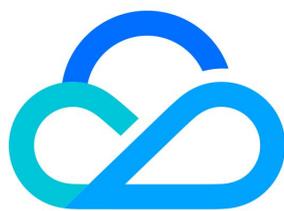


万象优图（旧） 开发者手册 产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

开发者手册

- 签名和鉴权文档

- RESTful API

- 返回码说明

- 版本说明

- OCR识别接口文档

- 人脸识别接口文档

- 图片标签接口文档

- 智能鉴黄接口文档

 - SDK文档

 - SDK文档

 - SDK下载

开发者手册

签名和鉴权文档

最近更新时间：2018-05-28 10:58:31

本文为万象优图V2版本和V2加强版本的鉴权文档，V1的鉴权文档参见[万象优图鉴权及签名文档-V1](#)。

注意：各版本只能使用属于其的CGI，请不要混用。具体版本说明请参照[版本说明](#)。

1 签名与鉴权

腾讯移动服务通过签名来验证请求的合法性。开发者通过将签名授权给客户端，使其具备上传下载及管理指定资源的能力。签名分为多次有效签名和单次有效签名。

多次有效签名：签名中绑定或者不绑定文件fileid，需要设置大于当前时间的有效期，有效期内此签名可多次使用，有效期最长可设置三个月。

单次有效签名：签名中绑定文件fileid，有效期必须设置为0，此签名只可使用一次，且只能应用于被绑定的文件。

具体适用场景参见[签名适用场景](#)。

注意：本页面介绍的签名算法适用于V2版本和V2加强版本，V1版本签名算法参加[万象优图鉴权及签名文档-V1](#)。

2 签名算法

2.1 获取签名所需信息

生成签名所需信息包括项目ID（appid），空间名称（bucket,图片资源的组织管理单元），项目的Secret ID和Secret Key。获取这些信息的方法如下：

- 1) 登录万象优图-图片空间，进入图片空间；
- 2) 如开发者未创建图片空间，可添加图片空间，获取项目ID（appid），空间名称（bucket）；如果开发者已经创建过空间，则可以直接获取项目ID和空间名称（bucket）；
- 3) 登录万象优图-项目设置，进入项目设置；
- 4) 如果开发者未添加密钥，则需添加密钥，获取项目的Secret ID和Secret Key，每个项目最多添加两对密钥；如果已经添加过密钥则直接获取项目的Secret ID和Secret Key。

注：

- (1) 添加图片空间可参考[添加图片空间](#)；
- (2) 添加密钥可参考[添加密钥](#)。

2.2 拼接签名串

拼接多次有效签名串：

```
a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&u=[userid]&f=[fileid]
```

拼接单次有效签名串：

```
a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&u=[userid]&f=[fileid]
```

注意：多次有效签名串中fileid为可选参数，为空不绑定资源，例如上传签名和下载签名（不绑定资源）；不为空，则为绑定资源，例如绑定资源的下载，只能对绑定的资源进行下载。

签名串中各字段含义如下：

字段	解释
a	开发者的项目ID，接入万象优图创建空间时系统生成的唯一标示项目的ID
b	图片空间名称bucket
k	项目的Secret ID
e	签名的有效期，是一个符合UNIX Epoch时间戳规范的数值，单位为秒；单次签名时，e必须设置为0
t	当前时间戳，是一个符合UNIX Epoch时间戳规范的数值，单位为秒，多次签名时，e应大于t
r	随机串，无符号10进制整数，用户需自行生成，最长10位
u	历史遗留字段，请填写为0
f	资源存储的唯一标识，单次签名必填；多次签名选填，如填写则会验证与当前操作的文件路径是否一致。

注：

拼接单次有效签名的签名串时，过期时间e必须设置为0，以保证此签名只能针对固定资源且只能使用一次；

删除和复制文件必须使用单次有效签名；

上传必须使用多次有效签名；

具体适用场景参见[签名适用场景](#)。

2.3 生成签名

1. 万象优图使用 HMAC-SHA1 算法对请求进行加密（SHA1算法加密后的输出必须是原始的二进制数据，否则签名失败）；
2. 签名串需要使用 Base64 编码。

即生成签名的公式如下：

$$\text{SignTmp} = \text{HMAC-SHA1}(\text{SecretKey}, \text{original})$$
$$\text{Sign} = \text{Base64}(\text{SignTmp})$$

其中SecretKey为项目密钥SecretKey，original为2.2节中拼接好的签名串，首先对original使用HMAC-SHA1算法进行签名，然后将original附加到签名结果的末尾，再进行Base64编码，得到最终的sign。

注：此处使用的是标准的Base64编码，不是urIsafe的Base64编码，请注意。

3 实例

本节介绍生成签名的算法实例，实例中使用PHP语言，如果开发者使用其他与开发，请使用对应的算法。

3.1 获取签名所需信息

获取得到的签名所需信息如下。

项目ID：10001290

空间名称（bucket）：tencentyun

Secret ID：AKIDgaoOYh2kOmJfWVdH4IpfXScG2zPLPGoK

Secret Key：nwOKDouy5JctNOInere4gkVoOUz5EYAb

3.2 拼接签名串

拼接的多次有效签名串如下(不绑定资源)：

```
a=10001290&b=tencentyun&k=AKIDgaoOYh2kOmJfWVdH4lpfxScG2zPLPGoK&e=1438669115&t=1436077115&r=11162&u=0&f=
```

拼接的多次有效签名串如下(绑定资源)：

```
a=10001290&b=tencentyun&k=AKIDgaoOYh2kOmJfWVdH4lpfxScG2zPLPGoK&e=1438669115&t=1436077115&r=11162&u=0&f=tencentyunSignTest
```

拼接的单次有效签名串如下：

```
a=10001290&b=tencentyun&k=AKIDgaoOYh2kOmJfWVdH4lpfxScG2zPLPGoK&e=0&t=1436077115&r=11162&u=0&f=tencentyunSignTest
```

```
$appid = "10001290";
$bucket = "tencentyun";
$secret_id = "AKIDgaoOYh2kOmJfWVdH4lpfxScG2zPLPGoK";
$secret_key = "nwOKDouy5JctNOInere4gkVoOUz5EYAb";
$expired = time() + 2592000;
$onceExpired = 0;
$current = time();
$rdm = rand();
$userid = "0";
$fileid = "tencentyunSignTest";

$srcStr = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$expired.'&t='.$current.'&r='.$rdm.'&u='
.$userid.'&f=';

$srcWithFile = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$expired.'&t='.$current.'&r='.$rdm.'&u='
.$userid.'&f='.$fileid;

$srcStrOnce = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$onceExpired.'&t='.$current.'&r='.$rdm
.&u='.$userid.'&f='.$fileid;
```

3.3 生成签名

SHA1算法加密后的输出必须是原始的二进制数据，否则签名失败。

```
$signStr = base64_encode(hash_hmac('SHA1', $srcStr, $secret_key, true).$srcStr);

$srcWithFile = base64_encode(hash_hmac('SHA1', $srcWithFile, $secret_key, true).$srcWithFile);

$signStrOnce = base64_encode(hash_hmac('SHA1', $srcStrOnce, $secret_key, true).$srcStrOnce);

echo $signStr."\n";

echo $srcWithFile."\n";

echo $signStrOnce."\n";
```

最终得到的多次有效签名为(不绑定资源)：

```
L9U0luDidww68urljeoq6Dlid8hhPTEwMDAxMjkwJmI9dGVuY2VudHI1biZrPUFLS
URnYW9PWWgya09tSmZXVmRINGxwZnhTY0cyelBMUEdvSyZlPTE0Mzg2NjkxMTUmdD
0xNDM2MDc3MTE1JnI9MTEwNjImdT0wJmY9
```

最终得到的多次有效签名为(绑定资源)：

```
Pzb65w5vL8tMPVBP0w0fCbww7vRhPTEwMDAxMjkwJmI9dGVuY2VudHI1biZrPUFLS
URnYW9PWWgya09tSmZXVmRINGxwZnhTY0cyelBMUEdvSyZlPTE0Mzg2NjkxMTUmdD
0xNDM2MDc3MTE1JnI9MTEwNjImdT0wJmY9dGVuY2VudHI1blNpZ25UZXR0
```

单次有效签名为：

```
DKWF806udLkHcbQXRp31KBmI8FhPTEwMDAxMjkwJmI9dGVuY2VudHI1biZrPUFLS
URnYW9PWWgya09tSmZXVmRINGxwZnhTY0cyelBMUEdvSyZlPTAmdD0xNDM2MDc3MT
E1JnI9MTEwNjImdT0wJmY9dGVuY2VudHI1blNpZ25UZXR0
```

4 签名适用场景

万象优图对签名的适用场景做了如下限制：

场景	适用签名
查询	不验证签名
下载（无防盗链）	不验证签名
上传	多次有效签名
下载（开启防盗链，绑定资源）	多次有效签名
下载（开启防盗链，不绑定资源）	多次有效签名
复制	单次有效签名
删除	单次有效签名
图片鉴黄	多次有效签名
OCR识别（身份证识别、名片识别）	多次有效签名
人脸识别（人脸检测与分析、五官定位、人脸比对与验证、人脸检索等）	多次有效签名
图片标签	多次有效签名

RESTful API

最近更新时间：2018-05-28 11:00:10

本页面为万象优图V2加强版的Restful API文档。

1. 旧版本（V1和V2版本）的Restful API参见[万象优图Restful API文档-V1](#)和[万象优图Restful API文档-V2](#)。
2. 如果您使用的是万象优图2.0，则APPID是以125为前三位的，请查看[新版文档](#)。

注意：各版本只能使用属于其的CGI，请不要混用。具体版本说明请参照[版本说明](#)。

1 基本概念

概念	解释
appid	接入万象优图创建空间时,系统生成的唯一项目ID,用于唯一标识接入项目, 申请链接: 图片空间
bucket	开发者添加的空间名称, 参见 图片空间
userid	即将废弃字段, 请使用默认值"0"
fileid	资源存储的唯一标识

2 鉴权

腾讯云·万象优图通过签名来验证请求的合法性。开发者通过将签名授权给客户端，使其具备上传下载及管理指定资源的能力。

签名分为单次签名和多次签名，区别为（可参见[签名适用场景](#)）：

如果针对资源进行写操作(资源删除和资源复制)，那么这个签名必须是单次有效的，重复使用该签名则会返回签名失败；

如果是上传一个新的资源，那么这个签名可以是多次有效的。有效时长最多为三个月。

开发者可以通过[服务器SDK文档](#)生成签名，也可以参考我们的签名函数自行生成签名，具体生成方式详见[鉴权签名方法](#)。

3 图片上传

说明：如果您使用的是万象优图2.0，则APPID是以125为前三位的，您需要使用COS的上传接口，请查看[新版文档](#)。

3.1 直接上传

功能: 直接上传单张图片, 只支持POST表单(multipart/form-data)方式, 目前只支持20M以内的图片。

接口: `http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]` (自定义fileid)

接口: `http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/` (由后台自动生成fileid)

其中fileid为可选参数，可以包含除了'\0'外的任意字符，需要对fileid进行urlencode，utf8字符集，最大长度支持128字节，建议控制在64字节内。

注意：userid为即将废弃字段，请使用默认值"0"，不要依赖该字段进行业务判断。

方法: POST

请求参数HTTP头部信息:

参数名称	必选	类型	描述
Host	是	String	图片云服务器域名, 固定为web.image.myqcloud.com
Content-Length	是	Int	整个multipart/form-data内容的总长度, 单位: 字节 (Byte)。
Content-Type	是	String	标准的multipart/form-data 格式, 参见 rfc1867
Authorization	是	String	多次有效签名,用于鉴权, 具体生成方式详见 鉴权签名方法

HTTP包头

参数名称	必选	类型	描述
MagicContext	否	String	业务附加信息,当配置回调时, 腾讯云·万象优图会透传给开发者的服务器.
FileContent	是	Binary	文件内容
Md5	否	String	图片的Md5值,如果提供,服务会对上传的文件做Md5校验及秒传

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息
data	具体查询数据. 内容见下表

data里面字段描述:

字段名称	描述	格式
url	资源url(用于restful api交互, 如查询,复制,删除资源)	<code>http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]</code>
download_url	生成的下载资源 url(用于下载)	<code>http://[bucket]-[appid].image.myqcloud.com/[fileid]</code>
fileid	生成的资源唯一标识符	
info	图片的具体信息, 见下表	

参数名称	必选	类型	描述
MagicContext	否	String	业务附加信息,当配置回调时,腾讯云·万象优图会透传给开发者的服务器.
Sha	是	String	图片的sha1值
Op	是	String	固定填: upload_slice, 与普通上传区分
FileSize	是	Int	图片总大小
Session	否	String	断点续传功能. 上一次未成功传输的session id
Slice_size	是	String	单位:字节. 约定分片大小, 最大值为 3MB. 如果不设定, 自动设置为512KB

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息
data	具体查询数据. 内容见下表

data里面字段描述:

字段名称	必选	类型	描述
session	否	String	唯一标识此文件传输过程的 id. (图片命中秒传则不返回此项)
offset	否	Int	开始传输的文件位移(图片命中秒传则不返回此项)
slice_size	否	Int	分片大小(图片命中秒传则不返回此项)
url	否	String	图片命中秒传或上传完成则返回。 <code>http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]</code>
download_url	否	string	图片命中秒传或上传完成则返回。 <code>http://[bucket]-[appid].image.myqcloud.com/[fileid]</code>
fileid	否	String	图片命中秒传或上传完成则返回. 资源的唯一标识符

参数名称	必选	类型	描述
MagicContext	否	String	业务附加信息,当配置回调时,腾讯云·万象优图会透传给开发者的服务器.
FileContent	是	Binary	文件内容

参数名称	必选	类型	描述
Md5	否	String	图片的Md5值,如果提供,服务会对上传的文件做Md5校验及秒传

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息
data	具体查询数据. 内容见下表

data里面字段描述:

字段名称	描述	格式
url	资源url(用于restful api交互, 如查询,复制,删除资源)	<code>http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]</code>
download_url	生成的下载资源 url(用于下载)	<code>http://[bucket]-[appid].image.myqcloud.com/[fileid]</code>
fileid	生成的资源唯一标识符	
info	图片的具体信息, 见下表	

info里面字段描述 :

