

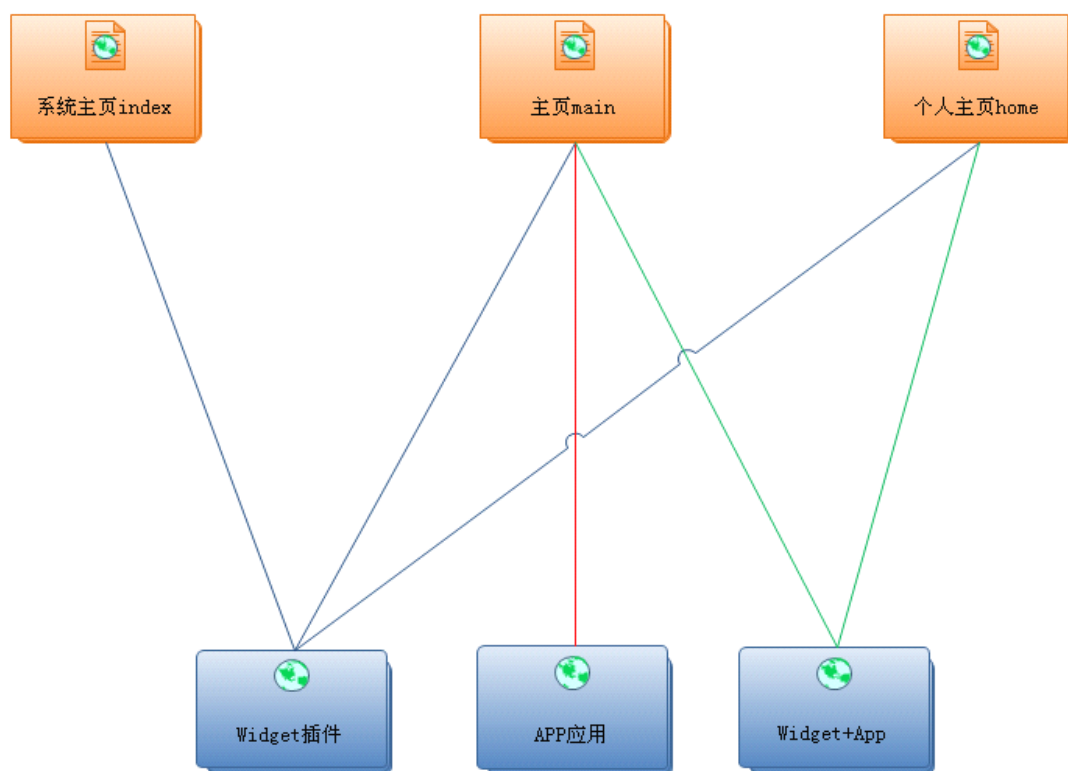
iWebMALL 插件规范(beta 版)

插件简介

插件就是为了满足个性化需求按照 **iWebMALL** 插件开发规范编写的可插拔程序，虽然可以直接对 **iWebMALL** 进行二次开发实现插件同样的功能，但是这样做势必影响到系统的升级和稳定性。采用插件的方式，可以随时进行停用和卸载，而且对系统毫无影响。

插件的类型

插件显示位置和分布如下。



插件布局分布图

插件规范及注意： 我们这里以插件的标识符 `myplugin` （建议唯一）为例说明。

- 1、`myplugin` 应该放在 `plugins` 目录下；
- 2、`myplugin` 必需包含 `plugin.xml` 文件，此文件用来配制插件的基本信息，可以通过工具自动生成；
- 3、如果插件有复杂的交互功能，还需在 `myplugin` 目录下创建 `actions.xml` 文件，并且编写相应的脚本文件。

插件开发步骤：

- 1、明确自己需要的功能，先认真看清楚 **iWebMALL** 是否已经包含；
- 2、插件命名（中英文皆可）。插件的标识符（必需为英文）建议唯一，这点非常重要，建议在开发插件的时候一定要遵守下面的规范；
- 3、明确插件要显示的位置（参见上面的图示），以确定插件的类型。

插件的分类有：

- 1)、以嵌入的方式来显示，交互主要是通过 Ajax 来实现。此类插件为系统的 Widget 插件；
 - 2)、通过页面的主显区域显示，实现较为复杂的交互功能。此类插件为系统的 APP 应用；
 - 3)、在两者之间，部分嵌入页面，部分显示在主显区域，即系统的 APP+Widget 插件。
- 4、如果开发需要有 API 的支持，请参阅 API 文档。
 - 5、编写开发完成后，生成自己的 Plugin.xml 插件的安装信息文件。此文件的创建由 createplugin.php 文件生成，该文件可从官方下载，或者运行系统 plugins 目录下的 createplugin.php 文件生成，如果插件涉及到交互脚本，还需要创建自己的 Actions.xml 文件，下面会有详细的介绍 Actions.xml 的编写规范。
 - 6、开发的插件如果涉及到数据表的操作，则应该先创建好自己的数据表，然后用 phpmyadmin、SQLyog 导出，如果自己编写一定要遵守 SQL 文件的编写规范,并把这些文件配制到 plugin.xml 中。
 - 7、调试通过后，插件即可发布。

