



云存储 PHP SDK

接口文档

1.0

发布日期： 2013年7月8日

百度开发者中心

（版权所有，翻版必究）

目录

第 1 章 使用说明 3

第 2 章 接口列表 3

2.1 CREATE_BUCKET 4

2.2 GET_BUCKET_ACL 4

2.3 DELETE_BUCKET 5

2.4 LIST_BUCKET 5

2.5 SET_BUCKET_ACL 5

2.6 CREATE_OBJECT 6

2.7 CREATE_OBJECT_BY_CONTENT 7

2.8 CREATE_OBJECT_SUPERFILE 7

2.9 COPY_OBJECT 8

2.10 GET_OBJECT 9

2.11 GET_OBJECT_INFO 9

2.12 GET_OBJECT_ACL 10

2.13 DELETE_OBJECT 10

2.14 LIST_OBJECT 11

2.15 LIST_OBJECT_BY_DIR 11

2.16 SET_OBJECT_META 12

2.17 SET_OBJECT_ACL 13

2.18 GENERATE_GET_OBJECT_URL 13

2.19 GENERATE_PUT_OBJECT_URL 13

2.20 UPLOAD_DIRECTORY 14

2.21 IS_OBJECT_EXIST 15

第1章 使用说明

通过 `ak`、`sk`、`host` 构造出 `BaiduBS` 类对象以后，直接调用其中提供的接口，各个接口的返回值类型均为 `ResponseCore`，其中包含了接口调用的所有结果信息，可以通过返回的 `ResponseCore` 中的 `isOK()` 接口来查看接口调用是否执行成功。

`PHP_SDK` 抛出异常时不对异常进行处理，需上层调用者对异常进行捕捉处理。

第2章 接口列表

按照操作对象的不同，`PHP_SDK` 中的接口主要分为 `Bucket`、`Object`、`ACL` 三部分，同时也提供了一些满足其它需求的接口：

- **Bucket 相关**
 - `create_bucket` -- 创建 bucket
 - `delete_bucket` -- 删除 bucket
 - `list_bucket` -- 获取用户的 bucket 列表
- **Object 相关**
 - `create_object` -- 创建 object
 - `create_object_superfile` -- 创建 superfile
 - `create_object_by_content` -- 从接口参数中携带文件内容
 - `copy_object` -- 以拷贝的方式创建 object
 - `delete_object` -- 删除 object
 - `get_object` -- 获取 object
 - `get_object_info` -- 获取文件信息
 - `set_object_meta` -- 设置 object 的 meta 信息
 - `list_object` -- 获取用户的 object 列表
 - `list_object_by_dir` -- 获取某文件夹下的 object 列表
- **ACL 相关**
 - `get_bucket_acl` -- 获取 bucket 的 acl policy
 - `set_bucket_acl` -- 设置 bucket 的 acl policy
 - `get_object_acl` -- 获取 object 的 acl policy
 - `set_object_acl` -- 设置 object 的 acl policy
- **其它**
 - `generate_get_object_url` -- 生成 `get_object` 的 url
 - `generate_put_object_url` -- 生成 `put_object` 的 url
 - `is_object_exist` -- 判断 object 是否存在

- upload_directory -- 将目录中的所有文件进行上传，每个文件为单独 object

2.1 create_bucket

- 功能:

创建新 bucket，bucket 是遵循如下命名规则：

- 由小写字母，数字和横线'-'组成，长度为 6~63 位
- 不能以数字作为 Bucket 开头
- 不能以 '-' 作为 Bucket 的开头或者结尾

- 函数原型:

```
public function create_bucket($bucket, $acl = NULL, $opt = NULL)
```

- 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false

- 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$bucket | bucket 名 | 是 |
| \$acl | acl_type 可在创建 bucket 时通过 acl_type（参见 Create Object）指定 bucket 的 acl 权限设置 常用的值有： private,public-read, public-write, public-read-write,public-control | 否 |

- 注意事项:

- bucket 名必须遵循 bucket 命名规则
- acl 参数只能选择 acl_type 中提供的

2.2 get_bucket_acl

- 功能:

为获取 bucket 的 acl，PHP_SDK 提供了 get_bucket_acl 接口

- 函数原型:

```
public function get_bucket_acl($bucket, $opt = NULL)
```

- 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false；返回的 acl 信息在 ResponseCore 的 body 属性中

- 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|---|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$opt | 可选参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

