

SSMS应用开发框架API文档 v1.0

一些值得注意的

- cache和upfiles需要可读写权限,
- 配置文件:根目录下的config.php

超级全局变量

- define

```
BASEPATH    //定义应用根目录的绝对路径

SYSDIR      //system核心目录
```

- 配置文件

```
$GLOBALS['config']
```

- 应用路径

```
$GLOBALS['path_app']
```

- 当前controller

```
$GLOBALS['current_c']
```

- 当前controller的method

```
$GLOBALS['current_m']
```

- 数据库操作实例

```
$GLOBALS['db']
```

- 缓存操作实例

```
$GLOBALS['cache']    //默认  
$GLOBALS['cache_file']    //文件缓存  
//$GLOBALS['cache_memcache']    //配置了就有
```

全局方法(在/system/common.php中)

- 友好格式的打印输出

```
dump(arg1,arg2...);
```

- 打印log文件日志

```
logs($file,$msg);
```

- 显示消息，并支持定时跳转

```
showMessage($msg,$url,$timeout=-1);
```

- JS分页(没有使用a标签)

```
multiPages($curpage,$psize,$itemcount,$params=false)
```

应用内方法

- controller中调用 model

```
$this->model->load('modelname');
```

- controller中调用 view（模板）

```
$this->view->load($viewname,$datas=array());
```

- controller中变更当前页的模板版本号，适用于多版本混合

```
$this->view->setVersion($ver='default');
```

- controller中变更当前页的模板版本号，适用于多模板引擎混合

```
$this->view->setEngine($engine=false);
```

Uri类

path:/system/core/uri.class.php

info: 文件内有详细注释

- 解析后的uri参数结构

```
Uri::getParams();  
//结构如下  
array('_c'=>'home','_m'=>'index',params=array(...))
```

- 根据规则组建url 强烈建议页面资源访问链接都这么生成，否则写死了的话，规则改了 所有页面都要改 该方法的最后一个参数决定了是返回(默认)还是直接跳转 (true)

```
Uri::build('home','index',array('param1','param2',...),$redirect=true)  
;
```

- 跳转

```
Uri::redirect('/home/index/');
```

- 分页

```
Uri::pages($curpage,$pagesize,$itemcount,$url);
```

- 手动设置前一页,默认当前页

```
Uri::setPrevPage($url);
```

- 获取设置的前一页,并删除，如果取不到返回根目录

```
Uri::getPrevPage();
```

- 文件路径转网站路径,注意只转换站内文件, 外部文件会出现问题

```
Uri::path2url($path);
```

- 安全get,目前只去除html,xml标签, 并使用反斜线引用字符串

```
Uri::get($key,$default='');
```

- 安全post,目前只去除html,xml标签, 并使用反斜线引用字符串

```
Uri::post($key,$default='');
```

- 安全request,目前只去除html,xml标签, 并使用反斜线引用字符串

```
Uri::request($key,$default='');
```

缓存

path:/system/libs/cache.class.php

info: 文件内有详细注释。存加入组（命名空间）的概念, `get,set,delete,clear`的第一个参数都是组

- 获取test组下的 a键

```
$GLOBALS['cache']->get('test','a');  
$GLOBALS['cache_file']->get('test','a');  
$GLOBALS['cache_memcache']->get('test','a');
```

- 设置test组下的 a键 的值（过期时间（秒）, -1代表使用系统配置的默认缓存时间, 0代表永不过期）

```
$GLOBALS['cache']->set('test','a',array(1,2,3),-1);
```

- 删除test组下的 a键

```
$GLOBALS['cache']->delete('test','a');
```

- 清空test组下的所有键

```
$GLOBALS['cache']->clear('test');
```

- 清空所有组下的所有键

```
$GLOBALS['cache']->clear();
```

数据库操作

path:/system/libs/db.class.php

info: 文件内有详细注释。

- 数据库连接

```
$GLOBALS['db']->connect();
```

- 关闭连接

```
$GLOBALS['db']->close();
```

- 选择操作数据库

```
$GLOBALS['db']->select_db($dbname);  
$GLOBALS['db']->create_db($dbname);
```

