

又拍云存储 API 操作文档 v1.5

章节

1. API 使用简介

2. HTTP REST API 接口

API 认证方式
接口支持的 API 操作方法
上传文件
下载文件
删除文件
创建目录
删除目录
获取目录文件列表
获取空间使用情况
获取文件信息

图片类空间特别规定
附录: 错误代码表

3. FTP 接口

4. 表单 API 接口

附录: 错误代码表

一、API 使用简介

又拍云存储, 用户可根据需要使用 FTP 或 API 进行操作。建议使用我们提供的 SDK 进行 API 开发, SDK 下载页面:

<https://www.upyun.com/intro/custom.php>

-----API 帐号信息-----

以下为示例使用的帐号信息, 使用时请用实际的帐号信息代替。

访问域名:	<code>http://static.demo.com</code>	(使用用户绑定的独立域名)
Api 接口地址:	<code>http://v1.api.upyun.com</code>	(电信)
	<code>http://v2.api.upyun.com</code>	(联通网通)
	<code>http://v3.api.upyun.com</code>	(移动铁通)
	<code>http://v0.api.upyun.com</code>	(自动判断)

API 接口同时支持 HTTPS 操作

Bucketname:	<code>demobucket</code>	(空间名称)
Username:	<code>demouser</code>	(操作员用户名, 非管理帐号。 操作员帐号可在 空间管理后台 添加)
Userpass:	<code>demopass</code>	(操作员密码)

为加快你的开发效率, 可使用我们提供的 [SDK 开发包](#) 进行二次开发, 更简单快速。

二、HTTP REST API 接口

a. 认证方式:

1. HTTP Basic Auth 验证

采用标准 http basic auth 验证方式。

请求方法:

```
> GET /demobucket/ HTTP/1.1
> Host: v0.api.upyun.com
> Authorization: Basic YWFhOmFw
```

代码示例(PHP):

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/');
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
```

```
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

2、签名验证

1) 签名格式:

Authorization: UpYun demouser:signature

其中:

UpYun 为固定字符, 与 username 之间以空格间隔
demouser 空间的操作用户帐号
signature 签名内容

2) 签名内容生成

`md5(METHOD&URI&DATE&CONTENT_LENGTH&md5(PASSWORD))`

各组成内容介绍:

&	字符串连接符
METHOD	本次请求的方式
URI	URI, 必须 http 协议标准 (中文文件名或目录需进行 rawurlencode 编码 RFC1738)
DATE	日期格式: Wed, 24 Aug 2011 09:13:05 GMT (RFC 1123)格式 请注意服务器时间必须跟世界时间一致, 否则签名失败, 并返回 Date offset error 错误。
CONTENT_LENGTH content	长度, 需与请求 header 中 Content-Length 一致 (GET 方法, 长度为 0)
md5 (PASSWORD)	帐号密码进行 md5 加密

请求方法:

```
> GET /demobucket/ HTTP/1.1
> Host: v0.api.upyun.com
> Authorization: UpYun demouser:62c428533830d84fd8bc77bf402512fc
> Date: Wed, 24 Aug 2011 07:33:47 GMT
```

代码示例 (PHP):

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/');
$date = gmdate('D, d M Y H:i:s \G\M\T');
$sign = md5('GET&/demobucket/&'.$date.'&0&'.md5(userpass));
curl_setopt($process, CURLOPT_HTTPHEADER, array(
    'Expect:', "Authorization: UpYun demouser:".$sign));
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
```

b、HTTP REST 接口支持的 API 操作方法

上传文件 (2012-02 加入自动创建父级目录功能)
 下载文件
 删除文件
 创建目录 (2012-02 加入自动创建父级目录功能)
 删除目录
 获取目录文件列表
 获取空间使用情况
 获取文件信息 (2012-02 新增)
 上传图片做缩略图、加水印功能 (2012-05 新增)

上传文件：**HTTP PUT**

上传文件操作。

请求方法：

```
> PUT /demobucket/a.txt HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com

> Content-Type: image/jpg      /// 可选
/// 设置待上传文件的 Content-Type 类型 (如不设置, 又拍云存储将使用存储的文件名中扩展名
进行自动设置)

> Content-MD5: cf3e1ce231e15a9e3a42ff688074a628      /// 可选
/// 设置待上传文件的 Content-MD5 值 (如又拍云服务端收到的文件 MD5 值与用户设置的不一致,
将回报 406 Not Acceptable 错误)

