

目录

1.导入到项目	2
2.生成代码.....	2
2.1 代码生成器的 url.....	2
2.2 生成的文件.....	3
2.3 生成文件的位置	3
2.4 功能大全.....	3
2.5 数据表中字段定义.....	4
2.6 图片增删改功能	4
2.7 表单验证规则	5
2.8 字段类型和表单输入类型的关系	6
2.9 列表页搜索字段的生成.....	7
2.10 代码生成举例.....	7
2.11 生成 RABC.....	11
3.公共函数.....	12
3.1 图片相关.....	12
3.2 coreseek 相关.....	13
3.3 字符串相关	13
3.4 发送 Email.....	15
3.5 其他.....	15
4.DYCModel 类方法.....	16
4.1 表单验证.....	17
4.2 数据库便捷查询	18
4.3 回收站相关	18
4.4 图片相关	19
4.5 数据表锁.....	19
5.生成的模型类方法	19
5.1 无限分类相关	19
5.2 非无限分类相关	21
5.3 扩展属性类.....	22
5.4 其他.....	22

ThinkPHP 后台代码生成器使用说明

1. 导入到项目

下载后解压，将以下文件和文件夹复制到项目根目录下。

- ▶ Admin
- ▶ Common
- ▶ Gii
- ▶ Model
- ▶ Public
- ▶ ThinkPHP
- ▶ index.php

Linux 和 Mac OS 请执行以下命令修改权限。

```
chmod -R 777 你的项目根目录
```

在 index.php 中添加以下常量。

```
define('BASE', '/'); //网站根目录

define('PUB', BASE.'Public/');
define('ADMIN_PUB', BASE.'Admin/View/Public/');
define('HOME_PUB', BASE.'Home/View/Public/');
define('GII_PUB', BASE.'Gii/View/Public/');
define('COMMON', BASE.'Common/Common/');
```

在项目根目录/Common/Conf/config.php 添加以下配置，值为空的需要自己设置。

```
/* 数据库 */
'DB_TYPE'          => 'mysql',    // 数据库类型
'DB_HOST'          => '',         // 服务器地址
'DB_NAME'          => '',         // 数据库名
'DB_USER'          => '',         // 用户名
'DB_PWD'           => '',         // 密码
'DB_PORT'          => '',         // 端口
'DB_PREFIX'        => '',         // 数据库表前缀

'TMPL_ENGINE_TYPE' => 'Smarty',   // 模板引擎

'MODULE_ALLOW_LIST' => array('Admin', 'Gii'), // 允许访问模块

'SPHINX_HOST'       => '',        // coreseek服务器ip或主机名
'SPHINX_PORT'       => '',        // coreseek服务器端口

'UPLOAD_PATH'       => '',        // 上传图片根目录,相对于index.php
'LOCK_PATH'         => '',        // 数据表锁文件目录,相对于index.php

/* 发邮件 */
'MAIL_USER'         => '',        // 用户名
'MAIL_PWD'          => '',        // 密码
'MAIL_HOST'         => '',        // 邮件服务器ip或主机名

/* 验证码 */
'CAPTCHA_OPTION'    => array(
    'imageH'  => 30,
    'imageW'  => 100,
    'length'  => 4,
    'fontSize' => 14
)
```

2. 生成代码

2.1 代码生成器的 url

<http://你的主机名/index.php/Gii/Index/index>

2.2 生成的文件

代码生成器可以根据指定的数据表生成如下文件：

--控制器类(controller.php)

--模型类(model.php)

--添加页(add.html)

--修改页(edit.html)

--列表页(index.html)

--详情页(detail.html)

--回收站(trashcan.html)

如果指定了和数据表关联的图集表，可以生成：

--图集列表(gallery.html)

--图集回收站(gallerytrashcan.html)

如果指定了和数据表关联的扩展属性表，可以生成对应的 `extend.html` 文件，既可以浏览所有扩展属性，又可以修改，添加，删除扩展属性。

2.3 生成文件的位置

生成的 `model.php` 存放在项目根目录/`Model` 下，`controller.php` 存放在项目根目录/指定模块/`Controller` 下，静态页面存放在项目根目录/指定模块/`View`/控制器名。如果在生成器页面没有指定模块，默认会将 `Admin` 作为指定模块。

2.4 功能大全

--支持增删改查操作，查询分为列表页和每条记录的详情页

--支持放入回收站和从回收站恢复

--支持批量删除，批量放入回收站，批量从回收站恢复

--支持图集和扩展属性的增删改查

--支持外键。当执行放入回收站，从回收站恢复或删除的操作时，外键表中相应的记录也会放入回收站，从回收站恢复或删除；列表页可以跳转到外键表列表页查看对应的记录，如下图，主键表是分类，外键表是商品

🏠 [首页](#) > [分类管理](#)

[+ 新增](#) [回收站](#)

☐ 全选 [批量移到回收站](#)

名称	操作
<input type="checkbox"/> + 服装	详情 商品 属性 修改 移到回收站

--为了防止跨站伪造表单，POST 提交数据需添加 `_csrf` 字段进行验证，如下所示。

```
<input id="_csrf" type="hidden" name="_csrf" value="{$_smarty.session.csrf}">
```

