

## 盛大云存储 PHP SDK 使用说明

1.	系统要求.....	- 2 -
2.	基本概念.....	- 2 -
2.1	AccessKey && AccessSecret .....	- 2 -
2.2	Bucket.....	- 2 -
2.3	Object.....	- 3 -
3.	PHP SDK 概述.....	- 3 -
4.	PHP SDK 文档.....	- 4 -
4.1.	初始化 GrandCloudStorage 对象.....	- 4 -
4.2.	设置 AccessKey 和 AccessSecret .....	- 5 -
4.3.	设置默认 Bucket .....	- 5 -
4.4.	设置请求头部 (Header) .....	- 6 -
4.5.	获取 HTTP 响应代码 (HTTP Status Code) .....	- 6 -
4.6.	获取 HTTP 响应头部 (Header) .....	- 7 -
4.7.	获取 HTTP 响应体长度 (Content-Length) .....	- 8 -
4.8.	获取 HTTP 响应体 (Body) .....	- 9 -
4.9.	获取错误.....	- 9 -
4.10.	获取所有 Bucket .....	- 10 -
4.11.	新建 Bucket .....	- 11 -
4.12.	删除指定 Bucket .....	- 12 -
4.13.	列出指定 Bucket 下所有 Objects .....	- 12 -
4.14.	新建 Object .....	- 14 -
4.15.	获取 Object .....	- 15 -
4.16.	获取 Object 的 meta 信息.....	- 16 -
4.17.	删除 Object .....	- 17 -

## 1. 系统要求

盛大云存储 PHP SDK 需要 CURL 库支持, 在使用前请检查您系统中安装的 PHP 是否已支持 PHP CURL。

## 2. 基本概念

### 2.1 AccessKey & AccessSecret

盛大云存储所有 API 操作都需要对请求做 AccessSecret 签名的校验, 摘要字段将作为请求头发送给云存储系统, 所以在使用 PHP SDK 时必须先设定有效的 AccessKey 和 AccessSecret。

AccessSecret 涉及您存储资料的安全性, 所以请妥善保存您的 AccessSecret, 不要泄漏给第三方。如果 AccessSecret 不慎发生泄漏, 请立即登录盛大云存储用户信息管理站点重新生成新的 AccessSecret, 并废弃已泄漏的 AccessSecret。

**注意:** 任何时候 AccessSecret 都不应作为请求头或内容发送给盛大云存储系统。

#### 1) AccessKey

AccessKey 由盛大云存储颁发, 用于标识用户的唯一身份。AccessKey 在所有的操作中都会被使用, 并且以明文形式传输。

AccessKey 通过云计算网站的云存储用户信息管理获得:

<http://cs-console.grandcloud.cn/access> (需要登录)。

#### 2) AccessSecret

AccessSecret 由盛大云存储颁发, 用于请求头部的签名。AccessSecret 总是随同 AccessKey 一起分发, 一个 AccessKey 对应一个 AccessSecret。

AccessSecret 通过云计算网站的云存储用户信息管理获得:

<http://cs-console.grandcloud.cn/access> (需要登录)。

### 2.2 Bucket

Bucket 是数据存储的基本容器。你可以把 Bucket 想象成文件系统中的目录, 与目录不同的是 Bucket 不支持嵌套。

在用户空间内, 用户可以根据需要创建不同的 Bucket。盛大云存储系统中每个用户空间内最多只能创建 1000 个 Bucket。

**注意:** 盛大云存储系统中 Bucket 名称全局唯一, 即在云存储系统中任何 Bucket 名称都不能重复。例如: 用户 A 创建了名为 “a\_bucket” 的 Bucket 后, 用户 B 尝试创建名字同样为 “a\_bucket” 的 Bucket 将会失败。

#### 1) Bucket 命名规则

- 由小写字母或数字或点号(.)或下划线(\_)或破折号(-)组合而成
- 开头必须是小写字母或数字
- 长度必须大于等于 3 字节且小于等于 255 字节
- 不能是一个 IP 地址形式, 如 192.168.1.1 是不允许的
- 不能以 snda 作为 Bucket 名称前缀

- 如果希望以后提供 DNS 解析，则 Bucket 命名还必须需符合 DNS 主机名的命名规则

## 2.3 Object

Object 是数据存储的基本单元。用户在盛大云存储中存储的任何内容都以 Object 形式存在。每个 Object 由两个部分组成：Object Data 和 Object Meta。Object Data 即存储对象的实际内容数据，Object Meta 为包含对象属性的一系列 Key-Value 键值对。Object 必须存储在 Bucket 下。

