Application Platform 应用平台开发手册

Shanmin Labs.

2016年3月21日

# 目录

1. 开发、运行环境 ..........................................................10
   1. 标准开发环境 ....................................................11
   2. 开发服务器配置 ..................................................12
   3. 运行环境 ........................................................13
   4. 开发标准 ........................................................14
   5. 界面模式说明 ....................................................15
      1. 标准模式 ....................................................15
      2. 经典模式 ....................................................15
      3. 经典单界面模式 ..............................................15
      4. 平板模式 ....................................................15
2. 编码规范 ................................................................16
   1. 关于缩进 ........................................................17
   2. PHP .............................................................18
      1. 关于php文件 ................................................18
      2. 禁止使用的方式 ..............................................18
      3. 关于单引号和双引号使用的建议 ................................18
      4. 关于字符串中变量的引用 ......................................18
      5. 关于json\_encode的调用 ......................................19
      6. 关于@符号的使用 .............................................20
   3. Javascript ......................................................21
      1. 关于兼容性 ..................................................21
      2. 基础应用平台及核心模块中不要使用JQuery ......................21
      3. 在.js文件中，存在其他页面调用的情况时的处理 .................21
      4. 函数中使用this的处理 .......................................22
      5. 函数中使用JSON数据传参的处理 ...............................22
      6. 使用标准JSON格式 ...........................................23
      7. 关于JQuery和JQuery UI的使用说明 ...........................23
      8. 标准页面结构 ................................................23
3. 数据库约定 ..............................................................25
   1. 表类型及编码 ....................................................26
   2. 表名称 ..........................................................27
   3. 关于mid字段 ....................................................28
   4. 关于uid/uname字段 ..............................................29
   5. 关于oc字段 .....................................................30
   6. 关于视图 ........................................................31
   7. 关于SQL的折行问题 ..............................................32
   8. 关于表的别名 ....................................................33
   9. 关于列名的使用 ..................................................34
   10. PHP代码中的SQL书写规则 .........................................35
   11. 数据库管理工具 ..................................................36
   12. 系统表的调用 ....................................................37
   13. 与SQL Server的兼容性相关说明 ...................................38
       1. 不要使用password()函数 ..................................38
       2. 不要使用unix\_timestamp() 取时间戳 .......................38
       3. 不要使用 PDOStatement::rowCount() 取数据数量 ............38
       4. 尽量不要正数数据类型 ....................................39
4. PHP公共变量、函数 .......................................................40
   1. $db 默认数据库连接 ..............................................41
      1. 带有参数的查询 ..............................................41
      2. 分页调用 ....................................................42
      3. 参数为数组的调用 ............................................43
      4. like调用 ....................................................43
      5. 一种参数调用方式 ............................................43
      6. $db->row读取单行数据 ........................................44
      7. $db->rows读取多行数据 .......................................45
      8. $db->col读取首列数据 ........................................45
      9. $db->cols读取列数据集 .......................................46
      10. 导航栏 ..................................................47
   2. $\_SESSION[] .....................................................50
   3. $SKIP\_SIGN 跳过用户登录认证 .....................................51
   4. acl判断当前用户是否拥有指定权限 .................................52
   5. acl2判断当前用户是否拥有指定模块的指定权限 ......................55
   6. wlog 写日志 .....................................................56
   7. dbtools::updown 数据上移下移操作 ................................57
   8. dept::get\_email读取部门邮箱 .....................................58
   9. fav::set 设置关注标记 ...........................................59
   10. fav::get 读取关注标记 ...........................................60
   11. fav::del 删除关注标记 ...........................................61
   12. fav::get\_list\_uid 获取指定条件的用户列表 ........................62
   13. file::delete 删除指定文件 .......................................63
   14. file::dlnk 生成临时下载路径 .....................................64
   15. file::alias 生成临时别名下载路径 ................................65
   16. ~~file::download下载指定路径文件~~ ..................................66
   17. file::download2下载指定路径文件 .................................67
   18. file::upfile 标准文件上传处理 ...................................68
   19. file::upfiles标准文件批量上传处理 ...............................69
   20. file::upload\_mkdir 创建目录 .....................................70
   21. file::rcopy 拷贝目录 ............................................71
   22. file::rmdir 删除目录 ............................................72
   23. file::save 保存指定文件内容到标准路径 ...........................73
   24. file::tempdir 创建临时目录 ......................................74
   25. ini::set 设置INI项 .............................................75
   26. ini::get 读取INI项 .............................................76
   27. ini::del 删除INI项 .............................................77
   28. msg::send 当前用户向指定人发送信息 ..............................78
   29. msg::sys 系统账号向指定人发送信息 ...............................79
   30. msg::ignore忽略指定信息 .........................................80
   31. options::dd输出部门列表 .........................................81
   32. options::topdept输出顶层部门 ....................................82
   33. options::ud 输出部门及用户列表 ..................................83
   34. role::check\_parent\_dept\_enabled 检查上层部门是否启用状态.........84
   35. role::dept\_edit 编辑部门信息 ....................................85
   36. role::dept\_new 创建新部门 .......................................86
   37. role::dept\_top 使用部门名称读取顶层部门ID .......................87
   38. role::edit 角色编缉 .............................................88
   39. role::icon\_add 增加角色图标 .....................................89
   40. role::icon\_del 删除指定角色图标 .................................90
   41. role::id 读取指定角色ID .........................................91
   42. role::id2读取指定角色ID ........................................92
   43. role::mid 使用路径、权限读取模块编号 ............................93
   44. role::ispriv 判断指定模块是否存在指定权限 .......................94
   45. role::is\_child 判断指定角色是否在指定父层角色下面 ...............95
   46. role::get\_all\_id 读取所有符合条件的ID ...........................96
   47. role::get\_all\_mid 读取所有符合条件的mid .........................97
   48. role::get\_all\_child\_id 读取所有子角色编号 .......................98
   49. role::get\_all\_child\_info 读取所有子角色信息 .....................99
   50. role::get\_all\_top\_info 读取所有顶层角色信息 .................... 100
   51. role::get\_all\_parent\_id 读取所有上层角色编号 ................... 101
   52. role::getdept读取指定用户或部门的上层部门(停用) ................ 102
   53. role::getdept2 读取指定用户的上层部门列表 ...................... 103
   54. ~~role::getuser\_from\_priv 取回拥有指定权限的所有用户~~ ............. 104
   55. role::getuserid\_from\_priv 取回拥有指定权限的所有用户编号........ 105
   56. role::getusername\_from\_priv 取回拥有指定权限的所有用户名称 ...... 106
   57. role::get\_info 读取指定角色信息 ................................ 107
   58. role::name 读取指定角色名称 .................................... 108
   59. role::parent\_deptname 读取指定角色的上层部门名称 ...............109
   60. role::prefix 返回指定角色的前缀 ................................ 110
   61. role::get\_parent\_id 读取上层角色编号 ........................... 111
   62. role::get\_parent\_info获取上层角色信息 .......................... 112
   63. role::get\_parents\_info获取上层角色信息 ......................... 113
   64. ~~role::get\_ud读取指定用户上级部门编号~~ ........................... 114
   65. role::relation\_add 增加角色关系 ................................ 115
   66. role::relation\_exists指定角色关系是否存在 ...................... 116
   67. role::relation\_remove 移除指定角色关系 ......................... 117
   68. ~~role::sign 用户登录(已停用)~~ .................................... 118
   69. role::sign2 用户登录 ........................................... 119
   70. role::user\_dept\_add指定用户增加到指定部门下 .................... 120
   71. role::user\_dept\_remove从指定部门移除指定用户 ................... 121
   72. ~~role::user\_edit编辑用户信息(已停用)~~ ............................ 122
   73. role::user\_edit2编辑用户信息 ................................... 123
   74. role::user\_enabled用户启用禁用 ................................. 124
   75. role::user\_password\_reset重置指定用户密码 ...................... 125
   76. sms::get\_last读取发送给指定电话号码的最后一条短信内容 .......... 126
   77. sms::send发送短信 .............................................. 128
   78. sms::send\_all群发短信 .......................................... 129
   79. weibo::short\_url生成短链接 ..................................... 130
   80. tools::date6时间戳转换为6位日期显示 ........................... 131
   81. tools::date6time从6位日期类型转换为时间戳 ..................... 132
   82. pdf::merge多文件合并 ........................................... 133
   83. Office相关操作 ................................................. 134
       1. word::doc2mht格式转换 .................................. 134
       2. word::doc2pdf格式转换 .................................. 135
       3. word::doc2pdf2格式转换(自动接受修订) ................... 136
       4. word::doc2split5五书拆分 ............................... 137
       5. word::mht2doc格式转换 ..................................138
       6. word::mht2docx格式转换 ................................. 139
       7. word::mht2pdf格式转换 .................................. 140
       8. word::mht\_replace内容替换 .............................. 141
       9. word::mht\_replace\_file文件内容替换 ..................... 142
       10. word::mht\_replace2变量替换 ............................. 143
       11. word::mht\_replace3变量精准替换 ......................... 144
       12. excel::xml2xls格式转换 ................................. 145
   84. E-Mail相关 ..................................................... 147
       1. email::send发送邮件 .................................... 147
   85. Aliyun相关（阿里云） ........................................... 149
       1. aliyun::wsls写日志 ..................................... 149
       2. aliyun::oss\_upload向OSS上传文件 ....................... 150
       3. aliyun::oss\_download从OSS上下载文件 ................... 150
       4. aliyun::oss\_exists判断OSS上是否存在指定文件 ........... 151
5. Javascript公共函数 ..................................................... 153
   1. $ajax AJAX访问 ................................................. 154
   2. $af\_classList 让节点支持classList操作 ......................... 156
   3. $ap\_home 平台根路径 ............................................ 157
   4. $confirm 确认窗口 .............................................. 158
   5. $contextmenu上下文菜单 ......................................... 159
   6. $ddupfile 拖拽上传[停用] ....................................... 161
   7. $ddupfile2文件拖拽上传 ......................................... 162
   8. $el 创建节点 ................................................... 165
   9. $get读取URL中的参数 ........................................... 167
   10. $getElementsByClassName 模拟getElementsByClassName操作......... 168
   11. $jtree 树形结构显示 ............................................ 169
   12. $jctree树形选择框 .............................................. 172
   13. $msgbox 显示消息窗口（模式一） ................................. 176
   14. $msgbox 显示消息窗口（模式二） .................................177
   15. $topmsg 显示提示信息 ........................................... 178
   16. $table2 ........................................................ 179
   17. $topmost 节点zIndex置最高 ..................................... 181
   18. $window 创建窗口 ............................................... 182
   19. $tselect文本选择框 ............................................. 184
   20. ~~$email邮件编辑（停用）~~ ......................................... 186
   21. $email2邮件编辑器 .............................................. 189
6. 第三方组件 ............................................................. 192
   1. PHPExcel 1.8.0 ................................................. 193
7. CSS公共类 .............................................................. 195
   1. .info 提示信息区 ............................................... 196
   2. .warning 警告信息区 ............................................ 197
8. 权限模型的说明 ......................................................... 198
   1. 模块的形式 ..................................................... 199
   2. 页面中权限的判断 ............................................... 200
9. TABLE作为Excel文件保存时的说明 ........................................ 201
10. 固定文件格式 ....................................................... 202
    1. \_\_sysconfig.php ................................................ 203
    2. \_\_icon.png ..................................................... 204
    3. \_\_module.php ................................................... 205
    4. \_\_startup.js ................................................... 206
    5. \_\_tick.php ..................................................... 207
    6. \_\_update.php 模块升级文件 ...................................... 208
       1. 数据库列名关键字 ....................................... 208
       2. update::column\_add 增加列 .............................. 208
       3. update::column\_exists 判断列是否存在 ................... 209
       4. update::column\_info读取列信息 .......................... 210
       5. update::procedure\_exists 指定存储过程是否存在 .......... 210
       6. update::table\_create 创建表 ............................211
       7. update::table\_delete 删除表 ............................ 213
       8. update::table\_exists 判断表是否存在 .................... 213
       9. update::index\_exists 判断指定索引是否存在 .............. 214
       10. MySQL常用判断 .......................................... 214
11. 模块打包 ........................................................... 215

# 1开发、运行环境

1.1标准开发环境

操作系统：Debian 7.6 64bit 数据库：MySQL 5.5 语言：PHP 5.4

1.2开发服务器配置

Acer AOD270

1.3运行环境

标准运行环境：

Debian 7.x

Apache

MySQL 5.5

PHP 5.4、5.6

其他需要支持的运行环境

Windows 2008、Windows 7、Windows 2012、Debian、AIX、Solaris、FreeBSD

数据库：MySQL 5.5及以上 SQL Server 语言：PHP 5.4、5.6

1.4开发标准

页面标准：HTML5+各浏览器支持 编码标准：UTF-8

PHP版本：5.4、5.5、5.6

1.5界面模式说明

1.5.1标准模式

模拟多任务模式，对于高级用户拥有更自由的控制方式。

1.5.2经典模式

采用经典OA界面模式，使用起来更直观。

1.5.3经典单界面模式

采用经典单界面模式，每个功能共用一个区域。

1.5.4平板模式

模拟单任务模式的操作方式，对触摸屏操作进行优化。

# 2编码规范

2.1关于缩进

大括号的缩进：

if(ok) {

}

严禁使用K&R风格，即写成

if(ok){ }

2.2PHP

## 2.2.1关于php文件

当单文件大于1000行代码时，建议拆分文件。当单文件大于10000行时，必须拆分文

件。因为到达这个数量级后单文件的维护问题需要考虑。

## 2.2.2禁止使用的方式

list($a,$b)=exlpode(‘,’,$arr);

这是因为，Zend Guard编译后，这样编写的代码会发生错误。

## 2.2.3关于单引号和双引号使用的建议

如果字符串中不存在变量或转义符，必须使用单引号，例如：

$s=’abcdefg’;

只有在字符串中存在变量时，才使用双引号，例如：

$s=”abc{$str}defg”;

强烈建议存在变量的字符串使用双引号方式，避免使用链接拼接，不推荐的方式：

$s=’a’.$a1.’b’;

建议的方式

$s=”a{$a1}b”;

使用双引号的方式，要比拼接方式有更好的代码可读性。

## 2.2.4关于字符串中变量的引用

字符串中变量引用，一律加{}限定符，严禁不使用{}限定符：正确的方式

$s=”abcdefg{$str}”;

严禁使用的方式

$s=”abcdefg$str”;

对于基于键的数组访问，一定要加引号，例如:

$arr=array(‘a’=>’AAAA’,’b’=>’BBBB’,’c’=>’CCCC’);

如果要输出b的值时，使用下面的方式：

echo $arr[‘b’];

严禁使用下面的方式：

echo $arr[b];

## 2.2.5关于json\_encode的调用

在JSON数据保存数据库时，使用json\_encode时，建议加上JSON\_UNESCAPED\_UNICODE,这

样可以减少数据库保存长度。例如下列数据内容：

$json=array(

“修改时间”=>1234567,

“修改内容”=>”修改内容说明修改内容说明修改内容说明修改内容说明”

);

使用json\_encode($json)返回:

{"\u4fee\u6539\u65f6\u95f4":1234567,"\u4fee\u6539\u5185\u5bb9":"\u4fee\u6539\u5185\u5bb9\u8b f4\u660e\u4fee\u6539\u5185\u5bb9\u8bf4\u660e\u4fee\u6539\u5185\u5bb9\u8bf4\u660e\u4fee\u6539

\u5185\u5bb9\u8bf4\u660e"}

共返回211字节数据。

使用json\_encode($json,JSON\_UNESCAPED\_UNICODE)返回：

{"修改时间":1234567,"修改内容":"修改内容说明修改内容说明修改内容说明修改内容说明"}

共返回114字节数据（按UTF-8计算）。

## 2.2.6关于@符号的使用

尽量避免使用@符号隐藏错误显示，因为这样当发生错误时无法及时发现。

2.3Javascript

因为Javascript代码在最终发布时，会使用Google Closure Compiler压缩，所以在编写Javascript时需要注意一些Google Closure Compiler的压缩规范。

2.3.1关于兼容性

element.hasAttribute在IE7上不支持，需要使用element.getAttribute替代。

## 2.3.2基础应用平台及核心模块中不要使用JQuery

为了照顾对多种浏览器兼容性问题，禁止在基础应用平台及核心模块内使用 JQuery。因为各个版本的JQuery对浏览器及平板操作的兼容性存在问题。 但，在其他普通模块的编写中不做限制。

## 2.3.3在.js文件中，存在其他页面调用的情况时的处理

如果.js文件中存在当前未引用的自定义函数，则应该使用window[]方式保存，例如：

function aa() //这个函数在当前文件中未被引用，而是在其他页面调用

{

……

}

el.onclick=function()