关于 actions.xml 文件编写：

格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE actions SYSTEM "../Actions.dtd"> //此行一定要引入，否则程序不能正常运行
<actions>
    <package name="test">
        <action name="show" file="action/PersonAction.php" class="Person" method="show">
            <return name="input">error.php</return>
            <return name="success">view/helloword_list.php</return>
        </action>
        <action name="add" file="action/PersonAction.php" method="add">
            <return name="input">error.php</return>
            <return name="success">view/helloword_list.php</return>
        </action>
    </package>
</actions>
```

```

        </action>

        <action name="del" file="action/PersonAction.php" method="del">

            <return name="input">error.php</return>

            <return name="success">view/helloword_list.php</return>

        </action>

    </package>

</actions>

```

关于此文件的详细说明如下：

根结点(actions): 指 XML 配制文件的根结点。

包结点(package): 指包含一类功能的包名，方便开发者把同一类功能的 action 放在一起，利于管理，此节点只包含一个 name 必选属性，指包的名字。

动作结点(action): 指每一个功能处理节点，此节点包含 name、file、class、method 等属性，name 是必选属性，功能名是方便开发者调用的；file 是必选属性，指用来处理此功能的文件；class 是可选属性，是开发者的类名，如果开发者没用到类，那么此属性不用填写。Method 是必选属性，处理此动作的功能函数。

返回节点(return): 指动作处理完成后要跳转地址，name 是必选属性是指 method 方法中处理完后的返回的字符串；内容是指此字符串对应的跳转地址，地址的路径是指开发者插件的根目录开发的。

如果开发的插件是包含对于数据库的操作，那么分为两类。一类是有自己的数据库服务器，那么开发者不必提供对应的 SQL 安装文件，也不用编写 actions.xml 文件；另一类是没有自己的数据库服务器，要用此系统的数据库，就要编写自己的 SQL 安装文件（[建议此文件是用 PHPMyAdmin 和 SQLyog 工具或者由 MySQL 直接导出的文件](#)），系统识别的注释标记有 “/**/” “--” “#” ，每句都要以 “;” 为结束符号，并且要在生成 plugin.xml 文件时填写上此 SQL 安装文件，为了防止冲突，开发者创建的表必需添加表前缀，并且在插件的根目录下建立一个 config.php 文件，文件中包含 \$table_prefix='表前缀名';开发者的程序中要引用此变量，是保证能系统发生冲突时自动解决的前提。

关于用户文件有编写：

在系统目录下，有一个 plugins 文件夹，这是存放用户所有插件的地方，此目录下 includes.php 文件，是用户开发插件的基础，也是用户调用 Api 的必需文件。

在引入此文件之前要引入以下变量，

\$getpost_power=true;开启统一读入\$_GET 和\$_POST 的方法，如果开发者不使用此功能不用引入此行代码。

\$iweb_power=true;开启对系统平台数据库的支持，如果只是调用 API 则不用开启此功能。

\$session_power=true;如果开发者要使用，session 那么需要开启此功能。

如果只引用些文件，前面不引入变量，那么系统自动会开启 API 的支持。

如果开发者开发的功能用不到以上功能，则此文件不必引入。

一些补充说明：

- 1、widget 插件的编写。此类型的插件一般为一些小型的程序，如一些广告、FLSAH、日历、天气预报、IP 查询等等的小型插件，编写开发都比较自由，显示的方式是网页嵌入式的方式，如果有内容的更新和切换，是由系统提供的 Ajax 类来实现的，用户直接调用即可，此插件可以安装到系统的首页，个人首页及个人主页的页面上。
- 2、Application(App)应用的开发，一般都涉及到交互式的操作，如：开心农场、关注好友日志、关注度排行榜等等。此应用插件是可以由用户自由定制的，此插件有显示位置如下图：



APP 应用开发的要求，如果此插件要在系统内建立表，那么开发者要提供自己的 MySQL 安装文件，并且配制到自己的 plugin.xml 文件中，建议用户在自己的插件根目录下建立 config.php 文件，并写入变量 \$table_prefix='表前缀名';此举主要是防止开发的插件出现同名冲突，以实现产生冲突时系统能自动分配唯一标识符，保证插件的正常运行。其它的文件完全由用户自由的开发。在 App 应用显示的位置（如图所示），如果有内容提交及跳转请求应使用系统提供的 do_action(__file__) 函数，使用此函数必需引用 includes.php 文件。

- 3、Widget+APP 式的插件，只能显示在个人主页及个人首页上，以嵌入式和交互式的方式显示，如：音乐盒等

程序。需要配制的文件和注意的事项同上。

开发样例：

下面我们以一个实例来讲述如何开发一个插件。

插件名为：日志排行榜 英文唯一标识：**blogranking**

建立一个名为 **blogranking**（即唯一标识名）文件夹。

开始编写文件，文件名为 **index.php**，代码如下：