2.5 数据表中字段定义

如果存在原图字段和相应的缩略图字段，取原图字段的注释作为图片字段的中文名。

每个数据表都需要创建如下字段。

```
`id` int(10) unsigned NOT NULL AUTO_INCREMENT
```

```
`rank` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序'
```

```
`is_trash` enum('0','1') NOT NULL DEFAULT '0' COMMENT '在回收站'
```

如果用户没有定义这些字段，系统会自动创建。

另外除了图集表和扩展属性表，往往需要创建一个 name 字段，用来标识记录的名称，如下。

```
`name` varchar(用户自定义长度) NOT NULL COMMENT '名称'
```

如果用户没有定义 name 字段，但在代码生成器页面勾选了 name 字段，如下图，系统会自动创建，varchar 长度默认为 20。

[生成RABC](#)

表名

☒ name 字段

如果要使用 coreseek 进行关键词搜索，请添加 is_update 字段，用于支持 update 后更新数据源，如下。

```
`is_update` tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '更新'
```

如果用户没有定义 is_update 字段，但在代码生成器勾选了关键词搜索，如下图，系统会自动创建。

☒ name 字段
☐ 此表是无限分
☒ ~~关键词搜索~~

数据表字段的注释(comment)指定字段的中文名。

2.6 图片增删改功能

生成的模型类(model.php)已实现了增删改操作中图片的增删改功能，用户只需在数据表中定义相关的字段。图片字段名称须以小写的 logo, pic 或 img 结尾，数据类型指定为 varchar，如下。

```
`logo` varchar(60) NOT NULL COMMENT '图片'
```

```
`xxxlogo` varchar(60) NOT NULL COMMENT '图片'
```

```
`pic` varchar(60) NOT NULL COMMENT '图片'
```

```
`xxxpic` varchar(60) NOT NULL COMMENT '图片'
```

```
`img` varchar(60) NOT NULL COMMENT '图片'
```

```
`xxximg` varchar(60) NOT NULL COMMENT '图片'
```

缩略图字段的名称还应满足如下正则表达式。下划线之前的数依次指定缩略图的长和宽。

```
t[0-9]+_[0-9]+_.*(logo|pic|img)
```

缩略图字段名示例。

```
`t700_600_logo` varchar(60) NOT NULL COMMENT '长 700 宽 600 的缩略图'
```

```
`t350_300_logo` varchar(60) NOT NULL COMMENT '长 350 宽 300 的缩略图'
```

在原图和缩略图中，若要指定后台显示的图片，请在字段注释加上后缀#b，如下。

```
`t50_50_logo` varchar(60) NOT NULL COMMENT '图片#b'
```

那么后台就会显示 50*50 的缩略图。

2.7 表单验证规则

1. 如果字段指定为 NOT NULL 并且没有指定默认值，将会生成 require 验证规则，如下。

```
array('name', 'require', '名称不能为空', 1)
```

2. 如果字段添加 UNIQUE 索引，将会生成 unique 附加规则，如下。

```
array('name', '', '名称已存在', 2, 'unique')
```

3. 对于 char 和 varchar 数据类型，如果指定长度，将会生成 length 附加规则，如下。

```
array('name', '1,20', '名称不能超过 20 个字符', 2, 'length')
```

4. 如果数据类型为整型(tinyint, smallint, mediumint, int, bigint), 将会生成 number 验证规则，如下。

```
array('rank', 'number', '排序必须是数字', 2)
```

5. 如果数据类型为 float, double 或 decimal, 将会验证输入是否为整数或小数，如下。

```
array('price', 'checkDecimal', '价格必须是整数或小数', 2, 'callback')
```

6. 如果数据类型为 float(10, 2), double(10, 2) 或 decimal(10, 2), 将会生成 currency 验证规则，如下。

```
array('price', 'currency', '价格非法输入', 2)
```

7. 如果数据类型被指定为 unsigned, 将会验证输入的值是否为非负数，如下。

```
array('price', 'checkUnsigned', '价格不能小于 0', 2, 'callback')
```

8. 另外对于特殊命名的字段，将会生成特定的验证规则。注意字段名均为小写。

(1) 如果字段名为 email, 将会验证 Email 格式是否正确，如下。

```
array('email', 'email', 'Email 格式不正确', 2)
```

(2) 如果字段名为 url, 将会验证 url 格式是否正确，如下。

```
array('url', 'checkUrl', '网址格式不正确', 2, 'callback')
```

(3) 如果字段名为 mobile, 将会验证手机号码格式是否正确，如下。

```
array('mobile', 'checkMobile', '手机号码格式不正确', 2, 'callback')
```

(4) 如果字段名为 tel, 将会验证固定电话号码格式是否正确，如下。

```
array('tel', 'checkTel', '固定电话号码格式不正确, 参照 0XXX-XXXXXXX,  
400XXXXXXX, 800XXXXXXX', 2, 'callback')
```

