是,2 不是

`ishot`　　是否热销 1 是,2 不是

`isspecial`　　是否特卖 1 是,2 不是

`keywords`　　关键字

`brief`　　商品简介

`content`　　商品介绍

`marketprice`　　市场价

`shopprice`　　商城价即所谓的优惠价

`unit`　　计量单位

`publishtime`　　上架时间

`publishopt`　　发布选项 1,生成静态页 2 动态浏览

`iscomment`　　评论权限 1 允许评论 ，2 不允许评论

`goodstpl`　　商品模板

`modification`　　最后修改时间

`sortname`　　分类名称

`brandname`　　品牌名称

`brandurl`　　品牌链接

`brandlogo`　　品牌 logo

`typename`　　类型名称

`image`　　商品相册（数组：同文章缩略图）

`attr`　　属性（$v[‘attr’][‘地区’]）

`url`　　链接

`curl`　　栏目链接

7）Category 栏目标签

值：

`id`　　分类 ID

`pid`　　分类父 ID

`url`　　链接

`catname`　　分类名称

`model`　　内容模型 ID

`filepath`　　文件保存目录

`indextpl`　　栏目首页模板

`columntpl`　　栏目列表页模板

`contenttpl`　　内容页模板

`isnav`　　是否在导航显示 1 在导航显示, 2 不再导航显示

`columnoption`　　栏目列表选项 1 链接到默认页, 2 链接到列表第一页, 3 使用动态页

`dirpath`　　目录保存位置 1 上级目录, 2CMS 根目录, 3 站点根目录

`columnattr`　　栏目属性 1 最终列表栏目(允许在本栏目发布文档，并生成文档列表), 2 频道封面(栏目本身不允许发布文档),3 外部连接(在 文件保存目录 处填写网址)

`columncross`　　栏目交叉,仅适用[最终列表栏目],1 不交叉,2 自动获取同名栏目内容,3 手工指定交叉栏目 ID(用逗号分开)

`crossid`　　栏目交叉交叉 id 以逗号分隔的

`ordernum`　　排序

`seo_title`　　SEO 标题

`seo_keywords`　　SEO 关键字

`seo_description`　　SEO 描述

`created`　　创建时间

8）Navigation 导航标签

值：

`id`　　分类 ID

`pid`　　分类父 ID

`url`　　链接

`catname`　　分类名称

`model`　　内容模型 ID

`filepath`　　文件保存目录

`indextpl`　　栏目首页模板

`columntpl`　　栏目列表页模板

`contenttpl`　　内容页模板

`isnav`　　是否在导航显示 1 在导航显示, 2 不再导航显示

`columnoption`　　栏目列表选项 1 链接到默认页, 2 链接到列表第一页, 3 使用动态页

`dirpath`　　目录保存位置 1 上级目录, 2CMS 根目录, 3 站点根目录

`columnattr`　　栏目属性 1 最终列表栏目(允许在本栏目发布文档，并生成文档列表), 2 频道封面(栏目本身不允许发布文档),3 外部连接(在 文件保存目录 处填写网址)

`columncross`　　栏目交叉,仅适用[最终列表栏目],1 不交叉,2 自动获取同名栏目内容,3 手工指定交叉栏目 ID(用逗号分开)

`crossid`　　栏目交叉交叉 id 以逗号分隔的

`ordernum`　　排序

`seo_title`　　SEO 标题

`seo_keywords`　　SEO 关键字

`seo_description` SEO 描述

`created` 创建时间

9）Position 推荐位标签

值：

`id` mediumint(8) ID

`headline` 推荐位标题

`cat_id` 栏目 ID

`ag_id` 关联文章 ID 或者商品 ID

`pos_id` 推荐位 ID

`model_id` 模型 ID

`alter_time` 操作时间

`sortby` 排序

`pos_img` 推荐位图片地址

`pos_info` 推荐位描述

`title` 推荐内容标题名称

`info` 推荐内容的相关信息：有文章和商品 $v[‘info’][‘title’]文章名称

$v[‘info’][‘goodsname’] 商品名称

10）Message 留言标签

值：

id 留言管理 id

title 内容标题

typeid 留言类别 id(mix_model 表中的 Id)

username 留言人

replymember 回复人 id

replyer 回复人

replytime 回复时间

isreply 是否回复(是：1，否：2)

ischeck 是否审核（1：审核成功，2:未审核,不通过:3）replayinfor 回复内容

leavetime 留言时间