数组名称	字段名称	描述
0	height	图片高度
	width	图片宽度

4 图片复制

功能: 将图片复制一份(保留原有图片)。

接口: `http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]/copy`

方法: POST

请求参数HTTP头部信息:

参数名称	必选	类型	描述
Host	是	String	图片云服务器域名, 固定为web.image.myqcloud.com
Authorization	是	String	单次有效签名,用于鉴权, 具体生成方式详见 鉴权签名方法

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息
data	具体查询数据. 内容见下表

data里面字段描述:

字段名称	描述	格式
url	资源url(用于restful api交互, 如查询,复制,删除资源)	http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]
download_url	生成的下载资源 url(用于下载)	http://[bucket]-[appid].image.myqcloud.com/[bucket]-[appid]/[userid]/[fileid]/original

请求示例:

```
POST /photos/v2/10001290/tencentyun/0/tencentyunRestfulAPITest/copy HTTP/1.1
```

Host: web.image.myqcloud.com

Authorization: VRx9mELRjO7ptkovqjE797CTV2RhPTEwMDAxMjkwJmI9dGVuY2VudHI1biZrPUFLSURnYW9PWWgyxwZnhTY0cyelBMUEdvSyZIPTAmdD0xNDM2MDgxNTAyJnI9MjI1MjImdT0mZj10ZW5jZW50eXVuUmVzdGZ1bEFQSVRlc3C

Content-Length: 0

5 图片查询

功能: 查看文件的属性信息, 包含: 文件哈希值、文件大小、上传时间等, 图片和视频返回的信息会有所不同。

接口: `http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]/`

方法: GET

请求参数HTTP头部信息:

参数名称	必选	类型	描述
Host	是	String	图片云服务器域名, 固定为web.image.myqcloud.com

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息
data	具体查询数据. 内容见下表

data里面字段描述:

字段名称	描述	格式
file_url	图片资源的下载 url	http:// [bucket]-[appid].image.myqcloud.com/[bucket]-[appid]/[userid]/[fileid]/original
file_fileid	图片资源的唯一 id	
file_upload_time	图片上传时间	Unix timestamp
file_size	图片的大小	单位: Byte
file_md5	图片的MD5值	16进制的MD5值
photo_width	图片的宽度	
photo_height	图片的长度	

请求示例:

```
GET /photos/v2/10001290/tencentyun/0/tencentyunRestfulAPITest HTTP/1.1
Host: web.image.myqcloud.com
```

6 图片删除

功能: 删除一个图片。

接口: `http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]/del`

方法: POST

请求参数HTTP头部信息:

参数名称	必选	类型	描述
Host	是	String	图片云服务器域名, 固定为 <code>web.image.myqcloud.com</code>
Authorization	是	String	单次有效签名,用于鉴权, 具体生成方式详见 鉴权签名方法

返回的HTTP 状态码:

- 成功: 200
- 失败: 400

返回内容(json):

字段名称	描述
code	服务器错误码, 0为成功
message	服务器返回的信息

请求示例:

```
POST /photos/v2/10001290/tencentyun/0/tencentyunRestfulAPITest/del HTTP/1.1
```

Host: `web.image.myqcloud.com`

Authorization: `CP+70pfzpNnldJoYn3K85DGNUBxhPTEwMDAxMjkwJmI9dGVuY2VudHI1biZrPUFLSURnYW9PWWgwZnhTY0cyelBMUEdvSyZlPTAmdD0xNDM2MDgyMzIzJnI9MjI0NzkmdT0mZj10ZW5jZW50eXVvUmVzdGZ1bEFQSVRlc3Q:`

Content-Length: 0

7 图片下载

图片下载可以是公开下载, 即使用图片的`download_url` 直接访问即可。

请求示例:

```
GET test0706-1000037.image.myqcloud.com/tencentyunRestfulAPITest HTTP/1.1
```

Host: `test0706-1000037.image.myqcloud.com`

如果在控制台上面设置了[空间样式](#)或者[样式下载别名](#), 并且设置了[样式分隔符](#), 则访问样式图片的方式如下:

`download_url+样式分隔符+样式名`。

注: 空间样式和样式下载别名的名字统称样式名。

例如设置了样式名“160x160.jpeg”, 样式分隔符“/”, 则样式图片的访问方式如下:

```
GET v2test-10000812.image.myqcloud.com/tencentyunRestfulAPITest/160x160.jpeg HTTP/1.1
```

Host: `v2test-10000812.image.myqcloud.com/`

若开启了token防盗链，图片下载只能是私密下载，即必须download_url +?sign=[签名]。

请求示例:

```
GET http://test0706-10000037.image.myqcloud.com/951b0e3b-db35-40e2-8c31-ed38dab5ae69?sign=Ea5aPdBMeVm5O
T332nSYh1nqyJhhPTEwMDAwMDM3JmI9dGVzdDA3MDYmaz1BS0IEcG9LQmZNSzdhWWNZTmxxeG5FdFIBMWFqQXFqa
MjE0NzMwJnQ9MTQ0MTg1NDczMCZyPTE0NDE4NTQ3MzAmdT0wJmY9 HTTP/1.1
Host: test0706-10000037.image.myqcloud.com
```

图片下载支持实时压缩等动态处理，详见以下 [8.图像处理](#)

注意：以下为V2加强版的Restful API实时处理接口，如果开发者使用的是V1和V2的接口请参考本文档头部的相应Restful API文档。

8 图像处理

8.1 基本图像处理 (imageView2)

imageView2是腾讯云·万象优图提供了一种使用简单，但功能强大的图像处理接口。开发者根据业务需求，只需在下载url后面附加相应的参数，就可以生成相应的缩略图。

开发者还可以为一组处理参数指定一个别名，具体参考[样式下载别名](#)。

8.1.1 接口形式

注意：请忽略下面代码中的回车。

```
download_url?imageView2/<mode>/w/<Width>/h/<Height>
/format/<Format>
/q/<Quality>
```

8.1.2 参数说明

参数	含义
/0/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边和短边的最大值分别为LongEdge和ShortEdge，进行等比压缩；如果只指定一边，则另一边自适应
/1/w/<Width>/h/<Height>	限定缩略图的宽和高的最小值分别为Width和Height，进行等比压缩，居中裁剪；如果只指定一边，则表示宽高相等的正方形；缩放后的图片的大小为<Width>x<Height>（其中一边多余的部分会被裁剪掉）
/2/w/<Width>/h/<Height>	限定缩略图的宽和高的最大值分别为Width和Height，进行等比压缩；如果只指定一边，则另一边自适应
/3/w/<Width>/h/<Height>	限定缩略图的宽和高的最小值分别为Width和Height，进行等比压缩；如果只指定一边，代表另外一边为同样的值
/4/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边和短边的最小值分别为LongEdge和ShortEdge，进行等比压缩；如果只指定一边，代表另外一边为同样的值
/5/w/<LongEdge>/h/<ShortEdge>	限定缩略图的长边和短边的最大值分别为LongEdge和ShortEdge，进行等比压缩，居中裁剪；如果只指定一边，则表示宽高相等的正方形；同模式1，缩放后其中一边多余的部分会被裁剪掉

参数	含义
/format/<Format>	目标缩略图图片格式，Format可为：jpg, bmp, gif, png, webp, yjpeg等，其中 yjpeg为万象优图针对jpeg格式进行的优化，本质为jpg格式；缺省为原图格式
/q/<Quality>	图片质量，取值范围0-100，默认值为原图质量；取原图质量和指定质量的最小值；<Quality> 后面加！，表示强制使用指定值

示例：

```

http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageView2/0/w/400/h/300
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageView2/1/w/400/h/600/q/85
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageView2/2/w/400/h/600/q/85!
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageView2/3/w/400/format/png
    
```

8.2 高级图像处理 (imageMogr2)

imageMogr2是腾讯云·万象优图为开发者提供的简单而功能强大的高级图像处理接口，包括旋转、裁剪等。

注：裁剪为缩放裁剪，即先将图片进行等比缩放，在进行裁剪。例如：原图为1500x1200，裁剪成600x600，会先将图片缩放为750x600，然后再裁剪成600x600。

8.2.1 接口形式

请忽略一下接口中的回车。

```

download_url?imageMogr2/auto-orient
    /thumbnail/<imageSizeGeometry>
    /strip
    /gravity/<gravityType>
    /crop/<imageSizeAndOffsetGeometry>
    /scrop/<imageSizeAndOffsetGeometry>
    /rotate/<rotateDegree>
    /format/<Format>
    /quality/<Quality>
    /cgif/<FrameNumber>
    /interlace/<Mode>
    
```

8.2.2 参数说明

参数	含义
/auto-orient	根据原图的exif信息自动把图片旋转回正。
/strip	去除不安全代码包括exif信息。
/gravity/<gravityType>	图片处理位置，影响其后的裁剪偏移参数，参见下面九宫格方位，默认值为中间：Center。
/thumbnail/<imageSizeAndOffsetGeometry>	参考下面的缩放操作参数表。
/crop/<imageSizeAndOffsetGeometry>	请参考下面的裁剪参数表，缺省不裁剪。

参数	含义
/scrop/<imageSizeAndOffsetGeometry>	基于人脸识别执行智能裁剪功能。裁剪区域根据人的头像位置自动确定。输出的裁剪后图片大小需要结合宽高参数指定。参考下面的智能裁剪参数表。
/rotate/<rotateDegree>	图片旋转角度，取值范围0-360，默认不旋转。
/format/<Format>	目标缩略图图片格式，Format可为：jpg, bmp, gif, png, webp, yjpeg等，其中yjpeg为万象优图针对jpeg格式进行的优化，本质为jpg格式；缺省为原图格式。
/quality/<Quality>	图片质量，取值范围0-100，默认值为原图质量；取原图质量和指定质量的最小值；<Quality>后面加！，表示强制使用指定值，如：90！。
/cgif/<FrameNumber>	只针对原图为gif格式，对gif图片格式进行的优化，降帧降颜色。分为以下两种情况：FrameNumber=1，则按照默认帧数30处理，如果图片帧数大于该帧数则截取；FrameNumber取值(1,100]，则将图片压缩到指定帧数FrameNumber。
/interlace/<Mode>	输出为渐进式jpg格式。Mode可为0或1,0表示不开启渐进式；1表示开启渐进式。该参数仅在输出图片格式为jpg格式时有效。如果输出非jpg图片格式，会忽略该参数，默认值0。

缩放操作表格：

参数	含义
/thumbnail/!<Scale>p	指定图片的宽高为原图的Scale%
/thumbnail/!<Scale>px	指定图片的宽为原图的Scale%，高度不变
/thumbnail/!x<Scale>p	指定图片的高为原图的Scale%，宽度不变
/thumbnail/<Width>x	指定目标图片宽度为Width，高度等比压缩
/thumbnail/x<Height>	指定目标图片高度为Height，宽度等比压缩
/thumbnail/<LongEdge>x<ShortEdge>	限定缩略图的长边和短边的最大值分别为LongEdge和ShortEdge，进行等比缩放
/thumbnail/! <LongEdge>x<ShortEdge>r	限定缩略图的长边和短边的最小值分别为LongEdge和ShortEdge，进行等比缩放
/thumbnail/<Width>x<Height>!	忽略原图宽高比例，指定图片宽度为Width，高度为Height，强行缩放图片，可能导致目标图片变形
/thumbnail/<Area>@	等比缩放图片，缩放后的图像，总像素数量不超过Area

示例：

```
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/!50p
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/!50px
```

```

http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/!x50p
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/200x
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/200x400!
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/thumbnail/35000@
    
```

九宫格方位图：



裁剪操作表格 (cropSize)：

参数	含义
/crop/<Width>x	指定目标图片宽度为Width，高度不变。Width取值范围为10-16383
/crop/x<Height>	指定目标图片高度为Height，宽度不变。Height取值范围为10-16383
/crop/<Width>x<Height>	指定目标图片宽度为Width，高度为Height。Width和Height取值范围都为10-16383

裁剪操作表格 (scropSize)：

参数	含义
/scrop/<Width>x	指定目标图片宽度为Width，高度不变。Width取值范围为10-16383
/scrop/x<Height>	指定目标图片高度为Height，宽度不变。Height取值范围为10-16383
/scrop/<Width>x<Height>	指定目标图片宽度为Width，高度为Height。Width和Height取值范围都为10-16383

开发者可以指定裁剪的具体偏移位置，如下表所示：

参数	含义
/crop/! {cropSize}a<dx>a<dy>	相对于偏移位置，水平向右偏移dx，同时垂直向下偏移dy。取值范围小于原图宽高即可
/crop/!{cropSize}- <dx>a<dy>	相对于偏移位置，从指定宽度中减去dx，同时垂直向下偏移dy。取值范围小于原图宽高即可
/crop/!{cropSize}a<dx>- <dy>	相对于偏移位置，水平向右偏移dx，同时从指定高度中减去dy。取值范围小于原图宽高即可

参数	含义
/crop/{cropSize}-<dx>-<dy>	相对于偏移位置，从指定宽度中减去dx个像素，同时从指定高度中减去dy个像素。取值范围小于原图宽高即可

例如：

/crop/{cropSize}-<dx>-<dy>: 表示从原图的坐标 (x,y) 为 (20,20) 的位置裁剪600x600的缩略图。

/crop/{cropSize}-20a20: 表示从原图的坐标 (x,y) 为 (0,20) 的位置裁剪580x600的缩略图。

注意：scrop 参数与 crop 参数同时使用，当智能裁剪没有识别到人脸时，会执行普通的裁剪。要求使用的宽高参数一致，否则输出图片宽高是两个宽高参数中的一个。

<http://v2enhance-10000812.image.myqcloud.com/tencentyunRestfulAPITest?imageMogr2/scrop/300x400/crop/300x400>

示例：

```

http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/crop/{cropSize}-<dx>-<dy>/quality/85
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/crop/{cropSize}-20a20/quality/85!
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/format/jpg/interlace/1
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageMogr2/scrop/300x400
    
```