*注意：同一 Bucket 下的 Object 名称必须唯一！*

### 1) Object 命名规则

- 使用 UTF-8 编码
- 长度必须大于等于 1 字节且小于等于 1024 字节

## 3. PHP SDK 概述

盛大云存储 PHP SDK 为单一文件，文件名为 GrandCloudStorage.php，只需在您的程序内包含即可使用所有 API。

代码范例：

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$access_key = 'your-access-key';
$access_secret = 'your-access-secret';

$client = new GrandCloudStorage('http://storage.grandcloud.cn');
$client->set_key_secret($access_key, $access_secret);

$bucket_name = "test_bucket";

// create new bucket
$client->put_bucket($bucket_name);

// select $bucket_name as default for following ops
$client->set_bucket($bucket_name);

// upload localfile.ext file to $bucket_name and assign name as
text.ext
$client->put_object("test.ext", "localfile.ext");

// get test.ext object's meta
$metas = $client->head_object("test.ext");
```

```
// download test.ext object as local tmp.ext file
$client->get_object("test.ext", "tmp.ext");

// delete test.ext object
$client->delete_object("test.ext");
?>
```

## 4. PHP SDK 文档

PHP SDK 定义了三个用于云存储的对象：

- GCBucket: 每个 GCBucket 对象表示一个盛大云存储 Bucket;
- GCOBJECT: 每个 GCOBJECT 对象表示一个盛大云存储 Object;
- GCEntity: 每个 GCEntity 对象表示一个盛大云存储 Bucket 及其包含的 Object 列表的集合;

*注意: PHP SDK 中所有的 API 调用只有在成功时才返回正确的数据, 其他任何错误都将以异常的形式抛出!*

### 4.1. 初始化 GrandCloudStorage 对象

接口定义:

```
/**
 * constructor
 * @param string $host storage host, no ending slash
 * @param string $bucket default bucket
 * @return $this object
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');

// do something...
?>
```

## 4.2. 设置 AccessKey 和 AccessSecret

接口定义:

```
/**
 * set access_key and access_secret
 * @param string $access_key
 * @param string $access_secret
 * @return $this object
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

// do something...

// you can also retrieve access_key and access_secret by
// $access_key = $storage->get_access_key();
// $access_secret = $storage->get_access_secret();
?>
```

## 4.3. 设置默认 Bucket

接口定义:

```
/**
 * set default bucket
 * @param string $name bucket's name
 * @return $this object
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');
```

```
// set default bucket for following ops
$storage->set_bucket('your-bucket-name');

// do something...
?>
```

## 4.4. 设置请求头部 (Header)

接口定义:

```
/**
 * set http request header fields
 * @param string $field http header field
 * @param string $value http header field value
 * @return $this object
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

$storage->set_header('header-field1', 'header-field1-value');
$storage->set_header('header-field2', 'header-field2-value');

// do something...

// you can also remove a header by
//$storage->remove_header('header-field1');
?>
```

## 4.5. 获取 HTTP 响应代码 (HTTP Status Code)

接口定义:

```
/**
 * get response code
 * @param void
```

```
* @return integer  
*/
```

代码示例:

```
<?php  
require_once dirname(__FILE__) . '/GrandCloudStorage.php';  
  
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');  
$storage->set_key_secret('your-access-key', 'your-access-secret');  
  
try {  
    $storage->put_bucket('your-new-bucket-name');  
    $response_code = $storage->get_response_code();  
  
    // do something...  
} catch(Exception $e) {  
    // error stuff  
    //$errno = $e->getCode(); // $errno ==  
$storage->get_response_code()  
    //$error = $e->getMessage();  
}  
?>
```

## 4.6. 获取 HTTP 响应头部 (Header)

接口定义:

```
/**  
 * get response header  
 * @param void  
 * @return string  
 */
```

代码示例:

```
<?php  
require_once dirname(__FILE__) . '/GrandCloudStorage.php';  
  
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');  
$storage->set_key_secret('your-access-key', 'your-access-secret');  
  
try {
```

```
$storage->put_bucket('your-new-bucket-name');  
$response_header = $storage->get_response_header();  
  