{

……

}

改为：

window[‘aa’]=function()

{

……

}

el[‘onclick’]=function()

{

|  |  |
| --- | --- |
| } | …… |

否则使用Google Closure Compiler压缩后，会发生未被引用的函数丢弃的情况。

## 2.3.4函数中使用this的处理

如果在函数中存在this处理，则在函数定义时做一个声明，例如：

var el=document.createElement(“div”);

/\*\* @this Element \*/ el.onclick=function()

{

this.className=”new”;

}

因为在onclick定义的函数中使用了this。所以，在函数定义之前声明this为一个节点类型。 如果不这样做一个声明，会在模块打包时出现警告信息。

## 2.3.5函数中使用JSON数据传参的处理

在调用时不要使用args.id这种形式，应使用args[‘id’]的方式引用，例如：

functioin proc(args)

{

if(args.id==”a1”)

{

……

}

……

}

应改为：

functioin proc(args)

{

if(args[‘id’]==”a1”)

{

……

}

……

}

## 2.3.6使用标准JSON格式

请使用下面的方式：

var args=[“id”:”win”,”name”:”test name”];

不要使用这种方式：

var args=[id:”win”,name:”test name”];

## 2.3.7关于JQuery和JQuery UI的使用说明

核心模块，不允许使用JQuery、JQueryUI类库。但在普通功能模块内不做限定。各类库，请直接保存在各自模块目录下。

## 2.3.8标准页面结构

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>页面标题</title>

<script src=”../inc.js”></script>

</head>

<body>

<table class=”toolbar”>

<tr>

<td>栏目左边内容</td>

<td align=”center”>中间内容</td>

<td align=”right”>右边内容</td>

</tr>

</table>

<table class=”table”>

<tr>

<th>编号</th>

<th>名称</th>

<th>操作</th>

</tr>

<tr>

<td>1</td>

<td>信息名称</td>

<td><a href=”show.php?id=1”>查看</a></td>

</tr>

</table>

</body>

</html>

# 3数据库约定

3.1表类型及编码

使用MySQL时： 如无特殊说明,默认使用表类型为:MyISAM。

默认数据库编码为utf-8。

因为本系统需要支持SQL Server，所以有些MySQL专用的SQL写法或命令请不要使用。

后面的章节会详细说明具有区别和注意的地方。

3.2表名称

如无特殊说明，所有数据表名称都以“t\_”开头，后面紧跟模块名称，例如：pub-info 模块的表名为：t\_pub\_info。如果模块存在多表的情况下，数据表名前缀不变，例如： t\_pub\_info\_log。

表名全部使用小写字母[[1]](#footnote-1)。各单词间使用下划线隔开。

注意：所有t\_sys\_\*的表为系统级的表。各应用不得随意修改这组表中的数据!

3.3关于mid字段

因为大部分的功能都将设计成可重复使用的模块，所有这样的模块的表中必须存在mid 字段来保存对应的模块编号。

3.4关于uid/uname字段

在设计表的时候，对于用户的引用，一般使用uid这样的形式。建议，保存uid的同时也保存一个uname，这样方便显示和查询。

注：平台用户多，这样可以减少数据库的查询次数及查询量。

3.5关于oc字段

如果表中存在排序字段，则排序字段统一名称为oc，数据类型int或bigint，里面直

接使用unix\_timestamp()填充。

$db->exec("insert into t\_cap\_test (name,oc) values ('{$username}',unix\_timestamp())");

3.6关于视图

在实际使用中，发现MySQL对视图支持不好，如不是特别需要，不建议使用视图。

注：支持度主要表现在数据的备份和恢复上。备份的视图在恢复的时候会恢复成一个表。

3.7关于SQL的折行问题

如果一条SQL非常长的时候，可以考虑折行书写，示例如下：

select \* from t\_sys\_test where w\_type=13 and school=’new school’ and year<=’2014’ and year>=’2013 and flag=0 order by addtime,w\_type\_info

在代码中如下：

|  |
| --- |
| $dr=$db->query("select \* from t\_sys\_test  where w\_type=13 and school=’new school’ and year<=’2014’ and year>=’2013’ and flag=0  order by addtime,w\_type\_info"); |

带有join的：

|  |
| --- |
| $dr=$db->query("select \* from t\_sys\_test  join t\_edu\_xxxx on t\_edu\_equipment.id=t\_edu\_xxxx.iid  where w\_type=13 and school=’new school’ and year<=’2014’ and year>=’2013’ and flag=0  order by addtime,w\_type\_info"); |

3.8关于表的别名

很多的教程使用了表别名，但是本平台的开发不推荐使用。因为复杂的SQL中表别名不容易让人快速的理解。当然，在需要使用的时候还是得使用。

不建议使用的示例：

|  |
| --- |
| $dr=$db->query("select \* from t\_sys\_test t1  join t\_sys\_xxxx t2 on t1.id=t2.iid where t1.w\_type=13 and t1.school=’new shool’ and t1.year<=’2014’ and t1.year>=’2013’ and t1.flag=0  order by t1.addtime, t1.w\_type\_info"); |

建议的书写示例：

|  |
| --- |
| $dr=$db->query("select \* from t\_sys\_test  join t\_sys\_xxxx on t\_sys\_test.id= t\_sys\_xxxx.iid where t\_sys\_test.w\_type=13 and t\_sys\_test.school=’new school and t\_sys\_test.year<=’2014’ and t\_sys\_test.year>=’2013’ and t\_sys\_test.flag=0  order by t\_sys\_test.addtime, t\_sys\_test.w\_type\_info"); |

注意!这个说明不是说不让用表别名，只是不推荐使用。例如在同一表在一个SQL中同时多次调用时，别名是必须使用的。

3.9关于列名的使用

以下列名不可使用: repeat MySQL关键字

3.10 PHP代码中的SQL书写规则

在代码中,所有在SQL中的变量参数,都不许以后置参数的方式传递，例如下面的代码是不允许使用的：

$dr=$db->query(“select \* from t\_test where uid=’{$\_GET[‘uid’]}’”);

应该使用下面代码替代：

|  |
| --- |
| $dr=$db->query(‘select \* from t\_test where uid={:uid}’,  Array(‘uid’=>$\_GET[‘uid’])  ); |

注1：这个主要的目的是为了防止SQL注入。注2：这个参数传递方式与PDO::prepare有区别！

3.11 数据库管理工具

使用MySQL时推荐使用MySQL Workbench。

3.12 系统表的调用

除特殊说明的情况以外，一律严谨对t\_sys\_开头的系统表进行读写操作！

同时，尽量避免直接对系统表进行读操作。

3.13 与SQL Server的兼容性相关说明

为了保证系统对各种数据库系统的支持，本节说明非常重要，请详细阅读!

## 3.13.1不要使用password()函数

这个命令SQL Server不支持，请使用PHP命令替代。

原:

$db->exec(“insert into t\_test (passwd) values (password(‘123456’))”);

替代方案:

$db->exec(‘insert into t\_test (passwd) values ({:passwd})’,

Array(‘passwd’=>'\*'.strtoupper(sha1(sha1('123456',true))))

);

## 3.13.2不要使用unix\_timestamp() 取时间戳

这个命令是MySQL特有的，在SQL Server中不存在。所以不要直接在SQL中使用这个命令。例如原：

$db->query("delete from t\_sys\_log\_sign where dt<unix\_timestamp()-3600");

替代方案：

|  |
| --- |
| $db->query(‘delete from t\_sys\_log\_sign where dt<{:time}’,  Array(‘time’=>time()-3600)  ); |

注：至于这个-3600是在PHP端计算，还是数据库服务器端计算，由开发者自己决定。推荐在PHP端计算。

## 3.13.3不要使用 PDOStatement::rowCount() 取数据数量

开发过程中发现，SQL Server服务器，在使用PDOStatement::rowCount()读取数据时返回的数据不正确。请使用$db->count(“select count(\*) from t\_sys\_role”)取代,详细参见$db的详细说明。

原

$dr=$db->query(“select \* from t\_sys\_role”); echo $dr->rowCount();

替代方案

echo $db->count(“select count(\*) from t\_sys\_role”);

## 3.13.4尽量不要正数数据类型

在MySQL中可以使用unsigned int来取得更大的数字支持，但是这种类型在SQL Server 中并不支持无符号类型。在开发基础模块时，在数据库设计时，尽量避免使用unsigned int 这类无符号类型数据。 如果用户环境已经定位使用MySQL这类支持uint类型的数据库，则完全可以使用。 如果数值类型太大，必须使用unsigned int这类数据，则建议使用bigint类型替代。

注：如果系统确认不需要跨平台，则请忽略此说明。

# 4PHP公共变量、函数

这部分变量、函数大部分都在inc.php文件中定义的，所以使用时需要包含这个文件。

推荐在模块中使用的包含代码：

include '../inc.php';

4.1$db 默认数据库连接

这是一个使用PDO封装的公共的数据库连接。如果没有特殊需要，所有的数据库连接都必须调用这个接口！ 对于存在参数的SQL，必须使用参数字段传递。这是为了防止SQL注入！

查询示例

$dr=$db->query(“select \* from t\_sys\_role”); while($row=$dr->fetch()) echo $row[‘name’];

执行示例

$db->exec(“insert into t\_sys\_role (name) values (‘new name’)”);

查询分页示例

$dr=$db->query(“select \* from t\_sys\_role”,20); while($row=$dr->fetch()) echo $row[‘name’]; echo $db->navigation(1);//显示导航样式1 echo $db->navigation(2);//显示导航样式2

说明：分页导航会自动添加一个$\_GET[‘\_\_page’]的参数来标识当前的页号显示样式



读取最后一次插入的id

$db->exec(“insert into t\_cap\_role (name) values (‘new name’)”);

$id=$db->lastInsertId();//返回正确插入的ID

注意：如果SQL语句中带有limit关键字，则$db不会进行分页。

取当前影响的数据行数

echo $db->count(“select count(\*) from t\_sys\_role”);

## 4.1.1带有参数的查询

通常的：

<?php

$db->exec(“insert into t\_test (a1) values (‘{$\_POST[‘a1’]}’)”);

?>

但是这样的代码，在实际情况下是非安全的。所以，新的编码规范要求必须使用下面的方式替代：

<?php

$db->exec(‘insert into t\_test (a1) values ({:a1})’, array(‘a1’=>$\_POST[‘a1’])

);

?>

注：{:a1}这种方式是本平台特有的方式，不是PHP标准用法。

多个参数的查询示例：

<?php

$dr=$db->query(‘select \* from t\_test where a1={:a1} and a2={:a2}’, array(‘a1’=>$\_POST[‘a1’],’a2’=>$\_POST[‘a2’])

);

?>

对一些数字的直接量，可以直接写在SQL中，但字符串的直接量不允许写在SQL中。

例如下面的方式是允许的：

<?php

$dr=$db->query(‘select \* from t\_test where a1=123 and a2={:a2}’, array(’a2’=>$\_POST[‘a2’])

);

?>

下面的方式是不允许的！！！：

<?php

$dr=$db->query(‘select \* from t\_test where a1=“abc” and a2={:a2}’, array(’a2’=>$\_POST[‘a2’])

);

?>

## 4.1.2分页调用

按照每页20条数据进行分页：

<?php

$dr=$db->query(‘select \* from t\_test where a1={:a1} and a2={:a2}’, array(‘a1’=>$\_POST[‘a1’],’a2’=>$\_POST[‘a2’]),20

);

42

?>

## 4.1.3参数为数组的调用

当使用in等操作时，会出现多个参数的情况，这时候可以直接对参数置数组，例如：

<?php

$arr=array(’1’,’2’,’3’,’4’,’5’);

$db->query(‘select \* from t\_test where id in ({:id})’, arrray(‘id’=>$arr)

);

?>

## 4.1.4like调用

当SQL中存在like调用时，%、？这类通配符需要放入参数中，例如：

|  |
| --- |
| <?php  $dr=$db->query(‘select \* from t\_test where name like {:name}’, array(‘name’=>’%张%’)  );  while($row=$dr->fetch())  {  ……  }  ?> |

## 4.1.5一种参数调用方式

本节写的只是一种参考方式，仅供参考使用：

<?php

$sql\_where[]=array(); $sql\_param[]=array();

$sql\_where[]=”type={:type}”;

$sql\_param[‘type’]=’分组’;

if(isset($\_GET[‘cc’]))

|  |
| --- |
| {  $sql\_where[]=”cc={:cc}”;  $sql\_param[‘cc’]=$\_GET[‘cc’]; }    $sql\_where=implode(‘ and ’,$sql\_where); if(!empty($sql\_where)) $sql\_where=" where {$sql\_where}";    $dr=$db->query(“select \* from t\_test {$where}”,$sql\_param);  ……  ?> |

## 4.1.6$db->row读取单行数据

定义

function row($statement,$params=array())

参数

$statement 执行的SQL $params 参数列表返回值 符合SQL的第一条数据记录。

说明 返回SQL执行后返回的第一条数据记录。

这个函数相当于

<?php

$dr=$db->query(‘select \* from t\_test where id={:id}’,array(‘id’=>$\_GET[‘id’]));

$row=$dr->fetch(PDO::FETCH\_ASSOC);

?>

示例

<?php

$row=$db->row(‘select \* from t\_test where id={:id}’,array(‘id’=>$\_GET[‘id’])); ?>

## 4.1.7$db->rows读取多行数据

定义

function rows($statement,$params=array())

参数

$statement 执行的SQL $params 参数列表返回值 符合SQL的数据记录集。

说明 返回SQL执行后返回数据记录集。 注意！！！本函数仅用于小的、有限结果集使用。对大量数据的读取请不要用！把结果集直接存储在数组中，会占用大量内存！ 这个函数相当于

<?php

$dr=$db->query(‘select \* from t\_test where id>{:id}’,array(‘id’=>$\_GET[‘id’]));

$rows=$dr->fetchAll(PDO::FETCH\_ASSOC);

?>

示例

<?php

$rows=$db->row(‘select \* from t\_test where id>{:id}’,array(‘id’=>$\_GET[‘id’]));

?>

## 4.1.8$db->col读取首列数据

定义

function col($statement,$params=array())

参数

$statement 执行的SQL $params 参数列表返回值 符合SQL的第一条数据的第一列数据。示例

<?php

$name=$db->col(‘select name from t\_test where id={:id}’,array(‘id’=>$\_GET[‘id’]));

?>

## 4.1.9$db->cols读取列数据集

定义

function col($statement,$params=array())

参数

$statement 执行的SQL $params 参数列表返回值 符合SQL的第一列数据的数组。

说明 此方法，相当于：

<?php

$dr=$db->query($statement,$params); return $dr->fetchAll(PDO::FETCH\_ASSOC);

?>

示例

<?php

$name=$db->col(‘select id from t\_test where id<{:id}’,array(‘id’=>$\_GET[‘id’])); print\_r($name);

?>

输出示例

Array

(

1. => 14007
2. => 9

……

)

## 4.1.10导航栏

定义

public function navigation2()

参数

无

返回值 输出字符串。

说明 此导航栏可以带着当前页name=”\_\_selected”的input元素值。具体使用，参见下面的示例部分。示例

|  |
| --- |
| <?php  include dirname(dirname(dirname($\_SERVER['SCRIPT\_FILENAME']))).'/inc.php';  ?><html>  <head>  <script src="../../inc.js"></script>  </head>  <body>  <pre><?php print\_r($\_GET); ?></pre>    <div style="background-color: #cccccc">  <?php    //获取已选择信息  if(isset($\_GET['\_\_selected']))  $\_\_selected=json\_decode($\_GET['\_\_selected']); else  $\_\_selected=array();  $\_\_selected\_count=count($\_\_selected);    //输出到隐藏域  foreach($\_\_selected as $v)  echo "<input type='hidden' name='\_\_selected' value='{$v}'>";    ?>  其他页已经选择数据 <span id="\_\_selected\_count"><?php echo $\_\_selected\_count; ?></span> 条。  </div> |
| <?php    //显示数据  $dr=$db->query('select \* from t\_sys\_role',20); while($row=$dr->fetch(PDO::FETCH\_ASSOC))  {  if(in\_array($row['id'],$\_\_selected))  {  $\_\_selected\_count--; //计算其他页数据选择  $checked='checked';  }  else  $checked='';  echo "<input type='checkbox' name='\_\_selected' value='{$row['id']}' {$checked} /> {$row['id']} {$row['login']} {$row['name']}<br>";  }  echo $db->navigation2();    //更新数量显示  echo  "<script>document.getElementById('\_\_selected\_count').innerHTML='{$\_\_selected\_count}';</scrip t>";  ?>  </body>  </html> |

