

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.	通用接口.....	- 5 -
4.2.	Service 相关接口.....	- 11 -
4.3.	Bucket 相关接口	- 12 -
4.4.	Object 相关接口	- 20 -

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，其中包括所有错误码、GCBucket 对象、GCOBJECT 对象、GCEntity 对象、GCMIME 对象和 GCError 对象的定义。通常你不需要关注 GC*对象，他们是对盛大云存储数据结构定义的内部封装。

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");

// delete bucket
$client->delete_bucket($bucket_name);
?>
```

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;
- **GCUUploadPart**: 每个 GCUUploadPart 表示一个盛大云存储的 Upload Part 对象
- **GCMultipartUpload**: 每个 GCMultipartUpload 表示一个盛大云存储的 Multipart Upload 对象
- **GCMultipartUploadEntity**: 每个 GCMultipartUploadEntity 表示一个盛大云存储指定 Bucket 下的 Multipart Upload 信息, 包 Multipart Upload 列表的集合。
- **GCPartsEntity**: 每个 GCPartsEntity 表示一个盛大云存储上指定 Multipart Upload 及其包含的 Part 列表信息。

PHP SDK 定义了一个通用的异常对象：

- **GCErrror**: 每个请求对应的非 20x 响应 (http status code 不在 200~299 之间) 都会转换成一个 GCErrror 异常对象并抛出;

PHP SDK 定义了一个文件类型辅助对象：

- **GCMIME**: 你可以调用 `GCMIME::get_type($file_extension)` 根据文件扩展名获取其对应的类型;

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态抛出!

4.1. 通用接口

4.1.1. 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...
?>
```

4.1.2. 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();  
?>
```

4.1.3. 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...  
?>
```

4.1.4. 设置默认 Bucket 公开访问域名

接口定义:

```
/**  
 * set default bucket cname  
 * @param string $cname bucket's cname  
 * @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 cname for public access  
$storage->set_bucket_cname('http://storage.sdcloud.cn');
```

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

4.1.5. 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');  
?>
```

4.1.6. 设置请求体 (Body)

接口定义:

```
/**  
 * set http request body  
 * @param string $content http request body  
 * @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_body('name=spring&age=28');

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

4.1.7. 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();
}

?>
```

4.1.8. 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();  
}  
?>
```

4.1.9. 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！

4.1.10. 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！

4.1.11. 获取 CURL 错误

接口定义：

```
/**
```

```
* get last curl 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();
}

?>
```

4.2. Service 相关接口

4.2.1. 获取所有 Bucket

该接口对应于 API 中的 GET Service, 可以通过该接口获得用户的所有 Bucket 信息
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. 3. Bucket 相关接口

4.3.1. 获取 Bucket meta 信息

目前该接口对应于 API 中的 GET Bucket Location, 通过该请求可以获取目标 Bucket 所在的区域 (Region) 信息

接口定义:

```
/**
 * get bucket's metas, now only return idc
 * @param string $bucket bucket's name
 * @return string bucket's idc info
 * @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-bucket-name';
    $bucket_local = $storage->head_bucket($bucket_name);

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

?>
```

4.3.2.新建 Bucket

该接口对应于盛大云存储 API 中的 PUT Bucket 接口，该接口可以创建一个新的 Bucket 接口定义：

```
/**
 * create new bucket
 * @param string $name bucket's name to create
 * @param string $local bucket's idc, huabei-1 or huadong-1, default
to huabei-1
 * @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';
    $bucket_local = 'huabei-1';
    $result = $storage->put_bucket($bucket_name, $bucket_local);

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

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

4.3.3. 删除指定 Bucket

该接口对应于盛大云存储 API 中的 DELETE Bucket 接口，可通过该接口删除指定的 Bucket，注意：必须确保要删除的 Bucket 中没有任何数据。

接口定义：

```
/**
 * delete specified bucket
 * @param string $name bucket's name to delete
 * @return true on success
 * @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-bucket-name';
    $result = $storage->delete_bucket($bucket_name);

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

4.3.4. 获取 Bucket Policy 信息

该接口对应于盛大云存储 API 的 GET Bucket Policy 接口，该接口用于获取指定 Bucket 的 Policy。

接口定义：

```
/**
```

```
* get bucket policy
* @param string $bucket bucket's name
* @return string bucket's policy json
* @exception see GCErrror
*/
```

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-bucket-name';
    $bucket_policy = $storage->get_bucket_policy($bucket_name);

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

?>
```

4.3.5. 设置 Bucket Policy

该接口对应于盛大云存储 API 中的 PUT Bucket Policy 接口, 该接口通过 policy 子资源来增加或替换指定的 Bucket 的 Policy。如果该 Bucket 已经存在了 Policy, 那么该操作会替换原有的 Policy。

接口定义:

```
/**
 * put bucket policy to storage
 * @param string $bucket bucket's name
 * @param array $policy bucket policy config
 * @return true on success
 * @exception see GCErrror
 */
```

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-bucket-name';
    $bucket_resource = "{$bucket_name}/*";
    $bucket_policy = array(
        array( // 允许匿名用户访问 $bucket_name 下所有对象
            'Sid' => 'public-get-object',
            'Effect' => 'Allow',
            'Principal' => array(
                'SNDA' => '*'
            ),
            'Action' => 'storage:GetObject',
            'Resource' => "srn:snnda:storage:::{$bucket_resource}"
        )
    );
    $result = $storage->put_bucket_policy($bucket_name,
    $bucket_policy);

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

?>
```

4.3.6. 删除 Bucket Policy

该接口对应于盛大云存储 API 中的 Delete Bucket Policy，用户可以通过该操作删除指定的 Bucket 的 policy。

接口定义：

```
/**
 * delete bucket policy
 * @param $bucket bucket's name
 * @return true on success
 * @exception see GCErrror
 */
```

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-bucket-name';
    $result = $storage->delete_bucket_policy($bucket_name);

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

?>
```

4.3.7. 列出指定 Bucket 下所有 Objects

该接口对应于盛大云存储 API 的 Get Bucket 接口，通过该接口可以获得指定 Bucket 中的 Object 信息列表，请求时可以通过一些查询条件来限制返回的结果。

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();
}
?>
```

4.3.8. 列出指定 Bucket 下的所有未完成的 Multipart Upload

该接口对应于盛大云存储 API 的 List Multipart Upload 接口，通过该接口可列出指定 Bucket 下的所有未完成的 Multipart Upload。可通过设置查询字符来限制返回结果。

接口的定义：

```
/**
 * Get all multipart (List Multipart Upload)
```

```
* @param string $bucket, your bucketname
* @param string $key_marker,the key to start with
* @param string $upload_id_marker,the uploadid to start with
* @param int $max_uploads,the maximum number of keys returned in the
response body
* @param string $prefix,the prefix parameter to the key of the multipart
upload you want to retrieve
* @param char $delimiter,the param you use to group keys
* @return GCMultipartUploadEntity object 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');

try {
    $bucket_name = 'your-new-bucket-name';
    $maxuploads = 1000;
    $keymarker = '';
    $uploadidmarker = '';
    $delimiter = '';
    $prefix = '';

    $result = $storage->get_all_multipart_upload($bucket, $keymarker,
        $uploadidmarker, $maxuploads, $prefix,$delimiter);
    $multipart_uploads = $result->get_object();

    foreach($multipart_uploads as $multipart_upload) {
        print_r($multipart_upload->to_array());
        // Do something..
    }

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

4. 4. Object 相关接口

4.4.1.新建 Object

该接口对应于盛大云存储 API 中的 PUT OBJECT 接口，该接口用来上传一个新的 Object 到指定的 Bucket 中，数据的最大长度限制为 5GB。

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())。

4.4.2. 获取 (下载) Object

该接口对应盛大云存储 API 的 GET OBJECT 接口, 可通过该接口获取指定 Object 内容
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())。

4.4.3. 获取 Object 资源链接 URI

该接口用于获取 Object 的资源链接 URI，可通过设置参数 `expire` 设定链接的过期时间。

接口定义：

```
/**
 * get object resource from storage
 * @param string $name object's name
 * @param integer $expire expire of resource
 * @return resource on success
 */
```

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_resource('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。

注意：如果未设置 bucket cname 则此方法只返回 uri path。

4.4.4. 拷贝 Object

该接口对应于盛大云存储 API 的 PUT Object-Copy 接口，通过该接口可以将一个盛大云存储上已存在的 Object 拷贝至指定 Bucket 下。可通过对应参数的设置定制新 Object 的属性。

接口定义：

```
/**
 * Copy Object(corresponds to "PUT Object - Copy" in API)
```

```
* @param $sbucket,name of source bucket
* @param $skey,name of source object
* @param $dbucket,name of destination bucket
* @param $dkey,name of destination object
* @param $content_meta,fields will be sended as request headers,
*           like x-snda-meta-XXXX or those headers do not necessary
* return new object info on success
* @exception see GCError
*/
```

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
$sbucket = 'your-source-bucket-name';
$skey = 'your-source-key-name';
$dbucket = 'your-destination-bucket-name';
$dkey = 'your-destination-key-name';
try {
    $result = $storage->copy_object($sbucket,$skey,$dbucket,$dkey);;

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

4.4.5. 获取 Object 的 meta 信息

该接口对应于盛大云存储 API 的 HEAD Object 接口, 通过该接口可以获取指定 Object 的元数据信息。

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。

4.4.6. 删除 Object

该接口对应于盛大云存储 API 的 DELETE Object 接口, 用于删除指定的 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。

4.4.7. 初始化一个 Multipart Upload

该接口对应于盛大云存储 API 的 Initiate Multipart Upload，用于初始化一个 Multipart Upload。
接口定义：

```
/**
 * Initiate multipart upload
 * @param string $bucket, bucket's name
 * @param string $key, object's name
 * @param string $content_meta see make_request()
 * @param string $content_type see make_request()
 * @param string $content_md5 see make_request()
 * @return $array, multipart upload info 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');
```