(5) 如果字段名为 phone，将会验证输入的固定电话号码或手机号码格式是否正确，如下。

```
array('phone', 'checkPhone', '电话号码格式不正确,可输入手机或固话,固话参照  
0XXX-XXXXXXX,400XXXXXXX,800XXXXXXX', 2, 'callback')
```

(6) 如果字段名为 idcard，将会验证输入的身份证号格式是否正确，如下。

```
array('idcard', 'checkIdcard', '身份证号格式不正确', 2, 'callback')
```

(7) 如果字段名为 password，将会在表单页 (add.html 和 edit.html) 生成确认密码字段 repassword，同时验证确认密码和设定密码输入是否一致，如下。

```
array('repassword', 'password', '两次输入的密码不一致', 1, 'confirm')
```

(8) 如果要验证输入的验证码，请在代码生成器页面表单验证码字段勾选 add 或 edit，如下图，将会分别在 add.html 或 edit.html 表单生成验证码字段，如果勾选 add，那么会在添加操作时验证输入的验证码，如果勾选 edit，那么会在修改操作时验证输入的验证码。

表单验证码

☐ add ☐ edit

模块

生成文件

☒ controller.pl

生成的验证规则分别如下。

```
array('captcha', 'checkCaptcha', '验证码错误', 1, 'callback', 1)  
array('captcha', 'checkCaptcha', '验证码错误', 1, 'callback', 2)
```

2.8 字段类型和表单输入类型的关系

1. 如果字段名以小写的 logo, pic 或 img 结尾，将被视为图片字段，如前面 2.6 节所述，会生成 `<input type="file">`。如果存在相应的缩略图字段，不会再生成上传缩略图的 `<input type="file">`。例如，如果数据表定义了 logo 和 t700_700_logo 两个字段，只会生成字段 logo 对应的 `<input type="file">`，不会生成 t700_700_logo 对应的 `<input type="file">`。
2. 如果字段类型定义为 enum(''.xxx')，将会生成 `<input type="checkbox">`。
3. 如果字段类型定义为整型 (tinyint, smallint, mediumint, int, bigint) 且在代码生成器页面指定为外键 (如下图)，将会生成 `<select>`。

表名

☒ name 字段

☐ 此表是无限分类

☐ 关键词搜索

外键

多个外键请用&隔开

4. 如果字段类型定义为 enum (非前面第 2 点所述情形)，将会生成 `<select>`。

5. 如果字段类型定义为 text，将会生成<textarea>，使用的是 UEditor 编辑器。
6. 如果字段名为 password，将被视为输入密码字段，如前面 2.7 节所述，会生成设置密码的<input type="password">和确认密码的<input type="password">。
7. 如果字段不满足上述 6 点情形，且名称不为 id，is_trash 或 is_update，将会生成<input type="text">。

2.9 列表页搜索字段的生成

1. 如果在代码生成器页面勾选关键词搜索，如下图，将会在列表页生成关键词搜索框。

☒name字段
☐此表是无限分
☒关键词搜索

2. 如果字段被定义为索引，并符合以下条件之一，将会在列表页生成搜索字段，输入类型为<select>或<input type="checkbox">。字段名为 id，is_trash 或 is_update 不会生成搜索字段。

(1) 字段类型定义为 enum。

(2) 已在代码生成器页面指定为外键，如下图。

表名
 t_goods
☒name字段
☐此表是无限分类
☐关键词搜索
 外键
 cat_id&brand_id
 多个外键请用&隔开

2.10 代码生成举例

1. 无限分类表

定义无限分类表 t_cat 如下。

```

CREATE TABLE `t_cat` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(20) NOT NULL COMMENT '名称',
  `parent_id` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '父类',
  `rank` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序',
  `is_trash` enum('0','1') NOT NULL DEFAULT '0' COMMENT '在回收站',
  `is_floor` enum('', '推荐到楼层') NOT NULL DEFAULT '' COMMENT '推荐到楼层',
  PRIMARY KEY (`id`),
  KEY `parent_id` (`parent_id`),
  KEY `is_trash` (`is_trash`)
) ENGINE=InnoDB AUTO_INCREMENT=56 DEFAULT CHARSET=utf8mb4 COMMENT='分类';
  
```

代码生成器页面填写如下。

表名
 t_cat
☒name字段
☒此表是无限分类 勾选这个，勾选之后不能使用关键词搜索
☐关键词搜索
 外键
 多个外键请用&隔开
 此表主键作为外键的表
 t_goods&t_attr 表t_goods和表t_attr都将cat_id作为外键

定义商品图集表 t_goods_img 如下。

```
CREATE TABLE `t_goods_img` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `rank` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序',
  `is_trash` enum('0','1') NOT NULL DEFAULT '0' COMMENT '在回收站',
  `goods_id` int(10) unsigned NOT NULL COMMENT '商品',
  `logo` varchar(60) NOT NULL COMMENT 'Logo',
  `t350_350_logo` varchar(60) NOT NULL COMMENT 'Logo',
  `t150_150_logo` varchar(60) NOT NULL COMMENT 'Logo',
  `t50_50_logo` varchar(60) NOT NULL COMMENT 'Logo#b',
  PRIMARY KEY (`id`),
  KEY `goods_id` (`goods_id`),
  KEY `is_trash` (`is_trash`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COMMENT='商品图集';
```