- 获取当前数据库详情

```
$GLOBALS['db']->get_db_info();
```

- 返回上一次sql操作语句

```
$GLOBALS['db']->getlastsql();
```

- 数据库执行语句，可执行查询添加修改删除等任何sql语句

```
$GLOBALS['db']->query($sql);
```

- 清空结果集

```
$GLOBALS['db']->free($result);
```

- 返回上一次insert操作生成的id

```
$GLOBALS['db']->insert_id();
```

- 返回insert,update,delete影响的记录数

```
$GLOBALS['db']->affected_rows();
```

- 返回select返回的记录数

```
$GLOBALS['db']->num_rows($result);
```

- 从结果集中获取一个行作为关联数组返回

```
$GLOBALS['db']->fetch_array($result);
```

- 从结果集中获取一个行作为数字索引数组返回

```
$GLOBALS['db']->fetch_assoc($result);
```

- 将整个结果集转为关联数组返回

```
$GLOBALS['db']->fetch_all($result,$index='');
```

- 根据传入的字段数组，在表中选择数据

```
$GLOBALS['db']->select($table,$conditions=array(),$index='', $orderby='', $page=1, $psize=20);
```

- 根据传入的字段数组，在表中插入一条数据

```
$GLOBALS['db']->insert($table,$arr);
```

- 简化的更新函数

```
$GLOBALS['db']->update($table,$arr,$condition='');
```

- 简化的删除函数

```
$GLOBALS['db']->delete($table,$condition='');
```

- 返回结果集中某行某列的值

```
$GLOBALS['db']->result($query, $row = 0, $field = 0);
```

- 获取上一个sql文本错误

```
$GLOBALS['db']->getlasterror();
```

- 构建where字符

```
$GLOBALS['db']->build_where($cond);
```

- 事务开始（要求表支持）

```
$GLOBALS['db']->transaction_begin();
```

- 事务提交（结束）（要求表支持）

```
$GLOBALS['db']->transaction_commit();
```

- 事务回滚（结束）（要求表支持）

```
$GLOBALS['db']->transaction_rollback();
```

Helper类

path:/liberay/helper.class.php

info: 全部都是静态方法，文件内有详细注释。

- 获取服务器系统信息

```
Helper::getSystemInfo();
```

- 获取站点相关信息

```
Helper::getSiteInfo();
```

- 获取客户端IP

```
Helper::getClientIp();
```

- 根据类名生成实例

```
Helper::refClass($classname,$args=array());
```

- 生成唯一标示

```
Helper::getUniqid();
```

- 设置session

```
Helper::setSession($key,$val);
```

- 获取session

```
Helper::getSession($key,$destory);
```

Captcha图片验证码类

path:/liberay/captcha.class.php

info: 文件内有详细注释。

- 生成验证码图片，值得注意的是buildAndExportImage方法直接生成图片（header改成图片）

```
$captcha = new Captcha();  
$captcha->buildAndExportImage();
```

- 检查验证码

```
Captcha::check($val);
```

Http类

path:/liberay/http.class.php

info: 全部都是静态方法，文件内有详细注释。

- 拼接URL

```
Http::buildUrl($baseUrl,$params=false);
```

- get请求

```
Http::get($url,$params=false);
```

- post请求

```
Http::post($url,$params=false);
```

Qrcoder类

path:/liberay/http.class.php

info: 封装第三方qrcode类，，

- 生成二维码图片并返回地址

```
Qrcoder::image($itype,$data,$saveurl,$errorCorrectionLevel="Q",$matrix  
PointSize=5,$margin=2);
```

SFile文件处理类

path:/liberay/sfile.class.php

info: 全部都是静态方法，文件内有详细注释。

- 创建多级目录

```
SFile::mkdirs($path, $mode = 0777);
```

- 复制文件

```
SFile::copy($src , $dst);
```

- 移动文件,不检查目录是否存在

```
SFile::move($src , $dst);
```

- 获取目录下的所以对象数组,带不带/结尾 无所谓

```
SFile::getPathList($path,$desc=false)
```

- 获取目录下所有文件类型的对象

```
SFile::getPathFiles($path,$ext='', $desc=false)
```

- 获取目录下所有目录类型的对象

```
SFile::getPathFolders($path,$desc=false)
```

- 清空目录

```
SFile::clearDir($path,$rmdir=false)
```

- 返回文件信息

```
SFile::getInfo($file)
```

- 返回文件后缀

```
SFile::getExt($file)
```

- 读取文件

```
SFile::read($file,$offset=0,$len=null)
```

- 写文件 成功的话返回写入的字节数否则false

```
SFile::write($file,$data,$append=true)
```

- 删除文件

```
SFile::remove($file)
```

- 清除文件状态缓存。防止多次操作同一个文件时，缓存文件的状态信息，比如大小，位移等等

```
SFile::clearState()
```

SForm表单创建类

path:/liberay/sform.class.php

info: 全部都是静态方法，文件内有详细注释。扩展中

- 创建input

```
SForm::build_input($name,$val='', $type='text', $extarr=false)
```