显示效果



4.2$\_SESSION[]

|  |  |
| --- | --- |
| $\_SESSION[‘ap\_uid’] | 当前登录用户的编号 |
| $\_SESSION[‘ap\_uname’] | 当前登录用户姓名 |

注意：由于本平台$\_SESSION 管理方式已经定制，所以不要往$\_SESSION 中存储任何内容[[2]](#footnote-2)。

4.3$SKIP\_SIGN 跳过用户登录认证

本变量在include ‘../inc.php’;之前设置方可生效，例如：

<?php

$SKIP\_SIGN=true; include ‘../inc.php’;

……

?>

带有此标识的页面，将不再判断用户是否登录。但，如果页面中使用了 acl(…)权限判断函数，则会返回无权限。

4.4acl判断当前用户是否拥有指定权限

定义

function acl($privname,$wlog=true,$check\_path=true)

参数

$privname [必须]要查询的权限名称

$wlog [可选]当没有权限时，是否需要进行记录[[3]](#footnote-3)

$check\_path [可选]是否检查模块路径[[4]](#footnote-4)

返回值

如果当前用户拥有$priv权限，则返回1，否则返回0。

说明

此函数会读取当前URL中的mid参数，所以调用时请保证在URL中mid参数必须存在。 如果调用本函数，只为了判断用户是否具有某权限，而不是限制页面访问，则必须填写

$log参数。因为，如果不填写时，系统会自动写log，这样会造成大量无用的log。具体参见示例3。

示例

如果没有权限，则停止执行:

acl(‘查看’) or die;

根据权限做不同处理:

if(acl(‘管理’,false)) echo ‘管理权限’; else

echo ‘无权限’;

示例2(多个权限多次使用的情况)：

|  |
| --- |
| …… if(acl('区管理')) $priv='admin'; else if(acl('校管理')) $priv='school'; else if(acl('查看')) $priv='read'; else  die;  ……  if($priv=='admin') echo '<a href="?action=del">删除</a>';  if($priv=='admin' || $priv=='school') echo '<a href="?action=edit">编辑</a>';  …… |

注：不推荐这样调用。在以前版本的AP上，这样调用是为了减少读库次数，新版的AP已经优化了此部分代码，所以不需要单独存到一个变量中，只需要在需要的地方直接调用即可。

示例3

if(acl('管理')) $priv\_admin=true; if(acl('流程')) $priv\_work=true; if(acl('涉外流程')) $priv\_en=true; if(acl('专利外内流程')) $priv\_en2=true; if(acl('商标内内流程')) $priv\_en1=true; if(acl('客服')) $priv\_service=true;

这段代码原意是把这些权限保存在一个本地变量中，但是这样调用会在“日志管理”模块会产生6条记录：



这6条记录，实际上是没有任何用处的。所以这样的调用一定要增加$log参数，改为：

if(acl('管理',false)) $priv\_admin=true; if(acl('流程',false)) $priv\_work=true; if(acl('涉外流程',false)) $priv\_en=true; if(acl('专利外内流程',false)) $priv\_en2=true; if(acl('商标内内流程',false)) $priv\_en1=true; if(acl('客服',false)) $priv\_service=true;

4.5acl2判断当前用户是否拥有指定模块的指定权限

定义

function acl2($mid,$priv)

参数

$mid [必须]模块编号或模块路径

$priv [必须]要查询的权限名称

返回值

如果当前用户拥有$priv权限，则返回1，否则返回0。

说明 本函数使用类似与acl。主要区别是，本函数需要在调用时提供模块ID或路径。

本函数是acl的一个补充，不存在替代关系。

需要注意的是，如果使用模块路径的方式，可能同一模块被多次引用，那么只要有一个引用存在$priv就会返回拥有权限。

示例

如果没有权限，则停止执行:

<?php

acl2(123,’一审’) or die;

?>

或

<?php

acl2(‘/pub-info/’,’一审’) or die;

?>

4.6wlog 写日志

定义

function wlog($title,$message,$iid)

参数

|  |  |
| --- | --- |
| $title | [必须]日志标题 |
| $message | [必须]日志内容，支持HTML字符串或数组格式。 |
| $iid | [可选]关联的数据ID，这个是用于用户自定义使用。 |

返回值

无

说明 本函数写入的日志信息，可以在日志管理(sys-log)模块中进行查看、管理。 推荐使用JSON格式写入LOG，因为这样在查看LOG时会更清晰一些。

示例

wlog(‘用户管理’,”接收到非法信息<br>用户编号:{$\_GET[‘uid’]}”);

wlog(‘会员管理’,”删除会员<br>会员编号:{$\_GET[‘id’]}”,123);

wlog(‘会员管理’,array(‘操作’=>’删除会员’,’会员编号’=>$\_GET[‘id’]));

4.7dbtools::updown 数据上移下移操作

定义

function updown($updown,$table,$id,$id\_col='id',$oc\_col='oc',$where='')

参数

$updown = up | down

$table = 所在表名

$id = id

$id\_col = id列，列名

$oc\_col = 排序列列名 $where = 附加条件返回值

无

说明 对指定表的指定数据进行上移、下移处理。

示例

dbtools::updown('down','t\_pub\_info',"{$\_GET['id']}",'id','oc',"mid='{$\_GET['mid']}'");

4.8dept::get\_email读取部门邮箱

定义

function get\_email($dept\_ids=array())

参数

$dept\_ids [可选]部门ID列表。如果不填写，则系统自动读取当前用户所在的部门。

返回值 指定部门（或当前用户）的上层部门邮箱。返回的是一个邮箱字符串数组。 如果未找到设置的邮箱，则会返回一个空数组。

说明 当用户同时属于多个部门的时候，将返回最近一层部门的邮箱列表，如果同时多个同层部门都设有部门邮箱，则会同时返回。

示例

<?php

$em=dept::get\_email(); print\_r($em);

?>

一个邮箱设置时返回：

Array ( [0] => lym510416@163.com )

多个邮箱设置时返回：

Array ( [0] => 0123@163.com [1] => lym510416@163.com )

4.9fav::set 设置关注标记定义

|  |  |
| --- | --- |
| function set($uid,$cid=0,$mid=0) | |
| 参数  $uid | [必须]设置的用户ID。 |
| $cid | [可选]设置数据项ID。 |
| $mid 返回值  无  说明  无  示例 | [可选]设置对应mid |
| fav::set($\_SESSION(‘ap\_uid’));  fav::set($\_SESSION(‘ap\_uid’),0,$\_GET[‘mid’]); | |

上面示例中的两项效果是等同的。

4.10 fav::get 读取关注标记

定义

function get($uid,$cid=0,$mid=0)

参数

|  |  |
| --- | --- |
| $uid | [必须]设置的用户ID。 |
| $cid | [可选]设置数据项ID。 |
| $mid | [可选]设置对应mid |

返回值 存在返回true。说明

无

示例

fav::get($\_SESSION(‘ap\_uid’)); fav::get($\_SESSION(‘ap\_uid’),0,$\_GET[‘mid’]);

上面示例中的两项效果是等同的。

4.11 fav::del 删除关注标记定义

|  |  |
| --- | --- |
| function del($uid,$cid=0,$mid=0) | |
| 参数  $uid | [必须]设置的用户ID。 |
| $cid | [可选]设置数据项ID。 |
| $mid 返回值  无  说明  无  示例 | [可选]设置对应mid |
| fav::del($\_SESSION(‘ap\_uid’));  fav::del($\_SESSION(‘ap\_uid’),0,$\_GET[‘mid’]); | |

上面示例中的两项效果是等同的。

4.12 fav::get\_list\_uid 获取指定条件的用户列表

定义

function get\_list\_uid($cid=0,$mid=0)

参数

|  |  |
| --- | --- |
| $cid | [可选]设置数据项ID。 |
| $mid | [可选]设置对应mid,如果不设置，则默认为：$\_GET[‘mid’] |

返回值 符合参数指定条件的，包含uid参数的数组。

说明

无

示例

|  |
| --- |
| <?php  $list=fav::get\_list\_uid(); print\_r($list);  ?> |

运行后显示效果

Array

(

1. => 12
2. => 34
3. => 56
4. => 78

)

4.13 file::delete 删除指定文件

定义

function delete($filename)

参数

$filename upload目录下的文件名称。以upload目录为基路径。

返回值

无

说明 删除上传的指定文件

示例

<?php

file::delete(‘20150708/1234567/a.docx’); ?>

4.14 file::dlnk 生成临时下载路径

定义

function dlnk($filename,$shortname=false)

参数

$filename upload目录下的文件名称。以upload目录为基路径。 $shortname 是否启用短路径。启用短路径后，会返回一个a.\*的下载路径。返回值 相对于ap根目录的下载路径。

说明 主要是把上传文件路径转换为下载路径。

使用下载路径的好处是，24 小时后下载路径会自动删除。防止了文件实际路径的长时间暴漏。示例

<?php

echo file::dlnk($row[‘file’]); ?>

4.15 file::alias 生成临时别名下载路径

定义

function alias($filename,$alias)

参数

$filename [必须]upload目录下的文件名称。以upload目录为基路径。

$alias [必须]新的文件名称。

返回值 相对于ap根目录的下载路径。

说明 主要是把上传文件路径转换为下载路径。

使用下载路径的好处是，24 小时后下载路径会自动删除。防止了文件实际路径的长时间暴漏。

示例

<?php

echo file::alias($file,’新文件’);

?>

如果$file=201611/12345612/old.doc，则返回路径为:/download/1234567/新文件.doc。

注意，此函数会自动生成文件到临时目录。

~~4.16 file::download下载指定路径文件~~

此函数已经file::download2替代。因为此函数传递

urlencode后的文件路径时会发生错误。这个问题已经在

file::download2中修复。

定义

function download($url)

参数

$url [必须]下载的文件路径

返回值 相对于根upload目录的标准下载路径。

说明 本函数的主要功能是从指定网址下载文件，并保存在服务器默认路径下。

如果文件过大，可能会出现文件超时的情况，请根据实际情况，自行设置limit\_time 时间。

示例

<?php echo file::download('http://ubmcmm.baidustatic.com/media/v1/0f0000uKQpx.jpg');

?>

执行结果

201612/20161140/0f0000uKQpx.jpg

4.17 file::download2下载指定路径文件

定义

function download2($url)

参数

$url [必须]下载的文件路径，支持urlencode后的文件名。

返回值 相对于根upload目录的标准下载路径。

说明 本函数的主要功能是从指定网址下载文件，并保存在服务器默认路径下。

如果文件过大，可能会出现文件超时的情况，请根据实际情况，自行设置limit\_time 时间。

示例

<?php echo file::download2('http://ubmcmm.baidustatic.com/media/v1/0f0000uKQpx.jpg');

?>

执行结果

201612/20161140/0f0000uKQpx.jpg

4.18 file::upfile 标准文件上传处理

定义

function upfile($postname='file')

参数

$postname POST中FILES字段，默认为:file 返回值 如果上传成功，则返回上传后的文件路径。 如果上传失败，则返回空字符串。

说明 在标准模式下，返回路径为/upload/目录的相对路径。

示例

HTML部分

<form method="post" enctype="multipart/form-data" name="form1" id="form1">

<input type="file" name="icon" id="icon">

<input type="submit" name="submit" id="submit" value="保存">

</form>

PHP部分

<?php

$icon=file::upfile('icon'); echo $icon;

?>

运行结果

201402/28194634/untitled.png

4.19 file::upfiles标准文件批量上传处理

定义

function upfiles($postname='file')

参数

$postname POST中FILES字段，默认为:file。注意，这个名称不带有[]符号。

返回值 如果上传成功，则返回上传后的文件路径数组。 如果上传失败，则返回空数组。

说明 在标准模式下，返回路径为/upload/目录的相对路径。

示例

HTML部分

<form method="post" enctype="multipart/form-data" name="form1" id="form1">

<input type="file" name=”file[]” id="file1">

<input type="file" name=”file[]” id="file2">

<input type="file" name=”file[]” id="file3">

<input type="submit" name="submit" id="submit" value="保存">

</form>

PHP部分

<?php

$icon=file::upfiles('file'); print\_r($icon);

?>

运行结果

|  |
| --- |
| [  “201402/28194634/untitled1.png”,  “201402/28194634/untitled2.png”,  “201402/28194634/untitled3.png”  ] |

4.20 file::upload\_mkdir 创建目录

定义

function upload\_mkdir()

参数

无

返回值 在upload创建一个唯一名称的目录。

说明 生成一个唯一的临时目录。

示例

<?php echo file::upload\_mkdir();

?>

输出

/var/www/ap/upload/201609/01121212/

4.21 file::rcopy 拷贝目录

定义

function rcopy($src,$dst)

参数

|  |  |
| --- | --- |
| $src | 源目录 |
| $dst | 目标目录 |

返回值

无

说明 源目录中的目录全部拷贝到目标目录中。

示例

<?php

file::rcopy(‘../download/t1234567890’,’../upload/m123’); ?>

4.22 file::rmdir 删除目录

定义

function rmdir($dir)

参数

$dir 要删除的目录返回值

无

说明 这个删除与PHP中的rmdir的区别是，PHP中的只能删除空目录，这个mkdir会自动删除目录下的所有文件。

示例

<?php

file::rmdir(‘../upload/1234567’); ?>

4.23 file::save 保存指定文件内容到标准路径

定义

function save($filename,$content)

参数

$filename [必须]文件名 $content [必须]文件内容返回值 保存后文件的标准路径。

说明

无

示例

|  |
| --- |
| <?php  $content=’11111111112222222223333333333’;  $fn=file::save(‘1.txt’,$content); echo $fn  ?> |

4.24 file::tempdir 创建临时目录

定义

function tempdir()

参数

无

返回值 临时目录的绝对路径。

说明 生成一个唯一的临时目录。此临时路径前缀为t。 增加t的原因是，为了不与dlnk生成的临时路径产生冲突。

示例

<?php echo file::tempdir();

?>

输出

/var/www/ap/download/t1409937471/

4.25 ini::set 设置INI项

定义

function set($uid,$name,$value)

参数

$uid 当前用户ID，系统级设置为0。如果为-1，则系统默认取当前用户ID。

$name 设置项名称 $value 设置项值返回值

无

说明 为安全考虑，如果$uid传递的不是数字，则函数会直接调用die。

示例

ini::set($\_SESSION(‘ap\_uid’),’bg’,$file); ini::set(-1,’bg’,$file);

上面示例中的两项效果是等同的。

4.26 ini::get 读取INI项

定义

function get($uid,$name,$def=’’)

|  |  |
| --- | --- |
| 参数 |  |
| $uid | 当前用户ID，系统级设置为0。如果为-1，则系统默认取当前用户ID。 |
| $name | 设置项名称 |
| $def | 默认值。如果未找到指定设置，则返回此值。默认:’’ |

返回值 读取的INI设置值。

说明 为安全考虑，如果$uid传递的不是数字，则函数会直接调用die。

示例

ini::get($\_SESSION[‘ap\_uid’],’bg’);

ini::get($\_SESSION[‘ap\_uid’],’bg’,’/images/bg.jpg’);

4.27 ini::del 删除INI项

定义

function del($uid,$name)

参数

$uid 当前用户ID，系统级设置为0。如果为-1，则系统默认取当前用户ID。

$name 设置项名称返回值

无

说明 为安全考虑，如果$uid传递的不是数字，则函数会直接调用die。

示例

ini::del($\_SESSION[‘ap\_uid’],’bg’);

4.28 msg::send 当前用户向指定人发送信息

定义

function send($to,$title,$message,$file,$dt\_timer='')

参数

|  |  |
| --- | --- |
| $to | [必须]接收人id。如果是多人接收，则为array类型 |
| $title | [必须]信息标题 |
| $message | [必须]信息内容 |
| $file | [可选]附件地址 |
| $dt\_timer | [可选]定时消息的定时时间 |

返回值

无

说明 如果$file参数为“http://” 开头的，则在向企业微信号发送信息时，作为展示路径发送的。

示例

<?php

msg::send(9,’提示信息’,’这是提示信息内容’);

msg::send(9,’提示信息’,’发送一个指定时间到达的信息’,’2014-09-25 10:10’); msg::send(array(9,10,11),’提示信息’,’向多人同时发送信息’);

?>

4.29 msg::sys 系统账号向指定人发送信息

定义

function sys($to,$title,$message,$file,$dt\_timer='',$mid=0,$opid=0)

参数

|  |  |
| --- | --- |
| $to | [必须]接收人id。如果是多人接收，则为array类型 |
| $title | [必须]信息标题 |
| $message | [必须]信息内容 |
| $file | [可选]附件地址 |
| $dt\_timer | [可选]定时消息的定时时间 |
| $mid | [可选]关联模块 |
| $opid | [可选]关联ID |