定义商品扩展属性表如下。

```
CREATE TABLE `t_goods_attr` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `goods_id` int(10) unsigned NOT NULL COMMENT '商品',
  `attr_id` int(10) unsigned NOT NULL COMMENT '属性',
  `value` varchar(30) NOT NULL COMMENT '属性值',
  `rank` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序',
  `is_trash` enum('0','1') NOT NULL DEFAULT '0' COMMENT '在回收站',
  PRIMARY KEY (`id`),
  KEY `goods_id` (`goods_id`),
  KEY `attr_id` (`attr_id`),
  KEY `is_trash` (`is_trash`)
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8mb4 COMMENT='商品属性';
```

定义商品会员价格表如下。

```
CREATE TABLE `t_memberprice` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(20) NOT NULL COMMENT '名称',
  `goods_id` int(10) unsigned NOT NULL COMMENT '商品',
  `memberlevel_id` int(10) unsigned NOT NULL COMMENT '会员级别',
  `price` decimal(10,2) unsigned NOT NULL COMMENT '价格',
  `rank` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序',
  `is_trash` enum('0','1') NOT NULL DEFAULT '0' COMMENT '在回收站',
  PRIMARY KEY (`id`),
  KEY `goods_id` (`goods_id`),
  KEY `memberlevel_id` (`memberlevel_id`),
  KEY `is_trash` (`is_trash`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COMMENT='会员价格';
```

为商品图集表 t_goods_img 生成模型。代码生成器页面填写如下。

表名	<input type="text" value="t_goods_img"/>
<input type="checkbox"/> name字段 图集表没有name字段	
<input type="checkbox"/> 此表是无限分类	
<input type="checkbox"/> 关键词搜索	
外键	<input type="text" value="goods_id"/>
多个外键请用&隔开	
此表主键作为外键的表	<input type="text"/>
多个表请用&隔开	
图集表	<input type="text" value="勾选了图集表此处无需填写"/>
<input checked="" type="checkbox"/> 此表是图集表	
扩展属性表	<input type="text" value="勾选了图集表此处无需填写"/>
多个表请用&隔开	
和扩展属性表关联的属性表	<input type="text" value="勾选了图集表此处无需填写"/>
多个表请用&隔开,顺序和扩展属性表的顺序对应	
扩展属性值字段	<input type="text" value="勾选了图集表此处无需填写"/>
多个字段请用&隔开,顺序和扩展属性表的顺序对应	

扩展属性值字段
☒ 勾选了图集表此处无需填写

多个字段请用&隔开,顺序和扩展属性表的顺序对应
☐ 此表是扩展属性表 ☒ 图集表和扩展属性表不能同时勾选

每页显示记录数
10

表单验证码
☐ add ☐ edit

模块

生成文件 ☐ 只勾选model.php 其他文件不勾选
☐ controller.php ☒ model.php ☐ add.html ☐ index.html ☐ trashcan.html ☐ edit.html ☐ detail.html ☐ gallery.html ☐ gallerytrashcan.html ☐ extend.html ☐ 全选

生成

为商品扩展属性表 t_goods_attr 生成模型。代码生成器页面填写如下。为会员价格 t_memberprice 生成模型的方法类似。会员价格也可以当作扩展属性，因为商品和会员等级是多对多的关系。存在多对多的两个外键的表均可当作扩展属性表。

表名
t_goods_attr

☐ name字段 扩展属性表没有name字段
☐ 此表是无限分类
☐ 关键词搜索

外键
goods_id&attr_id

多个外键请用&隔开
此表主键作为外键的表

多个表请用&隔开
图集表
☒ 勾选了扩展属性表此处无需填写
☐ 此表是图集表 图集表和扩展属性表不能同时勾选

扩展属性表
☒ 勾选了扩展属性表此处无需填写

多个表请用&隔开
和扩展属性表关联的属性表
t_attr

多个表请用&隔开,顺序和扩展属性表的顺序对应
扩展属性值字段
value

多个字段请用&隔开,顺序和扩展属性表的顺序对应
☒ 此表是扩展属性表
每页显示记录数 扩展属性不做分页显示

表单验证码
☐ add ☐ edit

模块

生成文件 ☐ 只勾选model.php 其他文件不勾选
☐ controller.php ☒ model.php ☐ add.html ☐ index.html ☐ trashcan.html ☐ edit.html ☐ detail.html ☐ gallery.html ☐ gallerytrashcan.html ☐ extend.html ☐ 全选

生成

最后生成商品表 t_goods 的后台管理代码。代码生成器页面填写如下。

表名
t_goods

☒ name字段
☐ 此表是无限分类
☒ 关键词搜索

外键
cat_id

多个外键请用&隔开
此表主键作为外键的表
t_goods_img&t_goods_attr&t_memberprice

多个表请用&隔开
图集表
t_goods_img

☐ 此表是图集表
扩展属性表
t_goods_attr&t_memberprice 会员价格也可以当作扩展属性 因为商品和会员等级是多对多的关系 存在多对多的两个外键的表均可以当作扩展属性表