8.3 图片水印 (watermark)

腾讯云·万象优图支持实时图片水印处理功能，目前，水印图片必须指定为已存储于万象优图中的图片。

8.3.1 接口形式

请忽略一下接口中的回车。

```

download_url?watermark/1
    /image/<encodedImageURL>
    /gravity/<gravity>
    /dx/<distanceX>
    /dy/<distanceY>
    
```

8.3.2 参数说明

参数	含义
/image/<encodedImageURL>	水印源图片地址，需要经过URL安全的Base64编码。指定的水印图片必须存在于万象优图中。
/gravity/<gravity>	文字水印位置，九宫格位置，参见8.2.2节的九宫格方位图，默认值SouthEast
/dx/<distanceX>	水平（横轴）边距，单位为像素，缺省值为0
/dy/<distanceY>	垂直（纵轴）边距，单位像素，默认值为0

示例：

```

http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?watermark/1
    /image/aHR0cDovL3Rlbnmd4dW55dW4tMTAwMDQ0ODYuaW1hZ2UubXlxY2xvdWQuY29tL3NodWI5aW5fMi5wbmc=
    
```

8.4 文字水印 (watermark)

腾讯云·万象优图支持实时文字水印功能，即通过在图片URL后面附加参数为图片实时打文字水印。

8.4.1 接口形式

请忽略一下接口中的回车。

```
download_url?watermark/2
  /text/<encodedText>
  /font/<encodedFontName>
  /fontsize/<fontSize>
  /fill/<encodedTextColor>
  /dissolve/<dissolve>
  /gravity/<gravity>
  /dx/<distanceX>
  /dy/<distanceY>
```

8.4.2 参数说明

参数	含义
/text/<encodedText>	水印内容，需要经过URL安全的Base64编码
/font/<encodedFontName>	水印字体，需要经过URL安全的Base64编码，默认值tahoma.ttf。水印字体列表参考 支持字体列表
/fontsize/<fontSize>	水印文字字体大小，单位是：磅，缺省值13
/fill/<encodedTextColor>	字体颜色，缺省为白色，RGB格式，可以是颜色名称（如blue）或者十六进制（如#FF0000），参考 RGB编码表 ，需经过URL安全的Base64编码，默认值#3D3D3D
/dissolve/<dissolve>	文字透明度，取值1-100，默认100（完全不透明）
/gravity/<gravity>	文字水印位置，九宫格位置，参见8.2.2节的九宫格方位图，默认值SouthEast
/dx/<distanceX>	水平（横轴）边距，单位为像素，缺省值为0
/dy/<distanceY>	垂直（纵轴）边距，单位像素，默认值为0

示例：

```
http://v2test-10000812.image.myqcloud.com/tencentyunRestfulAPITest?watermark/2
  /text/6lW-6K6v5LqRwrfkulfosaHkvJilm74=/fill/d2hpdGU=/fontsize/100/dissolve/50/gravity/northeast/dx/20/dy/20
```

8.5 获取图片基本信息 (imageInfo)

获取图片的基本信息，包括图片的格式、长、宽等。

接口形式：

```
download_url?imageInfo
```

示例：

```
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageInfo
```

8.6 获取图片exif (exif)

接口形式：

download_url?imageInfo

示例：

```
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?exif
```

8.7 获取图片主色调 (imageAve)

接口形式：

download_url?imageAve

示例：

```
http://v2test-10000812.image.myqcloud.com/tencentyunRestAPITest?imageAve
```

9 返回码定义

错误码	含义
-5999	参数错误
-5998	签名格式错误
-5997	后端网络错误
-5996	HTTP请求方法错误
-5995	文件大小错误
-5994	url参数解析不匹配
-5993	multipart/formdata参数错误
-5992	请求参数错误
-5991	分片过大
-5990	找不到filecontent
-5989	上传失败
-5988	cgi初始化错误
-5987	wup编码失败
-5986	wup解码失败

错误码	含义
-5985	获取路由失败
-5984	sha1不匹配
-5983	错误的session
-5982	建立连接错误
-5981	建立连接错误

其他返回码参见 [返回码说明](#)。

返回码说明

最近更新时间：2017-03-09 11:15:25

1 图片下载错误

图片下载失败时，返回包headers中的X-ErrNo字段错误码说明。

错误码	含义
-6101	图片不存在、图片没有上传或者图片已经删除
-5062	图片涉嫌违禁
-902	镜像存储功能把请求转发到开发商源站，但没有收到响应，超时了
-100	未知错误，某些场景或者逻辑未定义
-101	存储文件失败
-103	无效的请求；请求报文无法识别
-104	无效的appid。Url中包含的appid无效,或者域名没有和appid绑定
-105	无效的样式名。Url中指定的样式名或者别名没有配置
-106	无效的URL。Url格式不符合格式要求
-107	无效的Host头域
-108	无效的Referer
-109	无效的样式名ID。没有找到该样式名对应的图片。可能是上传该图片后，新增的样式名，因此图片上传时不能生成该样式名对应的数据
-110	该图片在黑名单中
-111	http服务器内部错误
-120	回源到源站获取数据时，源站返回的数据有异常，无法正常获取到图片数据
-121	http服务器内部错误
-122	图片数据没有修改，客户端可以使用缓存数据
-123	图片数据没有修改，客户端可以使用缓存数据
-124	下载偏移错误。Http请求的Range 断点续传偏移量可能设置错误
-129	无法预料的错误
-130	http服务器内部错误
-140	http服务器内部错误

2 图片插件错误码说明

错误码	含义
-2000	服务过载
-1900	框架handle process错误
-1899	存储文件失败
-1898	校验md5失败
-1897	秒传失败
-1896	编码失败
-1895	解码失败
-1894	业务ID错误
-1893	图片数据异常，压缩库无法处理
-1892	上传失败，服务器错误
-1891	解码biz_req失败
-1890	编码biz_req失败
-1889	存储文件超时
-1888	压缩超时
-1887	校验sha1失败
-1886	图片fileid已经存在

3 文件缓存错误码说明

错误码	含义
-300	服务过载
-299	命令字未知
-298	解包失败
-297	框架handle process错误
-296	打包失败
-295	文件数据异常
-294	文件数据异常
-293	通知插件失败

错误码	含义
-292	文件缓存服务器错误
-291	编码session失败
-290	无效的session
-289	插件拒绝上传
-288	process打包失败
-287	process解包失败
-286	解码session失败
-285	文件过大
-284	分片大小不一致
-283	分片过小

4 cmd错误码说明

错误码	含义
-199	文件移动失败
-198	重定向错误
-197	查无此文件
-196	网络请求失败
-195	网络请求失败
-194	后端打包失败
-193	返回包打包失败
-192	请求包解析失败
-191	url参数解析不匹配
-190	文件删除失败
-189	输入参数错误：download_url empty
-187	从url中解析参数失败
-186	暂不支持视频文件复制
-185	业务预处理失败
-184	业务后处理失败

错误码	含义
-183	获取路由失败
-182	参数检验失败

5 proxy错误码说明

错误码	含义
-99	proxy读取配置失败
-98	调用签名服务失败
-97	非法签名
-96	签名过期
-95	消息缺少session信息
-94	携带错误session信息
-89	proxy转发cmd服务失败
-88	编码cmd服务消息失败
-87	解析cmd服务消息失败
-86	proxy转发process服务失败
-85	解析process服务应答失败
-84	获取process L5失败
-83	签名服务解包失败
-82	不存在此appid
-81	签名为空
-80	非法的业务ID
-79	secret id不存在
-78	SDK协议不匹配,请升级
-77	单次性签名已不可用
-76	单次签名没有url
-75	不支持此操作
-74	多次签名-过期时间为0
-73	单次签名-过期时间不为0

错误码	含义
-72	签名失败
-71	操作太频繁,请稍后再试
-70	appid/userid 与签名不匹配
-69	输入参数错误:download_url empty
-68	ip直通车打包失败
-67	ip直通车解包失败
-65	断点续传不支持携带数据包
-64	该业务已经被屏蔽

版本说明

最近更新时间：2017-11-09 17:28:45

1. 客户端版本说明

Android SDK：版本号V1.1.0为V1版本；版本号V1.1.1为V2版本和V2版本加强版。

iOS SDK：版本号V1.1.0为V1版本；版本号V1.1.1为V2版本和V2版本加强版。

注：V2加强版中，自定fileid支持特殊字符，如'/'。

2. 服务端SDK版本说明

SDK	V1版本	V2版本	V2加强版本
Java	2.0以下	2.0.0-2.0.3	2.0.4-2.1.0
PHP	1.3.0以下	1.3.0-2.0.0	2.0.0
Python	2.0.0以下	2.0.0-2.1.3	2.1.4
Nodejs	2.0.0以下	2.0.0-2.0.2	2.0.2
Go	2.0.0以下	2.0.0	2.0.1

3. 控制台不同版本的区别

V2加强版本：V2加强版本空间名称是一串字符串，无特殊标记；



V2版本：V2版本空间名称前面会加一个  图标予以标示；



V1版本：V1版本空间名称前面会加一个  图标予以标示。

4. 各版本的区别

下面从功能、Restful API上传接口、下载url等几个方面对三个版本的区别进行说明。

4.1 功能

V1版本：万象优图第一个版本，无bucket概念，控制台创建的是应用；

V2版本：万象优图第二个版本，首次提出bucket概念；

V2加强版：万象优图第三个版本，有bucket概念，采用新的实时处理风格，fileid支持特殊字符，支持回源镜像、样式下载别名、样式分隔符。

注：

- 1、相应的功能请参考图片空间->管理和样式，以及相应的wiki说明
- 2、A风格和B风格实时处理请分别参考A风格实时处理文档和B风格实时处理文档
- 3、下表中"√"表示支持，"×"表示不支持；
- 4、fileid(特殊字符)表示是否支持自定fileid

功能	V1版本	V2版本	V2加强版
自定义fileid	√	√	√
fileid包含特殊字符	×	×	√
控制台图片样式	√	√	√
图片空间 (bucket)	×	√	√
A风格实时处理	√	√	×
B风格实时处理	×	×	√
样式下载别名-A风格	×	×	×
样式下载别名-B风格	×	×	√
样式分隔符	×	×	√
镜像回源	×	×	√
USERID	√	后台置0处理	后台置0处理
APPID含义	应用名称	项目名称	项目名称

4.2 Restful API上传接口

版本	上传接口	fileid说明
V1版本	<code>http://web.image.myqcloud.com/photos/v1/[appid]/[userid]/[fileid]</code>	可选，由字母,数字, -, _等字符组成，最长128字节
V2版本	<code>http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]</code>	可选，由字母,数字, -, _等字符组成，最长128字节
V2加强版本	<code>http://web.image.myqcloud.com/photos/v2/[appid]/[bucket]/[userid]/[fileid]</code>	可选，由除了\0外的任意字符组成，需要对fileid进行urlencode，最长128字节

4.3 下载url

版本	下载url
V1版本	<code>http://[appid].image.myqcloud.com/[appid]/[userid]/[fileid]/original</code>
V2版本	<code>http://[bucket]-[appid].image.myqcloud.com/[bucket]-[appid]/[userid]/[fileid]/original</code>

版本	下载url
V2加强版本	http://[bucket]-[appid].image.myqcloud.com/[fileid]

注：

- 1、表中的下载url为原图的下载url
- 2、V1和V2版本访问相应的样式图只需要将下载url中的"original"改成开发者设置的样式名即可
- 3、V2加强版访问相应的样式图方法为：下载url+样式分隔符+样式名

5. 历史版本及更新记录

5.1 Android SDK历史版本及更新日志

版本号：Android SDK V1.1.3.3

发布时间：2015-12-17

版本说明：

1. 支持查询图片底色
2. 优化代码，修改bug

下载地址：[Android SDK V1.1.3.3（含说明文档）](#) | [Android 体验 Demo](#)

版本号：Android SDK V1.1.3

发布时间：2015-09-15

版本说明：

1. 签名接口技术优化
2. 上传sdk和下载sdk去掉userid

下载地址：[Android SDK V1.1.3](#) | [Android 体验demo](#)

版本号：Android SDK V1.1.1

发布时间：2015-07-08

版本说明：

1. 图片上传支持Bucket
2. 图片上传支持自定义Fileid
3. 上传数据分片校验
4. 多链接并发上传
5. 修复上传SDK长时间空闲之后,再次上传超时的问题
6. 优化上传性能
7. 图片上传支持内存byte[]

下载地址：[Android SDK V1.1.1.71](#) | [Android 体验demo](#)

版本号：Android SDK V1.1.0

发布时间：2015-05-15

版本说明：

1. 上传性能优化；
2. 增加SDK日志上报接口。

下载地址：[Android SDK V1.1.0](#) | [Android 体验Demo](#)

5.2 iOS SDK历史版本及更新日志

版本号：iOS SDK V1.1.3.3

发布时间：2015-12-17

版本说明：

1. 增加了查询图片底色方法属性（TXYFileInfo 增加了photoRBG属性）
2. 上传SDK添加支持了 bitcode 版本
3. 修改bug

下载地址：[iOS SDK V1.1.3.3（含说明文档）](#) | [iOS 体验 Demo](#)

版本号：iOS SDK V1.1.3

发布时间：2015-09-15

版本说明：

1. 上传图片加入内存直接上传接口
2. 签名跟task任务绑定
3. 上传数据分片校验
4. 上传sdk去掉authorize接口
5. 上传sdk去掉userid参数
6. 上传任务持久化，签名经过加密后持久化

下载地址：[iOS SDK V1.1.3-真机](#) | [iOS SDK V1.1.3-真机&模拟器](#) | [iOS 体验Demo](#)

版本号：iOS SDK V1.1.1

发布时间：2015-07-08

版本说明：

1. 图片上传支持Bucket
2. 图片上传支持自定义FileId
3. 上传数据分片校验
4. 多链接并发上传
5. 下载添加域名黑名单
6. 修复进度跳跃问题

下载地址：[iOS SDK V1.1.1-真机](#) | [iOS SDK V1.1.1-真机&模拟器](#) | [iOS 体验Demo](#)