返回值

无

说明 如果$file参数为“http://” 开头的，则在向企业微信号发送信息时，作为展示路径发送的。

示例

<?php

msg::sys(9,’提示信息’,’这是提示信息内容’);

msg::sys(9,’提示信息’,’发送一个指定时间到达的信息’,’2014-09-25 10:10’); msg::sys(array(9,10,11),’提示信息’,’向多人同时发送信息’);

?>

4.30 msg::ignore忽略指定信息

定义

function ignore($mid,$opid,$uid=0)

参数

|  |  |  |
| --- | --- | --- |
| $mid |  | [必须]模块编号 |
| $opid |  | [必须]发送信息时填的opid |
| $uid |  | [可选]指定人 |

返回值

无

说明 本函数将会把指定的信息，自动置为已读。

示例

<?php

//将指定opid=1000的信息置为已读

msg::ignore($\_GET[‘mid’],1000);

//将指定人的opid的信息置为已读

msg::ignore($\_GET[‘mid’],1000,9); ?>

4.31 options::dd输出部门列表

定义

function dd($def,$parent\_id=0)

参数

$def 默认选中的部门编号

$parent\_id 上层部门编号，默认值为顶层部门。说明 直接输出option标签返回值

无

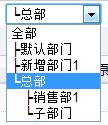
示例

<?php

options::dd($\_GET[‘dept\_id’]);

?>

显示效果：



4.32 options::topdept输出顶层部门

定义

function topdept($def,$parent\_id=0)

参数

$def 默认选中的部门编号

$parent\_id 上层部门编号，默认部门为顶层部门说明 直接输出option标签返回值

无

示例

<?php

options::topdept($\_GET[‘dept\_id’]); ?>

4.33 options::ud 输出部门及用户列表

定义

public function ud($def,$parent\_id=0)

参数

$def 默认

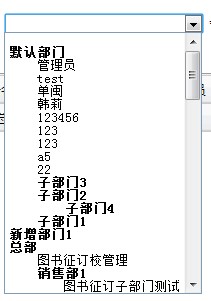
$parent\_id 上层部门编号

说明 输出以指定部门为基准的所有下层部门及用户option字符串

返回值

无

示例



4.34 role::check\_parent\_dept\_enabled 检查上层部门是

否启用状态

定义

function check\_parent\_dept\_enabled($dept\_list)

参数

$dept\_list [必须]部门列表数组。如果是字符串则自动转为数组进行处理。

返回值 如果上层部门启用则返回true,否则返回false。

说明 这个判断会从当前部门一直判断到顶层部门。 如果具有多个上层部门，则只要有一个部门启用就返回true。

示例

<?php

echo role::check\_parent\_dept\_enabled(123); echo role::check\_parent\_dept\_enabled(array(123)); echo role::check\_parent\_dept\_enabled(array(123,456)); ?>

4.35 role::dept\_edit 编辑部门信息

定义

function dept\_edit($id,$name,$enabled,$parent\_id=-1)

参数

$id [必须]部门编号

$name [必须]部门名称

$enabled [必须]是否启用

$parent\_id [可选]上层部门。如果不传递此参数（或传递-1），则保持原有上层部门关系。如果传递0，则当前部门作为最上层部门处理。

返回值

无

说明 编辑部门信息。

4.36 role::dept\_new 创建新部门

定义

public function dept\_new($name,$parent\_id)

参数

$name 新部门名称

$parent\_id 上层部门编号。如果此参数为NULL（即不传递），则为顶层部门。返回值 新建部门ID 说明

创建新部门。

4.37 role::dept\_top 使用部门名称读取顶层部门ID

定义

public function dept\_top($name)

参数

$name 部门名称

返回值

如果存在返回部门ID，如果不存在返回空。

说明

这个只读取顶层部门。如果顶层有多个同名部门，则返回第一个部门。

4.38 role::edit 角色编缉

定义

function edit($id,$name,$type,$enabled=1,$args=’’)

|  |  |
| --- | --- |
| 参数 |  |
| $id | [必须]角色编号，=0时为新建 |
| $name | [必须]角色名称 |
| $type | [必须]角色类型 |
| $enabled | [可选]启用状态,默认：1启用 |
| $args | [可选]附加参数,如果是数组，则自动转换为json |

返回值 新增或编辑角色的ID。

说明

无

示例

<?php

role::edit(0,’新角色’,’r’); ?>

4.39 role::icon\_add 增加角色图标

定义

function icon\_add($id)

参数 $id 角色编号隐藏参数

$\_FILES['icon'] 图标上传名称

返回值 成功返回true，失败返回false。

说明 这个功能会自动更改数据库内的内容。会自动在 t\_sys\_role.args 字段中增加

{“icon”:”icon……”}内容。

注意 如果原来t\_sys\_role.args字段中存在icon内容，则本命令会覆盖之前的设置。并且，原icon指定的文件也会一并删除。

4.40 role::icon\_del 删除指定角色图标

定义

function icon\_del($id)

参数 $id 角色编号返回值 正确删除图标返回true，否则返回false。

说明 本函数，在删除图标配置信息的同时，会同时删掉原图标文件。

4.41 role::id 读取指定角色ID

定义

function id($name,$type)

参数

$name [必须]角色名称

$type [可选]角色类型。如果不设置，则从所有角色类型中查找。

返回值 如果存在指定角色，返回角色ID。 如果不存在返回空。

说明

无

调用示例

echo role::id(“#行政单位”,”d”);

4.42 role::id2读取指定角色ID

定义

function id2($name,$type='m')

参数

$name [必须]角色名称。支持以“/”字符分割的分级。

$type [可选]角色类型。默认:m。

返回值 如果存在指定角色，返回角色ID。 如果不存在返回空。

说明 这方法与role::id相比,支持分级的角色读取,例如存在上下级的部门、模块等。

调用示例

<?php

echo role::id2(‘测试模块’); echo role::id2(‘测试模块/测试模块下层’);

?>

4.43 role::mid 使用路径、权限读取模块编号

定义

function mid($path,$priv=’’)

参数

$path [必须]模块路径

$priv [可选]当前用户具有的权限

返回值

模块编号

说明

这个函数通过模块路径标识返回模块编号。

因为同一个路径会被多次引用，所以如果不使用$priv可能会引起返回的mid不确定的问题。请在使用此函数前确定这个不会影响现有程序的执行。

如果设置了$priv 参数，则本函数返回符合$path 指定的模块路径，且当前用户具有

$priv权限的模块编号。

注意，模块路径标识前后要有“/”符号，具体参见示例。

示例

<?php

echo role::mid(‘/test-test/’);

?>

具有权限的模块读取

<?php

echo role::mid(‘/test-test/’,’查看’);

?>

4.44 role::ispriv 判断指定模块是否存在指定权限

定义

function ispriv($priv,$mid)

参数

$priv [必须]权限名称或权限编号 $mid [可选]模块编号。如果未设置则为当前模块编号。返回值 如果存在返回true，否则返回false。

示例

if(role::ispriv(‘管理’))

{

……

}

if(role::ispriv(‘管理’,100))

{

……

}

4.45 role::is\_child 判断指定角色是否在指定父层角色下面

定义

function is\_child($role\_id,$parent\_id,$relation\_type)

参数

$role\_id 要查询的角色编号

$parent\_id 上层目标角色编号

$relation\_type 角色关系

返回值

如果$role\_id属于$parent\_id下面，返回true，否则返回false。

示例

if(role::is\_child(123,45,’dd’))

echo ‘123部门是45的下属部门’;

else

echo ‘123部门不是45的下属部门’;

4.46 role::get\_all\_id 读取所有符合条件的ID

定义

function get\_all\_id($name,$type)

参数

|  |  |
| --- | --- |
| $name | 角色名称 |
| $type | 角色类型 |

返回值 符合条件的角色ID数组。

说明 此方法不支持分层操作。

示例

<?php

$ret=role::get\_all\_id(‘/pub-info/’,’m’);//读取所有pub-info模块

$ret=role::get\_all\_id(‘办公室’,’d’); //读取所有名称为“办公室”的部门

?>

4.47 role::get\_all\_mid 读取所有符合条件的mid

定义

function get\_all\_mid($module)

参数

$module [必须]模块路径

返回值 符合条件的模块ID数组。

说明

无。

示例

<?php

$ret=role::get\_all\_mid(‘/pub-info/’);//读取所有pub-info模块

?>

4.48 role::get\_all\_child\_id 读取所有子角色编号

定义

function get\_all\_child\_id($id,$relation\_type=’dd’,$root\_flag=false,$count=10)

参数

$id [必须]父层角色编号或数组

$relation\_type [可选]角色关系或数组，默认：dd

$root\_flag [可选]是否包含根节点数据，默认：false

$count [可选]读取几层子角色，默认：10，最小1

返回值 包含子角色编号的数组。

示例

<?php

//单参数

$ret=role::get\_all\_child\_id(123,’dd’);

//数组

$arr1=array(123,456); $arr2=array(‘ud’,’dd’);

$ret=role::get\_all\_child\_id($arr1,$arr2);

?>

4.49 role::get\_all\_child\_info 读取所有子角色信息

定义

function get\_all\_child\_info($id,$relation\_type=’dd’,$root\_flag=false,$count=10)

参数

$id [必须]父层角色编号

$relation\_type [可选]角色关系，默认：dd

$root\_flag [可选]是否包含根节点数据，默认：false

$count [可选]读取几层子角色，默认：10

返回值 所有子角色的数组，数据格式：

|  |  |
| --- | --- |
| [        ] | {“id”:”123”,”name”:”角色名称1”,”enabled”:”1”}, {“id”:”345”,”name”:”角色名称2”,”enabled”:”0”},  …… |

示例

<?php

$ret=role::get\_all\_child\_info(123,’dd’);

print\_r($ret); ?>

4.50 role::get\_all\_top\_info 读取所有顶层角色信息

定义

function get\_all\_top\_info($type=’d’,$relation\_type=’dd’)

参数

$type [可选]角色类型，默认:d

$relation\_type [可选]角色关系，默认：dd

返回值 所有子角色的数组，数据格式：

|  |  |
| --- | --- |
| [        ] | {“id”:”123”,”name”:”角色名称1”}, {“id”:”345”,”name”:”角色名称2”},  …… |

示例

<?php

//读取所有顶层部门

$ret=role::get\_all\_top\_info();

print\_r($ret); ?>

4.51 role::get\_all\_parent\_id 读取所有上层角色编号

定义

function get\_all\_parent\_id($id,$relation\_type=’dd’,$root\_flag=false,$count=1)

参数

$id [必须]角色编号

$relation\_type [可选]与父层角色关系，默认：dd

$root\_flag [可选]是否包含$id点数据，默认：false

$count [可选]读取几层子角色，默认：1

说明 查询部门的上层部门$relation\_type=dd 查询用户的上层部门$relation\_type=ud

返回值 包含父角色编号的数组。

示例

<?php

$ret=role::get\_all\_parent\_id(123,’dd’);

?>

4.52 role::getdept读取指定用户或部门的上层部门(停用)

由于用户允许存在多个上层部门，所以此函数的返回会引起数据的二义性。所以将彻底停用此函数！请使用role::get\_all\_parent\_id来替代！

定义

function getdept($id=0,$top=false)

参数

$id 指定的用户或部门，如果不指定($id=0)用户或部门,则默认为当前用户。

$top 是否读取顶层部门

返回值

上层部门或顶层部门（由$top决定）的编号

说明 因为当前系统，允许一个用户同时存在于多个部门。所以$id指定的是一个用户时，并

且这个用户有2个或2个以上的部门，则会随机返回一个上层部门信息。

示例

echo role::getdept(9);

4.53 role::getdept2 读取指定用户的上层部门列表

定义

function getdept2($id=0)

参数

$id 指定的用户或部门，如果不指定($id=0)用户或部门,则默认为当前用户。

返回值 返回值是一个数组类型，包含指定用户的上层部门列表。

说明

示例

echo role::getdept2(9);

~~4.54 role::getuser\_from\_priv 取回拥有指定权限的所有~~

~~用户~~

请不要再调用此函数！

此函数将被role::getuserid\_from\_priv()替代！

定义

function getuser\_from\_priv($priv,$mid=null)

参数

$priv [必须]权限名称

$mid [可选]模块编号，如果不传递，则默认为$\_GET[‘mid’] 返回值 拥有指定权限的所有用户编号

示例

<?php

$ulist=role::getuser\_from\_priv(‘管理’);

$ulust=role::getuser\_from\_priv(‘管理’,101);

?>

4.55 role::getuserid\_from\_priv 取回拥有指定权限的所

有用户编号

定义

function getusername\_from\_priv($priv,$mpath=0)

参数

$priv [必须]权限名称,支持数组。 $mpath [必须]模块路径或模块编号，默认当前模块。返回值 拥有指定权限的所有用户ID的数组。

示例

<?php

print\_r(role::getuserid\_from\_priv(‘管理’,’/test\_module/’));

?>

显示示例:

Array

(

1. => 14007
2. => 9
3. => 203
4. => 13451
5. => 14402
6. => 12235
7. => 14454
8. => 13993
9. => 13571
10. => 14424
11. => 13459
12. => 13992
13. => 12027
14. => 14892
15. => 14893
16. => 15208 [18] => 15796

)

4.56 role::getusername\_from\_priv 取回拥有指定权限的

所有用户名称

定义

function getusername\_from\_priv($priv,$mpath=0)

参数

$priv [必须]权限名称,支持数组。 $mpath [必须]模块路径或模块编号，默认当前模块。返回值 拥有指定权限的所有用户名称，以逗号隔开。

示例

<?php

echo role::getusername\_from\_priv(‘管理’,’/test\_module/’);

?>

显示示例:

管理员,张三,李四

4.57 role::get\_info 读取指定角色信息

定义

function get\_info($id,$cols=array('name'))

参数

|  |  |
| --- | --- |
| $id | 读取角色的编号 |
| $cols | 读取数据列 |

返回值 含有指定数据的数组对象。

说明

无

示例

<?php

print\_r(role::get\_info(123));

?>

输出

{‘name’=>’角色名称’}

4.58 role::name 读取指定角色名称

定义

function name($id)

参数

$id 角色编号

返回值 如果指定的角色存在，则返回角色的名称。

如果指定的角色不存在，则返回NULL。

说明 这里所指的“角色”是指包括“用户”、“模块”、“角色”、“部门”、“权限”等。

为安全考虑，如果$id传递的不是数字，则函数会直接调用die。

示例

echo role::name(9);

4.59 role::parent\_deptname 读取指定角色的上层部门名称

定义

function parent\_deptname($id)

参数 $id 角色编号返回值 上层部门名称

说明 这里的指定角色一般为用户或部门。否则无法读出上层部门的名称。

示例

<?php

echo role::parent\_deptname(123); ?>

4.60 role::prefix 返回指定角色的前缀

定义

function prefix($role\_id,$relation\_type)

参数

$role\_id 角色ID

$relation\_type 角色前缀关系类型，多个关系类型使用”,”隔开返回值 以” >> ”分割的关系字符串。

说明 这里所指的“关系类型”是指系统标准角色关系类型中的类型。常用的关系类型有： ur 用户 -> 角色 rr 角色 -> 角色 ud 用户 -> 部门 dd 部门 -> 部门 mm 模块 -> 模块 pm 权限 -> 模块 为安全考虑，如果$role\_id传递的不是数字或$relation\_type为空，则函数会直接调用die。示例取用户编号为9的用户的部门信息

echo role::prefix(9,’ud’);

运行结果

默认部门 >> 管理员

4.61 role::get\_parent\_id 读取上层角色编号

定义

function get\_parent\_id($id,$relation\_type='dd')

参数

$id [必须]角色编号

$relation\_type [可选]角色关系，默认:dd

返回值 如果存在指定上层关系，返回上层ID；否则返回空。

说明 常用角色关系： dd 查找部门的上层部门 mm 查找模块的上层模块 ud 查找用户的上层部门

示例

<?php

echo role::get\_parent\_id(123); //查找123的上层部门，因为默认$relation\_type=’dd’ echo role::get\_parent\_id(234,’mm’); //查找234模块的上层模块

?>

4.62 role::get\_parent\_info获取上层角色信息

定义

function get\_parent\_info($id,$relation\_type='dd')

参数

$id [必须]当前角色编号

$relation\_type [可选]当前角色与上层角色的关系。默认值：dd。

返回值 如果存在上层角色，则返回上层角色信息；否则返回空。

返回值结构：

{

“id”:”123”, //上层角色编号

“name”:”部门名称”, //上层角色名称

“args”:[] //上层角色附加信息（JSON）

}

说明

无。

示例

$ret=role::get\_parent\_info(101);

4.63 role::get\_parents\_info获取上层角色信息

定义

function get\_parents\_info($id,$relation\_type='dd')