- 注意事项:

bucket 必须存在且有权限

2.3 delete_bucket

- 功能：
PHP_SDK 提供了 delete_bucket 接口来满足删除 bucket 的需求
- 函数原型：

```
public function delete_bucket($bucket, $opt = NULL)
```

- 返回值：
返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|---|------|
| \$bucket | 需要删除的 bucket | 是 |
| \$opt | 可选参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

- 注意事项：
bucket 必须存在且有权限

2.4 list_bucket

- 功能：
为获取当前密钥对拥有者的 bucket 列表，PHP_SDK 提供了 list_bucket 接口
- 函数原型：

```
public function list_bucket($opt = NULL)
```

- 返回值：
返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false；返回的 bucket list 信息在 ResponseCore 的 body 属性中
- 参数说明：

| 参数名称 | | 描述 | 是否必需 |
|-------|----------------------|--------------------------------------|------|
| \$opt | import_bs_log_method | 用户的自定义日志处理函数，函数定义如 function f(\$log) | 否 |

- 注意事项：
若无用户自定义日志处理函数，则会调用默认的日志处理函数

2.5 set_bucket_acl

- 功能：
为设置 bucket 的 acl ，PHP_SDK 提供了 set_bucket_acl 接口
- 函数原型：

```
public function set_bucket_acl($bucket, $acl, $opt = NULL)
```

- 返回值：

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false

● 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$acl | 用户设置的 acl 数据，设置 bucket 的 acl 支持三种模式： 1. 设置详细 json 格式的 acl 1.1 acl 为 json 的 array 1.2 acl 为 json 的 string 2. 通过 acl_type 字段进行设置 | 是 |
| \$opt | import_bs_log_method 用户的自定义日志处理函数，函数定义如 function f(\$log) | 否 |

● 注意事项：

- bucket 必须存在且有权限
- acl 参数内容必须符合 acl 规则

2.6 create_object

● 功能：

对 Create Object 情景， PHP_SDK 提供了 2 个接口：

- create_object -- 上传文件，创建 object
- create_object_by_content -- 直接把内容附带在参数中无需再上传文件，创建 object

● 函数原型：

```
public function create_object($bucket, $object, $file, $opt = NULL)
```

● 返回值：

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false

● 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|---|------|
| \$bucket | bucket 名 由小写字母，数字，‘-’组成，长度为 6~63 位，不能以数字开头，不能以‘-’开头或结尾 | 是 |
| \$object | object 名 以“/”开头的 object 名，不超过 255 字节 | 是 |
| \$file | 待上传的文件的相对路径或绝对路径 | 是 |
| \$opt | filename | 否 |
| | acl | |
| | seekTo | |
| | length | |
| | Headers | |

● 代码例子：

create_object.php

```
<?php
require_once './bcs.class.php';
$ak = "accesskey"; // ak&sk 请在管理网站上获取
$sk = "secretkey";
$host = 'bcs.duapp.com';
$baidu_bcs = new BaiduBCS ( $ak, $sk, $host );
$bucket = "sample";
$object = "/create_object.php";
$file = "./create_object.php";
$response = $baidu_bcs->create_object ( $bucket, $object, $file );
echo $response->isOK () ? "OK\n" : "NOT OK\n";
echo 'Status:' . $response->status . "\n";
echo 'Body:' . $response->body . "\n";
foreach ( $response->header as $key => $value ) {
    echo "Http Header -> $key:$value\n";
}
?>
```

- 注意事项:
 - bucket 必须存在且有权限
 - file 必须存在
 - opt 参数中的 acl 只能使用 acl_type ， 即 "public-read" | "public-write" | "public-read-write" | "public-control" | "private"

2.7 create_object_by_content

- 功能:

根据内容创建 object

- 函数原型:

```
public function create_object_by_content($bucket, $object, $content, $opt = NULL)
```

- 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false
- 参数说明:

| 参数名称 | 描述 | 是否必需 |
|------------------------|-------------------------------|------|
| \$content | 字符串，直接作为 object 的内容，无需再指定上传文件 | 是 |
| 其它参数同 create_object 接口 | | |

- 注意事项:
 - content 必须为非空的字符串
 - 其它同 create_object 接口

2.8 create_object_superfile

- 功能:

Create Superfile 是通过 superfile 的方式上传文件，因为在云存储中 superfile 是一种特殊的

object ，所以这也可以看做是 Create Object 的一种特殊情况。
 PHP_SDK 提供了 create_object_superfile 接口

● 函数原型:

```
create_object_superfile($bucket, $object, $file, $opt = NULL)
```

● 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false

● 参数说明:

| 参数名称 | | 描述 | 是否必需 |
|----------|-----------------|---|------|
| \$bucket | | object 所属的 bucket 名 | 是 |
| \$object | | object 名 以"/"开头的 object 名，不超过 255 字节 | 是 |
| \$file | | 待上传的文件的相对路径或绝对路径 | 是 |
| \$opt | filename | 可以事先指定下载时，文件的文件名 | 否 |
| | sub_object_size | 指定子文件的划分大小，单位 B，建议以 256KB 为单位进行子 object 划分，默认为 1MB 进行划分 | |
| | Headers | 设定文件的一些属性，现在支持以下几种类型： x-bs-acl , Content-Type , Cache-Control, Content-Disposition, Content-Encoding, Content-MD5, Expires 其中 x-bs-acl 共五种： public-read, public-write, public-read-write, pubic-control, private | |

● 注意事项:

- bucket 必须存在且有权限
- file 必须存在

2.9 copy_object

● 功能:

Copy Object 是以拷贝的方式创建 object ，无需重新上传文件
 PHP_SDK 提供了 copy_object 接口

● 函数原型:

```
public function copy_object($source, $dest, $opt = NULL)
```

● 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false

● 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$source | object 来源，是包含 bucket 和 object 的数组 | 是 |
| \$dest | 待 copy 的目标 object，是包含 bucket 和 object 的数组 | 是 |
| \$opt | 否参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

● 注意事项:

- source 必须存在且有权限

- bucket 必须存在且有权限
- file 必须存在

2.10 get_object

● 功能:

云存储中 Get Object 包括 3 种情况，PHP_SDK 提供了对应的 3 个接口:

- get_object -- 获取 object
- get_object_info -- 获取文件信息
- get_object_acl -- 获取 object 的 acl 信息

● 函数原型:

```
public function get_object($bucket, $object, $opt = NULL)
```

● 返回值:

- 返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false
- 若指定了 fileWriteTo 参数，则获取到的 object 数据将写入该文件；若未设置，则可从 ResponseCore 的 body 属性中读取结果

● 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|---|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$object | object 名 | 是 |
| \$opt | object 名 否参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 需要通过参数 fileWriteTo 指定将请求结果写入该文件 | 是 |

● 注意事项:

bucket 和 object 都必须存在且有权限

2.11 get_object_info

● 功能:

获取 object 信息

● 函数原型:

```
public function get_object_info($bucket, $object, $opt = NULL)
```

● 返回值:

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false；返回的 info 信息在 ResponseCore 的 header 属性中

● 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|---------------------|------|
| \$bucket | object 所属的 bucket 名 | 是 |

| | | |
|----------|--|---|
| \$object | object 名 | 是 |
| \$opt | 否参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

- 注意事项：
bucket 和 object 都必须存在且有权限

2.12 get_object_acl

- 功能：
获取 object 的 ACL 列表信息
- 函数原型：

```
public function get_object_acl($bucket, $object, $opt = NULL)
```

- 返回值：
返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false；返回的 acl 信息在 ResponseCore 的 body 属性中
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$object | object 名 | 是 |
| \$opt | 否参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

- 注意事项：
bucket 和 object 都必须存在且有权限
- 说明
PHP_SDK 提供了 delete_object 接口来满足删除 object 的需求

2.13 delete_object

- 功能：
删除 object
- 函数原型：

```
public function delete_object($bucket, $object, $opt = NULL)
```

- 返回值：
返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$object | object 名 | 是 |
| \$opt | 否参数的数组，其中可包含 source_tag 指定拷贝对象的版本号，以及用户自定义日志处理函数 | 否 |

- 注意事项:
bucket 和 object 都必须存在且有权限
- 说明:
对于 List Object 场景, PHP_SDK 提供了两个接口:
 - list_object -- 获取 bucket 中 object 列表
 - list_object_by_dir -- 获取 bucket 中指定目录下的 object 列表, 通过这个接口能获取到比 list_object 更精细的结果

2.14 list_object

- 功能:
获取 object 列表
- 函数原型:

```
public function list_object($bucket, $opt = NULL)
```