版本号：iOS SDK V1.1.0

发布时间：2015-05-15

版本说明：

1. 上传性能优化；
2. 增加SDK日志上报接口。

下载地址：[iOS SDK V1.1.0-真机](#) | [iOS SDK V1.1.0-真机&模拟器](#) | [iOS 体验Demo](#)

5.3 服务器SDK历史版本及更新日志

服务器SDK历史版本及更新日志见相应的github项目。

3.1 Java SDK

github地址：<https://github.com/tencentyun/java-sdk.git>

3.2 PHP SDK

github地址：<https://github.com/tencentyun/php-sdk>

3.3 Python SDK

github地址：<https://github.com/tencentyun/python-sdk>

3.4 Nodejs SDK

github地址：<https://github.com/tencentyun/nodejs-sdk>

3.5 Go SDK

github地址：<https://github.com/tencentyun/go-sdk>

OCR识别接口文档

最近更新时间：2018-05-28 11:01:16

1. 体验与说明

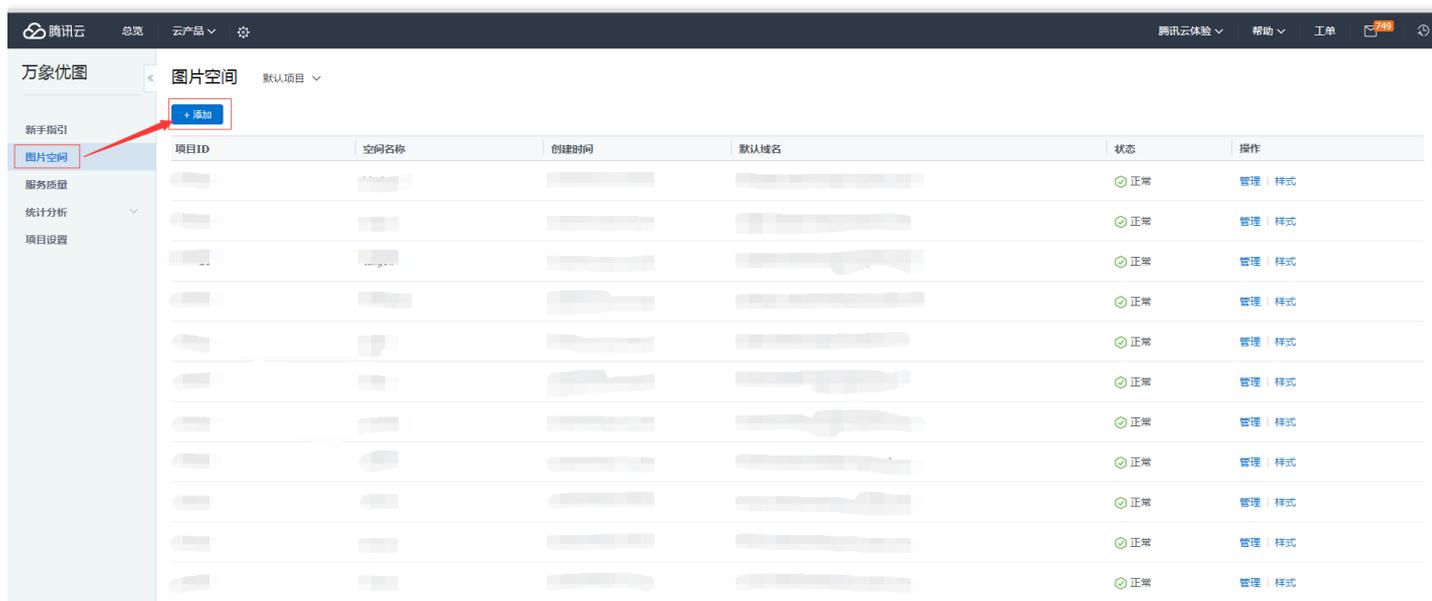
1.1 开发准备

开发者使用OCR功能之前，需要先注册腾讯云账号，并创建图片空间，添加密钥，具体操作步骤如下：

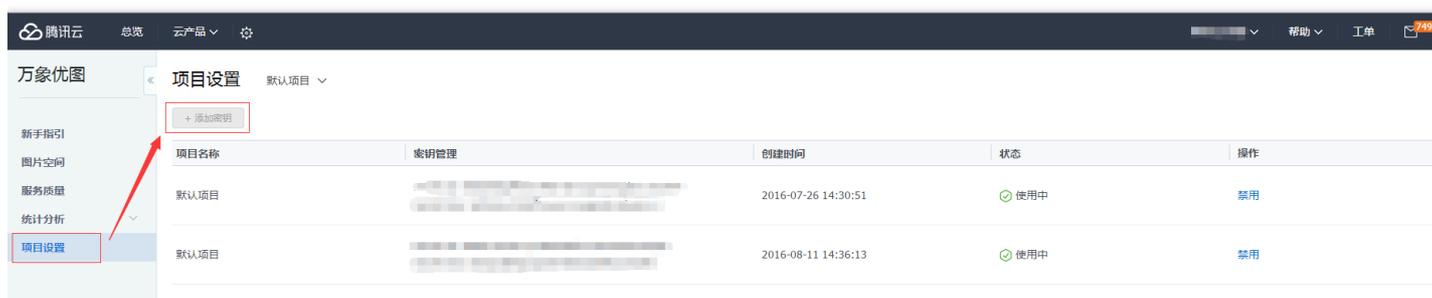
1) 前往腾讯云·万象优图控制台注册账号；如果已经注册账号，请跳过此步骤；



2) 在腾讯云·万象优图控制台创建一个图片空间，获取空间名称（bucket）和项目ID；如果已创建过图片空间，请跳过此步骤；



3) 在腾讯云·万象优图控制台项目设置中添加密钥，获取SecretID和SecretKey；如果已经添加过密钥，请跳过此步骤。



基本概念：

概念	解释
appid	项目ID, 用于唯一标识接入项目
bucket	开发者添加的空间名称

2. OCR_CGI

2.1 身份证OCR接口

OCR接口采用http协议，支持多URL和多本地图片文件,每个请求最多支持20张图片或url。

接口：<http://service.image.myqcloud.com/ocr/idcard>

方法：POST

2.1.1 身份证OCR-图片URL

请求语法:

POST /ocr/idcard HTTP/1.1

Authorization: Signature

Host: service.image.myqcloud.com

Content-Length: ContentLength

Content-Type: "application/json"

```
{
  "appid": appid,
  "bucket": "bucket",
  "card_type": type,
  "url_list": [
    "url",
    "url"
  ]
}
```

请求包http header:

参数	是否必选	描述
Host	是	访问域名, service.image.myqcloud.com
Authorization	是	鉴权签名, 详见下面鉴权章节
Content-Type	是	标准application/json
Content-Length	是	http body总长度

请求包http body:

参数	是否必选	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
card_type	是	int	0为身份证有照片的一面, 1为身份证有国徽的一面
url_list	是	string	图片url列表

响应http body (json格式):

参数	类型	类型
result_list	json数组	具体查询数据, 内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码, 0为成功

参数	类型	描述
message	string	服务器返回的信息
url	string	当前图片的url
data		具体查询数据，内容见下表

data字段具体内容（身份证有照片的一面）

参数	类型	描述
name	string	姓名
sex	string	性别
nation	string	民族
birth	string	出生日期
address	string	地址
id	string	身份证号
name_confidence_all	array(int)	证件姓名置信度
sex_confidence_all	array(int)	性别置信度
nation_confidence_all	array(int)	民族置信度
birth_confidence_all	array(int)	出生日期置信度
address_confidence_all	array(int)	地址置信度
id_confidence_all	array(int)	身份证号置信度

data字段具体内容（身份证反面）

参数	类型	描述
authority	string	发证机关
valid_date	string	证件有效期
authority_confidence_all	array(int)	发证机关置信度
valid_date_confidence_all	array(int)	证件有效期置信度

注：置信度的值为区间在[0,100]的整数

2.1.2 身份证OCR-图片URL示例

http请求：

POST /ocr/idcardHTTP/1.1

Authorization: FL26MsO1nhrZGuXdin10DE5tnDdhPTEwMDAwMDEmYj1xaW5pdXRlc3QyJms9QUtJRG1PNWNQVz

Host: service.image.myqcloud.com

Content-Length: 302

Content-Type: "application/json"

```
{
  "appid":11111,
  "bucket":"test",
  "card_type":0,
  "url_list":[
    "http://www.test.com/aaa.jpg",
    "http://www.test.com/bbb.jpg"
  ]
}
```

响应httpbody (application/json格式) :

```
{
  "result_list": [
    {
      "code": 0,
      "message": "OK",
      "url": " http://www.test.com/aaa.jpg",
      "data": {
        "name": "某某",
        "sex": "男",
        "nation": "汉",
        "birth": "2000/1/1",
        "address": "某地",
        "id": "123456200001011234",
        "name_confidence_all": [
          38,28
        ],
        "sex_confidence_all": [
          28
        ],
        "nation_confidence_all": [
          34
        ],
        "birth_confidence_all": [
          38, 38, 20, 46, 50, 49
        ],
        "address_confidence_all": [
          13, 30,
        ],
        "id_confidence_all": [
          49, 50, 58, 63, 51, 52, 55, 48, 48, 47, 58, 47, 48, 56,
          45, 55, 54, 47
        ]
      }
    ]
  }
}
```

```
},
{
  "code": 0,
  "message": "OK",
  "url": " http://www.test.com/bbb.jpg",
  "data": {
    "name": "某某某",
    "sex": "女",
    "nation": "汉",
    "birth": "2001/1/1",
    "address": "某地",
    "id": "123456200101011234",
    "name_confidence_all": [
      38,28,55
    ],
    "sex_confidence_all": [
      28
    ],
    "nation_confidence_all": [
      34
    ],
    "birth_confidence_all": [
      38, 38, 20, 46, 50, 49
    ],
    "address_confidence_all": [
      13, 30,
    ],
    "id_confidence_all": [
      49, 50, 58, 63, 51, 52, 55, 48, 48, 47, 58, 47, 48, 56,
      45, 55, 54, 47
    ]
  }
}
]
```

2.1.3 身份证OCR-图片文件

图片文件OCR使用HTML表单上传一个或多个文件，文件内容通过多重表单格式（multipart/form-data）编码。

请求语法:

```
POST /ocr/idcard HTTP/1.1
Content-Type:multipart/form-data;boundary=-----acebdf13572468
Authorization: Signature
Host: service.image.myqcloud.com
Content-Length: ContentLength

-----acebdf13572468
Content-Disposition: form-data; name="appid";

appid
-----acebdf13572468
```

```

Content-Disposition: form-data; name="bucket";

bucket
-----acebdf13572468
Content-Disposition: form-data; name="card_type";

1
-----acebdf13572468
Content-Disposition: form-data; name="image[0]"; filename="image_1.jpg"
Content-Type: image/jpeg

image_content
-----acebdf13572468
Content-Disposition: form-data; name="image[1]"; filename="image_2.jpg "
Content-Type: image/jpeg

image_content
-----acebdf13572468--
    
```

请求包http header:

参数	是否必选	描述
Host	是	访问域名, service.image.myqcloud.com
Authorization	是	鉴权签名, 详见下面鉴权章节
Content-Type	是	标准的multipart/form-data
Content-Length	是	http body总长度

表单域:

参数	是否必选	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
card_type	是	int	0为身份证有照片的一面, 1为身份证有国徽的一面
image	是	image/jpeg 等	图片文件, 支持多个。参数名须为 "image[0]"、"image[1]"等image开头的字符串。响应http body中会按照该字符串的字典序排列。每张图片需指定filename, filename的值为可为空, 响应http body中会返回用户设置的filename值。

响应http body (json格式)

参数	类型	描述
result_list	json数组	具体查询数据, 内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码，0为成功
message	string	服务器返回的信息
filename	string	当前图片的filename，与请求包中filename一致
data		具体查询数据，内容见下表

data字段具体内容（身份证有照片的一面）

参数	类型	描述
name	string	姓名
sex	string	性别
nation	string	民族
birthday	string	出生日期
address	string	地址
id	string	身份证号
name_confidence_all	array(int)	证件姓名置信度
sex_confidence_all	array(int)	性别置信度
nation_confidence_all	array(int)	民族置信度
birth_confidence_all	array(int)	出生日期置信度
address_confidence_all	array(int)	地址置信度
id_confidence_all	array(int)	身份证号置信度

data字段具体内容（身份证反面）

参数	类型	描述
authority	string	发证机关
valid_date	string	证件有效期
authority_confidence_all	array(int)	发证机关置信度
valid_date_confidence_all	array(int)	证件有效期置信度

注：置信度的值为区间在[0,100]的整数

2.1.4 身份证OCR-图片文件示例

http请求：

```
POST /ocr/idcard HTTP/1.1
Content-Type:multipart/form-data;boundary=-----acebdf13572468
Authorization: FCHXddYbhZEBfTeZ0j8mn9Og16JhPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJiUkp3Sk5zbTM
Host: service.image.myqcloud.com
Content-Length: 61478

-----acebdf13572468
Content-Disposition: form-data; name="appid";

11111
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

testbucket
-----acebdf13572468
Content-Disposition: form-data; name="card_type";

1
-----acebdf13572468
Content-Disposition: form-data; name="image[0]"; filename="1.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\185839ggh0oedgnog04g0b.jpg.thumb.jpg*@>
-----acebdf13572468
Content-Disposition: form-data; name="image[1]"; filename="2.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\200132svnmybmbhbmgbmga.jpg.thumb.jpg*@>
-----acebdf13572468
```

响应httpbody (application/json格式)：

```
{
  "result_list": [
    {
      "code": 0,
      "message": "success",
      "filename": "1.jpg",
      "data": {
        "authority": "某市派出所",
        "valid_date": "2012.01.01-2022.01.01",
        "authority_confidence_all": [
          42, 36, 40, 49, 41
        ],
        "valid_date_confidence_all": [
          44, 50, 63, 44, 47, 42, 43, 53, 48, 52, 44, 47, 48, 45, 50, 58
        ]
      }
    }
  ],
  {
```

```

"code": 0,
"message": "success",
"filename": "2.jpg",
"data": {
  "authority": "某市派出所",
  "valid_date": "2012.01.01-2022.01.01",
  "authority_confidence_all": [
    42, 36, 40, 49, 41
  ],
  "valid_date_confidence_all": [
    44, 50, 63, 44, 47, 42, 43, 53, 48, 52, 44, 47, 48, 45, 50, 58
  ]
}
}
]
}