// do something...  
} catch(Exception $e) {  
    // error stuff  
    //$errno = $e->getCode();  
    //$error = $e->getMessage();  
}  
?>
```

## 4.7. 获取 HTTP 响应体长度 (Content-Length)

接口定义:

```
/**  
 * get response content length  
 * @param void  
 * @return integer  
 */
```

代码示例:

```
<?php  
require_once dirname(__FILE__).'/GrandCloudStorage.php';  
  
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');  
$storage->set_key_secret('your-access-key', 'your-access-secret');  
  
try {  
    $storage->get_allbuckets();  
    $response_length = $storage->get_response_length();  
  
    // do something...  
} catch(Exception $e) {  
    // error stuff  
    //$errno = $e->getCode();  
    //$error = $e->getMessage();  
}  
?>
```

**注意:** 有些请求不存在响应体, 此时响应体长度未定义!



## 4.8. 获取 HTTP 响应体 (Body)

接口定义:

```
/**
 * get response content
 * @param void
 * @return string
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $storage->get_allbuckets();
    $response_body = $storage->get_response_body();

    // do something...
} catch(Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}

?>
```

*注意: 有些请求不存在响应体!*

## 4.9. 获取错误

接口定义:

```
/**
 * get last error message
 * @param void
 * @return string
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';
```

```
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $storage->put_bucket('your-new-bucket-name');
    $error = $storage->get_last_error();

    // you can also get last curl error if any by
    //$curl_error = $storage->get_curl_error();

    // do something...
} catch (Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage(); // $error ==
    $storage->get_last_error()
}
?>
```

## 4.10. 获取所有 Bucket

接口定义:

```
/**
 * get all buckets
 * @param void
 * @return GCBucket objects list
 * @exception throw exception when response invalid
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $buckets = $storage->get_allbuckets();

    foreach ($buckets as $bucket) {
```

```
// code stuff
// $bucket_idc = $bucket->get_idc();
// $bucket_name = $bucket->get_name();
// $bucket_ctime = $bucket->get_ctime();

// you can also use array style by
// $arr = $bucket->to_array(); // array('idc'=>', 'name'=>',
'ctime'=>')
}

// do something...
} catch(Exception $e) {
    // error stuff
    // $errno = $e->getCode();
    // $error = $e->getMessage();
}
?>
```

## 4. 11. 新建 Bucket

接口定义:

```
/**
 * create new bucket
 * @param string $name bucket's name to create
 * @return true if success
 * @exception throw exception when bucket conflicted or response invalid
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $bucket_name = 'your-new-bucket-name';
    $result = $storage->put_bucket($bucket_name);

    // do something...
} catch(Exception $e) {
```

```
// error stuff
//$errno = $e->getCode();
//$error = $e->getMessage();
}
?>
```

## 4. 12. 删除指定 Bucket

接口定义:

```
/**
 * delete specified bucket
 * @param string $name bucket's name to delete
 * @return true on success, false others
 * @exception throw exception when bucket is not empty or response
invalid
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__).'/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $bucket_name = 'your-new-bucket-name';
    $result = $storage->delete_bucket($bucket_name);

    // do something...
} catch (Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}
?>
```

## 4. 13. 列出指定 Bucket 下所有 Objects

接口定义:

```
/**
 * get all objects of specified bucket
 * @param string $bucket bucket's name
 * @param integer $maxkeys max response objects number of per-request
 * @param string $marker response objects offset
 * @param string $prefix response objects name filter
 * @return GCEntity object
 * @exception throw exception when response invalid
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

try {
    $bucket_name = 'your-new-bucket-name';

    // get first 10 objects of $bucket_name
    $entity = $storage->get_allobjects($bucket_name, 10);
    $objects = $entity->get_objectarray();
    if (empty($objects)) {
        // empty stuff
        //echo "No object in {$bucket_name} bucket";
        //exit;
    }

    // if there are more than 10 objects in bucket,
    // here is how to get following 10 objects code stuff
    //if ($entity->get_istruncated()) {
    // $tmpentity = $storage->get_allobjects($bucket_name, 10,
    $entity->get_marker());
    // $tmpobjects = $tmpentity->get_objectarray();
    //}

    foreach ($objects as $object) {
        // code stuff
        //$object_key = $object->get_key();
        //$object_size = $object->get_size();
        //$object_lastmodified = $object->get_last_modified();
    }
}
```

```
// $object_etag = $object->get_etag();