参数

$id [必须]当前角色编号

$relation\_type [可选]当前角色与上层角色的关系。默认值：dd。

返回值 如果存在上层角色，则返回所有上层角色信息；否则返回空。

返回值结构： [

{

“id”:”123”, //上层角色编号

“name”:”部门名称”, //上层角色名称

“args”:[] //上层角色附加信息（JSON）

},

……

]

说明

无。

示例

$ret=role::get\_parents\_info(101);

~~4.64 role::get\_ud读取指定用户上级部门编号~~

本函数已被停用！由于系统支持一个用户多部门，所以此函数存在二义性。

定义

function get\_ud($uid=0)

参数

$uid 用户编号。默认值：0

返回值 上级部门编号。

说明 如果$uid传递0，则返回当前用户的上级部门。

示例返回当前用户的上级部门编号

<?php

$dept\_id=role::get\_ud();

?>

返回指定用户的上级部门编号

<?php

$dept\_id=role::get\_ud(123);

?>

4.65 role::relation\_add 增加角色关系

定义

function relation\_add($role\_id,$parent\_id,$relation\_type)

参数

$role\_id 下层角色

$parent\_id 上层角色

$relation\_id 角色关系

返回值

无

说明 创建指定角色关系

示例

<?php

role::relation\_add(100,200,’dd’); ?>

4.66 role::relation\_exists指定角色关系是否存在

定义

function relation\_exists($role\_id,$parent\_id,$relation\_type)

参数

$role\_id 下层角色

$parent\_id 上层角色

$relation\_id 角色关系

返回值 存在返回true；否则返回false。

说明 本方法只是判断指定关系是否存在，没有其他操作

示例

|  |
| --- |
| <?php  include ‘../inc.php’;    if(role::relation\_exists(100,10,’dd’))  echo ‘角色关系存在’; else  echo ‘角色关系不存在’;  ?> |

4.67 role::relation\_remove 移除指定角色关系

定义

function relation\_remove($role\_id,$parent\_id,$relation\_type)

参数

|  |  |
| --- | --- |
| $role\_id | 下层角色 |
| $parent\_id | 上层角色 |
| $relation\_id | 角色关系 |

返回值

无

说明 删除指定角色关系

示例

<?php

role::relation\_remove(100,200,’dd’); ?>

~~4.68 role::sign 用户登录(已停用)~~

此接口已被停用，请使用role::sign2替代。

定义

function sign($uid,$pwd)

参数

$uid 登录名称

$pwd 登录密码

返回值 登录正确返回true，登录失败返回错误信息。

示例

|  |
| --- |
| <?php  $ret=role::sign(‘user’,’123123’); if($ret===true)  echo ‘登录成功’; else  echo “登录失败，失败原因：{$ret}”;  ?> |

4.69 role::sign2 用户登录

定义

function sign2($uid,$pwd)

参数

$uid [必须]登录名称

$pwd [必须]登录密码

返回值 返回状态数组，格式如下：

[

‘success’:true|false, //正常登录为true，否则false

‘info’:’提示信息’, //当登录失败时，这里为失败原因

‘ap\_sid’:’abcdefg’, //当前session\_id

‘ap\_uid’:123, //当前登录用户ID

‘ap\_uname’:’姓名’, //当前登录用户姓名

‘ap\_logo’:’my.png’ //当前登录用户头像地址

]

说明 支持OTP登录，OTP密码直接传到$pwd参数即可。

示例

<?php

$ret=role::sign2(‘user’,’123123’); if($ret[‘success’])

echo “登录成功，用户编号:{$ret[‘ap\_uid’]}”; else

echo “登录失败，失败原因:{$ret[‘info’]}”;

?>

4.70 role::user\_dept\_add指定用户增加到指定部门下

定义

public function user\_dept\_add($id,$dept)

参数

$id 用户编号

$dept 部门编号

返回值 正常返回true,否则返回错误信息

说明 当前用户支持存在于多部门，这个就是增加指定用户到指定部门的方法。

示例

4.71 role::user\_dept\_remove从指定部门移除指定用户

定义

public function user\_dept\_remove($id,$dept)

参数

$id 指定用户编号

$dept 指定部门编号

返回值 如果正常返回true，否则返回错误信息。

说明 本方法将指定用户从指定的部门中移除。

示例

<?php

user\_dept\_remove(9,101); ?>

~~4.72 role::user\_edit编辑用户信息(已停用)~~

此方法停用!!!请使用user\_edit2替代。

定义

public function user\_edit($id,$login,$name,$enabled,$new\_dept,$old\_dept)

参数

$id [必须]用户编号。如果为0则新增用户。

$login [必须]用户登录名称

$name [必须]用户姓名

$enabled [必须]是否启用，取之：1或0

$new\_dept[] [可选]所属部门编号数组，如不传递则不进行部门相关操作

$old\_dept[] [可选]要删除的之前所属部门编号数组，如不传递则不删除之前部门关系

返回值

如果执行正常返回操作用户的ID，否则返回错误信息文本。

说明 新增用户信息时，如果新增正确返回新的用户ID； 编辑用户信息时，如果编辑正确返回编辑用户ID。

当新增(编辑)发生错误时，直接返回错误信息文本。

示例

|  |
| --- |
| $ret=role::user\_edit($\_POST['uid'],$\_POST['login'],$\_POST['name'],1,$\_POST['dept\_id']); if(is\_numeric($ret))  {  ……正常处理  } else  {  ……出错处理  } |

4.73 role::user\_edit2编辑用户信息

定义

function user\_edit2($id,$login,$name,$enabled=1,$args,$dept\_array)

参数

$id [必须]用户编号。如果为0则新增用户。

$login [必须]用户登录名称

$name [必须]用户姓名

$enabled [可选]是否启用，取值：1或0。默认：1

$args [可选]附加参数

$dept\_array[] [可选]所属部门编号数组，如不传递则不进行部门相关操作

返回值 如果执行正常返回操作用户的ID，否则返回错误信息文本。

说明 新增用户信息时，如果新增正确返回新的用户ID； 编辑用户信息时，如果编辑正确返回编辑用户ID。 当新增(编辑)发生错误时，直接返回错误信息文本。

示例

|  |
| --- |
| $ret=role::user\_edit2($\_POST['uid'],$\_POST['login'],$\_POST['name'],1,’’,$\_POST['dept\_id']); if(is\_numeric($ret))  {  ……正常处理  } else  {  ……出错处理  } |

4.74 role::user\_enabled用户启用禁用

定义

|  |  |
| --- | --- |
| public function user\_enabled($id,$enabled) | |
| 参数  $id | [必须]用户ID |
| $enabled    返回值  无    示例禁用指定用户 | [必须]是否启用，取值：1启用/0禁用 |
| role::user\_enabled($\_GET['uid'],0); | |

启用指定用户

role::user\_enabled($\_GET['uid'],1);

4.75 role::user\_password\_reset重置指定用户密码

定义

public function user\_password\_reset($id)

参数

$id 用户ID

返回值

无

说明 重置指定用户的密码为“123123”。

示例

role::user\_password\_reset($\_GET['uid']);

4.76 sms::get\_last 读取发送给指定电话号码的最后一条

短信内容

定义

function get\_last($mobile)

参数

$mobile 电话号码

返回值 短信相关信息数组： {

‘sms’=>’短信内容’,

‘result’=>’短信平台返回的信息’,

‘dt\_created’=>’发送时间’,

‘addo’=>’附加内容’

}

说明 此方法只能读取单发的短信信息。如果是群发的短信信息，则需要提供所有按半角逗号隔开的号码才能取到。

对应模块 此功能需要安装：pub-sms模块

示例

<?php

$ret=sms::get\_last(‘13501272822’); print\_r($ret);

?>

输出结果：

|  |
| --- |
| Array  (  [sms] => wp短信测试  [result] => result=0&message=短信发送成功&smsid=2015092604182877  [dt\_created] => 2015-09-26 04:17:51  [addo] => ) |

4.77 sms::send发送短信

定义

function send($mobile,$sms,$addo='')

参数

$mobile 手机号码，多个手机号码时使用半角逗号隔开

$sms 短信内容

$addo 写短信记录时的附加信息，例如短信的验证码等。没有时请不要填写

返回值

无

说明 此方法通过平台向指定用户发送短信。

对应模块 此功能需要安装：pub-sms模块

示例

<?php

sms::send(‘13501272822’,’短信测试’); ?>

4.78 sms::send\_all群发短信

定义

function send\_all($mobile,$sms)

参数

$mobile 手机号码，多个手机号码时使用半角逗号隔开

$sms 短信内容

返回值 无

说明

此方法通过平台向批量用户群发短信。

群发接口，支持一次支持的用户量与短信服务提供商有关，请参阅短信服务提供商提供的文档。

群发与单发的区别：单发短信时，每条短信作为一个记录处理。群发短信时，一组短信作为一条记录处理。

对应模块

此功能需要安装：pub-sms模块

示例

<?php

sms::send\_all(‘13501272822,18607426646’,’短信测试’);

?>

4.79 weibo::short\_url生成短链接

定义

function short\_url($url)

参数

$url 网址

返回值 生成的t.cn短链接

说明 此方法通过平台向weibo.com进行申请得到的短链接。

示例

<?php

echo weibo::short\_url(‘http://baidu.com’); ?>

4.80 tools::date6时间戳转换为6位日期显示

定义

function date6($ts)

参数

$ts [必须]指定显示的时间戳

返回值

无

说明 此方法将直接显示6位的日期，并且生成title属性。在title属性中，显示完整的时间。

示例

<?php

tools::date6(1234567890);

?>

显示效果



4.81 tools::date6time从6位日期类型转换为时间戳

定义

function date6time($d)

参数

$d [必须]指定时间

返回值 指定参数给定的时间戳。

说明 此方法将直接把6位的日期，转换为时间戳，例如：输入为170320，则返回1489939200。

示例

<?php

tools::date6time(‘170320’); ?>

4.82 pdf::merge多文件合并

定义

function merge($files,$outfilename)

参数

$files [必须]要合并的文件列表

$outfilename [可选]输出的文件名称，如果不设置则系统自动生成一个文件名。

返回值 生成的pdf文件路径。

说明 此方法可以合并pdf文件、图片文件。

示例

<?php

echo pdf::merge(array('1.pdf','000001.tif','2.pdf')) ?>

4.83 Office相关操作

此部分的函数，必须在安装ap\_win模块后才可使用。使用前，请确认系统中已经安装了相应的模块。

使用时，一定记得引入定义文件：使用Word相关引用的文件：

include ‘../ap\_win/word.php’;

使用Excel相关引用的文件：

include ‘../ap\_win/excel.php’;

## 4.83.1word::doc2mht格式转换

定义

function doc2mht($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.mht” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把doc文件转换为mht文件。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如： 201608/15143210/123.doc。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::doc2mht($f); print\_r($ret); ?>

## 4.83.2word::doc2pdf格式转换

定义

function doc2pdf($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.pdf” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把doc文件转换为pdf文件。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.doc。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::doc2pdf($f); print\_r($ret); ?>

## 4.83.3word::doc2pdf2格式转换(自动接受修订)

定义

function doc2pdf2($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file [必须]标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.pdf” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把doc文件转换为pdf文件。 与word::doc2pdf2的区别在于，此转换，在转换前会首先接受文档中的所有修订，之后才进行转换。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.doc。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::doc2pdf2($f); print\_r($ret); ?>

## 4.83.4word::doc2split5五书拆分

定义

function doc2split5($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |  |
| --- | --- | --- |
| { | “error”:”错误信息”, //出错时，返回错误信息  “url1”:”201608/15143210/100001.pdf” //转换后的文件标准路径。 | |
|  | “url2”:”201608/15143210/100002.pdf” | //转换后的文件标准路径。 |
|  | “url3”:”201608/15143210/100003.pdf” | //转换后的文件标准路径。 |
|  | “url4”:”201608/15143210/100004.pdf” | //转换后的文件标准路径。 |
| } | “url5”:”201608/15143210/100005.pdf” //转换后的文件标准路径。 | |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明

此方法会自动把doc文件转换为5个标准pdf文件。 注意1：上传的文件必须是分成5节的文件，否则无法正常生成。

注意 2：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.doc。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::doc2split5($f); print\_r($ret); ?>

## 4.83.5word::mht2doc格式转换

定义

function mht2doc($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.doc” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明

此方法会自动把mht文件转换为doc文件。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.mht。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::mht2doc($f); print\_r($ret); ?>

## 4.83.6word::mht2docx格式转换

定义

function mht2docx($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.docx” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把mht文件转换为docx文件(Office 2007标准文件格式)。注意：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.mht。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::mht2docx($f); print\_r($ret);

?>

## 4.83.7word::mht2pdf格式转换

定义

function mht2pdf($file)

引用文件

include ‘../ap\_win/word.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.pdf” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把mht文件转换为pdf文件。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如： 201608/15143210/123.mht。

示例

<?php

include ‘../ap\_win/word.php’;

$f=file::upfile('file'); $ret=word::mht2pdf($f); print\_r($ret); ?>

## 4.83.8word::mht\_replace内容替换

定义

function mht\_replace($search,$replace,$content)

引用文件

include ‘../ap\_win/word.php’;

|  |  |
| --- | --- |
| 参数 |  |
| $search | [必须]搜索内容，类型：数组。 |
| $replace | [必须]替换内容，类型：数组。 |
| $content | [必须]将被替换的内容 |

返回值 返回替换完成的内容。

说明 此方法会自动把$content中的$search中的数据替换成$replace中的数据。

示例

<?php

include ‘../ap\_win/word.php’;

$search=array(‘1’,’2’,’3’);

$replace=array(‘一','二’,’三’);

$content=file\_get\_contents('test.mht');

$ret=word::mht\_replace($search,$replace,$content); echo $ret; ?>

## 4.83.9word::mht\_replace\_file文件内容替换

定义

function mht\_replace\_file($search,$replace,$file)

引用文件

include ‘../ap\_win/word.php’;

|  |  |
| --- | --- |
| 参数 |  |
| $search | [必须]搜索内容，类型：数组。 |
| $replace | [必须]替换内容，类型：数组。 |
| $file | [必须]替换文件的标准路径 |

返回值 成功返回true；否则返回false。

说明 此方法会自动把$file中的$search中的数据替换成$replace中的数据。

示例

<?php

include ‘../ap\_win/word.php’;

$search=array(‘1’,’2’,’3’);

|  |
| --- |
| $replace=array(‘一','二’,’三’);  if(word::mht\_replace($search,$replace,’test.mht’))  echo ‘ok’; else echo ‘fail’;  ?> |

## 4.83.10word::mht\_replace2变量替换

定义

function mht\_replace2($search,$replace,$content)

引用文件

include ‘../ap\_win/word.php’;

|  |  |
| --- | --- |
| 参数 |  |
| $search | [必须]搜索内容（变量），类型：数组。 |
| $replace | [必须]替换内容，类型：数组。 |
| $content | [必须]将被替换的内容 |

返回值 返回替换完成的内容。

说明 此方法会自动把$content中的$search中的数据替换成$replace中的数据。

此方法与mht\_replace的区别是，变量需要放在{:}中，例如：{:test}。$search中只需要设置test项，而替换的$cotnent中需要写成{:test}样式。

使用{:}括起来，是为了保证变量的完整性。

示例

<?php

include ‘../ap\_win/word.php’;

$search=array(‘1’,’2’,’3’);

$replace=array(‘一','二’,’三’);

$content=file\_get\_contents('test.mht');

$ret=word::mht\_replace2($search,$replace,$content); echo $ret; ?>

## 4.83.11word::mht\_replace3变量精准替换

定义

function mht\_replace3($search,$replace,$content)

引用文件

include ‘../ap\_win/word.php’;

|  |  |
| --- | --- |
| 参数 |  |
| $params | [必须]替换内容，类型：键对数组。键保存查询内容，值保存替换内容。 |
| $content | [必须]将被替换的内容 |

返回值 返回替换完成的内容。

说明 之前使用的方法word::mht\_replace和word::mht\_replace2方法，采用了忽略mht文

件中无效标签的方式。但是这种方式会造成相近标签的混乱。所以，重新设计了识别方式，提高变量替换的精准度。 此方法会自动把$params中的键数据替换成$params中的值数据。

$params的键，可以带有{:}变量标识，也可以不带。本方法将自动识别。 使用{:}括起来，是为了保证变量的完整性。

示例

<?php

include ‘../ap\_win/word.php’;

$params=array(‘1’=>’一’,’2’=>’二’,’3’=>’三’);

$search=array(‘1’,’2’,’3’);

$replace=array(‘一','二’,’三’);

$content=file\_get\_contents('test.mht');

$ret=word::mht\_replace2($search,$replace,$content); echo $ret; ?>

## 4.83.12excel::xml2xls格式转换

定义

function xml2xls($file)