```

2.2 名片OCR接口

OCR接口采用http协议，支持多URL和多本地图片文件,每个请求最多支持20张图片或url。

接口：http://service.image.myqcloud.com/ocr/namecard

方法：POST

2.2.1 名片OCR-图片URL

请求语法:

```

POST /ocr/namecard HTTP/1.1
Authorization: Signature
Host: service.image.myqcloud.com
Content-Length: ContentLength
Content-Type: "application/json"

```

```

{
  "appid": appid,
  "bucket": "bucket",
  "ret_image": ret,
  "url_list": [
    "url",
    "url"
  ]
}

```

请求包http header:

参数	是否必选	描述
Host	是	访问域名，service.image.myqcloud.com
Authorization	是	鉴权签名，详见下面鉴权章节
Content-Type	是	标准application/json

参数	是否必选	描述
Content-Length	是	http body总长度

请求包http body:

参数	是否必选	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
ret_image	是	int	0不返回图片, 1返回图片
url_list	是	string	图片url列表

响应http body (json格式) :

参数	类型	描述
result_list	json数组	具体查询数据, 内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码, 0为成功
message	string	服务器返回的信息
url	string	当前图片的url
data		具体查询数据, 内容见下表

data字段具体内容

参数	类型	描述
name	string	姓名
phone	string	手机号
uin	string	QQ号
name_confidence	double	姓名置信度
phone_confidence	double	手机号置信度

注: 如未识别出某字段 (如name) , 则该字段对应的置信度 (如name_confidence) 为-1

2.2.2 名片OCR-图片URL示例

http请求：

```
POST /ocr/namecard HTTP/1.1
Authorization: FL26MsO1nhrZGuXdin10DE5tnDdhPTEwMDAwMDEmYj1xaW5pdXRlc3QyJms9QUtJRG1PNWNQVzNMREc
Host: service.image.myqcloud.com
Content-Length: 302
Content-Type: "application/json"
```

```
{
  "appid":11111,
  "bucket":"test",
  "ret_image":0,
  "url_list":[
    "http://www.test.com/aaa.jpg",
    "http://www.test.com/bbb.jpg"
  ]
}
```

响应httpbody (application/json格式)：

```
{
  "result_list": [
    {
      "code": 0,
      "message": "OK",
      "url": " http://www.test.com/aaa.jpg",
      "data": {
        "name": "某某",
        "phone": "12345678998" ,
        "uin": "10000",
        "name_confidence": 0.9,
        "phone_confidence": 0.9,
        "uin_confidence": 0.9
      }
    },
    {
      "code": 0,
      "message": "OK",
      "url": " http://www.test.com/bbb.jpg",
      "data": {
        "name": "某某某",
        "phone": "12345678999" ,
        "uin": "10001",
        "name_confidence": 0.99,
        "phone_confidence": 0.99,
        "uin_confidence": 0.99
      }
    }
  ]
}
```

2.2.3 名片OCR-图片文件

图片文件OCR使用HTML表单上传一个或多个文件，文件内容通过多重表单格式（multipart/form-data）编码。

请求语法

```

POST /ocr/namecard HTTP/1.1
Content-Type:multipart/form-data;boundary=-----acebdf13572468
Authorization: Signature
Host: service.image.myqcloud.com
Content-Length: ContentLength

-----acebdf13572468
Content-Disposition: form-data; name="appid";

appid
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

bucket
-----acebdf13572468
Content-Disposition: form-data; name="ret_image";

0
-----acebdf13572468
Content-Disposition: form-data; name="image[0]"; filename="image_1.jpg"
Content-Type: image/jpeg

image_content
-----acebdf13572468
Content-Disposition: form-data; name="image[1]"; filename="image_2.jpg "
Content-Type: image/jpeg

image_content
-----acebdf13572468--
    
```

请求包http header:

参数	是否必选	描述
Host	是	访问域名，service.image.myqcloud.com
Authorization	是	鉴权签名，详见下面鉴权章节
Content-Type	是	标准的multipart/form-data
Content-Length	是	http body总长度

表单域:

参数	是否必选	类型	描述
appid	是	uint	业务id

参数	是否必选	类型	描述
bucket	是	string	图片空间
ret_image	是	int	0不返回图片，1返回图片
image	是	image/jpeg 等	图片文件，支持多个。参数名须为“image[0]”、“image[1]”等image开头的字符串。响应http body中会按照该字符串的字典序排列。每张图片需指定filename，filename的值为可为空，响应http body中会返回用户设置的filename值。

响应http body (json格式)

参数	类型	描述
result_list	json数组	具体查询数据，内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码，0为成功
message	string	服务器返回的信息
filename	string	当前图片的filename，与请求包中filename一致
data		具体查询数据，内容见下表

data字段具体内容

参数	类型	描述
name	string	姓名
phone	string	手机号
uin	string	QQ号
name_confidence	double	姓名置信度
phone_confidence	double	手机号置信度
uin_confidence	double	QQ号置信度
image	string	返回图片的base64编码 (ret_image=1时返回)

注：如未识别出某字段（如name），则该字段对应的置信度（如name_confidence）为-1

2.2.4 名片OCR-图片文件示例

http请求：

```

POST /ocr/namecard HTTP/1.1
Content-Type:multipart/form-data;boundary=-----acebdf13572468
Authorization: FCHXddYbhZEBfTeZ0j8mn9Og16JhPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJiUkp3Sk5zbTM
Host: service.image.myqcloud.com
Content-Length: 61478

-----acebdf13572468
Content-Disposition: form-data; name="appid";

11111
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

testbucket
-----acebdf13572468
Content-Disposition: form-data; name="ret_image";

0
-----acebdf13572468
Content-Disposition: form-data; name="image[0]"; filename="1.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\185839ggh0oedgnog04g0b.jpg.thumb.jpg*@>
-----acebdf13572468
Content-Disposition: form-data; name="image[1]"; filename="2.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\200132svnmybmbhbmgbmga.jpg.thumb.jpg*@>
-----acebdf13572468
    
```

响应httpbody (application/json格式) :

```

{
  "result_list": [
    {
      "code": 0,
      "message": "success",
      "filename": "1.jpg",
      "data": {
        "name": "某某",
        "phone": "12345678998" ,
        "uin": "10000",
        "name_confidence": 0.9,
        "phone_confidence": 0.9,
        "uin_confidence": 0.9
      }
    },
    {
      "code": 0,
      "message": "success",
      "filename": "2.jpg",
    }
  ]
}
    
```

```

    "data": {
      "name": "某某某",
      "phone": "12345678999" ,
      "uin": "10001",
      "name_confidence": 0.99,
      "phone_confidence": 0.99,
      "uin_confidence": 0.99
    }
  }
]
}
    
```

2.3 鉴权

腾讯云·万象优图通过签名来验证请求的合法性。

2.3.1 签名算法

1 获取签名所需信息

生成签名所需信息包括项目ID（appid），空间名称（bucket,图片资源的组织管理单元），项目的Secret ID和Secret Key。获取这些信息的方法如下：

- a) 登录 万象优图-图片空间, 进入图片空间；
- b) 如开发者未创建图片空间，可添加图片空间，获取项目ID（appid），空间名称（bucket）；如果开发者已经创建过空间，则可以直接获取项目ID和空间名称（bucket）；
- c) 登录万象优图-项目设置，进入项目设置；
- d) 如果开发者未添加密钥，则需添加密钥，获取项目的Secret ID和Secret Key，每个项目最多添加两对密钥；如果已经添加过密钥则直接获取项目的Secret ID和Secret Key。

注：

- (1) 添加图片空间可参考添加图片空间；
- (2) 添加密钥可参考添加密钥。

2 拼接签名串original

```
a=[appid]&b=[bucket]&k=[SecretID]&t=[currentTime]&e=[expiredTime]
```

注意：如果开发者使用的是V1版本，a字段为appid，b字段的值置空

字段名称	解释
a	开发者的项目ID，接入万象优图创建空间时系统生成的唯一标示项目的项目ID
b	图片空间名称
k	项目的Secret ID
t	当前时间，UNIX时间戳
e	签名过期时间，UNIX时间戳

3 生成签名

- a) 万象优图使用 HMAC-SHA1 算法对请求进行签名；
- b) 签名串需要使用 Base64 编码。

即生成签名的公式如下：

$$\text{SignTmp} = \text{HMAC-SHA1}(\text{SecretKey}, \text{original})$$
$$\text{Sign} = \text{Base64}(\text{SignTmp}.\text{original})$$

其中SecretKey为项目的密钥SecretKey，original为1节中拼接好的签名串，首先对original使用HMAC-SHA1算法进行签名，然后将original附加到签名结果的末尾，再进行Base64编码，得到最终的sign。

注：此处使用的是标准的Base64编码，不是urIsafe的Base64编码，请注意。

2.3.2 签名示例

本节介绍生成签名的算法实例，实例中使用PHP语言，如果开发者使用其他与开发，请使用对应的算法。

1. 获取签名所需信息

获取得到的签名所需信息如下。

项目ID：1000027

空间名称 (bucket)：test333

Secret ID：AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek

Secret Key：oHL5srXW3Fkn8xtugG0BDhUfOOsC9DAd

2. 拼接签名串original

...

```
a=1000027&b=test333&k=AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek&t=1443434355&e=1443434365
```

```
$appid = "1000027";
```

```
$bucket = "test333";
```

```
$secret_id = "AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek";
```

```
$secret_key = "oHL5srXW3Fkn8xtugG0BDhUfOOsC9DAd";
```

```
$expired = time() + 10;
```

```
$current = time();
```

```
$srcStr = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&t='.$current.'&e='.$expired;
```

1. 生成签名

```
$signStr = base64_encode(hash_hmac('SHA1', $srcStr, $secret_key, true).$srcStr);
```

```
echo$signStr."\n";
```

最终得到的签名为：

```
xBx0lpW/tdlr1vfksAt3GpfXo9phPEwMDAwMjcmYj10ZXN0MzZjM5QWUjRG5YOTExNzJCczJOSzRTUDIBZDIKRFZZcG03THgyTmVrJnQ9MTQ2NjM5MzQ4OSZlPTE0NjYzOTM0OTk=
```

...

2.4 错误码

错误码	含义
3	错误的请求

错误码	含义
4	签名为空
5	签名串错误
6	appid/bucket/url不匹配
7	签名编码失败（内部错误）
8	签名解码失败（内部错误）
9	签名过期
10	appid不存在
11	secretid不存在
12	appid不匹配
13	重放攻击
14	签名失败
15	操作太频繁，触发频控
16	Bucket不存在
17	url为空
18	没有图片或url
19	图片数过多，单次请求最多支持20个url或文件
20	图片过大，单个文件最大支持1MB
21	无效的参数
200	内部打包失败
201	内部解包失败
202	内部链接失败
203	内部处理超时
-1300	图片为空
-1308	url图片下载失败
-1400	非法的图片格式
-1403	图片下载失败
-1404	图片无法识别
-1505	url格式不对

错误码	含义
-1506	图片下载超时
-1507	无法访问url对应的图片服务器
-5062	url对应的图片已被标注为不良图片，无法访问（专指存储于腾讯云的图片）
-5103	OCR识别失败

人脸识别接口文档

最近更新时间：2017-11-29 15:29:52

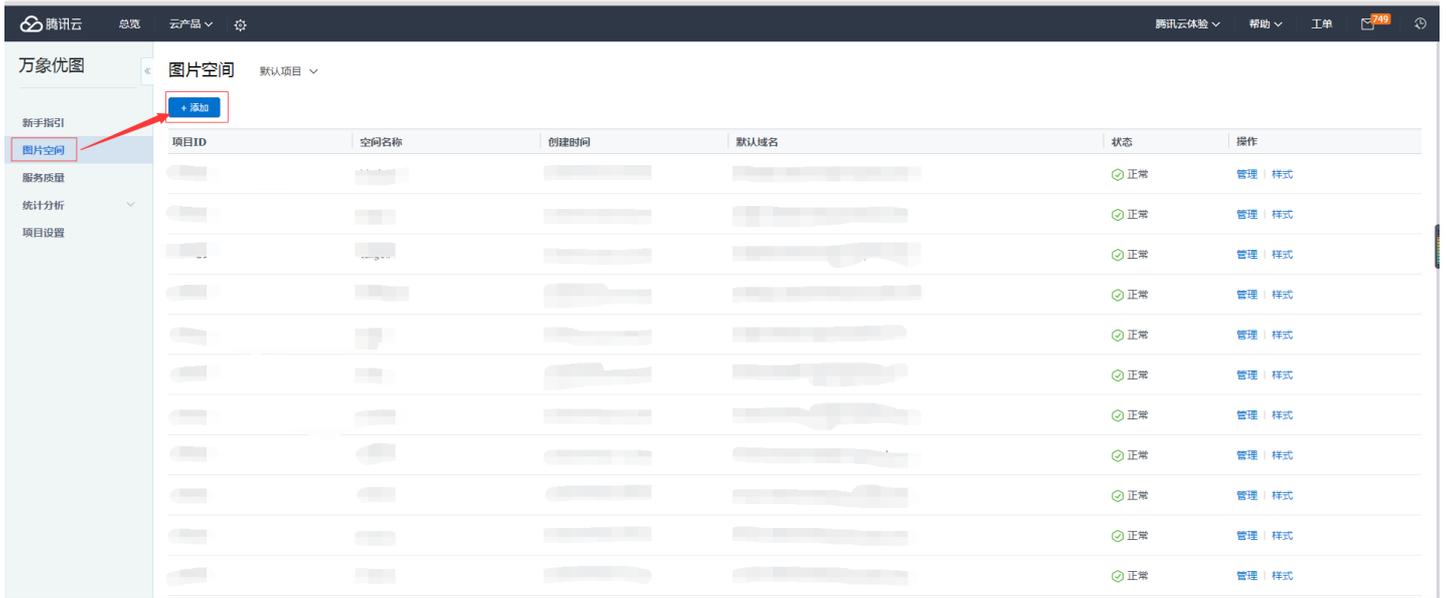
1. 说明

开发者使用人脸识别功能之前，需要先注册腾讯云账号，并创建图片空间，添加密钥，具体操作步骤如下：

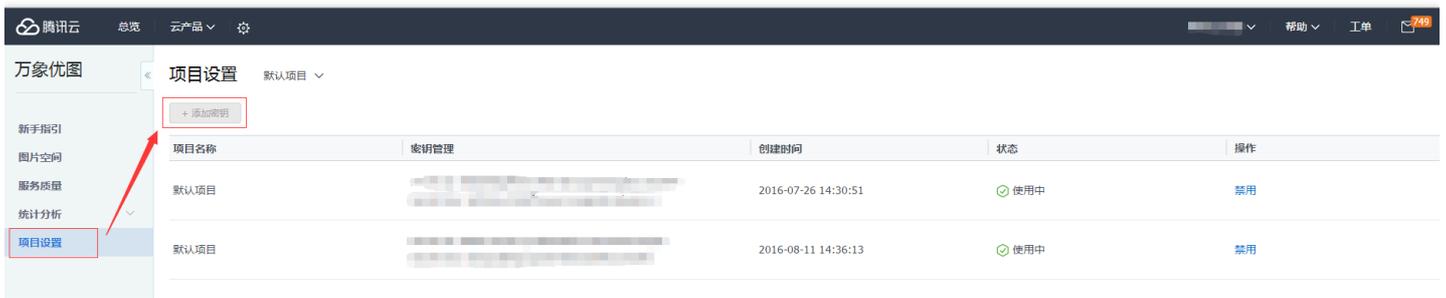
1) 前往[腾讯云·万象优图控制台](#)控制台注册账号；如果已经注册账号，请跳过此步骤；