```
<?php

include(dirname(__file__)."/../includes.php");//引入调用 API 的必要条件
?>

<link href="<?php echo self_url(__file__)?>test.css" rel="stylesheet" type="text/css" />

//调用系统方法得到当前路径

<div id="blogranking">//用户开发插件自定义 ID，也是用户定义的工作区

<div id="bar" style="background:url(<?php echo self_url(__file__)?>image/top.gif); height:29px; align='center'"><a
href="<?php echo self_url(__file__)?>index.php?type=0" name="ajax" target=""blogranking" >日</a>

//

<a href="<?php echo self_url(__file__)?>index.php?type=1" name="ajax" target=""blogranking" >周</a>      <a
href="<?php echo self_url(__file__)?>index.php?type=2" name="ajax" target=""blogranking" >月</a></div>

<table width="100%" height="20" border="0" align="center" cellspacing="0" >

<tr style=" background-color:#c7ebf9 ;height:20px">

    <td bgcolor=""> 日志标题</td>

    <td bgcolor=""> 发表者</td>

    <td bgcolor=""> 点击数</td>

</tr>

<?php

function change($type) /根据自己的需要编写自己的函数
{

    date_default_timezone_set("Asia/ShangHai");

    $date=date('Y-m-d');
```

```

$week=604800;
$month=2592000;
switch(intval($type))
{
    case 1: return date('Y-m-d',strtotime($date)-$week); break;
    case 2: return date('Y-m-d',strtotime($date)-$month); break;
    default:
    {
        return $date; break;
    }
}
}
$date=change(0);
if(isset($_POST['type'])) $date=change($_POST['type']);
$blogs=Api_Proxy("blog_self_by_date","log_title,user_name,hits,user_id,log_id",$date."~",'hits'); //调用系统的 API
函数，下面会讲述此函数的调用。
$num=0;
foreach($blogs as $blog)
{
    $num=(++$num)%2;
    if(get_sess_username())$blog_url="href='modules.php?app=blog_show&id=$blog[log_id]' target='frame_content'";
    else $blog_url="href='home.php?h=$blog[user_id]&app=blog_show&id=$blog[log_id]'";
    echo<<<EOD
    <tr class="tr {$num}">
    <td><a $blog_url >$blog[log_title]</a></td>
    <td>$blog[user_name]</td>
    <td class="num">$blog[hits]</td>
    </tr>
    EOD;
}

```

?>

</table>

</div>

test.css

@charset "utf-8";

/* CSS Document */

#blogranking table

{

border-bottom:#999999 1px dotted;

}

#blogranking td

{

height:24px;

padding-left:4px;

border-top:#999999 1px dotted;

}

#blogranking #bar a

{

float:right;

color:#FF0000;

padding-top:4px;

padding-right:2px;

}

.tr1

{

background:#fafafa;

}

.tr0

{