引用文件

include ‘../ap\_win/excel.php’;

参数

$file 标准文件路径。

返回值 返回类型：数组

|  |  |
| --- | --- |
| {      } | “error”:”错误信息”, //出错时，返回错误信息  “url”:”201608/15143210/123.xls” //转换后的文件标准路径。 |

如果不出错误，则不会返回error字段，可以使用判断此字段是否存在来识别是否正确返回。

说明 此方法会自动把Excel标准xml文件转换为xls文件。

注意：所谓标准文件路径，是本系统自动生成的文件路径，例如：

201608/15143210/123.xml。

示例

<?php include ‘../ap\_win/excel.php’;

$f=file::upfile('file'); $ret=excel::xml2xls($f); print\_r($ret); ?>

4.84E-Mail相关

## 4.84.1email::send发送邮件

依赖模块

sys\_email

调用文件

include ‘../sys\_email/class.php

定义

function send($args)

参数

$args.from [可选]指定发送人

|  |  |
| --- | --- |
| $args.to | [必须]接收人email地址。多个联系人时，使用半角分号隔开。 |
| $args.cc | [可选]抄送人email地址。多个联系人时，使用半角分号隔开。 |
| $args.subject | [必须]邮件标题 |
| $args.body | [必须]邮件内容 |

$args.attachment [可选]附件地址数组返回值

{

success:true|false, //是否发送成功

error:’错误信息’, //如果发送失败，则存在此字段 eml:’……’ //发送的邮件完整内容

}

示例：

<?php

include '../sys\_email/class.php';

$ret=email::send(array(

'to'=>'0123@163.com',

'subject'=>'PHP Mail Test',

'body'=>'PHP Mail Body

测试汉字内容

<b>HTML粗体</b>'

|  |
| --- |
| ));  print\_r($ret);  ?> |

示例输出内容

Array

(

[success] => 1

[eml] => Date: Tue, 7 Jun 2016 12:19:35 +0800

To: 0123@163.com

From: sm2@cnkip.com

Subject: PHP Mail Test

Message-ID: <0ba00d116a61e739fb33eaf8391d1018@192.168.6.254>

X-Mailer: cnkip\_oa

MIME-Version: 1.0

Content-Type: text/html; charset=utf-8

Content-Transfer-Encoding: 8bit

PHP Mail Body

测试汉字内容

HTML粗体

)

4.85Aliyun相关（阿里云）

## 4.85.1aliyun::wsls写日志

依赖模块

aliyun\_sls

定义

function wsls($topic,$json)

参数

$topic 日志主题

$json 日志内容。如果内容为字符串，则自动转换为array(‘content’=>$json)

返回值

无

说明使用此方法，可以把日志内容提交到sls缓冲池，由aliyun\_sls模块统一提交到Aliyun SLS服务器上。如果服务器未安装aliyun\_sls模块，则此方法调用等同于wlog调用(会自动进行array 转换)。

字符串日志示例

<?php

aliyun::wsls(‘测试’,’日志内容’); ?>

JSON日志示例

<?php

aliyun::wsls(‘测试’, array(‘请求地址’=>$\_SERVER[‘REQUEST\_URI’],’content’=>’请求内容’)

);

?>

## 4.85.2aliyun::oss\_upload向OSS上传文件

依赖模块

aliyun\_oss

定义

function oss\_upload($object,$content)

参数

$object 上传的路径名称和文件名称

$content 上传内容

返回值

无

说明使用此方法，可以把指定内容上传到OSS服务器的指定路径。

示例

<?php

aliyun::oss\_upload(‘test/test.txt’,’日志内容’); ?>

## 4.85.3aliyun::oss\_download从OSS上下载文件

依赖模块

aliyun\_oss

定义

function oss\_download($object)

参数

$object OSS服务器上的文件路径

返回值 文件内容。

说明使用此方法，可以读回OSS服务器上指定文件内容。

示例

<?php

$file=aliyun::oss\_download(‘test/test.txt’);

?>

## 4.85.4aliyun::oss\_exists判断OSS上是否存在指定文件

依赖模块

aliyun\_oss

定义

function oss\_exists($object)

参数

$object OSS服务器上的文件路径

返回值存在返回true，否则返回false。

说明使用此方法，可以读回OSS服务器上指定文件的属性。

示例

<?php

$exists=aliyun::oss\_exists(‘test/test.txt’); if($exists) echo ‘文件存在’;

?>

# 5Javascript公共函数

调用公共Javascript函数，需要引用inc.js文件，例如：

<script src=”../inc.js”></script>

5.1$ajax AJAX访问

定义

function $ajax(args)

参数

|  |  |
| --- | --- |
| args.url:”……” | [必须]访问路径。 |
| args.post:”POST内容” | [可选]POST内容 |
| args.dataType:”json” | [可选]返回JSON格式 |

args.success:function(data){……}

[可选]访问成功时执行，data为服务器返回的字符串数据。

args.error:function(ex){……} [可选]访问失败时执行，ex为错误信息。说明 这个函数用于向指定路径发送GET或POST请求。当设置args.post参数后，函数自动把请求方式设置为POST。

示例仅访问，不考虑执行状态

$ajax({url:"tick.php?action=tick"});

使用返回的数据

$ajax({url:”tick.php”,success:function(data){document.write(data);}});

执行结果

{“status”:”ok”}

出错时进行处理

|  |
| --- |
| $ajax({ url:”tick.php”,  success:function(data){document.write(data)}, error:function(ex){document.write(ex)}  }); |

执行结果（如果页面不存在，则返回404）

404

返回json数据的使用

|  |
| --- |
| $ajax({  url:“role.php?type=dd”, dataType:’json’,  success:function(data){ //data is JSON Object }  ); |

向指定URL发送POST信息

|  |
| --- |
| $ajax( {  url:”test.php”, post:”abcdefg”  }); |

5.2$af\_classList 让节点支持classList操作

定义

function $af\_classList(el)

参数 el 需要支持的节点或节点ID

说明

IE9以及更老的版本，不支持节点classList操作。使用本函数后指定的节点就可以支持classList的相关操作。

示例

$af\_classList(document.body); document.body.classList.add("c"); document.body.classList.remove("c"); document.body.classList.toggle("c"); document.body.classList.contains("c");

节点支持示例

|  |
| --- |
| <html>  <body>  <div id=”test”></div>  <script>  //使用id设置  $af\_classList(‘test’);  document.getElementById(‘test’).classList.add(‘c’);    //使用节点设置  $af\_classList(document.getElementById(‘test’)); document.getElementById(‘test’).classList.add(‘c’);  </script>  </body>  </html> |

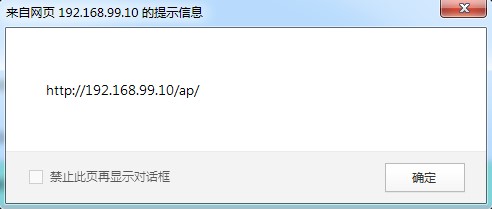
5.3$ap\_home 平台根路径

此变量保存当前平台的根路径。

示例：

alert($ap\_home);

示例结果：



5.4$confirm 确认窗口

定义

function $confirm(args)

参数 args.text 显示文本 args.btn1 按钮1文本，默认文本：确认 args.btn1click 按钮1响应，默认处理，关闭当前窗口。

args.btn1click=function(obj){……} obj为调用$confirm节点

args.btn2 按钮2文本，默认文本：取消 args.btn2click 按钮2响应，默认处理：关闭当前窗口。

args.btn2click=function(obj){……} obj为调用$confirm节点

返回值

无

说明

示例

5.5$contextmenu上下文菜单

定义

function $contextmenu(args)

参数

{

args.id=’’, //[可选]上下文菜单标识。如果不设置，则生成一个随机ID。

args.className:’’, //[可选]上下文菜单使用的样式类。默认值：contextmenu。 args.parent:object, //[可选]上下文菜单的上层节点。默认值：document.body。

args.data:object, //[可选]上下文菜单数据。

//如果使用show(…)显示菜单时，会覆盖此变量内容。

args.onclose=function(){…}, //[可选]上下文菜单关闭时调用 args.submenu: //[必须]菜单内容

[

{

“id”:””, //菜单标识

“icon”:””, //图标

“html”:””, //显示内容

“onclick”:function(){…}, //[可选]点击事件

‘data’:object, //[可选]用户自定义数据

‘submenu’:[…] //[可选]子菜单

},

……

]

}

返回值 上下文菜单对象。

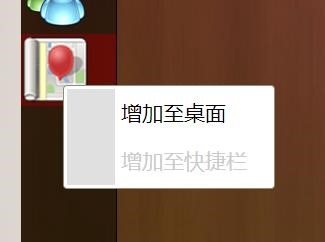
说明

本方法，在HTML页面内显示上下文菜单。

示例

|  |
| --- |
| $contextmenu({'submenu':  [  {  'id':'desktop\_add',  'html':'增加至桌面',  'onclick':function(){……}  },  {  'id':'toolbar\_add',  'html':'增加至快捷栏',  'onclick':function(){……}  }  ]}); |

显示效果



5.6$ddupfile 拖拽上传[停用]

此方法不再推荐使用,请使用$ddupfile2替代

定义

function $ddupfile(args)

参数 args.name =file[] //POST字段名称 args.url =window.location //POST路径

args.success //上传完成后的处理函数

返回值

无

说明

示例

5.7$ddupfile2文件拖拽上传

定义

function $ddupfile2(args)

参数 args.el 操作节点 args.el\_dd [可选]拖动操作节点，默认为$args['el'] args.name 上传文件的文件域名称 args.url 接收文件处理的路径 args.success 上传完成后的回调。

args.dataType 上传完成后的回调参数类型。

如果是Json类型则需要设置dataType=‘json’

args.text 显示的提示文本。默认值为“文件拖拽到这里，即可直接上传。” args.input [可选]是否显示浏览式文件上传。默认：显示 args.weixin [可选]是否支持微信上传，默认：false。 args.count [可选]限制上传文件数量限制，默认：无限制 args.files [可选]已上传文件列表。如果设置，则自动显示已上传文件的列表项。

返回值

无

说明 通过新浏览器的文件拖拽功能实现文件上传。需要注意的是，在早期、老版本浏览器上是不支持此功能的。

示例

<div id="test1" style="position: absolute; left: 10px; top: 380px; width: 200px; height: 200px; background-color: #ccccff"></div>

<div id="test2" style="position: absolute; left: 220px; top: 380px; width: 200px; height:

200px; background-color: #ccffcc"></div>

|  |
| --- |
| <div id="test3" style="position: absolute; left: 430px; top: 380px; width: 200px; height: 200px; background-color: #ffcccc"></div>  <div id="test4" style="position: absolute; left: 640px; top: 380px; width: 200px; height:  200px; background-color: #ffccff"></div>    <table width="600" border="1">  <tr>  <td height="100">单元格11</td>  <td height="100">单元格12</td> <td height="100">单元格13</td>  </tr>  <tr>  <td height="100">单元格21</td>  <td height="100"><div style="position:relative; width:100%; height:100%" id="test5"> 单元格22</div></td>  <td height="100">单元格23</td>  </tr>  <tr>  <td height="100">单元格31</td>  <td height="100">单元格32</td> <td height="100">单元格33</td>  </tr>  </table>    <script>  $ddupfile2({'el':document.getElementById("test1"),'success':function(data){alert(data)}, 'text':'提示文本1'  });  $ddupfile2({'el':document.getElementById("test2"),'success':function(data){alert(data)}, 'text':'提示文本2'  });  $ddupfile2({'el':document.getElementById("test3"),'success':function(data){alert(data)}, 'text':'提示文本3'  });  $ddupfile2({'el':document.getElementById("test4"),'success':function(data){alert(data)},  'text':'提示文本4s'  });  $ddupfile2({'el':"test5",'el\_dd':'aa','url':'../../upfile.php',  'text':'提示文本4s',  'input':true,  'name':'name',  ‘weixin’:true,  "files":['201605/12121212/123.txt','201605/12121212/3333.doc'] }); |

</script>

显示效果：

 其中，单元格22和下面四个色块是可以直接拖拽的。

5.8$el 创建节点

定义

function $el(args)

参数以下参数都是可选的，根据需要填写。

|  |
| --- |
| args  {  parent:document.body, //上层节点 tagName:’div’, //节点名称。默认：div。 id:’’, //节点ID  backgroundImage:””, //背景图片路径 className:””, //节点CSS类名称 cursor:””, //CSS中cursor属性 data:……, //附加数据  display:””, //el.style.display html:””, //el.innerHTML left:””, //el.style.left width:””, //el.style.width src:””, //el.src  title:””, //el.title zIndex:””, //el.style.zIndex; onclick:function(){}, //onclick事件，鼠标点击时触发。 onmousedown:function(){}, //onmousedown事件，鼠标按下时触发。 onmousemove:function(){}, //onmousemove事件，鼠标移动时触发。 onmouseup:function(){}, //onmouseup事件，鼠标按起时触发。 onmouseout:function(){}, //onmouseout事件，鼠标移入时触发。  } |

返回值 返回创建的节点或选择的节点。

说明 这个函数实际上是简化了document.createElement()命令。示例

$el({parent:document.body,className:”stext”,html:”test”});

输出结果

<html>

<head>……</head>

<body>

<div class=”stext”>test</div>

</body>

</html>

5.9$get读取URL中的参数

定义

function $get(name)

参数 name [必须]参数名称

返回值 如果参数存在，则返回参数内容；如果参数不存在则返回空。

说明

无

示例

假设当前URL= http://192.168.6.254/ap/?mid=123&name=456

<script> var n=$get(“mid”); alert(n);

</script>

运行后，显示“123”。

5.10$getElementsByClassName 模 拟

getElementsByClassName操作

定义

function $getElementsByClassName(name,node)

参数 name [必须]样式名称 node [可选]操作节点

返回值

节点列表，与document.getElementsByClassName返回相同。

说明

IE6不支持getElementsByClassName，此函数可以模拟支持。

示例

var els=$getElementsByClassName(‘test’);

5.11$jtree 树形结构显示

定义

function $jtree(args)

参数

args

{

el:'', //插入的节点 dataurl:'', //网络数据读取的路径 url:'', //默认链接路径,如果节点数据存在a，则忽略此参数。

自动会在url后面加i

def:'', //默认数据项,这个是与i比较得出的

onclick:function(){}, //点击事件 onmousedown:function(){},//菜单事件 data:

[ {

i:'', //节点id t:'', //显示的文本 g:'', //图标 a:'' //节点URL，这个优先于URL

},

......