- 创建option下拉菜单,基于二维数组

```
SForm::build_options($items,$valkey,$namekey,$selectval='')
```

- 创建option下拉菜单,基于一维数组

```
SForm::build_options($items,$selectval='')
```

- 创建checkbox或者radio组,基于二维数组

```
SForm::build_checks($items,$name,$valkey,$namekey,$selectval='', $type='checkbox')
```

- 创建checkbox或者radio组,基于一维数组

```
SForm::build_checks_simple($items,$name,$selectval='', $type='checkbox')
)
```

FormVerify 表单验证类

path:/libaray/formverify.class.php

info: 摘自canphp, 做了些小修改, 全静态方法,网上有详细的文档。

- 用于设置验证规则, 并进行验证

```
FormVerify::rule($array=array())
```

- 检查字符串长度, 按字节计算

```
FormVerify::len($str,$min=0,$max=255)
```

- 检查字符串是否为空

```
FormVerify::must($str)
```

- 检查两次输入的值是否相同

```
FormVerify::same($str1,$str2)
```

- 检查用户名

```
FormVerify::userName($str,$len_min=0,$len_max=255,$type='ALL')
```

- 检查密码

```
FormVerify::password($str,$len_min=0,$len_max=255)
```

- 验证邮箱

```
FormVerify::email($str)
```

- 验证手机号码

```
FormVerify::mobile($str)
```

- 验证固定电话

```
FormVerify::tel($str)
```

- 验证qq号码

```
FormVerify::qq($str)
```

- 验证邮政编码

```
FormVerify::zipCode($str)
```

- 验证ip

```
FormVerify::ip($str)
```

- 验证身份证(中国)

```
FormVerify::idCard($str)
```

- 验证网址

```
FormVerify::url($str)
```

SImage图片处理类

path:/libaray/simage.class.php

info: 全部都是静态方法，文件内有详细注释。依托于Imagick，请安装相关扩展

- 获取图片信息

```
SImage::getImgInfo($pic)
```

- 给图片增加水印,不生成图片，无返回

```
SImage::addMark($pic , $water , $offx , $offy)
```

- 不改动原图，在同目录下生成缩放裁切后生成指定尺寸的缩略图

```
SImage::resize($src,$type , $width , $prefix='')
```

- 不改变原图，在同目录下生成根据宽度等比缩放的缩略图

```
SImage::psize($src,$width)
```

- 不改变原图，在同目录下生成根据宽度剪裁成正方形,然后缩放的缩略图

```
SImage::csize($src,$width)
```

SUpload文件上传类

path:/libaray/supload.class.php

info: 如果发生读写错误，请记得更新文上传目录的【权限】

- 初始化并上传

```
$up = new SUpload($formName='', $dirPath='', $maxSize=8388608);  
$ret = $up->upload($fileName = '');
```

- 获取上传后的文件名

```
$up->UpFile()
```

- 上传后文件的目录

```
$up->UpFilePath()
```

- 上传后文件的路径

```
$up->filePath()
```

- 获取文件大小

```
$up->getSize($format = 'B')
```

- 获取文件类型

```
$up->getExt()
```

- 获取原文件名称

```
$up->getName()
```

- 根据时间戳新建一个文件名，后缀不变

```
$up->newName()
```

- 显示错误参数

```
$up->Err()
```

Useragent

path:/liberay/useragent.class.php

info: copy的codeigniter的类，详情看文件

业务逻辑部分

第三方登陆

- 去登陆逻辑，根据后台配置自动去登陆

```
// $key = qq|weibo|douban|renren  
Module_ThirdLogin::gotoAppLogin($key);
```

- 框架实现的第三方登陆html代码,用户可自己修改自己的

```
Module_ThirdLogin::getThirdLoginHTML();
```

。 。 。

支付

- 去支付逻辑，根据后台配置调用支付接口

```
// $key = alipay  
Module_ThirdLogin::gotoAppLogin($key);
```