background:#f2f2f2;

}

关于插件中的 a 连接和 form 表单的作用说明：

1、a 连接：

如：<a href="<?php echo self_url(__file__);?>a.php" name="ajax" target="targetid">AJAX 连接

如果一个 A 连接的 name 属性声明为 "ajax" 并且 target 属性也已经声明，那么系统会自动对此标签进行 AJAX 处理，target 属性有两种方式，一种是在 "targetid"，此 targetid 是开发者自定义的 ID，也就是开发者定义的工作区 ID，连接请求的内容会展示的工作区内，另一种是 "targetid"，此 targetid 是开发者定义的函数，系统会自动回调此函数，此函数的格式为：function functionname (content){……code}，content 变量就是请求连接的回显内容，在函数体内开发者可以对连接返回的信息进行再加工处理，注意这里的引号是半角引号。

2、form 表单：

如<form name="ajax" action="<?php echo self_url(__file__);?>php.php" target="hello_word">表单内容</form>，处理的方式和上面的 A 连接方式是一样的，这里要注意，目前不支持对文件有上传。

然后打开插件信息生成器，填写如下信息，

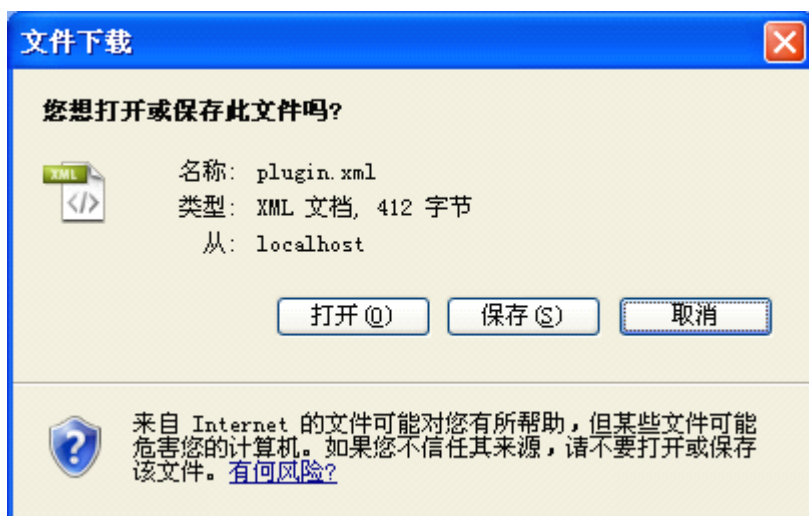
插件信息生成器

插件名称：	<input type="text" value="日志排行榜"/>	*
版本号：	<input type="text" value="1.0"/>	*
作者：	<input type="text" value="jooyea.net"/>	*
缩略图：	<input type="text"/>	
插件描述：	<div><div>一个小的测试插件</div></div>	
后台权限：	<div></div>	
前台权限：	<div></div>	
开发者主页：	<input type="text" value="www.jooyea.com"/>	
数据库安装SQL文件：	<input type="text"/>	
入口文件：	<input type="button" value="增加"/>	
主页朋友圈下侧	<input type="text" value="index.php"/>	*
<input type="button" value="提交"/>		

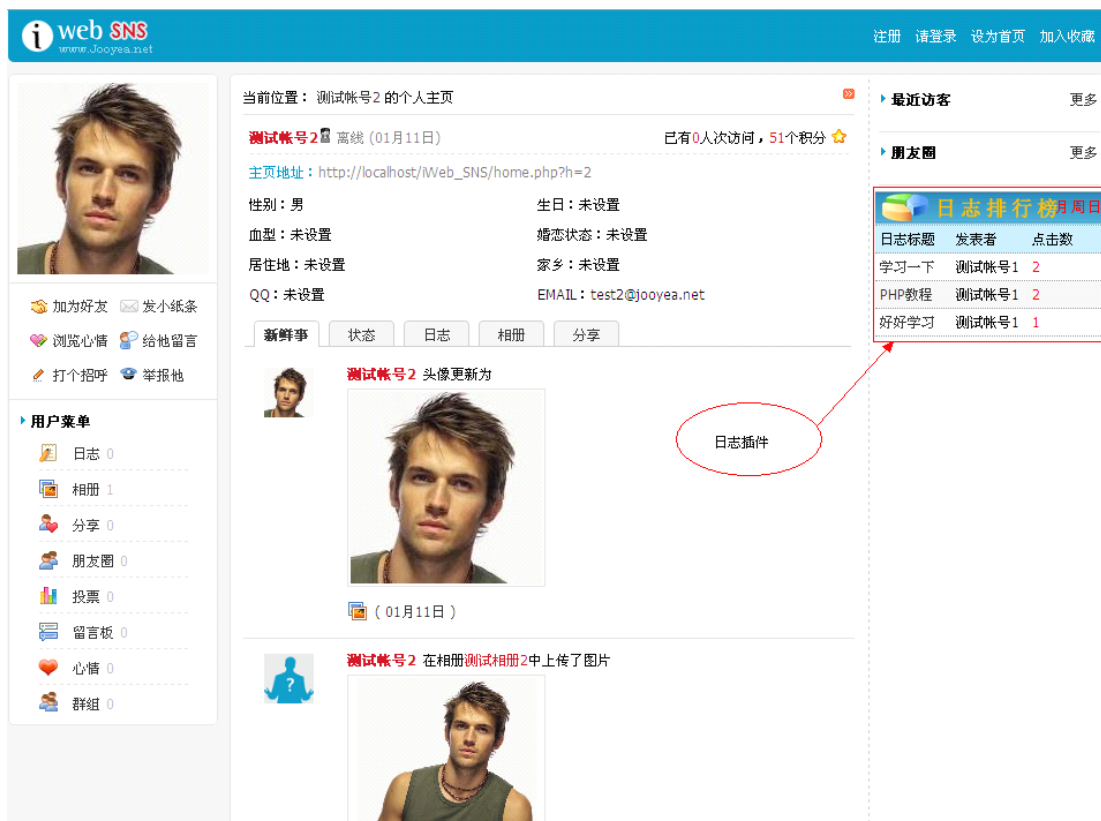
说明：关于权限的格式：

如：add:添加记录,del:删除记录，前面为权限码，后面为权限名，各权限之间用逗号分开，用户如果想实现权限的验证属性实用 plugin_check_right(__file__,"add")，通过返回 true，未通过返回 false，这里的冒号和逗号者是英文半角,如果无权限功能，则不必添加。

点击提交：



把文件保存到自己的插件目录下，一个插件就完成了，后台安装后前台显示如图：



重要文件及函数说明：

1、Includes.php 文件，这是开发插件的基础文件，此文件有几个重要开关变量：

\$session_power 控制是否开启 session，值如果为 true 为开启，不开启则不用写此变量；

\$iweb_power 控制着是否开启对系统数据库的支持，并且兼容系统的平滑 SI 的支持，开启为 true，不开启则不用此变量；

\$getpost_powe控制是否开启系统对\$_GET 和\$_POST 进行封装的支持，调用函数为 get_args(name),系统会自动取 GET 和 POST 上来的变量，如果有同名的变量，函数会返回 POST 的上来的变量，如果查询不到返回 NULL，开启为 true，不开启则不用写此变量。

2、关于 Api_Proxy(参数 1, 参数 2,……)的使用:

用户调用的所有的 API 都是通过此方法，参数 1 是 API 方法的名称，参数 2、参数 3……是对应的 API 的参数。

3、关于路径问题:

用户自己也可以编写自己的 CSS 文件的 JS 文件，也可以引用自己的图片，在引用这些文件的时候，一定调用 self_url(__file__)函数，此函数返回到开发的插件的根目录，例如开发者要引用自己目录下的 CSS 目录的 test.cee 文件，应该写成<link href="<?php echo self_url(__file__);?>css/test.css" rel="stylesheet" type="text/css" />说明一下，在 css/test.css 前不要添加 “/” 符号。引用 JS 文件格式同上。

4、关于权限验证的问题:

用户自定义的

5、关于提交信息问题:

如果开发者开发的插件(这里是指 APP 应用和 APP+Widget)有信息的提交和页面的跳转，要先在自己的 actions.xml 文件里配制上相应的信息，然后在开发的文件里应用系统提供的 do_action(__file__)函数，格式:

```
<form action="<?php echo do_action(__file__);?>test_add" method="post">
```

```
1 <?xml version="1.0" encoding="UTF-8"?>↓
2 <!DOCTYPE actions SYSTEM "../Actions.dtd">↓
3 <actions>↓
4   <package name="test">↓
5     <action name="show" file="action/PersonAction.php" class="Person" method="show">↓
6       <return name="input">error.php</return>↓
7       <return name="success">view/helloword_list.php</return>↓
8     </action>↓
9     <action name="add" file="action/PersonAction.php" method="add">↓
10      <return name="input">error.php</return>↓
11      <return name="success">view/helloword_list.php</return>↓
12    </action>↓
13    <action name="del" file="action/PersonAction.php" method="del">↓
14      <return name="input">error.php</return>↓
15      <return name="success">view/helloword_list.php</return>↓
16    </action>↓
17  </package>↓
18 </actions>↑
```

首先是调用<?php echo do_action(__file__);?>，然后是包名_Action 动作名，如上图开发者 Actions.xml 的配制信息的标注。

对应的 action 节点 file 属性文件 PersonAction.php 的内容

```
1 <?php
2 //$_getpost_power=true;
3 $iweb_power=true;
4 $session_power=true;
5 include(dirname(__file__)."/../../includes.php");
6 function lis()
7 {
8     return "success";
9 }
10 function add()
11 {
12     if(!check())return "input";
13     $db=new dbex();
14     dbplugin('w');
15     $db->exeUpdate("insert into helloworld(msg) values ('".$_get_args('msg')."')");
16     return "success";
17 }
18 function del()
19 {
20     if(!check())return "input";
21     $db=new dbex();
22     dbplugin('w');
23     $db->exeUpdate("delete from helloworld where id=".$_get_args('id'));
24     return "success";
25 }
26 function check()
```

对应的 return 节点 helloworld_list.php 文件内容