]

}

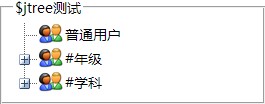
data 当前处理的数据,如果不传递则自动载入args.data pre 显示的前导信息

返回值 生成的jtree对象。说明

按照指定格式生成树形结构示例

|  |
| --- |
| <script src=”../inc.js”></script>  <div id=”jtree”></div>  <script>  $jtree(  {  "el":"jtree",  "target":"edit",  "url":"edit.php?mid=123&id=", "data":  [  {"i":"8","t":"普通用户","g":"../sys-role/\_\_icon.png"},  {"i":"145","t":"#年级","g":"../sys-role/\_\_icon.png",  "c":  [  {"i":"147","t":"初一(2014级)","g":"../sys-role/\_\_icon.png"},  {"i":"146","t":"初二(2013级)","g":"../sys-role/\_\_icon.png",  "c":  [  {"i":"148","t":"1班","g":"../sys-role/\_\_icon.png"},  {"i":"149","t":"2班","g":"../sys-role/\_\_icon.png"}  ]  },  {"i":"153","t":"初三(2012级)","g":"../sys-role/\_\_icon.png"}  ]  },  {"i":"150","t":"#学科","g":"../sys-role/\_\_icon.png",  "c":  [  {"i":"151","t":"语文","g":"../sys-role/\_\_icon.png"},  {"i":"152","t":"数学","g":"../sys-role/\_\_icon.png"}  ]  }  ]  });  </script> |

运行效果：



5.12$jctree树形选择框

定义

function $jctree(args)

参数

args

{

el:'' //插入的节点名称，如果不设置则返回jctree对象 el\_id:’节点id’, //当前控制节点id el\_replace:false, //是否直接使用指定的节点，而不是插入。

//设置此参数，则必须设置el

el\_insert\_before:false, //插入到节点内最前面，默认是插入到最后

//此参数与el\_replace是互斥的 name:'' //自动生成的input节点名称 mode:1|2 //1单选2多选,默认:1单选

selectleaf:true //只能选择叶子 selected: //默认选项

[

{ i:”标识编号”, t:”显示文本”

},

……

],

width:”160px”, //设置选择区域宽度 popup\_width:”160px”, //设置弹出区域宽度，如果未设置，则与width相同。

dataurl:'' //数据读取地址(异步读取，参数与data互斥) data:[...] //参见jtree.data属性

}

返回

生成的jctree节点对象。

说明(el\_replace)

el\_replace属性，关系到生成的节点层次，例如：

<div id="jctree1"></div>

<script>

$jctree(

{

"el":"jctree1",

"name":"jctree1\_id",

"dataurl":"../role.php?type=du&enabled=1" //设置数据读取位置

});

</script>

生成的HTML代码：

|  |
| --- |
| <div id="jctree1">  <div class="jctree">  <span class="jctree\_inputs">&nbsp;</span>  <div class="jctree\_popup" style="z-index: 2;">  <div class="jtree">……</div>  </div>  </div>  </div> |

当设置el\_replace=true时，例如代码：

<div id="jctree2"></div>

<script>

$jctree(

{

"el":"jctree2",

"el\_replace":true,

"name":"jctree1\_id",

"dataurl":"../role.php?type=du&enabled=1" //设置数据读取位置

});

</script>

生成的HTML代码

<div id="jctree2" class="jctree">

<span class="jctree\_inputs">&nbsp;</span>

<div class="jctree\_popup" style="display: none;">

<div class="jtree">……</div>

</div>

</div>

也就是说，如果没有特殊需求时，使用el\_replace=true会使$jctree生成的HTML代码更精炼。所以，建议使用el\_replace=true属性。

注 1：el\_replace 参数不默认为 true，是为了照顾老版本的模块的兼容性，防止老版本模块出现未知错误。

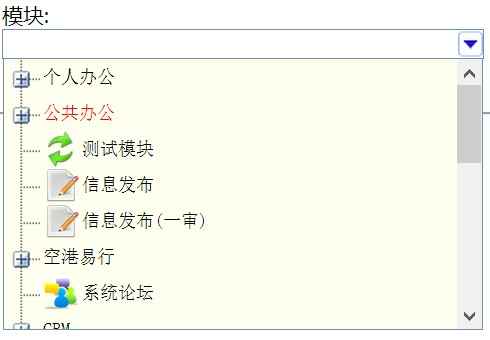
注2：此参数与el\_insert\_before是互斥的，因为使用el\_replace后，不存在插入位置的问题

示例（模块读取）

动态从数据文件中读取显示数据：

|  |
| --- |
| <label>模块:</label>  <div id="m\_id\_zone"></div> <script>  $jctree(  {  'el':'m\_id\_zone',  'name':'m\_id',  'mode':1,  'selectleaf':true,  'dataurl':'m.php?mid='+$get('mid') });  <script> |

显示示例:



示例(用户选择)

<div id="jctree">常用选项:

<a

href="javascript:document.getElementById('jctree\_id\_id').jtree\_selected([{'i':'123','t':' 选项1'},])">选项1</a>

<a

href="javascript:document.getElementById('jctree\_id\_id').jtree\_selected([{'i':'234','t':' 选项2'}])">选项2</a>

</div>

<script>

$jctree(

{

"el":"jctree",

"el\_id":"jctree\_id\_id",

"el\_insert\_before":true,

"name":"jctree\_id",

"mode":1,

"selected":[{"i":"9","t":"管理员"}], //设置默认值

"selectleaf":true,

"width":"200px",

"dataurl":"../role.php?type=du&enabled=1" //设置数据读取位置

});

</script>

显示样式



5.13 $msgbox 显示消息窗口（模式一）

模式一只是为了兼容之前的代码，请使用模式二替代。

定义

function $msgbox(msg，func)

参数 msg 显示的内容 func 当点击时指定的函数返回值

无

说明 显示一个模式提示框。

示例

<a href="javascript:$msgbox('test test !!');">$msgbox</a>

运行结果



5.14 $msgbox 显示消息窗口（模式二）

定义

function $msgbox(args)

参数 args.msg [必须]显示的HTML内容 args.clickclose [可选]是否点击自动关闭 args.onclose [可选]当点击关闭时运行的函数返回值

$msgbox对象。

说明 显示一个全屏模式提示框。

返回值，支持close()和show(html)操作。

示例

<a href="javascript:$msgbox({‘msg’:'test test !!'});">$msgbox</a>

运行结果



5.15 $topmsg 显示提示信息

定义

function $topmsg(txt,flag,timeout)

参数

|  |  |
| --- | --- |
| txt | [必填]显示文本，可以为HTML格式内容。 |
| flag | [可选]是否着重显示，即：红色、加粗显示。 |

timeout [可选]显示时长，单位：毫秒，默认值：5000

返回值

无

说明 在界面右上角显示提示信息。

示例

<a href="javascript:$topmsg('显示内容');">topmsg显示内容</a>

<a href="javascript:$topmsg('重要信息',true);">topmsg显示重要信息</a>

<a href="javascript:$topmsg('信息显示10秒',false,10000);">topmsg显示重要信息</a>



5.16 $table2

备注：此部分需要与PHP部分中的table2配合使用。

示例页面

<form action="data.php" name="form1" id="table2">

//action属性标识读取数据的文件

<div id=”navigation1”></div>

<input type=”hidden” name=”mid” id=”mid” value=”<?php echo $\_GET[‘mid’]; ?>”>

<input type=”text” name=”args1” id=”args1”> //附加参数

<table>

<thead>

<tr>

<th>No</th>

<th sort=”id” class=”sort2”>ID</th>

//class标识默认排序，sort1正序,sort2反序

<th sort=”name”>Name</th> //sort标识列排序，sort值为列名

<th>OP</th>

</tr>

</thead>

<tbody>

<tr id=”table2datarow”> //id标识数据行

<td>{:\_\_rowindex}</td> //行号

<td>{:id}</td> //数据字段名称

<td>{:name}</td>

<td><a href=”show.php?mid=<?php echo $\_GET[‘mid’]; ?>&id={:id}</td>

</tr>

</tbody>

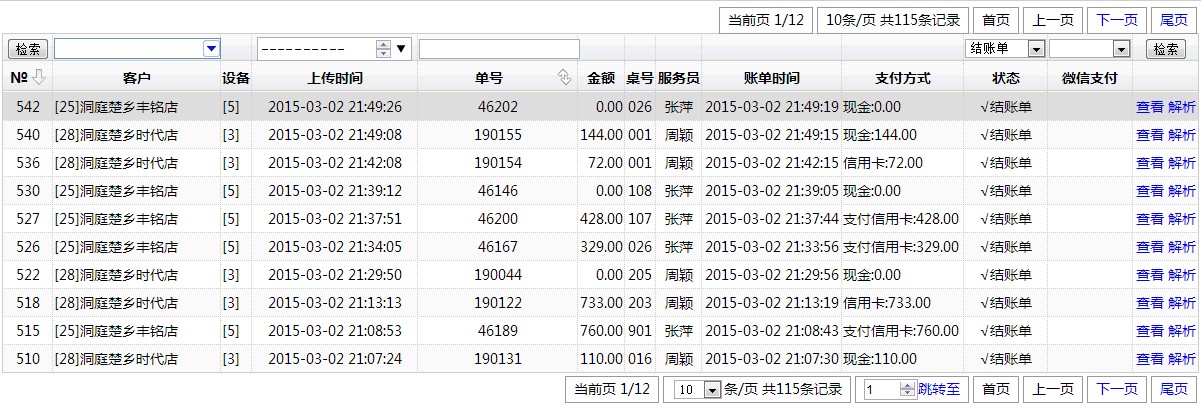
</table>

<div id="navigation2"></div> //导航拦

</form>

<script>$table2("table2");</script>

显示示例：



5.17$topmost 节点zIndex置最高

定义

function $topmost(obj)

参数 obj 节点返回值

无

示例

$topmost(el);

5.18 $window 创建窗口

定义

function $window(args)

参数

args

{

id:'', //窗口名称，如果没有提供，则自动生成一个

className:'window', //使用的窗口类 parent:document.body, //上层节点 btn\_refresh:true|false, //是否显示"刷新"按钮 btn\_min:true|false, //是否显示“最小化”按钮 btn\_max:true|false, //是否显示“最大化”按钮 center:true|false, //窗口居中显示 modal:true|false, //是否为模式窗口，是的话为true,默认false url:'...', //显示的页面 html:HtmlElement, //窗口内显示的html内容，这个不能与url同时使用。因为url 使用iframe,html使用div。 title:'...', //窗口标题 width:800, //窗口默认宽度 height:480, //窗口默认高度 onclose:function(){..}, //当onclose事件发生时触发 data:object //用户自定义数据

}

返回值

无

说明

在窗口内，调用 window.frameElement.close(); 来关闭窗口。

示例简单的模式窗口

$window(

{

"modal":true,

"title":"用户增加",

"width":300,

"height":280,

"url":"uedit.php" });

5.19 $tselect文本选择框

引用文件

<script src=”../Scripts/tselect.js”>

<link rel="stylesheet" type="text/css" href="../Scripts/tselect.css">

定义

function $tselect(args)

参数

args

{

'el\_select':el\_select, //[必须]选择显示节点

'el\_value':el\_value, //[可选]保存ID的字段

'selected:true|false, //[可选]是否完全符合选择。默认值：false

'url':'tselect.php', //[可选]数据读取程序，与data\_url必须有一个设置。

“data\_url”:”data.php?key=”, //[可选]动态数据读取程序，与url必须有一个设置

'onload':function(){...}, //[可选]装载数据后，

'onselected':function(){...}, //[可选]选中后的事件

|  |  |
| --- | --- |
| “ondelete”:function(){…} | //[可选]点删除按钮的事件 |
| “notfound”:”” | //[可选]找不到数据时的提示文本。 |

如果不设置则显示默认提示文本。

}

返回值

无。

说明 当使用data\_url参数时，此函数会在用户输入内容时，自动去data\_url指定的地址去读取信息。并且，会把当前输入的内容附加到data\_url后面。例如：

<script>

$tselect(

{

"el\_select":"textfield",

"el\_value":"textfield1",

"url":"tselect\_data.php",

"data\_url":"tselect\_data.php?key="

});

</script>

则当输入关键字：“A”时，会自动去 tselect\_data.php?key=A这个地址去读取信息。

data\_url参数文件返回数据格式

|  |  |
| --- | --- |
| [        ] | {id:1,code:”aaa”,name:”AAAAA”}, {id:2,code:”bbb”,name:”BBBBB”},  …… |

注:name是必须的，id和code为可选数据。

参数notfound示例代码

<form id="form1" name="form1" method="post">

<label for="tselect1">Text Field:</label>

<input type="text" name="tselect1" id="tselect1">

<input type="text" name="tselect1a" id="tselect1a">

</form>

<script>

$tselect(

{

"el\_select":"tselect1",

"el\_value":"tselect1a",

"url":"tselect\_data.php?key=a",

'notfound':'<font color=“#ff0000">无指定信息</font>'

});

</script>

显示效果

如果想不显示提示信息

，

则

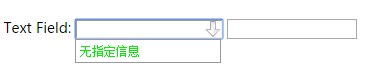
not

found=

’’

即可

。



~~5.20 $email邮件编辑（停用）~~

引用文件

<script src=”../sys\_email/email.js”>

定义

function $email(args)

参数

args

{

“parent”:””, //[必须]上层节点名称

“name”:”email”. //字段名称前缀

“to”:[{“title”:”类”,“address”:”012@16.com”,“name”:”张三”},…], //收件人列表

“cc”:[{“title”:”类”,“address”:”123@16.com”,”name”:”李四”},…], //抄送人列表

“contact”:[{“address”:”2345@163.com”,”name”:”王五”},…], //联系人列表

“subject”:””, //邮件标题

“body”:””, //邮件内容

“attachment”:[“1.png”,……], //邮件附件列表

“attachment\_checked”:[“1.png”,……], //选中的附件列表

“template”:[{“name”:”模板名称”,”subject”:””,”body”:””}] //邮件模板列表

“template\_selected”:0 //使用的模板（如果选择了，则忽略subject和body参数）

}

参数中，未写必须的参数，请根据需要选填。

返回值

无。

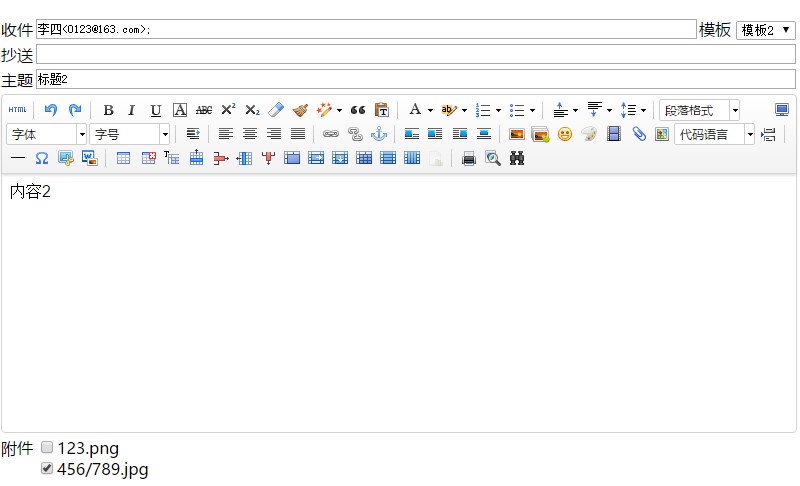
说明 此方法将返回完整的Email编辑界面，在POST时，各字段名称分别为： name\_to //收件人 name\_cc //抄送人

|  |  |  |
| --- | --- | --- |
| name\_subject |  | //主题 |
| name\_body |  | //内容 |
| name\_attachment[] |  | //附件列表 |

使用示例

|  |
| --- |
| <html>  <head>  <title>Email发送测试</title>  <script src="../inc.js"></script>  <script src="email.js"></script>  </head>  <body>  <div id="email"></div>  <script>  $email(  {  'parent':'email',  'to':[{'address':'0123@163.com','name':'李四'}],  'contact':[{'address':'sm2@cnkip.com','name':'单闽'},  {'address':'0123@163.com','name':'张三'}],  'name':'posttest',  'attachment':['123.png','456/789.jpg'], //显示的附件  'attachment\_checked':['456/789.jpg'],//默认选中的附件  'template':[{'name':'模板1','subject':'标题1','body':'内容1'},  {'name':'模板2','subject':'标题2','body':'内容2'}],  'template\_selected':1  });  </script>  </body> |

显示样式



5.21$email2邮件编辑器

引用文件

<script src=”../sys\_email/email2.js”>

定义

function $email2(args)

参数

args

{

“parent”:””, //[必须]上层节点名称

“from”:[“a@a.com”,”b@b.com”], //[可选]邮件发送邮箱名称列表。

“name”:”email”. //[必须]字段名称前缀

“to”:[{“title”:”类”,“address”:”012@16.com”,“name”:”张三”},…], //收件人列表

“cc”:[{“title”:”类”,“address”:”123@16.com”,”name”:”李四”},…], //抄送人列表

“contact”:[{“address”:”2345@163.com”,”name”:”王五”},…], //联系人列表

“subject”:””, //邮件标题

“body”:””, //邮件内容

“attachment”:[“1.png”,……], //邮件附件列表

“attachment\_checked”:[“1.png”,……], //选中的附件列表

“template”:[{“name”:”模板名称”,”subject”:””,”body”:””}] //邮件模板列表

“template\_selected”:0 //使用的模板（如果选择了，则忽略subject和body参数）

}

参数中，未写必须的参数，请根据需要选填。

返回值

无。

说明 此方法将返回完整的Email编辑界面，在POST时，各字段名称分别为：

{name}\_from

{name}\_to //收件人

{name}\_cc //抄送人

{name}\_subject //主题

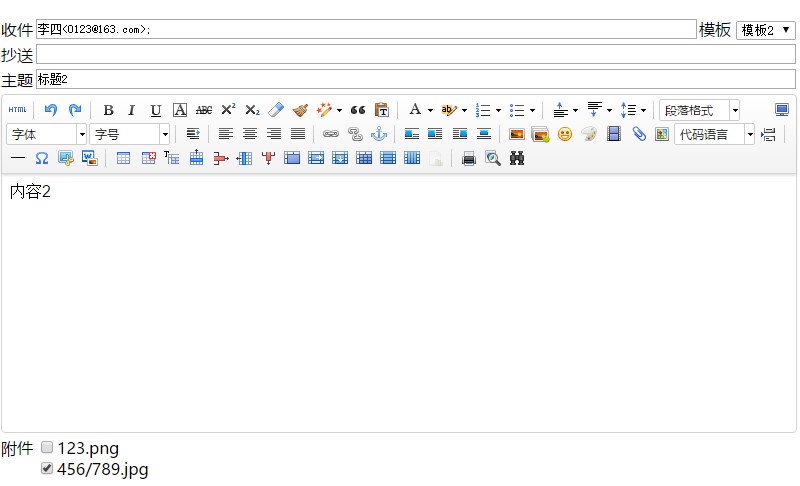
{name}\_body //内容

{name}\_attachment[] //附件列表

使用示例

|  |
| --- |
| <html>  <head>  <title>Email发送测试</title>  <script src="../inc.js"></script>  <script src="email.js"></script>  </head>  <body>  <div id="email"></div>  <script>  $email(  {  'parent':'email',  'to':[{'address':'0123@163.com','name':'李四'}],  'contact':[{'address':'sm2@cnkip.com','name':'单闽'},  {'address':'0123@163.com','name':'张三'}],  'name':'posttest',  'attachment':['123.png','456/789.jpg'], //显示的附件  'attachment\_checked':['456/789.jpg'],//默认选中的附件  'template':[{'name':'模板1','subject':'标题1','body':'内容1'},  {'name':'模板2','subject':'标题2','body':'内容2'}],  'template\_selected':1  });  </script>  </body> |