2) 在[腾讯云·万象优图控制台](#)创建一个图片空间，获取空间名称（bucket）和项目ID；如果已创建过图片空间，请跳过此步骤；



3) 在[腾讯云-万象优图控制台](#)项目设置中添加密钥，获取SecretID和SecretKey；如果已经添加过密钥，请跳过此步骤。



基本概念：

概念	解释
appid	项目ID, 用于唯一标识接入项目
bucket	开发者添加的空间名称
group_id	个体(person)以组 (group) 的形式存储，一个组可以包含多个个体，一个个体也可以存在于多个组。group_id即用来标识group。组(group)没有专门的创建接口，创建个体(person)时，指定group_id则会自动创建。
person_id	人脸以个体 (person) 的形式存储，一个个体下可以存储多张人脸。person_id即用来标识person
face_id	标识每张人脸的id

说明：如果开发者使用的是V1版本，appid为其当时生成的appid

2. 人脸识别CGI

人脸识别接口采用http协议，支持指定图片URL和 上传本地图片文件两种方式进行识别。

限制说明：

- (1) 一个appid下建立的group_id数量限制为5000个。
- (2) 一个group_id下建立的person_id数量限制为10000个。
- (3) 一个person_id下建立的人脸数量限制为20个。
- (4) 每个请求的包体大小限制为1.5MB。
- (5) 所有接口都为POST方法。
- (6) 不支持 .gif这类的动图。

协议头部：所有请求都要求含有下表列出的头部信息

参数名	值	描述
Host	service.image.myqcloud.com	万象优图服务器域名
Content-Length	包体总长度	整个请求包体内容的总长度，单位：字节（Byte）
Content-Type	Application/json或者Multipart/form-data	根据不同接口选择
Authorization	鉴权签名	用于鉴权的签名， 详情

2.1 个体创建

1) 接口

`http://service.image.myqcloud.com/face/newperson`

2) 描述

创建一个Person，并将Person放置到group_ids指定的组当中，不存在的group_id会自动创建。

3) 参数

使用image则使用 multipart/form-data格式

不使用image则使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	string	空间名称
group_ids	必须	Array(String)	加入到组的列表
person_id	必须	String	指定的个体id
image	可选	Binary	图片内容
url	可选	String	图片的url, image和url只提供一个即可,如果都提供,只使用url
person_name	可选	String	名字
tag	可选	String	备注信息

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.suc_group	Int	成功被加入的group数量
data.suc_face	Int	成功加入的face数量
data.person_id	String	相应person的id
data.face_id	String	创建所用图片生成的face_id
data.group_ids	Array(String)	加入成功的组id
code	Int	返回码
message	String	返回错误消息

5) 样例

使用url的请求包:

```

POST /face/newperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 187
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test",
  "group_ids": [ "tencent", "qq" ],
  "person_id": "person0",
  "url": "http://test-123456.image.myqcloud.com/test.jpg"
}
    
```

使用image的请求包:

```

POST /face/newperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 735
Content-Type: multipart/form-data;boundary=-----acebdf13572468

-----acebdf13572468
Content-Disposition: form-data; name="appid";

123456
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

test
-----acebdf13572468
    
```

```
Content-Disposition: form-data; name="group_ids[0]";

tencent
-----acebdf13572468
Content-Disposition: form-data; name="group_ids[1]";

qq
-----acebdf13572468
Content-Disposition: form-data; name="person_id";

person0
-----acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----acebdf13572468--
```

回包:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 204
Content-Type: application/json

{
  "data":{
    "person_id":"person0",
    "suc_group":2,
    "suc_face":1,
    "session_id": "",
    "face_id":"1009550071676600319",
    "group_ids":["tencent", "qq"]
  },
  "code":0,
  "message":"OK"
}
```

2.2 删除个体

1) 接口

http://service.image.myqcloud.com/face/delperson

2) 描述

删除一个Person。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
-----	------	----	------

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	待删除个体ID

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.person_id	String	成功删除的person_id
data.deleted	Int	成功删除的Person数量
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```

POST /face/delperson HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 66
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person0"
}
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 114
Content-Type: application/json

{
  "data": {
    "deleted": 1,
    "person_id": "person0",
    "session_id": ""
  },
  "code": 0,
}
    
```

```
"message": "OK"
}
```

2.3 增加人脸

1) 接口

http://service.image.myqcloud.com/face/addface

2) 描述

将一组Face加入到一个Person中。注意，一个Face只能被加入到一个Person中。一个Person最多允许包含20个Face；加入几乎相同的人脸会返回错误。

3) 参数

使用image则使用 multipart/form-data格式；不使用image则使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	待增加人脸的个体id
images	可选	Array(Binary)	图片内容
urls	可选	String	图片的urls, images 和urls只提供一个就可以了,如果都提供,只使用urls
tag	可选	String	备注信息

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.added	Int	成功加入的face数量
data.face_ids	Array(String)	增加的人脸ID列表
data.ret_codes	Array(Int)	每张图片增加人脸的返回码
code	Int	返回状态码
message	String	返回错误消息

5) 样例

使用url的请求包:

```
POST /face/addface HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 191
```

Content-Type: application/json

```
{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person0",
  "urls": [
    "http://test-123456.image.myqcloud.com/face1.jpg",
    "http://test-123456.image.myqcloud.com/face2.jpg"
  ]
}
```

使用image的请求包:

POST /face/addface HTTP/1.1

Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==

Host: service.image.myqcloud.com

Content-Length: 641

Content-Type: multipart/form-data; boundary=-----acebdf13572468

-----acebdf13572468

Content-Disposition: form-data; name="appid";

123456

-----acebdf13572468

Content-Disposition: form-data; name="bucket";

test

-----acebdf13572468

Content-Disposition: form-data; name="person_id";

person0

-----acebdf13572468

Content-Disposition: form-data; name="images[0]"; filename="face1.jpg"

Content-Type: image/jpeg

yy

-----acebdf13572468

Content-Disposition: form-data; name="images[1]"; filename="face2.jpg"

Content-Type: image/jpeg

xx

-----acebdf13572468--

回包:

HTTP/1.1 200 OK

Connection: keep-alive

Content-Length: 168

Content-Type: application/json

```
{
```

```

"data":{
  "added":2,
  "session_id":"",
  "face_ids":["1009550071676600319","109387462576600319"],
  "ret_codes":[0, 0]
},
"code":0,
"message":"OK"
}
    
```

2.4 删除人脸

1) 接口

http://service.image.myqcloud.com/face/delface

2) 描述

删除一个person下的face，包括特征，属性和face_id。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	待删除人脸的person ID
face_ids	必须	Array(String)	删除人脸id的列表

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.deleted	Int	成功删除的face数量
data.face_ids	Array(String)	成功删除的face_id列表
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```

POST /face/delface HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
    
```

```
Content-Length: 125
Content-Type: application/json
```

```
{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person0",
  "face_ids": [
    "199d1efd19ce4ee67a7ec7655f859b1a"
  ]
}
```

回包:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 150
Content-Type: application/json
```

```
{
  "data": {
    "deleted": 1,
    "face_ids": [
      "199d1efd19ce4ee67a7ec7655f859b1a"
    ],
    "session_id": ""
  },
  "code": 0,
  "message": "OK"
}
```

2.5 设置信息

1) 接口

<http://service.image.myqcloud.com/face/setinfo>

2) 描述

设置Person的name。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	相应person的id
person_name	可选	String	新的name

参数名	是否必须	类型	参数说明
tag	可选	String	备注信息

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.person_id	String	相应person的id
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```

POST /face/setinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 97
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person0",
  "person_name": "NicolasCage"
}
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 96
Content-Type: application/json

{
  "data": {
    "person_id": "person0",
    "session_id": ""
  },
  "code": 0,
  "message": "OK"
}
    
```

2.6 获取信息

1) 接口

`http://service.image.myqcloud.com/face/getinfo`

2) 描述

获取一个Person的信息, 包括name, id, tag, 相关的face, 以及groups等信息。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	待查询个体的ID

4) 返回值说明

字段	类型	说明
data.person_name	String	相应person的name
data.person_id	String	相应person的id
data.face_ids	Array(String)	包含的人脸列表
data.group_ids	Array(String)	包含此个体的组id
data.session_id	String	相应请求的session标识符
data.tag	String	为此个体设置的备注信息
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```

POST /face/getinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 66
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person0"
}
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 293
Content-Type: application/json

{
  "data":{
    "person_id": "person0",
    "person_name": "NicolasCage",
    "face_ids": [
      "199d1efd19ce4ee67a7ec7655f859b1a",
      "19321efd19ce4ee67a7127655f85934a"
    ],
    "group_ids":[
      "tencent"
    ],
    "tag": "",
    "session_id": ""
  },
  "code":0,
  "message":"OK"
}
    
```

2.7 获取组列表

1) 接口

<http://service.image.myqcloud.com/face/getgroupids>

2) 描述

获取一个AppId下所有group列表。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称

4) 返回值说明

字段	类型	说明
data.group_ids	Array(String)	相应appid的group_id列表
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```

POST /face/getgroupids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 41
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test"
}
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 136
Content-Type: application/json

{
  "data": {
    "group_ids": [
      "tencent",
      "tencent-beijing",
      "tencent-shanghai"
    ]
  },
  "code": 0,
  "message": "OK"
}
    
```

2.8 获取人列表

1) 接口

<http://service.image.myqcloud.com/face/getpersonids>

2) 描述

获取一个组Group中所有person列表。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
group_id	必须	String	组id

4) 返回值说明

字段	类型	说明
data.person_ids	Array(String)	相应person的id列表
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```
POST /face/getpersonids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 65
Content-Type: application/json
```

```
{
  "appid": "123456",
  "bucket": "test",
  "group_id": "tencent"
}
```

回包:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 120
Content-Type: application/json
```

```
{
  "data": {
    "person_ids": [
      "person0",
      "person1",
      "person2"
    ]
  },
  "code": 0,
  "message": "OK"
}
```

2.9 获取人脸列表

1) 接口

`http://service.image.myqcloud.com/face/getfaceids`

2) 描述

获取一个组person中所有face列表。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	String	空间名称
person_id	必须	String	个体id

4) 返回值说明

字段	类型	说明
data.face_ids	Array(String)	相应face的id列表
code	Int	返回状态码
message	String	返回错误消息

5) 样例

请求包:

```
POST /face/getfaceids HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 66
Content-Type: application/json
```

```
{
  "appid": "123456",
  "bucket": "test",
  "person_id": "person1"
}
```

回包:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 154
Content-Type: application/json
```

```
{
  "data": {
    "face_ids": [
      "1005338790489817087",
      "1005338790489817088",
      "1005338790489817089"
    ]
  }
}
```

```

    },
    "code":0,
    "message":"OK"
}
    
```

2.10 获取人脸信息

1) 接口

<http://service.image.myqcloud.com/face/getfaceinfo>

2) 描述

获取一个face的相关特征信息。

3) 参数

使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	string	空间名称
face_id	必须	String	人脸id

4) 返回值说明

字段	类型	说明
data.face_info	Faceltem	人脸信息
code	Int	返回状态码
message	String	返回错误消息

Faceltem说明

字段	类型	说明
face_id	String	人脸标识
x	Int	人脸框左上角x
y	Int	人脸框左上角y
width	Float	人脸框宽度
height	Float	人脸框高度
gender	Int	性别 [0(female)~100(male)]
age	Int	年龄 [0~100]
expression	Int	微笑[0(normal)~50(smile)~100(laugh)]

字段	类型	说明
glass	Bool	是否有眼镜 [true,false]
pitch	Int	上下偏移[-30,30]
yaw	Int	左右偏移[-30,30]
roll	Int	平面旋转[-180,180]

5) 样例

请求包:

```

POST /face/getfaceinfo HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 76
Content-Type: application/json

{
  "appid": "123456",
  "bucket": "test",
  "face_id": "1005338790489817087"
}
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 313
Content-Type: application/json

{
  "data": {
    "face_info": {
      "face_id": "1005338790489817087",
      "x": 44,
      "y": 33,
      "height": 64.0,
      "width": 64.0,
      "pitch": 5,
      "roll": 0,
      "yaw": 6,
      "age": 37,
      "gender": 99,
      "glass": true,
      "expression": 6
    }
  },
  "code": 0,
}
    
```

```
"message": "OK"
}
```