```
1 <?php
2 $iweb_power=true;
3 $session_power=true;
4 include(dirname(__file__)."/../../includes.php");
5 $t_hello = "helloworld";
6 $db=new dbex();
7 dbplugin('r');
8 $sql="select * from $t_hello";
9 $result=$db->getRs($sql);
10 ?>
11 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
12 <html xmlns="http://www.w3.org/1999/xhtml">
13 <head>
14 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
15 <title></title>
16 <link rel="stylesheet" type="text/css" href="skin/default/css/row_menus.css">
17 <link rel="stylesheet" type="text/css" href="skin/default/css/right.css">
18 </head>
19 <body>
20 <div id="test_helloworld">
21 <h1 style="text-align:center"><?php if(!is_null(get_sess_username()))echo get_sess_username()."<h1>
22 <div class='main_right_title'>
23 <table width="100%">
24 <tr><td>编号</td><td>信息</td><td>操作</td></tr>
25 <tr><td><?php foreach( $result as $rs){?>
26 <tr><td><?php echo $rs['id'];?></td><td><?php echo $rs['msg'];?></td><td><a href="<?php echo do_action(__file__);?>test_del&id=<?php
echo $rs['id'];?>" name="ajax" target="test_helloworld">删除</a></td></tr>
```

实现了插件的 MVC 架构。

6、如果需要使自己的页面显示在个人主页 home 和主页 main 页面的主显区域：

1)、通过连接，设定 target 属性为 frame_content。

2)、通过 JS，用 document.getElementById("frame_content").src=url;的方式改变。

以上两的 Url 要用上面的动作处理格式(<?php echo do_action(__file__);?>包名_Action 动作名)来实现，不要自己

写地址，否则会找不到你填写的地址的。

相关 API:

说明：api 函数中绿色字表示返回的数据类型。

地址 api

\$fields 字段名称

area_id	地区 id
parent_id	地区父 id
area_name	地区名称
area_type	地区类型 0:country,1:province,2:city,3:district

（注：标有“*”红色星号的参数表示必填参数）

areas_list_by_parentid(array or str \$fields,array or str \$parent_id,str \$by_col,str \$order)

说明：根据地区父 id 值取得其地区下的地区值。

返回值：Array

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为*。

\$parent_id 地区父 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

新闻 api

\$fields 字段名称

article_id	文章 id
cat_id	分类 id
title	标题
content	内容
thumb	图片
admin_id	管理员 id
add_time	添加时间
is_link	0 不跳转，1 跳转
link_url	跳转 url
is_show	1 显示，0 隐藏

（注：标有“*”红色星号的参数表示必填参数）

article_list_by_catid(array or str \$fields,array or str \$cat_id,str \$by_col,str \$order)

说明：根据分类 id 值取得其分类下的新闻。

返回值：Array

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为*。

\$cat_id 分类 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

article_list_all(array or str \$fields,array or str \$article_id,str \$by_col,str \$order)

说明： 根据文章 id 值取得其新闻内容。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$article_id 文章 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

新闻分类 api

\$fields 字段名称

cat_id	分类 id
cat_name	分类名称
parent_id	父 id ， -1 为系统分类
sort_order	分类排序

(注：标有“*”红色星号的参数表示必填参数)

articlecat_list_all(array or str \$fields,array or str \$cat_id,str \$by_col,str \$order)

说明： 根据分类 id 值取得分类内容。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$cat_id 分类 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

articlecat_list_by_parentid(array or str \$fields,array or str \$parent_id,str \$by_col,str \$order)

说明： 根据父分类 id 值取得其下的分类内容。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$parent_id 父分类 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

品牌 api

\$fields 字段名称

brand_id	分类 id
brand_name	品牌名称
brand_logo	品牌 logo
brand_desc	品牌介绍
site_url	网址
is_show	1 为显示

(注：标有“*”红色星号的参数表示必填参数)

brand_list_all(array or str \$fields,str \$by_col,str \$order)

说明： 取得品牌列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

商品分类 api

\$fields 字段名称

cat_id	分类 id
cat_name	分类名称
parent_id	父分类 id: 0 为根分类
sort_order	分类排序
goods_num	产品数量，需要定时更新

(注：标有 “*” 红色星号的参数表示必填参数)

goods_category_by_parentid(array or str \$fields,array or str \$parent_id,str \$by_col,str \$order)

说明： 根据父分类 id 值取得其下的分类列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$parent_id 父分类 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

goods_category_by_catid(array or str \$fields,array or str \$cat_id,str \$by_col,str \$order)

说明： 根据分类 id 值取得分类内容。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$parent_id 分类 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

商品图片 api

\$fields 字段名称

img_id	图片 id
goods_id	商品 id
img_url	图片 url
img_desc	图片介绍
thumb_url	图片缩略图
img_original	原始图片
is_set	设为产品显示图片，1 为设置

(注：标有 “*” 红色星号的参数表示必填参数)