显示样式



# 6第三方组件

6.1PHPExcel 1.8.0

关联模块: third\_phpexcel180

引用路径:

include\_once ‘../third\_phpexcel180/PHPExcel.php’;

或

include dirname(dirname($\_SERVER['SCRIPT\_FILENAME'])).'/ third\_phpexcel180/PHPExcel.php';

调用示例

|  |
| --- |
| <?php  include dirname(dirname($\_SERVER['SCRIPT\_FILENAME'])).'/inc.php'; //路径请根据具体情况设置 include dirname(dirname($\_SERVER['SCRIPT\_FILENAME'])).'/ third\_phpexcel180/PHPExcel.php';  //创建对象  $excel= new PHPExcel();  $excel->getActiveSheet()->freezePane("A2");//冻结表头行  $excel->getActiveSheet() //设置表头显示  ->setCellValue('A1', 'ID')  ->setCellValue('B1', '名称') ->setCellValue('C1', '数量')  //填充数据  $rcount=2;  $dr=$db->query(‘select \* from t\_test’); while($row=$dr->fetch(PDO::FETCH\_ASSOC)  {  $excel->getActiveSheet()  ->setCellValue(“A{$j}”,$row[‘id’])  ->setCellValue(“B{$j}”,$row[‘name’])  ->setCellValue(“C{$j}”,$row[‘num’])  $rcount++;  }  $excel->getActiveSheet()->getColumnDimension("A")->setAutoSize(true); //设置指定列自动宽度  $excel->getActiveSheet()->setTitle('新建页'); //设置页名称  //清除之前的缓冲区，发送文件信息  ob\_end\_clean();//清除缓冲区,避免乱码  header('Content-Type: application/vnd.ms-excel;charset=utf-8'); header("Content-Disposition: attachment;filename='excel.xls'"); header('Cache-Control: max-age=0');  //输出生成的文件内容 |

$objWriter = PHPExcel\_IOFactory::createWriter($excel, 'Excel5');

$objWriter->save('php://output');

?>

# 7CSS公共类

7.1.info 提示信息区

示例

<div class="info">.info CSS类示例</div>

显示样式



7.2.warning 警告信息区

示例

<div class="warning">.warning CSS类示例</div>

显示样式



# 8权限模型的说明

1.1模块的形式

本应用平台使用的权限控制为模块控制，任何模块都会分一个唯一的 mid。在模块内，

需要使用acl(…)来判断当前用户在当前模块下具有的权限。

例如：/pub-info/?mid=11

这里指示这个模块的编号是11。在程序里面使用acl(…)判断其具有的权限。

如：

acl('管理') or die; //判断权限,并且如果没有权限时停止执行。

//具体参见Acl函数的说明

注：这里所指的权限是在“模块管理”里面设置的权限。程序里面的权限名称一定要与设置的权限名称一致。

另外一个注意的地方是，绝大部分模块需要根据分给的 mid 的不同对数据进行不同处理，例如：

/pub-info/?mid=11 //为一中的通知

/pub-info/?mid=12 //为二中的通知

/pub-info/?mid=13 //为三中的通知

虽然pub-info中程序是相同的，但会根据mid的不同显示和管理的内容不同。另外，在“模块管理”中，对模块的权限需要单独添加。

1.2页面中权限的判断

页面中的权限判断使用acl[[5]](#footnote-5)进行判断。如无特殊需求，建议每页都进行权限判断，最大化的保证系统的安全性。

如果一个权限在页面中判断多次，建议使用变量保存权限，例如：

$priv\_admin=acl('管理');

……

if($priv\_admin) ……

……

if($priv\_admin) ……

……

# 9TABLE作为Excel文件保存时的说明

由于单元格处理不同，有些单元格需要进行独立的标识才可以。下面说明的附加样式，以style方式加到单元格内。例如：

<td style='mso-number-format:"yyyy-mm-dd hh:mm:ss"'>2014-12-12 12:34:56</td>

日期字段附加样式：

mso-number-format:"yyyy-mm-dd hh:mm:ss"

文本样式：

mso-number-format:"\@"

# 10固定文件格式

10.1 \_\_sysconfig.php

这个文件是平台软件主配置文件，主要记录数据连接等信息。

|  |  |  |
| --- | --- | --- |
| <?php  $AP\_TITLE='Application Platform (应用平台)';  $AP\_DB\_DSN='mysql:host=127.0.0.1;dbname=ap'; |  | 数据库连接字符串 |
| $AP\_DB\_UID='root'; |  | 数据库用户名 |
| $AP\_DB\_PWD=''; |  | 数据库密码 |
| ~~$\_DOCUMENT[‘DROOT’]=\_\_DIR\_\_;~~ |  | ~~网站地址~~ |
| ~~$\_DOCUMENT[‘VROOT’]='';~~  ?> |  | ~~网站虚拟目录~~ |

注释:最新版本已经不再支持$\_DOCUMENT[‘DROOT’]和$\_DOCUMENT[‘VROOT’]

10.2 \_\_icon.png

这个图片文件作为默认图标使用。通常图片分辨率为48px\*48px或96px\*96px，为适应高清平板的分辨率，建议使用96px\*96px或更高分辨率的图标。

10.3 \_\_module.php

一般情况下，每个模块都在一个单独目录中存在。模块的命名规则为：

模块类型-模块功能

每个模块都必须拥有一个\_\_module.php文件，这个文件主要保存每个模块的基本信息。

|  |
| --- |
| <?php  $\_\_module=array(  ‘code’=>’test-test’, [必须]模块标识  ‘name’=>’测试模块’, [必须]模块名称  ‘refcount’=>’1’, [必须]模块可被引用次数 ‘privs’=>array(‘查看’,’管理’), 模块权限列表  ‘privs\_opt’=>array(), 可选权限列表  ‘privs\_info’=>’’, 权限说明  ‘desc’=>’本模块仅为测试使用’, 模块说明  ‘ignore’=>array(‘test.js’,’test.css’), （可选）打包时忽略的文件列表  ‘through’=>array(‘scripts’), （可选）不进行加密的模块 ‘require’=>array( （可选）模块依赖。 array(‘rcode’=>’test-test1’,’rrev’=>123) 依赖的模块标识和版本。  ),  ‘rev’=>’123’ 模块版本号。这个由打包程序自动填写。  );  ?> |

关于require中rcode和rrev参数的说明：

使用这个参数名称，是为了与模块的code、rev参数不产生冲突。在使用打包工具时，

有可能两个参数产生歧义，为保证数据的完整性，所以这里的参数前面增加了一个r。

10.4 \_\_startup.js

本文件是模块中的一个自启动文件，脚本将在进入主界面后由系统自动调用。

需要注意的是，这个脚本文件会影响主界面脚本的执行，如果需要在window.onload事件中执行，则需要对原有的window.onload事件做处理，下面是一个windows.onload处理的示例，供参考使用：

|  |  |  |  |
| --- | --- | --- | --- |
| // JavaScript Document  (function()  {  var \_\_onload=window.onload; | | | //暂存onload方法 |
|  | window.onload=function() | | //重新定义onload方法 |
|  | { | \_\_onload();    //下面是自己的代码  $topmsg('\_\_startup.js'); | //模拟执行暂存的onload方法 |
| }  }()); | | |  |

如果不需使用window.onload事件，可以简单的写成下面的方式：

// JavaScript Document

(function()

{

//下面是自己的代码

$topmsg('\_\_startup.js');

}());

注：强烈建议使用(function(){}())闭包方式把这部分代码放入独立的作用域。这样可

以有效的防止与主代码的冲突。

10.5 \_\_tick.php

此文件，用户浏览器会每隔一段时间（按照 tick 刷新时间，一般 10 秒以内）调用一次，那些需要周期处理的数据可以放在这个文件中。

此文件中可使用的公共变量：

$data[‘msg’] 信息提醒数据，注意传值前判断是否已经存在，防止其他模块

数据被覆盖。

$data[‘msg’][‘u’] 点击信息时的URL

$data[‘msg’][‘t’] 显示文本

$data[‘badge’][mid] 设置模块badge显示内容。

mid为badge所属的模块编号。

10.6 \_\_update.php 模块升级文件

当安装或升级模块时，系统都将自动执行本文件。

本文件的内容应该是在安装、升级时对数据库、数据文件的操作。例如数据库表的生成

等。 系统包含文件中包含一些已经封装好的数据库操作，下面的章节将会详细讲解各函数的用法。

\_\_update.php文件示例

|  |  |  |
| --- | --- | --- |
| <?php  if(update::table\_exists(‘t\_test’)===false)  {  update::table\_create(‘t\_test’, array(  array(‘col’=>id’ ,type’=>’uint’ | | ,’nn’=>true,’ai’=>true,’pk’=>true), |
| array(‘col’=>name’ | ’type’=>’varchar’,’len’=>45 | ,’nn’=>true), |
| array(‘col’=>’oc’  ));  }  ?> | ’type’=>’uint’ | ,’nn’=true) |

## 10.6.1数据库列名关键字

key SqlServer关键字 percent SqlServer关键字

## 10.6.2update::column\_add 增加列

定义

|  |
| --- |
| function column\_add($table\_name,  $column\_name,$column\_type,$column\_len=0,$column\_nn=false,$column\_after='',  $def=NULL) |

参数

$table\_name 表名

$column\_name 列名

|  |  |
| --- | --- |
| $column\_type | 数据类型 |
| $column\_len | 数据长度 |
| $column\_nn | NOT NULL |
| $column\_after | 排序 |
| $column\_def | 默认值 |

返回值

无

说明 向指定表添加列。

示例

## 10.6.3update::column\_exists 判断列是否存在

定义

function column\_exists($table\_name,$column\_name)

参数

$table\_name 表名

$column\_name 列名返回值 如果存在返回true；不存在返回false；不确定返回null。

说明 判断指定表中是否存在指定列。

示例

if(update::column\_exists(‘t\_test’,’name’)===false)

{

……对没有name列的处理

}

## 10.6.4update::column\_info读取列信息

定义

|  |  |
| --- | --- |
| function column\_info($table\_name,$column\_name) | |
| 参数  $table\_name | 表名 |
| $column\_name | 列名 |
| 返回值  $info[‘type’] | 列类型 |
| $info[‘len’] | 长度 |
| $info[‘un’]  说明  读取列信息  示例 | 正整数 |
| <?php  echo update::column\_info('t\_pub\_info','id');  ?> | |

返回示例

Array

(

[type] => int

[len] => 10

[un] => 1

)

## 10.6.5update::procedure\_exists 指定存储过程是否存在

定义

function procedure\_exists($procedure\_name)

参数

$procedure\_name 存储过程名称

返回值

如果存在返回true；不存在返回false；不确定返回null。

说明 返回指定的存储过程是否存在。

当前此命令仅支持SQL Server。示例

if(update::procedure\_exists(‘ap\_proc’)===false)

{

……存储过程不存在时的处理

}

## 10.6.6update::table\_create 创建表

定义

function table\_create($tablename,$table)

参数

$tablename 要创建的表名

$table 表格式，本参数以数组方式保存表中的各列数据：

array(

array('col'=>'id' ,'type'=>'uint' ,'nn'=>true,'ai'=>true,'pk'=>true), array('col'=>'name','type'=>'varchar' ,'len'=>45 ,'nn'=>true),

……

);

$table中各列说明 col 列名 type 数据类型。支持的数据类型：

|  |  |  |
| --- | --- | --- |
| type值 | 对应MySQL数据类型 | 对应SQL Server数据类型 |
| datetime | datetime | datetime |
| tinyint | tinyint | tinyint |
| utinyint | tinyint unsigned | tinyint |
| int | int | int |
| uint | int unsigned | int |
| int64 | bigint | bigint |
| uint64 | bigint unsigned | bigint |
| char | char | nchar |
| varchar | varchar | nvarchar |

注：SQL Server不支持uint数据类型，所以设置为int。 len 对char、varchar这类具有长度的数据类型设置数据长度。

对int这类不需要指定数据类型的列设置数据长度，将被忽略此参数。

nn NOT NULL标志，如果不是NOT NULL列，则不要设置此参数 ai 自增列标志。当前默认自增列为1开始，步长+1 pk 主键列标志。

说明 此命令支持MySQL、SQL Server数据库的表创建。

示例

$table=array(

array('col'=>'id' ,'type'=>'uint' ,'nn'=>true,'ai'=>true,'pk'=>true), array('col'=>'name' ,'type'=>'varchar' ,'len'=>45 ,'nn'=>true), array('col'=>'login' ,'type'=>'varchar' ,'len'=>45 ,'nn'=>true), array('col'=>'paddword' ,'type'=>'char' ,'len'=>41 ,'nn'=>true), array('col'=>'enabled' ,'type'=>'utinyint' ,'nn'=>true), array('col'=>'type' ,'type'=>'char' ,'len'=>1 ,'nn'=>true), array('col'=>'oc' ,'type'=>'uint' ,'nn'=>true), array('col'=>'args' ,'type'=>'varchar' ,'len'=>1000 ,'nn'=>true)

);

update::table\_create('t\_sys\_role',$table);

或

|  |
| --- |
| update::table\_create('t\_sys\_role', array(  array('col'=>'id' ,'type'=>'uint' ,'nn'=>true,'ai'=>true,'pk'=>true), array('col'=>'name' ,'type'=>'varchar' ,'len'=>45 ,'nn'=>true), array('col'=>'login' ,'type'=>'varchar' ,'len'=>45 ,'nn'=>true), array('col'=>'paddword' ,'type'=>'char' ,'len'=>41 ,'nn'=>true), array('col'=>'enabled' ,'type'=>'utinyint' ,'nn'=>true), array('col'=>'type' ,'type'=>'char' ,'len'=>1 ,'nn'=>true), array('col'=>'oc' ,'type'=>'uint' ,'nn'=>true),  array('col'=>'args' ,'type'=>'varchar' ,'len'=>1000 ,'nn'=>true) )); |

## 10.6.7update::table\_delete 删除表

定义

function table\_delete($table\_name)

参数

$table\_name 表名

返回值

无

说明

删除指定表

示例

<?php

table\_delete(‘t\_test’); ?>

## 10.6.8update::table\_exists 判断表是否存在

定义

function table\_exists($tablename)

参数

$tablename 表名称返回值 表存在返回true，不存在返回false。 如果当前数据库不被支持，返回null。

说明 判断指定表是否存在。

因为当数据库不支持会返回null，所以判断是否存在时，请使用===。

示例

if(update::table\_exists('t\_sys\_role')===false)

{

……创建表

}

## 10.6.9update::index\_exists 判断指定索引是否存在

定义

function index\_exists($table\_name,$index\_name)

参数

$table\_name 表名

$index\_name 索引名

返回值 如果存在返回true，否则返回false。

说明 本方法当前仅支持MySQL。

10.6.10MySQL常用判断

# 11模块打包

1. 使用小写字母的原因是为考虑到移植的问题。在MySQL中，整个数据库作为一个目录存储，所有的表单独作为一个文件来存储，表名即为文件名称。这样在从Linux移植到Windows或者其他系统间互相移植时，会由于各系统对文件名称中的大小写字母的处理不同而增加移植的复杂度。统一使用小写字母将把移植时的成本降到最低。 [↑](#footnote-ref-1)
2. 系统会自动忽略所有保存到$\_SESSION的信息。 [↑](#footnote-ref-2)
3. 这个记录，可在sys-log模块中查看。 [↑](#footnote-ref-3)
4. $check\_path=false，只是及特殊的情况下才会使用，因为这个直接影响到整个系统的安全性，在实际开发中，能不使用的地方，尽量不要使用! [↑](#footnote-ref-4)
5. 具体参见acl函数的调用部分 [↑](#footnote-ref-5)