扩展属性表

t_goods_attr&t_memberprice 会员价格也可以当作扩展属性 因为商品和会员等级是多对多的关系 存在多对多的两个外键的表均可以当作扩展属性表

多个表请用&隔开

和扩展属性表关联的属性表

t_attr&t_memberlevel t_goods_attr的外键attr_id是t_attr的主键 t_memberlevel的外键memberlevel_id是t_memberlevel的主键

多个表请用&隔开,顺序和扩展属性表的顺序对应

扩展属性值字段

value&price t_goods_attr的属性值字段是value t_memberprice的会员价格字段是price

多个字段请用&隔开,顺序和扩展属性表的顺序对应

☐此表是扩展属性表

每页显示记录数

10

表单验证码

☐add ☐edit

模块

生成文件

☒controller.php ☒model.php ☒add.html ☒index.html ☒trashcan.html ☒edit.html ☒detail.html ☒gallery.html ☒gallerytrashcan.html ☒extend.html ☒全选

生成

生成的商品列表页如下。

首页 > 商品管理

关键字:

分类:

品牌:

☐精品 ☐热门 ☐新品 ☐推荐到楼层

☐促销

+ 新增 回收站

☐全选 批量移到回收站

名称	操作
<input type="checkbox"/> iPhone 6s	详情 商品图集 商品属性 会员价格 商品数量 购物车 商品印象 修改 移到回收站
<input type="checkbox"/> 苹果 6s Plus	详情 商品图集 商品属性 会员价格 商品数量 购物车 商品印象 修改 移到回收站
<input type="checkbox"/> HM连衣裙	详情 商品图集 商品属性 会员价格 商品数量 购物车 商品印象 修改 移到回收站
<input type="checkbox"/> ZARA上衣	详情 商品图集 商品属性 会员价格 商品数量 购物车 商品印象 修改 移到回收站

2.11 生成 RABC

点击代码生成器页面的生成 RABC，如下。

[生成RABC](#)

表名

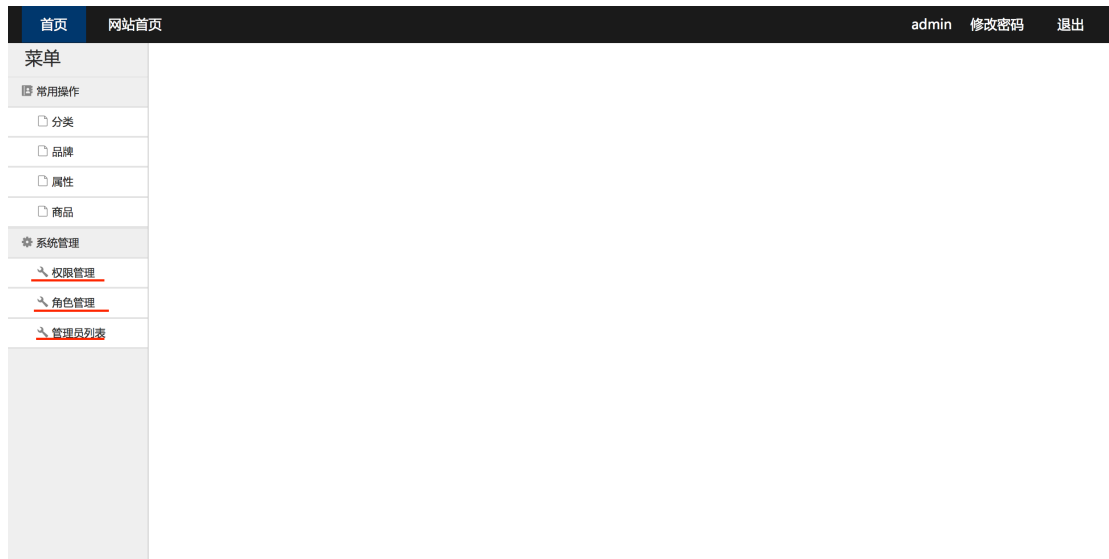
系统会创建超级管理员用户 admin，拥有所有权限，初始密码 admin。

后台首页地址如下。

http://你的主机名/Admin

如果没有登录，将会进入登录页面，登录成功后才能进入后台管理系统。

进入后台管理系统后，左侧菜单栏有权限管理、角色管理、管理员列表三项，如下图，默认只有 admin 用户才具有其操作权限。



3. 公共函数

公共函数存放在项目根目录/Common/Common/function.php。

3.1 图片相关

```
function uploadPic($name, $dir, $thumb, $maxSize = 409600, $exts = array('jpg', 'jpeg', 'png', 'gif'))
```