goods_gallery_list_by_goodsid(array or str \$fields,array or str \$goods_id,str \$by_col,str \$order)

说明： 根据商品 id 值取得商品图片列表。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$goods_id 商品 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

商品列表 api

\$fields 字段名称

goods_id	商品 id
shop_id	店铺 id
goods_name	商品名称
cat_id	分类 id
ucat_id	用户自定义分类 id
brand_id	品牌 id
type_id	属性类型 id
goods_intro	商品详情
goods_wholesale	批发说明
goods_number	库存量
goods_price	商品价格
transport_price	运费
keyword	关键字
is_delete	0 为已删除
is_best	1 为精品
is_new	1 为新品
is_hot	1 为热销
is_promote	1 为特价
is_on_sale	0 为下架，1 为上架
is_set_image	是否已设置图片
goods_thumb	缩略图
pv	关注度
favpv	被收藏次数
sort_order	排序
add_time	添加时间
last_update_time	最后修改时间
lock_flg	锁定，1 为锁定

（注：标有 “*” 红色星号的参数表示必填参数）

goods_list_by_terms(array or str \$fields,array or str \$condition, str \$by_col,str \$order)

说明： 根据 \$condition 值取得商品列表（新品 热销 精品 特价 最新上架等）。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$condition 您要取得的商品的条件，空值表示搜索全部（比如要搜索新品，则 \$condition = “is_new”）。

goods_list_by_shopid(array or str \$fields,array or str \$id,str \$by_col,str \$order)

说明： 根据店铺 id 值取得商品列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$id 店铺 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

goods_list_by_goodsid(array or str \$fields,array or str \$id,str \$by_col,str \$order)

说明： 根据商品 id 值取得商品信息或商品列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$id 商品 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

信用评价 api

order_credit_by_seller(array or str \$user_id,str \$by_col,str \$order)

说明： 为用户返回来自卖家的评价。

返回值: **Array**

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

order_credit_by_buyer(array or str \$user_id,str \$by_col,str \$order)

说明： 为用户返回来自买家的评价。

返回值: **Array**

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

order_credit_by_my(array or str \$user_id,str \$by_col,str \$order)

说明： 为用户返回我对他人的评价，返回数据需要进一步进行处理。

返回值: **Array**

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

订单 api

\$fields 字段名称

order_id	订单 id
payid	支付订单号
shop_id	店铺 id
pay_id	支付方式 id
pay_name	支付方式名称
user_id	用户 id
goods_id	产品 id
goods_name	产品名称

goods_price	产品价格
transport_price	运费
order_num	购买数量
order_amount	总订单价格
consignee	联系人
country	国家
province	省份
city	城市
district	地区
address	详细地址
zipcode	邮政编码
telephone	电话
mobile	手机
email	邮箱
shipping_name	物流公司名称
shipping_no	物流发货单号
shipping_type	发货运输类型
message	订单附言
order_time	订单时间
pay_time	支付时间
shipping_time	发货时间
receive_time	用户已接收时间
order_status	订单状态: 0 已取消订单, 1 用户已下订单, 2 商家已确认订单, 3 用户验货
pay_status	支付状态: 0 未支付, 1 已支付
transport_status	发货状态: 0 未发货, 1 已发货
trade_no	支付宝返回的, 支付宝交易号
seller_reply	买家是否已回复: 0 未回复, 1 回复
buyer_reply	卖家是否已回复: 0 未回复, 1 回复

(注: 标有 “*” 红色星号的参数表示必填参数)

order_info_by_orderid(array or str \$fields,array or str \$order_id,str \$by_col,str \$order)

说明: 根据订单 id 值取得订单信息。

返回值: **Array**

\$fields 要取得的字段值数据, 以数组或以 “,” 分割的字符串, 默认为 * 。

\$order_id 订单 id 值, 以数组或以 “,” 分割的字符串, 空值表示搜索全部。

order_info_by_shopid(array or str \$fields,array or str \$shop_id,str \$by_col,str \$order)

说明: 根据店铺 id 值取得店铺订单信息。

返回值: **Array**

\$fields 要取得的字段值数据, 以数组或以 “,” 分割的字符串, 默认为 * 。

\$shop_id 店铺 id 值, 以数组或以 “,” 分割的字符串, 空值表示搜索全部。

order_info_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明： 根据用户 id 值取得用户订单信息。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

店铺自定义分类 api

\$fields 字段名称

shop_cat_id	店铺分类 id
shop_id	店铺 id
shop_cat_name	店铺分类名称
shop_cat_unit	店铺分类数量单位
parent_id	父分类
sort_order	分类排序

(注：标有“*”红色星号的参数表示必填参数)

shop_category_by_shopid(array or str \$fields,array or str \$shop_id,str \$by_col,str \$order)

说明： 根据店铺 id 值取得店铺分类列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$shop_id 店铺 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

店铺评价 api

\$fields 字段名称

cid	评价 id
order_id	订单 id
goods_id	产品 id
buyer	买家 uid
buyer_credit	买家获取的评价 -1 差, 0 中, 1 好
buyer_evaluate	买家被评价的内容
buyer_evaltime	买家被评价时间
buyer_explanation	买家解释内容
buyer_exptime	买家解释时间
seller	卖家 uid
seller_credit	卖家获取的评价 -1 差, 0 中, 1 好
seller_evaluate	卖家被评价的内容
seller_evaltime	卖家被评价时间

seller_explanation	卖家解释内容
seller_exptime	卖家解释时间

（注：标有“*”红色星号的参数表示必填参数）

shop_credit_by_seller(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户作为卖家的评价列表。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