> Content-Secret: bac          /// 可选
/// 设置待上传文件的 访问密钥 (注意: 仅支持图片空间!, 设置密钥后, 无法根据原文件 URL 直接访问,
需带 URL 后面加上 (缩略图间隔标志符+密钥) 进行访问)
如缩略图间隔标志符为 !, 密钥为 bac, 上传文件路径为 /folder/test.jpg,
那么该图片的对外访问地址为: http://空间域名/folder/test.jpg!bac

/// 如果需要把原图进行缩小存储, 可使用以下 header 参数进行设置 (注意: 使用该功能将丢弃原图)
> x-gmkerl-thumbnail: 缩略图版本名称      /// 需图片空间
/// 如文件类空间要使用原图缩小保存功能, 必须设置 x-gmkerl-type 和 x-gmkerl-value
两个参数
如图片类空间还要在原缩略图版本设置的基础上进行调整, 也可使用以下参数

> x-gmkerl-type: fix_width      ///
(缩略图类型有:
    fix_max                    限定最长边, 短边自适应      像素值 例: 150
    fix_width_or_height        限定宽度和高度          像素值 例: 150x130
    fix_width                   限定宽度, 高度自适应      像素值 例: 150
    fix_height                  限定高度, 宽度自适应      像素值 例: 150
    square                      方块图, 固定高固定宽      像素值 例: 150
    fix_both                    固定宽度和高度          像素值 例: 150x130)

> x-gmkerl-value: 150          /// 如需缩小功能, 这必须输入 (缩略图宽度/像素)
> x-gmkerl-quality: 95        /// 可选 (图片压缩质量, 默认 95)
> x-gmkerl-unsharp: True      /// 可选 (是否进行锐化处理, 默认锐化)
```

```
/// 返回信息
> Content-Length: 48
>
> ...文件内容...
```

返回格式:

状态 HTTP 200 OK 表示文件上传成功, 如遇到错误请具体参考返回的 BODY 信息。

对于图片空间上传图片成功后, 还在 Header 包含图片信息如下:

```
x-upyun-width:          (int) 图片宽度
x-upyun-height:         (int) 图片高度
x-upyun-frames:         (int) 图片帧数
x-upyun-file-type:      图片文件类型
```

代码示例(PHP):

```
<?php
$data = 'demodata';
$process = curl_init('http://v0.api.upyun.com/demobucket/a.txt');
curl_setopt($process, CURLOPT_POST, 1);
curl_setopt($process, CURLOPT_POSTFIELDS, $data);
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HTTPHEADER, array('Expect:'));
/// 如果需要自动创建父级目录功能(仅限10级)
/// curl_setopt($process, CURLOPT_HTTPHEADER, array('Expect:', "Mkdir:
true"));
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 60);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($process, CURLOPT_HEADER, 1); /// 获取 header (如需获取图片上传
成功后的信息)
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

*注意:图片空间无法上传普通文件, 否则报 403 错误。

下载文件: **HTTP GET**

读取文件操作。

请求方法:

```
> GET /demobucket/a.txt HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com
```

代码示例(PHP):

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/a.txt');
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
```

```
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

删除文件：
删除文件操作。

请求方法：

```
> DELETE /demobucket/a.txt HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com
```

代码示例(PHP)：

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/a.txt');
curl_setopt($process, CURLOPT_CUSTOMREQUEST, 'DELETE');
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

创建目录：
创建目录操作。

请求方法：

```
> POST /demobucket/demo/ HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com
> Folder: create
```

代码示例(PHP)：

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/demo/');
curl_setopt($process, CURLOPT_POST, 1);
curl_setopt($process, CURLOPT_POSTFIELDS, '');
curl_setopt($process, CURLOPT_HTTPHEADER, array( 'Expect:',
                                                'Folder: true'));
/// 如果需要自动创建父级目录功能(仅限10级)
/// curl_setopt($process, CURLOPT_HTTPHEADER, array('Expect:',
///                                               'Folder: true', 'Mkdir: true'));

curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
```

```
curl_close($process);  
?>
```

删除目录：

删除目录，只允许删除空目录。

请求方法：

```
> DELETE /demobucket/demo/ HTTP/1.1  
> Authorization: Basic YWFhOmFw  
> Host: v0.api.upyun.com
```

代码示例(PHP)：