- 返回值:
返回 ResponseCore 类型, 调用其 isOK() 接口检查执行结果: 成功为 true, 失败则 false; 返回的 object list 信息在 ResponseCore 的 body 属性中
- 参数说明:

| 参数名称 | | 描述 | 是否必需 |
|----------|--------|--|------|
| \$bucket | | bucket 名 | 是 |
| \$opt | start | 获取 object list 的起点, 主要用于页面翻页显示功能, 用法同 sql 查询中 start 的用法 | 否 |
| | limit | 获取 object list 的数目, 主要用于页面翻页显示功能, 用法同 sql 查询中 limit 的用法 | |
| | prefix | 指定需要获取的 object 的前缀, prefix 必须 '/' 开头, 用法类似 sql 查询中的 like 'prefix%' | |

- 注意事项:
 - bucket 必须存在且有权限
 - prefix 必须以 '/' 开头

2.15 list_object_by_dir

- 功能:
- 函数原型:

```
public function list_object_by_dir($bucket, $dir = '/', $list_model = 2, $opt = NULL)
```

- 返回值:
返回 ResponseCore 类型, 调用其 isOK() 接口检查执行结果: 成功为 true, 失败则 false; 返回的 object list 信息在 ResponseCore 的 body 属性中

● 参数说明:

| 参数名称 | | 描述 | 是否必需 |
|---------------|--------|--|------|
| \$bucket | | bucket 名 | 是 |
| \$dir | | 目录名, 格式为必须以 '/' 开头和结尾, 默认为 '/' | 是 |
| \$list_models | | 目录展现形式, 值可以为 0,1,2, 默认为 2, 以下对各个值的功能进行描述: <ul style="list-style-type: none">0->只返回 object 列表, 不返回子目录列表1->只返回子目录列表, 不返回 object 列表2->同时返回子目录列表和 object 列表 | 是 |
| \$opt | start | 获取 object list 的起点, 主要用于页面翻页显示功能, 用法同 sql 查询中 start 的用法 | 否 |
| | limit | 获取 object list 的数目, 主要用于页面翻页显示功能, 用法同 sql 查询中 limit 的用法 | |
| | prefix | 指定需要获取的 object 的前缀, prefix 必须 '/' 开头, 用法类似 sql 查询中的 like 'prefix%' | |

● 注意事项:

- bucket 必须存在且有权限
- dir 必须以 '/' 开头和结尾

2.16 set_object_meta

● 功能:

Set Object 场景包括设置 object 的 meta 和 acl, PHP_SDK 提供了对应的接口

- set_object_meta -- 设置 object 的各种 meta 属性
- set_object_acl -- 设置 object 的 acl 信息

● 函数原型:

```
public function set_object_meta($bucket, $object, $meta, $opt = NULL)
```

● 返回值:

返回 ResponseCore 类型, 调用其 isOK() 接口检查执行结果: 成功为 true, 失败则 false

● 参数说明:

| 参数名称 | | 描述 | 是否必需 |
|----------|---------------------|---|------|
| \$bucket | | bucket 名 | 是 |
| \$object | | object 名 | 是 |
| \$meta | Content-Type | 指定文件类型 | 否 |
| | Cache-Control | 控制浏览器的 cache 策略 | |
| | Content-Disposition | 设置浏览器下载文件时候的文件名 | |
| | Content-Encoding | 文件编码方式 | |
| | Expires | GET 返回的 Header 中 Expires 值, 如果不设置, 将自动设为下载时间加 3 天 | |

● 注意事项:

- bucket 和 object 必须存在且有权限
- meta 数组中只能选择目前支持的 meta 种类

2.17 set_object_acl

- 功能：
- 函数原型：

```
public function set_object_acl($bucket, $object, $acl, $opt = NULL)
```

- 返回值：

返回 ResponseCore 类型，调用其 isOK() 接口检查执行结果：成功为 true，失败则 false
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|--|------|
| \$bucket | bucket 名 | 是 |
| \$object | object 名 | 是 |
| \$acl | <ul style="list-style-type: none"> 有三种方式设置 policy 设置详细 json 格式的 acl <ul style="list-style-type: none"> acl 为 json 的 array acl 为 json 的 string 通过 acl_type 字段进行设置 | 是 |

- 注意事项：
 - bucket 和 object 必须存在且有权限
 - acl 参数内容必须符合 acl 规则

2.18 generate_get_object_url

- 功能：

用户有时会需要上传或者下载某 object 的签名 url，为 PHP_SDK 对此提供 2 个相应接口

 - generate_get_object_url -- 生成 get_object 的 url
 - generate_put_object_url -- 生成 put_object 的 url