shop_credit_by_buyer(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户作为买家的评价列表。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

留言 api

\$fields 字段名称

gid	产品 id
shop_id	店铺 id
shop_name	店铺名称
user_id	用户 id
name	姓名
email	email
contact	其它联系方式
content	留言内容
add_time	添加时间
add_ip	添加时 ip
shop_del_status	0 代表店言已删除留言
user_del_status	0 代表用户已删除留言
reply	卖家留言

（注：标有“*”红色星号的参数表示必填参数）

shop_guestbook_by_shopid(array or str \$fields,array or str \$shop_id,str \$by_col,str \$order)

说明：根据店铺 id 值取得店铺获得的留言列表。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

\$shop_id 店铺 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

shop_guestbook_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明： 根据用户 id 值取得用户给出的留言列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$user_id 用户 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

图片 api

\$fields 字段名称

id	图片 id
uid	上传图片的用户 id
img_url	图片 url
img_size	图片大小
upl_time	图片上传时间

（注：标有 “*” 红色星号的参数表示必填参数）

shop_imgsize_list_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明： 根据用户 id 值取得用户图片列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$user_id 用户 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

图片 api

\$fields 字段名称

id	图片 id
uid	上传图片的用户 id
img_url	图片 url
img_size	图片大小
upl_time	图片上传时间

（注：标有 “*” 红色星号的参数表示必填参数）

shop_imgsize_list_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明： 根据用户 id 值取得用户图片列表。

返回值: **Array**

\$fields 要取得的字段值数据，以数组或以 “,” 分割的字符串，默认为 * 。

\$user_id 用户 id 值，以数组或以 “,” 分割的字符串，空值表示搜索全部。

店铺 api

\$fields 字段名称

shop_id	店铺 id
user_id	用户 id
shop_name	店铺名称
shop_country	店铺所在国家
shop_province	店铺所在省份
shop_city	店铺所在城市
shop_district	店铺所在区
shop_address	店铺所在详细地址
shop_images	店铺介绍中的图片
shop_logo	店铺 logo
shop_template_img	店铺模板大图
shop_template	店铺模板
shop_management	店铺主营
shop_intro	店铺介绍
shop_notice	店铺公告
shop_creat_time	店铺创建时间
goods_num	店铺产品数量
open_flg	店铺关闭，1 为关闭
lock_flg	店铺锁定，1 为锁定
map_x	地图经线坐标
map_y	地图纬线坐标
map_zoom	地图比例
count_imgsize	图片总大小

（注：标有“*”红色星号的参数表示必填参数）

shop_info_by_shopid(array or str \$fields,array or str \$id,str \$by_col,str \$order)

说明：根据商铺 id 值取得商品信息。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

\$user_id 商铺 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

shop_info_list_all(array or str \$fields,str \$by_col,str \$order)

说明：值取得商品列表信息。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

收货地址 api

\$fields 字段名称

address_id	地址 id
user_id	购物人 id
to_user_name	收件人姓名
full_address	详细地址
zipcode	邮政编码
telephone	电话
mobile	手机
user_country	国家
user_province	省
user_city	城市
user_district	地区
email	邮箱

（注：标有“*”红色星号的参数表示必填参数）

user_address_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户收货地址列表。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 *。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

用户信息 api

\$fields 字段名称

id	
user_id	用户 id
user_truename	真实姓名
user_ico	用户头像
user_marry	0 保密，1 未婚，2 已婚
user_gender	0 保密，1 男，2 女
user_mobile	用户手机号
user_telphone	用户电话
user_country	国家
user_province	省份
user_city	城市
user_district	区
user_zipcode	用户邮编
user_address	用户地址
user_birthday	用户生日

user_qq	用户 qq
user_msn	用户 msn
user_skype	用户 skype

（注：标有“*”红色星号的参数表示必填参数）

user_info_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户信息。

返回值：**Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为*。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

用户消息 api

\$fields 字段名称

rinfo_id	消息 id
user_id	用户 ID
remind_info	提醒内容
remind_time	发送提醒时间
isread	已读标志，1 为已读

（注：标有“*”红色星号的参数表示必填参数）

user_remind_by_buyer(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户消息列表。

返回值：**Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为*。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。

用户基本信息 api

\$fields 字段名称

user_id	用户 id
user_email	用户 email
user_name	用户名
user_passwd	用户密码
user_question	密码找回问题
user_answer	密码找回答案
user_lco	用户头像
reg_time	注册时间
last_login_time	最后登陆时间
last_ip	最后登陆 ip

email_check	邮件确认
email_check_code	邮件确认码
forgot_check_code	密码找回码
rank_id	用户级别
locked	用户锁定，1 为锁定

（注：标有“*”红色星号的参数表示必填参数）

users_by_uid(array or str \$fields,array or str \$user_id,str \$by_col,str \$order)

说明：根据用户 id 值取得用户基本信息。

返回值： **Array**

\$fields 要取得的字段值数据，以数组或以“,”分割的字符串，默认为 * 。

\$user_id 用户 id 值，以数组或以“,”分割的字符串，空值表示搜索全部。