```
<?php  
$process = curl_init('http://v0.api.upyun.com/demobucket/demo/');  
curl_setopt($process, CURLOPT_CUSTOMREQUEST, 'DELETE');  
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");  
curl_setopt($process, CURLOPT_HEADER, 0);  
curl_setopt($process, CURLOPT_TIMEOUT, 30);  
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);  
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);  
var_dump(curl_exec($process));  
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));  
curl_close($process);  
?>
```

获取目录文件列表：

获取列表。

请求方法：

```
> GET /demobucket/demo/ HTTP/1.1  
> Authorization: Basic YWFhOmFw  
> Host: v0.api.upyun.com
```

返回格式：(以 \t 间隔)

```
filename\ttype\tsize\tmodif  
demo.jpg\tN\t11383\t1306130450  
demo\tF\t14738423\t1306130450  
demo.js\tD\t1383\t1306130450
```

文件/目录名	类型	大小	最后修改时间
demo.jpg	N	11383	1306130450
demo	F	14738423	1306130450
demo.js	N	1383	1306130450

注：类型 N 表示文件、F 表示目录，目录大小表示当前目录下的文件大小（不含子目录）

代码示例(PHP)：

```
<?php  
$process = curl_init('http://v0.api.upyun.com/demobucket/demo/');  
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");  
curl_setopt($process, CURLOPT_HEADER, 0);  
curl_setopt($process, CURLOPT_TIMEOUT, 30);
```

```
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

获取空间使用情况

请求方法：

```
> GET /demobucket/?usage HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com
```

返回格式：

当前空间使用空间大小

代码示例(PHP)：

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/?usage');
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>
```

对于 **GET/PUT/DELETE** 操作，返回状态码 **200 OK** 为操作成功，**403** 为权限错误，**503** 为服务器错误。[详细错误描述请参考代码表](#)

获取文件信息

请求方法：

```
> HEAD /demobucket/test.jpg HTTP/1.1
> Authorization: Basic YWFhOmFw
> Host: v0.api.upyun.com
```

返回格式：

状态 HTTP 200 OK 表示查询的文件或目录存在，文件信息将存放于返回的 Header 中，如：

```
x-upyun-file-type: file
x-upyun-file-size: 159534
x-upyun-file-date: 1328768346
```

代码示例(PHP)：

```
<?php
$process = curl_init('http://v0.api.upyun.com/demobucket/test.jpg');
curl_setopt($process, CURLOPT_USERPWD, "demouser:demopass");
curl_setopt($process, CURLOPT_CUSTOMREQUEST, 'HEAD');
curl_setopt($process, CURLOPT_HEADER, 0);
curl_setopt($process, CURLOPT_TIMEOUT, 30);
curl_setopt($process, CURLOPT_RETURNTRANSFER, 1);
```

```

curl_setopt($process, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($process, CURLOPT_NOBODY, true);
curl_setopt($process, CURLOPT_HEADER, 1); /// 获取 header
var_dump(curl_exec($process));
var_dump(curl_getinfo($process, CURLINFO_HTTP_CODE));
curl_close($process);
?>

```

-----图片类空间特别规定-----

图片类空间支持缩略图调用。

1、缩略图调用路径格式：