- 函数原型：

```
public function generate_get_object_url($bucket, $object)
```

- 返回值：

返回字符串类型的可用于直接下载某 object 的签名 url
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|---------------------|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$object | object 名 | 是 |

- 注意事项：

要使用此 url 去下载对应的 object，需具有相应权限才能成功

2.19 generate_put_object_url

- 功能：

创建某 object 的 签名 URL

● 函数原型:

```
public function generate_put_object_url($bucket, $object)
```

● 返回值:

返回字符串类型的可用于创建某 object 的签名 url，比如在 linux 上的可直接使用 curl -T {local_file} -v {签名 url} 来创建 object

● 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|---------------------|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$object | object 名 | 是 |

● 注意事项:

要使用此 url 去创建对应的 object，需具有相应权限才能成功

● 说明:

用户有时会需要将某个文件夹中的所有文件上传，每个文件为单独 object，为避免用户逐个操作上传其中的文件，PHP_SDK 提供了 upload_directory 接口。

object 命令方式有两种，如有 /home/worker/a/b/c.txt 需上传目录为 \$dir=/home/worker/a

- object 默认命名方式为“子目录名 + 文件名”，如上述文件 c.txt，默认为 '/b/c.txt'
- 增强命名模式，在 \$opt 中有否参数进行配置
- 用户可自定义上传文件前的操作函数 pre_filter
- 用户可自定义上传文件后的操作函数 post_filter
- 详细说明可在 PHP_SDK 本接口的注释中查看

PHP_SDK 中提供的批量上传文件夹中文件的 shell 小工具就是调用此接口实现的，在 linux 上可直接使用。

2.20 upload_directory

● 功能:

● 函数原型:

```
public function upload_directory($bucket, $dir, $opt = NULL)
```

● 返回值:

返回数组形式的上传结果，该数组结构为

- *success -- 上传成功的文件数目
- *skipped -- 被跳过的文件
- *failed -- 上传失败的文件

● 参数说明:

| 参数名称 | 描述 | 是否必需 |
|----------|---------------------|------|
| \$bucket | object 所属的 bucket 名 | 是 |
| \$dir | 待上传的文件夹路径 | 是 |

| | | | |
|-------|-----------------------|--|---|
| \$opt | prefix | object 名前缀，上传的 object 名由此前缀加文件名构成 | 否 |
| | has_sub_directory | boolean 型（默认为 true），表示 object 命名中是否携带文件的子目录结构，若置为 false，请确认待上传的目录和所有子目录中没有重名文件，否则有可能会产生 object 覆盖问题 | |
| | import_bs_pre_filter | 用户可自定义上传文件前的操作函数 1. 函数参数列表顺序需为 (\$bucket,\$object,\$file,&\$opt)，注意 \$opt 为 upload_directory 函数传入的 \$opt 的拷贝，只对当前 object 生效 2. 函数返回值必须为 boolean，当 true 该文件进行上传，若 false 跳过上传 3. 如果函数返回 false，将不会进行 import_bs_post_filter 的调用 | |
| | import_bs_post_filter | 用户可自定义上传文件后的操作函数 1. 函数参数列表顺序需为 (\$bucket,\$object,\$file,&\$opt,\$response)，注意 \$opt 为 upload_directory 函数传入的 \$opt 的拷贝，只对当前 object 生效 2. 函数返回值无要求 | |
| | seek_object | 用户断点续传，需要为 object 名称，如果该 object 在目录中不存在，抛出异常，若存在则将该 object 和此后的 object 进行上传 | |
| | seek_object_id | 作用同 seek_object，只需要传入上传过程中日志中展示的[a/b]中 object 序号即可，注意 object 序号是以 1 开始计算的 | |

- 注意事项：
此接口中 opt 参数所支持的内容比较多，使用前请认真查看说明或注释
- 说明：
用户有时会需要判断某 object 是否存在，PHP_SDK 对此提供了 is_object_exist 接口

2.21 is_object_exist

- 功能：
判断 object 是否存在
- 函数原型：

```
public function is_object_exist($bucket, $object, $opt = NULL)
```

- 返回值：
返回 boolean 类型，object 存在则 true，否则 false
- 参数说明：

| 参数名称 | 描述 | 是否必需 |
|----------|----------|------|
| \$bucket | bucket 名 | 是 |
| \$object | object 名 | 是 |