ThinkOX开发手册

想天软件工作室

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订人 | 修订时间 | 备注 |
| 0.01 | 蔡培超、陈一枭 | 2014.4.4 | 初稿 |
| 0.02 | 陈一枭 | 2014.4.23 | 补充AdminBuilder搜索部分，JS常用函数部分，重新排版 |
| 0.03 | 陈一枭 | 2014.4.26 | 补充点赞插件 |
| 0.04 | 陈一枭 | 2014.5.4 | 补充模态窗口 |
| 0.05 | 陈一枭 | 2014.6.18 | 补充后台权限 |

# 目录

[目录 3](#_Toc17684)

[常用函数 4](#_Toc29892)

[PHP函数 4](#_Toc12382)

[JS函数 5](#_Toc6463)

[后台管理页面 5](#_Toc22656)

[后台权限约定 5](#_Toc1600)

[配置页面 5](#_Toc28394)

[列表页面 7](#_Toc481)

[排序页面 8](#_Toc14859)

[积分提示 9](#_Toc30349)

[点赞插件的使用 9](#_Toc10617)

[模态窗口的使用 9](#_Toc20749)

# 常用函数

PHP函数

is\_login()

返回当前登录的用户编号。如果没有登录，则返回0。

op\_t($text)

过滤html代码，防止XSS。

参数：$text 字符串。

返回值：经过过滤的字符串。

getPagination($totalCount, $countPerPage=10)

获取翻页html代码。

参数：

$totalCount：数据总数。

$countPerPage：每页显示的数据数量。

返回值：翻页的html代码。

用法举例：

class TestController extends Controller {

public function test($page=1){

$list = D('MyData')->page($page,10)->select();

$totalCount = D('MyData')->count();

$this->assign('totalCount', $totalCount);

$this->display();

}

}

test.html

<volist name="list" id="e">

...

</volist>

{:getPagination($totalCount,10)}

query\_user($fields,$uid=0)

获取用户的详细信息。

参数：

$fields，字符串数组，表示需要获取的字段。目前支持：uid username mobile email signature avatar32 avatar64 avatar128 avatar 256 avatar512 space\_url icons\_html title。

$uid，可选，表示用户编号。默认值是当前登录的用户编号。

举例：

$user = query\_user(array('username','avatar32'));

返回array('username'=>'test', 'avatar32'=>'/Uploads/Avatar/2014/03/31/5ab5d615a6\_32\_auto.jpg');

getThumbImage ($filename, $width, $height, $cut, $replace)

获取本地图片的缩略图。

参数：

$filename 图片的本地路径。

$width 需要的图片宽度

$height 需要的图片高度，可不填写，表示等比缩放。

$cut true表示裁剪图片，false表示拉伸图片。

$replace true表示重新缩放图片，false表示使用已经缩放的图片。

friendlyDate($sTime)

获取友好时间。

参数：$sTime时间戳。

JS函数

function op\_success(text, title)//成功提示

function op\_info(text, title) //信息提示

function op\_warning(text, title) //警告提示

function friendlyDate(sTime, cTime)//友好时间

function U(url, params, rewrite)//JS版U函数

# 后台管理页面

**后台权限约定**

**凡是涉及到表单的action url，尽可能提交到本页面，通过IS\_POST判断是否为表单提交。如不这么做，必须在菜单配置里加入该url的菜单，并设为隐藏，否则权限无法赋予。**

为了便于对后台进行二次开发，我们创造了AdminBuilder系列的辅助函数来快速创建后台管理页面。其中包括了配置页面（AdminConfigBuilder），列表页面（AdminListBuilder），以及排序页面（AdminSortBuilder），以后还会继续增加新的辅助页面，并完善各页面的功能。参考页面WeiboController，ForumController。

配置页面

AdminConfigBuilder

生成配置页。

方法：

title() 配置页的标题

填入字段：

keyHidden() 隐藏字段

keyReadOnly() 只读字段

keyText() 单行文本

keyTextArea() 多行文本

keyInteger() 整数字段

keyUid() UID字段

keyStatus() status字段

keySelect() 下拉框字段

keyRadio() 单选字段

keyCheckBox() 多选字段。逗号分隔。

keyEditor() 编辑器字段

keyTime() 时间字段

keyCreateTime() 创建时间字段

keyBool() 是否字段

keyUpdateTime() 更新时间字段

keyTitle() 标题字段

keyId() ID字段

keyMultiUserGroup() 多个用户组字段

keySingleUserGroup() 单个用户组字段

填入数据：

data($list)

添加按钮

buttonSubmit() 提交按钮

buttonBack() 返回按钮

用法示例：

class TestController extends AdminController {

public function config($id){

$data = D('MyData')->where(array('id'=>$id))->find();

$builder = new AdminConfigBuilder();

$builder->title('测试配置');

$builder->keyId()->keyTitle()->keyCreateTime()->keyText('test\_field','测试字段');

$builder->data($data);

$builder->buttonSubmit(U('myPostUrl'))->buttonBack();

$builder->display();