```

http://static.demo.com/原图路径+标志符+缩略图版本名称
例： http://static.demo.com/demo.jpg           (原图)
      http://static.demo.com/demo.jpg!200x100.jpg (缩略图)
      http://static.demo.com/demo.jpg!75x75.jpg  (缩略图)

```

2、缩略图间隔标志符：“ ! ” “ _ ” “ - ” 3个中选取一个作为标志符。

特别注意：选择的标志符不得出现在原图路径（文件名、目录名）中，否则会导致文件无法访问的 **404 错误**！

3、图片版本（缩略图）设置：

图片版本名称规范：可用字母、数字和半角点组成，不大于 20 个字符。
（如：75x75.jpg,200x100.jpg,small.jpg,large 等等）

缩略图类型规范

- 1 限定宽度，高度自适应
- 2 限定高度，宽度自适应
- 3 限定最长边，短边自适应
- 4 方块图，固定高固定宽
- 5 固定宽度和高度
- 6 限制宽度和高度

4、使用举例：

版本标志符设置“!”，版本名称使用“200x100.jpg”，版本规范：限定高度 200px

使用方式：

http://static.demo.com/demo.jpg!200x100.jpg

5、图片类空间有以下限制（性能规范，普通类空间中的图片无此限制）

- 1) 图片总像素（宽 x 高 x 帧数）：200000000 像素
- 2) 图片支持格式：jpg/png/gif/bmp/tif

-----出现的返回代码及含义-----

返回代码	含义	HTTP 状态码
OK	操作成功	200
Bad Request	错误请求（如 URL 缺少空间名）	400
Unauthorized	访问未授权	401
Sign error	签名错误（操作员和密码，或签名格式错误）	401
Need Date Header	发起的请求缺少 Date 头信息	401
Date offset error	发起请求的服务器时间错误 请检查服务器时间是否与世界时间一致	401

Not Access	权限错误 (如非图片文件上传到图片空间)	403
Not Found	获取文件或目录不存在或 上传文件或目录时上级目录不存在	404
System Error	系统错误	503
Not Acceptable(path)	目录错误: 创建目录时已存在同名 (目录或文件)或上传文件时存在 同名目录	406
File size too max	单个文件超出大小(100Mb 以内)	403
Not a Picture File	图片类空间错误码, 非图片文件或 图片文件格式错误(针对图片空间 只允许上传 jpg、png、gif、bmp、tif 格式)	403
Picture Size too max	图片类空间错误码, 图片尺寸太大 (针对图片空间, 图片总像素在 200000000 以内)	403
Bucket full	空间已用满	403
Bucket blocked	空间被禁用, 请联系管理员	403
User blocked	操作员被禁用	403

三、FTP 接口规范

云存储支持通过 FTP 方式管理文件, 支持标准 FTP 协议, 实现 FTP 常用操作。

FTP 服务其他规范:

a、UpYun FTP Server

可用操作列表:

- 登陆
- 上传文件
- 上传文件夹
- 创建文件夹
- 新建文件
- 文件改名
- 文件单个移动
- 文件批量移动
- 下载文件
- 下载文件夹
- 删除文件
- 删除文件夹

禁用的操作:

- 文件夹移动
- 文件夹改名

b、命令字符编码:

统一编码: UTF-8

FTP 客户端字符编码需设置成 utf-8

Windows 操作系统推荐使用客户端: FileZilla FtpRush

c、FTP 客户端模式: 被动模式 pasv

d、最大单个文件尺寸: 100Mb

- e、因 FTP 协议把文件传输连接关闭即表示该次传输过程完成，有可能会因网络质量而影响到文件上传的完整性。
如对文件内容一致性要求较高的业务，建议使用 HTTP API 进行上传操作（并可加入文件 MD5 校验的要求）

四、表单 API 上传功能

功能介绍：

表单 API 上传功能是专门为需要让客户网站的用户可直接上传到又拍云存储的功能。

表单 API 功能支持用户端（浏览器、客户端软件、手机 APP 等）直接上传文件到云存储，而不需要通过客户服务器进行中转。

a、浏览器 POST 到服务器内容：

- | | |
|----------------|------------------------------|
| 1、policy | [policy 内容 base64 值(无换行)] |
| 2、signature | md5(policy+&&+表单 API 验证密匙) |
| 3、file | |
| 4、POST 的目标 URL | http://v0.api.upyun.com/空间名/ |

b、policy 内容组成：

- 1、文件 KEY 命名规则：save-key [必须]

如：/filepath/filename

KEY 中可使用命名变量：

- | | |
|------------|---|
| (1)、绝对值： | 用户自定义字符 |
| (2)、时间类： | {year} {mon} {day} {hour} {min} {sec} |
| (3)、md5 类： | {filemd5} 文件内容 md5 值 |
| (4)、随机类： | {random} {random32} 可选 16 及 32 位随机字符及数字 |
| (5)、后缀类： | {suffix} {suffix} 上传文件的原始后缀 |

安全提示： 如果你的应用一次授权只允许上传 1 个文件，save-key 请使用绝对值，
如果你的应用一次授权在过期时间内允许上传多个文件，save-key 可使用一些命名变量。

- | | | |
|-----------|----------------------|--|
| 2、过期时间： | expiration | (UNIX TIME) [必须] (当前上传授权的过期时间) |
| 3、空间名： | bucket | [必须] (用户在又拍云存储的空间名) |
| 4、跳转 url： | return-url | [可选] (当上传完成后，跳转到该 URL) |
| 5、异步 url： | notify-url | [可选] (当上传完成后，云存储服务端主动把结 POST 到该 URL，为保障你的空间安全，建议启用异步回调) |
| 6、文件大小范围： | content-length-range | 格式：最小,最大 单位：Bytes [可选]
如：102400,1024000
(仅允许上传 100Kb 以上 1Mb 以下的文件) |
| 7、图片宽度范围： | image-width-range | 格式：最小,最大 单位：像素 [可选] |
| 图片高度范围： | image-height-range | 格式：最小,最大 单位：像素 [可选] |
| 8、指定文件类型： | Content-Type | [可选] (指定上传该文件后存储在云存储系统上的文件类型，一般系统通过扩展名自动识别) |
| 9、允许文件类型： | allow-file-type | jpg,jpeg,gif,png
[可选] (根据上传的文件名后缀进行判断) |
| 10、文件校验码： | content-md5 | 202cb962ac59075b964b07152d234b70 |

[可选] (根据上传的文件内容进行 md5 校验, 如上传内容的 md5 值不一致, 则该次请求失败)

11、原图访问密钥: `content-secret bac` [可选]
/// 设置待上传文件的 访问密钥 (注意: 仅支持图片空间!, 设置密钥后, 无法根据原文件 URL 直接访问,
/// 需带 URL 后面加上 (缩略图间隔标志符+密钥) 进行访问)
/// 如缩略图间隔标志符为 !, 密钥为 bac, 上传文件路径为 /folder/test.jpg ,
/// 那么该图片的对外访问地址为: `http://空间域名/folder/test.jpg!bac`

/// 如果需要把原图进行缩小存储, 可使用以下 **header** 参数进行设置 (注意: 使用该功能将丢弃原图)

12、缩略图版本名称: `x-gmkerl-thumbnail` 缩略图版本名称 /// 需图片空间
如文件类空间要使用原图缩小保存功能, 必须设置 `x-gmkerl-type` 和 `x-gmkerl-value` 两个参数
如图片类空间还要在原缩略图版本设置的基础上进行调整, 也可使用以下参数

13、缩略图类型:	<code>x-gmkerl-type</code>	<code>fix_width</code>	/// (缩略图类型有:
	<code>fix_max</code>	限定最长边, 短边自适应	像素值 例: 150
	<code>fix_width_or_height</code>	限定宽度和高度	像素值 例: 150x130
	<code>fix_width</code>	限定宽度, 高度自适应	像素值 例: 150
	<code>fix_height</code>	限定高度, 宽度自适应	像素值 例: 150
	<code>square</code>	方块图, 固定高固定宽	像素值 例: 150
	<code>fix_both</code>	固定宽度和高度	像素值 例: 150x130)

14、缩略图参数: `x-gmkerl-value` 150 /// 如需缩小功能, 这必须输入 (缩略图宽度/像素)

15、缩略图质量: `x-gmkerl-quality` 95 /// 可选 (图片压缩质量, 默认 95)

16、图片锐化: `x-gmkerl-unsharp` True /// 可选 (是否进行锐化处理, 默认锐化)

PHP 示例:

```
<?php
$policydoc = array(
    "bucket" => "demobucket", // 空间名
    "expiration" => 1323152967, // 该次授权过期时间
    "save-key" => "/{year}/{mon}/{random}{.suffix}", // 命名规则, /2011/12/随机.扩展名
    "allow-file-type" => "jpg,jpeg,gif,png", // 仅允许上传图片
    "content-length-range" => "0,102400", // 文件在 100K 以下
    "return-url" => "http://localhost/form-
test/return.php", // 回调地址
    "notify-url" => "http://localhost/form-test/notify.php"
);
// 异步回调地址

// json 编码后格式为 (注意不带换行):
/*
    {"bucket":"demobucket","expiration":1323152967,"save-key":"\
{year}\/{mon}\/{random}{.suffix}",
    "allow-file-type":"jpg,jpeg,gif,png","content-length-
range":"0,102400",
    "return-url":"http://localhost/form-test/return.php",
    "notify-url":"http://localhost/form-test/notify.php"}
*/

$policy = base64_encode(json_encode($policydoc));
// 注意 base64 编码后的 policy 字符串中不包含换行符! 否则无法通过验证
$sign = md5($policy.'&'. '表单 API 验证密钥');
// 表单 API 功能的密钥 (请访问又拍云管理后台的空间管理页面获取)
```

c、signature 签名

组成: md5(policy+'&'+表单 API 验证密匙)

/// 表单 API 功能的密匙 (请访问[又拍云管理后台的空间管理页面](#)获取)

一个标准的授权上传如下:

```
<form action="http://v0.api.upyun.com/空间名/" method="post"
```

```
enctype="multipart/form-data">
```

```
<input type="hidden" name="policy"
```

```
value="eyJidWNrZXQiOiJkZWlvIiwidXNlcm5hbWUiOiJlc2VybmFtZSI6ImV4cGlyYXRpb24iOjEzMjM0NTUyMDUyInNhdmUta2V5IjoieXB9eWVhcnlcl3ttb259XC97cmFuZG9tfXsuc3VmZml4fSI6ImFsbG93LWZpbGUtdHlwZSI6ImpwZyxqcGVnLGdpZixwbmciLCJjb250ZW50LWxlbmd0aC1yYW5nZSI6IjAsMTAyNDAwIiwicmV0dXJuLXVyYbCI6Imh0dHA6XC9cL2xvY2FsaG9zdFwvZm9ybS10ZXN0XC9yZXR1cm4ucGhwIiwibm90aWZ5LXVyYbCI6Imh0dHA6XC9cL2xvY2FsaG9zdFwvZm9ybS10ZXN0XC9ub3RpZnkucGhwIn0=">
```

```
<input type="hidden" name="signature"
```

```
value="cf3e1ce231e15a9e3a42ff688074a628">
```

```
<input type="file" name="file">
```

```
<input type="submit">
```

```
</form>
```

d、回调规则

当该次上传完成时, 如有设置 return-url 则自动使用 GET 模式跳转回用户指定的 URL。如:

<http://localhost/form-test/return.php?code=503&message=%E6%8E%88%E6%9D%83%E5%B7%B2%E8%BF%87%E6%9C%9F&url=%2F2011%2F12%2Ffd0e30047f81fa95.bz2&time=1332129461&non-sign=b11cb84538e884d63e14e52d35a7bd21>

回调地址中包括: code、message、url 和 sign(或 non-sign) 四个参数 (如是图片空间, 额外增加: image-width、image-height、image-frames 和 image-type 四个参数, 但不用于加密签名)。

其中 sign(或 non-sign) 是根据 code、message、url 进行加密签名而成

如:

```
md5("${_GET['code']}&${_GET['message']}&${_GET['url']}&${_GET['time']}&".表单 API 验证密匙) == $_GET['sign']
```

对于部分系统错误, 因未取得操作员信息, 会返回 non-sign 的签名。此时需特殊处理为:

```
md5("${_GET['code']}&${_GET['message']}&${_GET['url']}&${_GET['time']}&") == $_GET['non-sign']
```

对于有设置 notify-url 异步回调的情况, 又拍云存储服务端还将通过 POST 方式把上传结果回调到用户所指定的 URL。如:

```
POST http://localhost/form-test/notify.php
```

```
code=503&message=%E6%8E%88%E6%9D%83%E5%B7%B2%E8%BF%87%E6%9C%9F&url=%2F2011%2F12%2Ffd0e30047f81fa95.bz2&time=1332129461&non-sign=b11cb84538e884d63e14e52d35a7bd21
```

其签名规则与上面的 GET 模式一致 :)

如不设置 return-url 将在又拍云存储接受完上传操作后, 把结果信息输出在 body 中。

e、出现的返回代码及含义

返回代码	含义	HTTP 状态码
OK	上传成功	200
Not accept, Miss signature	不接受请求, 缺少签名	403

Not accept, Signature error	签名错误, a: 缺少 bucket、save-key 或 expiration 参数; b: 表单签名密钥错误	403
Not accept, POST URI error	不接受请求, POST 的 URL 错误 URL 中的空间名与表单 POST 的空间名不一致	403
System Error ... Retry again	系统错误, 请再尝试	503
Not accept, Bucket not exists	不接受请求, 空间不存在	400
Authorize has expired	不接受请求, 上传授权已过期 请检查服务器时间是否正确, 或设置的过期时间错误	400
Not accept, Bucket disabled	不接受请求, 空间被禁用	403
Not accept, Form API disabled	不接受请求, 表单 API 功能未打开	403
Not accept, No file data	不接受请求, 没有上传文件数据 请检查 POST 表单中文件域的名称是否为 file	403
Not accept, File too large	不接受请求, 上传文件过大	403
Not accept, File too small	不接受请求, 上传文件过小	403
Not accept, File type Error	不接受请求, 上传文件类型不允许	403
Not accept, Content-md5 error	不接受请求, 上传文件的内容 md5 校验错误	403
Not accept, Not a image file	不接受请求, 上传的不是图片文件	403
Not accept, Image width too small	不接受请求, 上传的图片宽度过小	403
Not accept, Image width too large	不接受请求, 上传的图片宽度过大	403
Not accept, Image height too small	不接受请求, 上传的图片高度过小	403
Not accept, Image height too large	不接受请求, 上传的图片高度过大	403