```
try {
    $bucket_name = 'your-bucket-name';
    $key = "your key name";
    /**添加自定义META信息*/
    $object_meta = 'x-snda-meta-project: grand cloud storage,
                    x-snda-meta-user: demo, x-snda-meta-user: guest';
    $result = $storage->initiate_multipart_upload($bucket_name,
    $key, $object_meta);
    printf("uploadId:{$result['UploadId']}\n");
    //do something...
} catch (Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}
?>
```

4.4.8. 上传一个 Part

该接口对应于盛大云存储 API 的 Upload Part 接口，用来上传一个 Part 到指定的 MultipartUpload 中。

接口定义：

```
/**
 * upload part to storage
 * @param string $bucketname bucket's name
 * @param string $key object's name
 * @param string $uploadid multipart upload's id
 * @param int $partnumber of this part
 * @param string $source local file path(/path/to/filename.ext) or
stream
 * @param long $contentlength the length of this content
 * @param string $content_md5 see make_request()
 * @return true on success
 * @exception throw exception when failed
 */
```

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-bucket-name';
    $key = "your key name";
    $filepath = dirname(__FILE__) . '/logo.jpg';
    $uploadid = "EZ05MD008D1DIYVEIO8BUUW44";
    $part_number = 1;
    $storage->upload_part($bucket_name, $key, $uploadid,
        $part_number, $filepath);
    //do something..
} catch(Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}
?>
```

4.4.9.拷贝上传一个 Part

该接口对应于盛大云存储 API 中的 Upload Part – copy,通过该接口可完成将一个云存储中已存在的 Object 拷贝给指定的 Multipart upload part 的一个 part,可通过对应参数的设置修改该 part 的属性。

接口定义:

```
/**
 * Upload Part - Copy (corresponds to "Upload Part - Copy" in API)
 * @param $sbucket,name of source bucket
 * @param $skey,name of source object
 * @param $dbucket,name of destination bucket
 * @param $dkey,name of destination object
 * @param $uploadid,uploadid of the multipart upload
 * @param $partnumber,partnumber of the part to create
 * @param $content_meta,fileds will be sended as request headers,
 *         like x-snda-meta-XXXX or those headers do not necessary
 * return part info on success
 * @exception see GCError
 */
```

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 = 'your-source-bucket-name';
$key = 'your-source-key-name';
$dbucket = 'your-destination-bucket-name';
$dkey = 'your-destination-key-name';
$uploadid = 'your-uploadid';
$part_number = 'your-part_number';

try {
    $result = $storage->upload_part_copy(
        $bucket, $key, $dbucket, $dkey, $uploadid, $part_number);

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

?>
```

4.4.10. 完成 Multipart upload

该接口对应于盛大云存储 API 中的 Complete Multipart Upload 接口，通过该接口可完成一个 Multipart Upload,合并其包含的所有 Part。

接口定义：

```
/**
 * complete multipartupload
 * @param string $bucket
 * @param string $key
 * @param string $uploadid
 * @param string $complete_xml,if is '',then we will get it by
list_parts($uploadid)
 * @throws Exception when failed
 * @return response body
 */
```

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-bucket-name';
    $key = "your key name";
    $filepath = dirname(__FILE__) . '/logo.jpg';
    $uploadid = "EZ05MD008D1DIYVEIO8BUUW44";
    $result = $storage->complete_multipartupload($bucket_name, $key,
        $uploadid);
    //do something..
} catch(Exception $e) {
    // error stuff
    //$errno = $e->getCode();
    //$error = $e->getMessage();
}

?>
```

4.4.11. 终止 Multipart Upload

该接口对应于盛大云存储 API 中的 Abort Multipart Upload 接口, 可通过该接口终止一个指定的 Multipart Upload

接口定义:

```
/**
 * abort multipart upload
 * @param string $bucket, bucket's name
 * @param string $key, object's name
 * @param string $uploadId, the uploadid of the multipart upload
 * @throws Exception when failed
 * @true on success
 */
```

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-bucket-name';
    $key = "your key name";
    $uploadid = "EZ05MD008D1DIYVEIO8BUUW44";
    $result = $storage->abortMultipartUpload($bucket_name, $key,
        $uploadid);
}
```

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

4.4.12. 列出 Multipart Upload 已上传的 Part

该接口对应于盛大云存储 API 中的 List Part 接口。通过该接口可获得指定 Multipart Upload 中已上传的 Part 信息。

接口定义:

```
/**  
 * list parts  
 * @param string,$bucket,bucket's name  
 * @param string,$key,object's name  
 * @param string $uploadId,the uploadid of the multipart upload  
 * @param int $max_parts, the maximum number of parts returned in the  
response body  
 * @param string $part_number_marker,the part to start with  
 * @return GCPartsEntity on success  
 * @exception,throw exception when failed  
 */
```

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-bucket-name';  
    $key = "your key name";  
    $uploadid = "EZ05MD008D1DIYVEIO8BUUW44";  
    $result = $storage->list_parts($bucket_name, $key,  
                                   $uploadid);  
    print_r($result->to_array());  
    //do something..  
} catch(Exception $e) {  
    // error stuff
```

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

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！