}

}

列表页面

AdminListBuilder

生成列表页。

基本配置

title() 页面标题

setStatusUrl() 设置状态的地址

buttonNew() “新增”按钮

buttonDisable() “禁用”按钮

buttonEnable() “启用”按钮

buttonDelete() “删除”按钮

buttonRestore() “还原”按钮

buttonSort() “排序”按钮

增加字段

keyText() 文本字段

keyHtml() Html字段

keyMap() 映射字段。将字段的值映射到另一个值。

keyId() ID字段

keyLink() 带链接的字段

keyStatus() status字段

keyYesNo() 是否字段

keyBool() 是否字段，同keyYesNo

keyTime() 时间字段

keyCreateTime() 创建时间字段

keyUpdateTime() 更新时间字段

keyUid() 用户编号字段

keyTitle() 标题字段

keyDoAction() 添加操作字段

keyDoActionEdit() 操作字段中添加编辑链接

keyDoActionRestore() 操作字段中增加还原链接

keyTruncText($name,$title,$length) 截断的字段。最多显示$length长度的字段。

加入翻页组件

pagination($totalCount,$listRow)

加入查询组件 //陈一枭

public function search($title = '搜索',$name='key', $type = 'text', $des = '', $attr) 加入一个文本类型搜索框

设置搜索表单提交的URL //陈一枭

setSearchPostUrl($url)，，默认情况下提交到当前页面，所以无需设置

传入数据

data()

用法演示：

class TestController extends Controller {

public function listTest($page=1,$r=20){

$list = D('MyTest')->page($page,$r)->select();

$totalCount = D('MyTest')->count();

$builder = new AdminListBuilder();

$builder->title('测试列表');

$builder->buttonNew(U('editTest'));

$builder->setStatusUrl(U('setTestStatus'))->buttonEnable()->buttonDisable()->buttonDelete()->buttonSort(U('Forum/sortForum'));

$builder->keyId()->keyCreateTime()->keyStatus()->keyDoActionEdit()

$builder->data($list);

$builder->pagination($totalCount,$r);

$builder->display();

}

}

排序页面

AdminSortBuilder

用于生成排序页面

用法：

class TestController extends Controller {

public function sortTest(){

$list = D('MyTest')->order('sort asc')->select();

$builder = new AdminSortBuilder();

$builder->title('测试排序');

$builder->data($list);

$builder->buttonSubmit(U('doSortTest'));

$builder->buttonBack();

$builder->display();

}

public function doSort($ids){

$builder = new AdminSortBuilder();

$builder->doSort('MyTest', $ids);

}

}

**积分提示**

1. 后台添加一个用户行为
2. 在代码中调用该行为

示例：

$score\_before = getMyScore();

$result = $model->addWeibo(is\_login(), $content);

$score\_after =getMyScore();

$this->success('发表微博成功。' . getScoreTip($score\_before,$score\_after));

扩展了一个getMyScore()函数来获取积分。

扩展了一个getScoreTip($before,$after)来获得积分差的中文表示。

**点赞插件的使用**

{:Hook('support',array('table'=>'weibo','row'=>$weibo['id'],'app'=>'Weibo','uid'=>$weibo['uid']，’jump’=>’no’))}

需要注意的是，app一般填写调用的应用名

table一般填写有需要赞的对象所在的表

row一般填写所赞对象的id

jump一般填写no,则表示点击消息跳转的链接为来源链接，如果不填写此变量，则默认会跳转到一个约定的URL，URL生成规则如下

U($appname . '/Index/' . $table . 'Detail', array('id' => $row))

如app=Weibo table=weibo row=1

则生成 U(‘Weibo/Index/weiboDetail’,array(‘id’=>1));

**模态窗口的使用**

ThinkOX采用magnific弹窗插件。可以应用于各类场合，支持功能比较丰富。

官方文档：<http://dimsemenov.com/plugins/magnific-popup/documentation.html>

比较常用的示例

这边就只展示一个常用的内部形式的弹窗，ajax、相册等形式需自行查阅官方文档。功能强大，能满足一般使用。

①引入对应的函数库和资源（默认为了节省开销，并不会引入该资源）

<link type="text/css" rel="stylesheet" href="\_\_STATIC\_\_/oneplus/js/ext/magnific/magnific-popup.css"/>

<script type="text/javascript" src="\_\_STATIC\_\_/oneplus/js/ext/magnific/jquery.magnific-popup.min.js"></script>

②加入页面样式

.white-popup {

position: relative;

background: #FFF;

padding: 20px;

width: auto;

max-width: 500px;

margin: 20px auto;

}

此弹窗与其他弹窗不同，需要定义这么一个类来实现弹窗的样式。

③加入弹窗的内容

<div id="test-popup" class="white-popup mfp-hide">

这里是内容区

</div>

④设置弹窗触发标签

<button class="btn btn-primary open-popup-link" href="#test-popup" ><i class="glyphicon glyphicon-send"></i>&nbsp;投稿</button>

⑤绑定触发事件

$('.open-popup-link').magnificPopup({

type:'inline',

midClick: true // Allow opening popup on middle mouse click. Always set it to true if you don't provide alternative source in href.

});

⑥enjoy it