上传图片

@param string \$name 表单中上传图片的字段名

@param string \$dir 上传后图片存在的路径 相对于上传图片存放的根目录

@param array \$thumb 生成缩略图的信息

每个元素是个一维数组 包含三个元素

第一个元素指定缩略图的长

第二个元素指定缩略图的宽

第三个元素指定缩略图的裁剪类型 可缺省 默认为 Image::IMAGE_THUMB_CENTER

例如

```
array(array(300, 200, Image::IMAGE_THUMB_SCALE), array(30, 20))
```

表示生成两张缩略图 第一张 300*200 等比例缩放类型 第二张 30*20 居中裁剪类型

@param int \$maxSize [optional] 上传图片限制的大小 字节为单位 默认最大为 400KB

@param array \$exts [optional] 上传图片允许接受的后缀名 默认为 jpg, jpeg, png, gif

@return array 若返回数组\$info

\$info['status']是上传的状态信息 true 表示上传成功 false 表示上传失败

\$info['images']是个数组 存放上传成功后原图和所有缩略图存放的路径 相对于上传图片根目录

\$info['error']是上传图片失败时的错误信息

```
function picUrl($pic)
```

获取图片的绝对路径

@param string \$pic 从数据库中获取的图片路径 相对于存放图片的根目录

return string 获取到的图片的绝对路径

```
function showPic($pic, $cls = '', $alt = '')
```

生成 img 标签

@param string \$pic 从数据库中获取的图片路径 相对于存放图片的根目录

@param string \$cls [optional] img 标签中指定的 class 属性值

@param string \$alt [optional] 占位图

@return string img 标签

3.2 coreseek 相关

```
function keySearch($key, $index, $ucfg = array())
```

使用 coreseek 进行关键词搜索

@param string \$key 要搜索的关键词

@param string \$index coreseek 数据源名称

@param array \$ucfg [optional] coreseek 搜索选项

\$ucfg['host']指定要连接的 coreseek 服务器的 ip 地址或主机名

\$ucfg['port']指定要连接的 coreseek 服务器的端口

@return array 所有符合搜索条件的记录的 id

```
function updateKeySearch($index, $id, $ucfg = array())
```

对数据库进行 update 操作后更新 coreseek 数据源

@param string \$index coreseek 数据源名称

@param int \$id 进行 update 操作的记录的 id

@param array \$ucfg [optional] coreseek 搜索选项

\$ucfg['host']指定要连接的 coreseek 服务器的 ip 地址或主机名

\$ucfg['port']指定要连接的 coreseek 服务器的端口

3.3 字符串相关

```
function contain($str, $needle)
```

判断字符串\$str 是否包含指定的子字符串\$needle

@param string \$str

@param string \$needle

@return bool true 表示包含 false 表示不包含

```
function hasPrefix($str, $needle)
```

判断字符串\$str 是否包含指定的前缀\$needle

```
@param string $str
```

```
@param string $needle
```

```
@return bool true 表示包含 false 表示不包含
```

```
function trimPrefix($str, $needle)
```

去除字符串\$str 中指定的前缀\$needle

```
@param string $str
```

```
@param string $needle
```

```
@return string 去除前缀后的字符串
```

```
function hasSuffix($str, $needle)
```

判断字符串\$str 是否包含指定的后缀\$needle

```
@param string
```

```
@param string
```

```
@return bool true 表示包含 false 表示不包含
```

```
function trimSuffix($str, $needle)
```

去除字符串\$str 中指定的后缀\$needle

```
@param string $str
```

```
@param string $needle
```

```
@return string 去除后缀后的字符串
```

```
function getClassNameFromTable($table)
```

由数据表名获得相应的类名

```
@param string $table 数据表名
```

```
@return string 类名
```

```
function getTableChineseName($table)
```

获得数据表中文名

```
@param string $table 数据表名
```

```
@return string 表中文名
```

```
function filterUrl($key, $url = __SELF__)
```

过滤 url 中指定的请求参数

```
@param string $key 要过滤的请求参数的 key
```

```
@param string $url [optional] 默认当前 url
```

```
@return string 过滤后的 url
```

3.4 发送 Email

```
function sendMail($address, $subject, $body)
```

发送 Email

@param string \$address 收件人

@param string \$subject 主题

@param string \$body 内容

@return bool true 表示发送成功 false 表示发送失败

3.5 其他

```
function getSessionExpire()
```

获取 session 的有效期限

@return string session 的有效期限

```
function backUrl($default = BASE)
```

获得要返回的 url 要返回的 url 存放在 session 中 如果为空 返回网站首页

@param string \$default

@return string [optional] 默认是网站首页

```
function removeXSS($data)
```

UEditor 编辑器 XSS 过滤

@param string \$data UEditor 编辑器的文本内容

@return string 过滤后的文本内容

```
function val($val, $default = '')
```

如果值为空 返回原值 否则返回默认值

@param string \$val 原值

@param string \$default [optional] 默认值

@return string

```
function setValue($key, $data = '')
```

用于表单字段值的设置

如果\$_REQUEST[\$key]不为空 返回\$_REQUEST[\$key] 否则返回指定的值

用法举例

```
<input type="text" name="name" value="<?=setValue('name', $data['name'])?>">
```

```
<textarea name="goods_desc"><?=setValue('goods_desc',
```

```
$data['goods_desc'])?></textarea>
```

@param string \$key

@param string \$data [optional]

\$_REQUEST[\$key]为空时指定的值

@return mixed|string

function setSelectValue(\$cur, \$key, \$data = '')