2.11 人脸验证

1) 接口

http://service.image.myqcloud.com/face/verify

2) 描述

给定一个图片和一个Person，检查是否是同一个人。

3) 参数

使用image则使用 multipart/form-data格式

不使用image则使用 application/json格式

参数名	是否必须**	类型**	参数说明**
appid	必须	String	项目ID
bucket	必须	string	空间名称
person_id	必须	String	待验证的Person
image	可选	Binary	图片内容
url	可选	String	图片的url, image 和url只提供一个就可以了,如果都提供,只使用url

4) 返回值说明

字段**	类型**	说明**
data.session_id	String	相应请求的session标识符，可用于结果查询
data.confidence	float	两个人的相似度
data.ismatch	Bool	两个输入是否为同一人的判断
code	Int	返回状态码
message	String	返回错误消息

5) 样例

使用url的请求包:

```
POST /face/verify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 123
Content-Type: application/json

{
```

```

"appid": "123456",
"bucket": "test",
"person_id": "person1",
"url": "http://test-123456.image.myqcloud.com/test.jpg"
}
    
```

使用image的请求包:

```

POST /face/verify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=-----acebdf13572468

-----acebdf13572468
Content-Disposition: form-data; name="appid";

123456
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

test
-----acebdf13572468
Content-Disposition: form-data; name="person_id";

Person1
-----acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----acebdf13572468--
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 409
Content-Type: application/json

{
  "data": {
    "session_id": "session_id",
    "confidence": 90.0,
    "ismatch": true
  },
  "code": 0,
  "message": "OK"
}
    
```

2.12 人脸检索

1) 接口

`http://service.image.myqcloud.com/face/identify`

2) 描述

对于一个待识别的人脸图片，在一个Group中识别出最相似的Top5 Person作为其身份返回，返回的Top5中按照相似度从大到小排列。

3) 参数

使用image则使用 multipart/form-data格式

不使用image则使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	string	空间名称
group_id	必须	String	候选人组id
image	可选	Binary	图片内容
url	可选	String	图片的url, image 和url只提供一个就可以了,如果都提供,只使用url

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符，可用于结果查询
data.candidates	Array(IdentifyItem)	识别出的top5候选人
code	Int	返回状态码
message	String	返回错误消息

IdentifyItem说明

字段	类型	说明
person_id	String	候选者person_id
face_id	String	候选者face_id
confidence	Float	候选者的置信度
tag	String	人脸备注信息

5) 样例

使用url的请求包:

```

POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 123
Content-Type: application/json
    
```

```

{
  "appid": "123456",
  "bucket": "test",
  "group_id": "tencent",
  "url": "http://test-123456.image.myqcloud.com/test.jpg"
}
    
```

使用image的请求包:

```

POST /face/identify HTTP/1.1
Authorization: FCHXdPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJi==
Host: service.image.myqcloud.com
Content-Length: 460
Content-Type: multipart/form-data;boundary=-----acebdf13572468
    
```

```

-----acebdf13572468
Content-Disposition: form-data; name="appid";

123456
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

test
-----acebdf13572468
Content-Disposition: form-data; name="group_id";

tencent
-----acebdf13572468
Content-Disposition: form-data; name="image"; filename="test.jpg"
Content-Type: image/jpeg

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----acebdf13572468--
    
```

回包:

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 409
Content-Type: application/json

{
  "data":{
    "session_id": "session_id",
    "candidates": [
    
```

```

{
  "person_id": "person3",
  "face_id": "1031567119985213439",
  "confidence": 54.90695571899414,
  "tag": "new tag"
},
{
  "person_id": "person1",
  "face_id": "1031587105968553983",
  "confidence": 54.86775207519531,
  "tag": "new tag"
}
]
},
"code": 0,
"message": "OK"
}
    
```

2.13 人脸比对

1) 接口

<http://service.image.myqcloud.com/face/compare>

2) 描述

计算两个Face的相似性以及五官相似度。

3) 参数

使用image则使用 multipart/form-data格式

不使用image则使用 application/json格式

参数名	是否必须	类型	参数说明
appid	必须	String	项目ID
bucket	必须	string	空间名称
imageA	可选	Binary	A图片内容
imageB	可选	Binary	B图片内容
urlA	可选	String	A图片的url, image 和url只提供一个就可以了,如果都提供,只使用url
urlB	可选	String	B图片的url, image 和url只提供一个就可以了,如果都提供,只使用url

4) 返回值说明

字段	类型	说明
data.session_id	String	相应请求的session标识符
data.similarity	float	两个face的相似度

回包:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 109
Content-Type: application/json
```

```
{
  "data":{
    "similarity":100.0,
    "session_id":""
  },
  "code":0,
  "message":"OK"
}
```

3. 返回码

错误码	含义
3	错误的请求
4	签名为空
5	签名串错误
6	签名中的appid/bucket与操作目标不匹配
9	签名过期
10	appid不存在
11	secretid不存在
12	appid和secretid不匹配
13	重放攻击
14	签名校验失败
15	操作太频繁，触发频控
16	Bucket不存在
21	无效参数
23	请求包体过大
107	鉴权服务不可用
108	鉴权服务不可用

错误码	含义
213	内部错误
-1101	人脸检测失败
-1102	图片解码失败
-1103	特征处理失败
-1104	提取轮廓错误
-1105	提取性别错误
-1106	提取表情错误
-1107	提取年龄错误
-1108	提取姿态错误
-1109	提取眼镜错误
-1200	特征存储错误
-1300	图片为空
-1301	参数为空
-1302	个体已存在
-1303	个体不存在
-1304	参数过长
-1305	人脸不存在
-1306	组不存在
-1307	组列表不存在
-1308	url图片下载失败
-1309	人脸个数超过限制
-1310	个体个数超过限制
-1311	组个数超过限制
-1312	对个体添加了几乎相同的人脸
-1400	非法的图片格式
-1403	图片下载失败

图片标签接口文档

最近更新时间：2017-11-29 15:32:11

1. 体验与说明

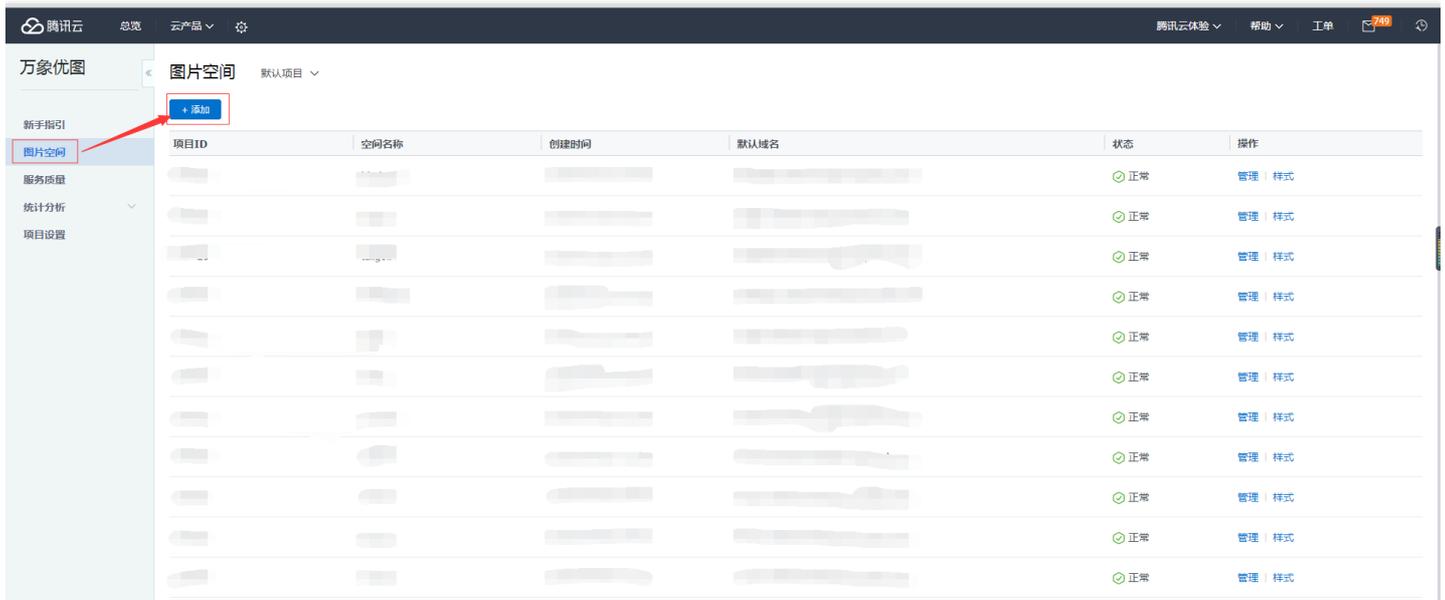
1.1 开发准备

开发者使用OCR功能之前，需要先注册腾讯云账号，并创建图片空间，添加密钥，具体操作步骤如下：

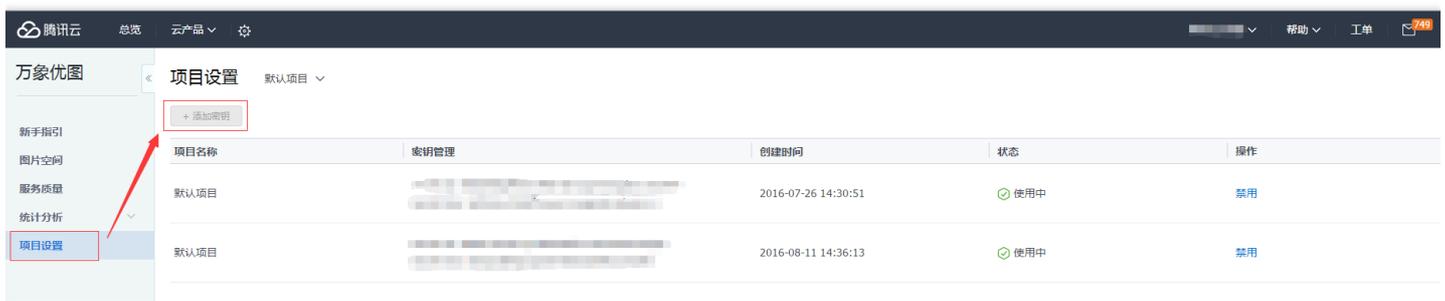
1) 前往腾讯云·万象优图控制台注册账号；如果已经注册账号，请跳过此步骤；



2) 在腾讯云·万象优图控制台创建一个图片空间，获取空间名称（bucket）和项目ID；如果已创建过图片空间，请跳过此步骤；



3) 在腾讯云·万象优图控制台项目设置中添加密钥，获取SecretID和SecretKey；如果已经添加过密钥，请跳过此步骤。



2. 接入层对外协议

2.1 接口和方法

接口：http://service.image.myqcloud.com/v1/detection/imagetag_detect

方法：POST

2.2 请求格式

请求包的http header

参数	是否必选	描述
Host	是	访问域名，service.image.myqcloud.com
Authorization	是	鉴权签名，见下节
Content-Type	是	标准的application/json
Content-Length	是	http body总长度

请求包body采用json格式，body具体内容见下表：

参数	是否必选	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
image	否	string	需要检测的图像base64编码，图像需要是JPG/PNG/BMP 其中之一的格式
url	否	string	图片可以下载的url, 如果url 和image 都提供, 仅使用url

注意：image和url必须填写一个

2.3 响应包格式

返回的结果为：

参数	类型	描述
code	int	服务器错误码, 0为成功
message	string	服务器返回的信息
tags	ImageTag	图像的分类标签ImageTag列表

其中ImageTag 具体内容为：

参数	类型	描述
tag_name	string	返回图像标签的名字
tag_confidence	int	图像标签的置信度,整形范围 0-100,越大置信度越高

示例：

```
{
  "code": 0,
  "message": "success",
  "tags": [
    {
      "tag_name": "天空",
      "tag_confidence": 27
    }
  ],
}
```

```

{
  "tag_name": "女孩",
  "tag_confidence": 47
}
]
}
    
```

3. 鉴权

腾讯云·万象优图通过签名来验证请求的合法性。

签名算法：

1) 拼接签名原文P

```
a=[appid]&b=[bucket]&k=[SecretID]&t=[currentTime]&e=[expiredTime]&l=[urlink]
```

注意:如果开发者使用的是 V1 版本，a 字段为 appid，b 字段的值置空

字段	解释
a	开发者的项目ID，接入万象优图创建空间时系统生成的唯一标示项目的项目ID
b	图片空间名称
k	项目的Secret ID
t	当前时间，UNIX时间戳
e	UNIX时间戳
l	图片url(如果没有置空)

2) 使用 HMAC-SHA1 算法对签名原文进行加密获得加密串C;

3) 对 (C + P) 进行base64编码，获得最终签名串

智能鉴黄接口文档

最近更新时间：2018-05-28 11:01:44

1 体验与说明

1.1 智能鉴黄体验

开发者可以现在[用户体验平台](#)体验智能鉴黄的效果(用户体验平台不需要注册账号)。

1.2 智能鉴黄开发准备

开发者使用智能鉴黄功能之前，需要先注册腾讯云账号，并创建图片空间，添加密钥，具体操作步骤如下：

1. 前往[腾讯云·万象优图控制台](#)注册账号；如果已经注册账号，请跳过此步骤；
2. 在[腾讯云·万象优图控制台](#)创建一个图片空间，获取空间名称（bucket）和项目ID；如果已创建过图片空间，请跳过此步骤；
3. 在[腾讯云·万象优图控制台项目设置](#)中添加密钥，获取SecretID和SecretKey；如果已经添加过密钥，请跳过此步骤。