// you can also use array style by
// $arr = $object->to_array(); // array('key'=>', 'size'=>',
'last_modified'=>', 'etag'=>')
}

// do something...
} catch (Exception $e) {
    // error stuff
    // $errno = $e->getCode();
    // $error = $e->getMessage();
}
?>
```

## 4. 14. 新建 Object

接口定义:

```
/**
 * put object to storage
 * @param string $name object's name
 * @param string $localfile local file path, /path/to/filename.ext
 * @param string $meta see make_request()
 * @param string $type see make_request()
 * @param string $md5 see make_request()
 * @return true on success
 * @exception throw exception when response invalid
 */
```

代码示例:

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

// set default bucket for following ops
$bucket_name = 'your-bucket-name';
$storage->set_bucket($bucket_name);

try {
```

```
$result = $storage->put_object('your-new-object-name.ext',  
    '/path/to/your/file.ext');  
  
    // do something...  
} catch(Exception $e) {  
    // error stuff  
    //$errno = $e->getCode();  
    //$error = $e->getMessage();  
}  
?>
```

注意：调用此方法前必须先调用 `set_bucket()` 设置默认 bucket。

注意：如果 Bucket 下已存在 `$name` 对象，此操作将会更新云存储中的对象（Object Data）。

注意：如果新建 Object 大小太大，此接口调用可能会由于超时意外退出，请正确设置脚本超时时间（`set_time_limit()`）。

## 4.15. 获取 Object

接口定义：

```
/**  
 * get object from storage  
 * @param string $name object's name  
 * @param string $localfile write to local file  
 * @return true on success  
 * @exception throw exception when response invalid  
 */
```

代码示例：

```
<?php  
require_once dirname(__FILE__).'/GrandCloudStorage.php';  
  
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');  
$storage->set_key_secret('your-access-key', 'your-access-secret');  
  
// set default bucket for following ops  
$bucket_name = 'your-bucket-name';  
$storage->set_bucket($bucket_name);  
  
try {  
    $result = $storage->get_object('your-object-name.ext',  
        '/tmp/tmp.ext');
```

```
// do something...  
} catch(Exception $e) {  
    // error stuff  
    //$errno = $e->getCode();  
    //$error = $e->getMessage();  
}  
?>
```

注意：调用此方法前必须先调用 `set_bucket()` 设置默认 bucket。

注意：如果 `$localfile` 已存在，此操作将会覆盖本地文件。

注意：如果 Object 大小太大，此接口调用可能会由于超时意外退出，请正确设置脚本超时时间 (`set_time_limit()`)。

## 4. 16. 获取 Object 的 meta 信息

接口定义：

```
/**  
 * get object's meta  
 * @param string $name object's name  
 * @return array('name'=>', 'meta'=>array(...), 'size'=>') when  
 success  
 * @exception throw exception when response invalid  
 */
```

代码示例：

```
<?php  
require_once dirname(__FILE__) . '/GrandCloudStorage.php';  
  
$storage = new GrandCloudStorage('http://storage.grandcloud.cn');  
$storage->set_key_secret('your-access-key', 'your-access-secret');  
  
// set default bucket for follow ops  
$bucket_name = 'your-bucket-name';  
$storage->set_bucket($bucket_name);  
  
try {  
    $result = $storage->head_object('your-object-name.ext');  
  
    // do something...  
    //echo "File size: {$result}";  
} catch(Exception $e) {
```



```
// error stuff
//$errno = $e->getCode();
//$error = $e->getMessage();
}
?>
```

注意：调用此方法前必须先调用 `set_bucket()` 设置默认 bucket。

## 4. 17. 删除 Object

接口定义：

```
/**
 * delete object from storage
 * @param string $name object's name
 * @return true on success
 * @exception throw exception when response invalid
 */
```

代码示例：

```
<?php
require_once dirname(__FILE__) . '/GrandCloudStorage.php';

$storage = new GrandCloudStorage('http://storage.grandcloud.cn');
$storage->set_key_secret('your-access-key', 'your-access-secret');

// set default bucket for follow ops
$bucket_name = 'your-bucket-name';
$storage->set_bucket($bucket_name);

try {
    $result = $storage->delete_object('your-object-name.ext');

    // do something...
} catch(Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}
?>
```

注意：调用此方法前必须先调用 `set_bucket()` 设置默认 bucket。

如果在使用中遇到任何问题，请到

<http://forum.grandcloud.cn/>反馈，我们将及时跟进。

谢谢！