用于设置<select>标签中当前的<option>是否被选中

如果\$_REQUEST[\$key]不为空 将<select>字段值设置为\$_REQUEST[\$key] 否则设置为指定的值

用法举例

```
<select name="cat_id">
<option value="">请选择</option>
<?php foreach($catData as $v2):?>
<option value="<?=$v2['id']?>" <?=setSelectValue($v2['id'],' cat_id',
$data['cat_id'])?><?=$v2['name']?></option>
<?php endforeach;?>
</select>
```

@param string \$cur 当前<option>选项的值

@param \$key

@param string \$data [optional]

\$_REQUEST[\$key]为空时指定的值

@return string

function setCheckValue(\$cur, \$key, \$data = '')

用于设置当前的单选按钮或复选框是否被选中

如果\$_REQUEST[\$key]不为空 将所在的字段值设置为\$_REQUEST[\$key] 否则设置为指定的值

用法举例

```
<input type="radio" name="is_best" value="精品" <?=setCheckValue('精品',' is_best',
$data[' is_best'])?>>
<input type="checkbox" name="is_best" value="精品" <?=setCheckValue('精品',' is_best',
$data[' is_best'])?>>
```

@param string \$cur 当前单选按钮或复选框的值

@param \$key

@param string \$data [optional]

\$_REQUEST[\$key]为空时指定的值

@return string

4.DYCModel 类方法

代码生成器生成的模型类继承自 DYCModel。

4.1 表单验证

```
function checkDecimal($arg)
```

验证值是否是整数或浮点数

@param string \$arg 待验证的值

@return bool true 是 false 不是

```
public function checkUnsigned($arg)
```

验证值是否为非负数

@param string \$arg 待验证的值

@return bool true 非负数 false 负数

```
public function checkMobile($arg)
```

验证手机号码格式是否正确

@param string \$arg 待验证的手机号码

@return bool true 正确 false 不正确

```
public function checkTel($arg)
```

验证固定电话格式是否正确

@param string \$arg 待验证的固定电话

@return bool true 正确 false 不正确

```
public function checkPhone($arg)
```

验证电话号码格式是否正确

@param string \$arg 待验证的电话号码 可以是手机号码或固定电话

@return bool true 正确 false 不正确

```
public function checkUrl($arg)
```

验证 url 格式是否正确

@param string \$arg 待验证的 url

@return bool true 正确 false 不正确

```
public function checkIdcard($arg)
```

验证身份证号格式是否正确

@param string \$arg 待验证的身份证号

@return bool true 正确 false 不正确

```
public function checkCaptcha($arg)
```

验证码格式是否正确

@param string \$arg 用户输入的验证码

@return bool true 正确 false 不正确

4.2 数据库便捷查询

```
public function getFromWhere($where = '', $fields = '', $order = '')
```

数据库查询操作

@param string|array \$where [optional] <p>

查询条件

@param string \$fields [optional] 查询字段 默认所有字段

@param string \$order [optional] 排序 如'id asc'

@return array 结果集如果只有一个字段 返回的是一维数组 如果有多个字段 返回的是二维数组

```
public function getIDs($where = '', $order = 'id desc')
```

按指定的顺序获取符合条件的所有记录的 id

@param string|array \$where [optional] 查询条件

@param string \$order [optional] 排序 如'id asc'

@return array

```
public function getNamesFromWhere($where = '', $order = '')
```

按指定的顺序获取符合条件的所有记录的 name

@param string|array \$where [optional] 查询条件 默认没有任何条件

@param string \$order [optional] 排序 如'id asc'

@return array

```
public function getFieldByID($id, $field)
```

获取指定 id 的记录中指定字段的值

@param \$id 指定 id

@param \$field 指定字段

@return string|null 如果字段值不为空 返回字段值 否则返回 null

```
public function getName($id)
```

获取指定 id 记录的名字

@param \$id 指定 id

@return string|null 如果 name 不为空 返回 name 否则返回 null

4.3 回收站相关

```
public function trash($where)
```

将符合条件的记录放入回收站

@param \$where string|array 条件

@return bool true 成功 false 失败

```
public function recover($where)
```

将符合条件的记录从回收站恢复

@param \$where string|array 条件

@return bool true 成功 false 失败

4.4 图片相关

```
public function deleteImgs($ims)
```

删除数组中指定的所有图片

@param array \$ims 存放要删除的图片的路径 相对于上传图片根目录

4.5 数据表锁

```
public function lock($lock = true)
```

数据表加锁/解锁

@param bool \$lock [optional] true 加锁 false 解锁

@return bool true 成功 false 失败

5.生成的模型类方法

5.1 无限分类相关

```
public function tree($parent_id = 0, $fields = '', $order = '')
```

获取无限分类树(没放到回收站的)

非递归包含 也就是说子分类和父分类是同维度的数组元素

@param int \$parent_id [optional] 父分类 id

@param string \$fields [optional] 查询字段 默认全部字段

@param string \$order [optional] 排序 默认'rank asc'

@return array 无限分类树

```
public function treeInTrash($fields = '', $order = '')
```

获取无限分类树(已放到回收站的)