基本概念：

概念	解释
appid	项目ID, 用于唯一标识接入项目
bucket	开发者添加的空间名称

说明：如果开发者使用的是V1版本，appid为其当时生成的appid

1.3 费用说明

图片智能鉴黄是一项收费服务，收费标准请参考[图片鉴黄收费标准](#)。

2 智能鉴黄CGI

2.1 智能鉴黄接口

智能鉴黄接口采用http协议，支持多URL和多本地图片文件,每个请求最多支持20张图片或url。

接口：http://service.image.myqcloud.com/detection/porn_detect

方法：POST

2.1.1 图片URL鉴黄

请求语法

POST /detection/porn_detect HTTP/1.1

Authorization: Signature

Host: service.image.myqcloud.com

Content-Length: ContentLength

Content-Type: "application/json"

```
{
  "appid": "appid",
  "bucket": "bucket",
  "url_list": [
    "url",
    "url"
  ]
}
```

请求包的http header :

参数	是否必须	描述
Host	是	访问域名, service.image.myqcloud.com
Authorization	是	鉴权签名, 详见下面鉴权章节
Content-Type	是	标准的application/json
Content-Length	是	http body总长度

请求包http body :

参数	是否必须	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
url	是	string	图片url列表

响应http body(json格式) :

参数	是否必须	类型
result_list	json数组	具体查询数据, 内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码, 0为成功

参数	类型	描述
message	string	服务器返回的信息
url	string	当前图片的url
data		具体查询数据，具体见下表

data字段具体内容：

参数	类型	描述
result	int	供参考的识别结果，0正常，1黄图，2疑似图片
confidence	double	识别为黄图的置信度，范围0-100；是normal_score, hot_score, porn_score的综合评分
normal_score	double	图片为正常图片的评分
hot_score	double	图片为性感图片的评分
porn_score	double	图片为色情图片的评分
forbid_status	int	封禁状态，0表示正常，1表示图片已被封禁（只有存储在万象优图的图片才会被封禁）

说明：

- 1、当result=0时，表明图片为正常图片；
- 2、当result=1时，表明该图片是系统判定的100%为违禁的图片；如果该图片存储在万象优图，则会直接被封禁掉；
- 3、当result=2时，表明该图片是疑似图片，即为黄图的可能性很大（目前confidence大于等于83小于91定为疑似图片）。

2.1.2 url鉴黄示例

http请求：

```
POST /detection/porn_detect HTTP/1.1
```

```
Authorization: FCHXddYbhZEBfTeZ0j8mn9Og16JhPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJiUkp3Sk5zbTM
```

```
Host: service.image.myqcloud.com
```

```
Content-Length: 238
```

```
Content-Type: "application/json"
```

```
{
  "appid": 10000379,
  "bucket": "testbucket",
  "url_list": [
    "http://www.bz55.com/uploads/allimg/140805/1-140P5162300-50.jpg",
    "http://img.taopic.com/uploads/allimg/130716/318769-130G60P30462.jpg"
  ]
}
```

响应httpbody（application/json格式）：

```

{
  "result_list": [
    {
      "code": 0,
      "message": "success",
      "url": "http://www.bz55.com/uploads/allimg/140805/1-140P5162300-50.jpg",
      "data": {
        "result": 0,
        "forbid_status": 0,
        "confidence": 12.509,
        "hot_score": 87.293,
        "normal_score": 12.707,
        "porn_score": 0.0
      }
    },
    {
      "code": 0,
      "message": "success",
      "url": "http://img.taopic.com/uploads/allimg/130716/318769-130G60P30462.jpg",
      "data": {
        "result": 0,
        "forbid_status": 0,
        "confidence": 14.913,
        "hot_score": 99.997,
        "normal_score": 0.003,
        "porn_score": 0.0
      }
    }
  ]
}
    
```

2.1.3 图片文件鉴黄

图片文件鉴黄使用HTML表单上传一个或多个文件，文件内容通过多重表单格式（multipart/form-data）编码。

请求语法：

```

POST /detection/porn_detect HTTP/1.1
Content-Type: multipart/form-data;boundary=-----acebdf13572468
Authorization: Signature
Host: service.image.myqcloud.com
Content-Length: ContentLength

-----acebdf13572468
Content-Disposition: form-data; name="appid";

appid
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

bucket
-----acebdf13572468
    
```

```
Content-Disposition: form-data; name="image[0]"; filename="image_1.jpg"
Content-Type: image/jpeg
```

```
image_content
```

```
-----acebdf13572468
```

```
Content-Disposition: form-data; name="image[1]"; filename="image_2.jpg "
Content-Type: image/jpeg
```

```
image_content
```

```
-----acebdf13572468--
```

请求包http header:

参数	是否必须	描述
Host	是	访问域名, service.image.myqcloud.com
Authorization	是	鉴权签名, 详见下面鉴权章节
Content-Type	是	标准的application/json
Content-Length	是	http body总长度

表单域

参数	是否必须	类型	描述
appid	是	uint	业务id
bucket	是	string	图片空间
image	是	image/jpeg	图片文件, 支持多个。参数名须为 "image[0]"、"image[1]"等, 每张图片需指定 filename

响应http body (json 格式)

参数	类型	描述
result_list	json数组	具体查询数据, 内容见下表

result_list (json数组) 中每一项的具体内容

参数	类型	描述
code	int	服务器错误码, 0为成功
message	string	服务器返回的信息
filename	string	当前图片的filename, 与请求包中filename一致
data		具体查询数据, 内容见下表

data字段具体内容：

参数	类型	描述
result	int	供参考的识别结果，0正常，1黄图，2疑似图片
confidence	double	识别为黄图的置信度，范围0-100；是normal_score, hot_score, porn_score的综合评分
normal_score	double	图片为正常图片的评分
hot_score	double	图片为性感图片的评分
porn_score	double	图片为色情图片的评分
forbid_status	int	封禁状态，0表示正常，1表示图片已被封禁（只有存储在万象优图的图片才会被封禁）

说明：

- a) 当result=0时，表明图片为正常图片；
- b) 当result=1时，表明该图片是系统判定的违禁的图片；
- c) 当result=2时，表明该图片是疑似图片，即为黄图的可能性很大（目前confidence大于等于83小于91定为疑似图片）。

2.1.4 文件鉴黄示例

http请求：

```

POST /detection/ porn_detect HTTP/1.1
Content-Type:multipart/form-data;boundary=-----acebdf13572468
Authorization:FCHXddYbhZEBfTeZ0j8mn9Og16JhPTEwMDAwMzc5Jms9QUtJRGVRZDBrRU1yM2J4ZjhRckJiUkp3Sk5zbTN
Host: service.image.myqcloud.com
Content-Length: 61478

-----acebdf13572468
Content-Disposition: form-data; name="appid";

10000379
-----acebdf13572468
Content-Disposition: form-data; name="bucket";

testbucket
-----acebdf13572468
Content-Disposition: form-data; name="image[0]"; filename="1.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\185839ggh0oedgnog04g0b.jpg.thumb.jpg*@>
-----acebdf13572468
Content-Disposition: form-data; name="image[1]"; filename="2.jpg"
Content-Type: image/jpeg

<@INCLUDE *D:\200132svnmybmbhbmgbmga.jpg.thumb.jpg*@>
-----acebdf13572468
    
```

响应httpbody(application/json格式)

```
{
  "result_list": [
    {
      "code": 0,
      "message": "success",
      "filename": "1.jpg",
      "data": {
        "result": 1,
        "forbid_status": 0,
        "confidence": 96.853,
        "hot_score": 0.0,
        "normal_score": 0.0,
        "porn_score": 100.0
      }
    },
    {
      "code": 0,
      "message": "success",
      "filename": "2.jpg",
      "data": {
        "result": 0,
        "forbid_status": 0,
        "confidence": 41.815,
        "hot_score": 19.417,
        "normal_score": 0.077,
        "porn_score": 80.506
      }
    }
  ]
}
```

2.2 鉴权

腾讯云·万象优图通过签名来验证请求的合法性。

2.2.1 签名算法

1 获取签名所需信息

生成签名所需信息包括项目ID（appid），空间名称（bucket,图片资源的组织管理单元），项目的Secret ID和Secret Key。获取这些信息的方法如下：

- 1) 登录 [万象优图-图片空间](#), 进入图片空间；
- 2) 如开发者未创建图片空间，可添加图片空间，获取项目ID（appid），空间名称（bucket）；如果开发者已经创建过空间，则可以直接获取项目ID和空间名称（bucket）；
- 3) 登录 [万象优图-项目设置](#), 进入项目设置；
- 4) 如果开发者未添加密钥，则需添加密钥，获取项目的Secret ID和Secret Key，每个项目最多添加两对密钥；如果已经添加过密钥则直接获取项目的Secret ID和Secret Key。

注：

- (1) 添加图片空间可参考[添加图片空间](#)；

(2) 添加密钥可参考[添加密钥](#)。

2 拼接签名串original

```
a=[appid]&b=[bucket]&k=[SecretID]&t=[currentTime]&e=[expiredTime]
```

注意：如果开发者使用的是V1版本，a字段为appid，b字段的值置空

字段名称	解释
a	开发者的项目ID，接入万象优图创建空间时系统生成的唯一标示项目的项目ID
b	图片空间名称
k	项目的Secret ID
t	当前时间，UNIX时间戳
e	签名过期时间，UNIX时间戳

3 生成签名

(1) 万象优图使用 HMAC-SHA1 算法对请求进行签名；

(2) 签名串需要使用 Base64 编码。

即生成签名的公式如下：

```
SignTmp = HMAC-SHA1(SecretKey, original)
Sign = Base64(SignTmp.original)
```

其中SecretKey为项目的密钥SecretKey，original为1节中拼接好的签名串，首先对original使用HMAC-SHA1算法进行签名，然后将original附加到签名结果的末尾，再进行Base64编码，得到最终的sign。

注：此处使用的是标准的Base64编码，不是ur-safe的Base64编码，请注意。

2.2.2 签名示例

本节介绍生成签名的算法实例，实例中使用PHP语言，如果开发者使用其他与开发，请使用对应的算法。

1 获取签名所需信息

获取得到的签名所需信息如下。

项目ID：1000027

空间名称 (bucket)：test333

```
Secret ID：AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek
Secret Key：oHL5srXW3Fkn8xtugG0BDhUfOOC9DAd
```

2 拼接签名串original

```
a=1000027&b=test333&k=AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek&t=1443434355&e=1443434365
```

```

$appid = "100027";
$bucket = "test333";
$secret_id = "AKIDnX91172Bs2NK4SP9Ad9JDVYpm7Lx2Nek";
$secret_key = "oHL5srXW3Fkn8xtugG0BDhUfOosC9DAAd";
$expired = time() + 10;
$current = time();
$srcStr = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&t='.$current.'&e='.$expired;
    
```

3 生成签名

```

$signStr = base64_encode(hash_hmac('SHA1', $srcStr, $secret_key, true).$srcStr);
echo $signStr."\n";
    
```

最终得到的签名为：

```
xBx0lpW/tdlr1vfksAt3GpfXo9phPEwMDAwMjcmYj10ZXN0MzMzMjms9QUtJRG5YOTExNzJCczJOSzRTUDIBZDIKRfZZcG03'
```

2.3 错误码

错误码	含义
3	错误的请求
4	签名为空
5	签名串错误
6	appid/bucket/url不匹配
7	签名编码失败（内部错误）
8	签名解码失败（内部错误）
9	签名过期
10	appid不存在
11	secretid不存在
12	appid不匹配
13	重放攻击
14	签名失败
15	操作太频繁，触发频控
16	内部错误
17	未知错误
200	内部打包失败
201	内部解包失败

错误码	含义
202	内部链接失败
203	内部处理超时
-1300	图片为空
-1308	url图片下载失败
-1400	非法的图片格式
-1403	图片下载失败
-1404	图片无法识别
-1505	url格式不对
-1506	图片下载超时
-1507	无法访问url对应的图片服务器
-5062	url对应的图片已被标注为不良图片，无法访问（专指存储于腾讯云的图片）

SDK文档

SDK文档

最近更新时间：2017-03-09 11:10:07

1. Android SDK 文档

[Android SDK文档](#)

2. iOS SDK 文档

[iOS SDK文档](#)

3. 服务器SDK文档

[Java SDK说明文档](#) | [PHP SDK说明文档](#) | [Python SDK说明文档](#) | [Nodejs SDK说明文档](#) | [Go SDK说明文档](#) | [C++ SDK说明文档](#) | [C# SDK说明文档](#)

注意：服务端SDK上传接口中的fileid，请参考对应版本的Restful API文档中的上传CGI。各版本的Restful API如下：

[V2加强版本](#) | [V2版本](#) | [V1版本](#)

SDK下载

最近更新时间：2017-03-09 11:12:55

说明：如果您使用的是万象优图2.0，则APPID是以125为前三位的，您需要使用COS的上传SDK，请查看[新版文档](#)。

1. Android SDK

版本号：Android SDK V1.1.3.3

发布时间：2015-12-17

版本说明：

1. 支持查询图片底色
2. 优化代码，修改bug

下载地址：[Android SDK V1.1.3.3](#) | [Android 体验demo](#)

历史版本：[Android SDK历史版本及更新日志](#)

2. iOS SDK

版本号：iOS SDK V1.1.3.3

发布时间：2015-12-17

版本说明：

1. 增加了查询图片底色方法属性（TXYFileInfo 增加了photoRBG属性）
2. 上传SDK添加支持了bitecode 版本

下载地址：[iOS SDK V1.1.3.3](#) | [iOS 体验Demo](#)

历史版本：[iOS SDK历史版本及更新日志](#)

3. 服务器SDK

服务器SDK的版本信息及具体内容详见相应的github。

3.1 Java SDK

github地址：<https://github.com/tencentyun/image-java-sdk>

3.2 PHP SDK

github地址：<https://github.com/tencentyun/image-php-sdk>

3.3 Python SDK

github地址：<https://github.com/tencentyun/image-python-sdk>

3.4 Nodejs SDK

github地址：<https://github.com/tencentyun/image-nodejs-sdk>

3.5 Go SDK

github地址：<https://github.com/tencentyun/image-go-sdk>

3.6 C# SDK

github地址：<https://github.com/tencentyun/image-dotnet-sdk>

3.7 C++ SDK

github地址：<https://github.com/tencentyun/image-cpp-sdk>