非递归包含 也就是说子分类和父分类是同维度的数组元素

@param string \$fields [optional] 查询字段 默认全部字段

@param string \$order [optional] 排序 默认'rank asc'

@return array 无限分类树

```
public function treeWithChildren($parent_id = 0, $fields = '', $order = '')
```

获取递归包含的无限分类树(没放到回收站的)

递归包含指的是子分类是父分类的一个数组元素

@param int \$parent_id [optional] 父分类 id

@param string \$fields [optional] 查询字段 默认全部字段

```

@param string $order [optional] 排序 默认'rank asc'
@return array 无限分类树
public function treeIDs($parent_id = 0, $order = '', $trash = false)
获取所有子分类 id
@param int $parent_id [optional] 父分类 id
@param string $order [optional] 排序 默认'rank asc'
@param bool $trash [optional] 是否已放入回收站
@return array

public function treeWithoutSubTree($subid, $parent_id = 0, $fields = '', $order = '')
获取不包含指定子树的分类树(没放到回收站的)
@param int $subid 指定子树根元素 id
@param int $parent_id [optional] 要获取的分类树的父分类 id
@param string $fields [optional] 查询字段 默认所有字段
@param string $order [optional] 排序 默认'rank asc'
@return array 分类树

public function filterData($fields = '', $order = '', $trash = false)
数据库查询操作
@param string $fields [optional] 查询字段 默认所有字段
@param string $order [optional] 排序 如'id asc'
@param bool $trash [optional] 查询的数据是否已放入回收站
@return mixed 查询结果

public function _tree($data, $parent_id = 0, $level = 0, $clear = true)
由数据库查询结果集获得无限分类树
非递归包含 也就是说子分类和父分类是同维度的数组元素
@param mixed $data 数据库查询结果集
@param int $parent_id [optional] 父分类 id
@param int $level [optional] 父分类层级
@param bool $clear [optional] 是否清空之前定义的无限分类树$tree(无限分类树$tree 定义为 static)
@return array 无限分类树

public function _treeInTrash($data)

```

由数据库查询结果集获得已放到回收站的无限分类树

@param mixed \$data 数据库查询结果集

@return array 无限分类树

```
public function _treeWithChildren($data, $parent_id = 0, $level = 0)
```

由数据库查询结果集获得递归包含的无限分类树

递归包含指的是子分类是父分类的一个数组元素

@param mixed \$data 数据库查询结果集

@param int \$parent_id [optional] 父分类 id

@param int \$level [optional] 父分类层级

@return array 无限分类树

```
public function getParents($id, $self = false, $fields = '')
```

获得所有父分类信息

@param int \$id 子分类的 id

@param bool \$self [optional] 返回结果是否包含子分类本身

@param string \$fields [optional] 要返回的字段 默认全部字段

@return array

```
public function getParentIDs($id, $self = false)
```

获得所有父分类的 id

@param int \$id 子分类的 id

@param bool \$self [optional] 返回结果是否包含子分类本身

@return array

5.2 非无限分类相关

```
public function search($fields = '', $where = array(), $order = '', $listRows = n, $trash  
= false)
```

分页搜索

@param string \$fields [optional] 查询字段 默认全部字段

@param array \$where [optional] 查询条件

@param string \$order [optional] 排序 默认'rank asc'

@param int \$listRows [optional] 每页显示的记录数 0 表示显示全部记录

@param bool \$trash [optional] 是否已放入回收站

@return array 返回数组\$data

\$data['data'] 存储搜索得到的所有记录

\$data['pageCount'] 存储总页数

`$data['page']` 存储分页 html 代码

5.3 扩展属性类

```
public function saveExtend($fkID1, $fkID2, ...)
```

指定外键值保存修改后的扩展属性值

例如 `public function saveExtend($goodsID)` 保存的是 id 为 `$goodsID` 的商品的扩展属性值

@param int \$fkIDn 外键值

```
public function search($fkID1, $fkID2, ...)
```

指定外键值获取所有的扩展属性值

例如 `public function search($goodsID)` 获取 id 为 `$goodsID` 的商品的扩展属性值

@param int \$fkIDn 外键值

@return array key 为属性名称

如果是唯一属性返回的是二维数组 如果是单选属性返回的是三位数组

属性的 id, 名称, 属性值 id, 属性值均会出现在返回数组中举例

```
array(  
    '颜色' => array(  
        array('id' => 1, 'name' => '颜色', 'value_type' => '单选', 'aid' => 1, 'value'  
=> '玫瑰金'),  
        array('id' => 1, 'name' => '颜色', 'value_type' => '单选', 'aid' => 2, 'value'  
=> '土豪金')  
    ),  
    '分辨率' => array('id' => 2, 'name' => '分辨率', 'aid' => 3, 'value' => '1080*1920')  
);
```

上述数组中 id 表示属性 id, name 表示属性名称, aid 表示属性值 id, value 表示属性值, value_type 表示属性是否单选(缺省表示唯一属性)

5.4 其他

```
public function getForeignKeys()
```

获得外键关联表所有记录的 id 和 name(无限分类表可以获得所有字段)

@return array