

阿里云 对象存储 OSS

数据处理

文档版本：20190212

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或惩罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。未经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
courier 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
##	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
<code>[]或者[a b]</code>]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{}或者{a b}</code> }	表示必选项，至多选择一个。	<code>switch {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 数据处理介绍.....	1
2 图片处理指南.....	3
2.1 快速使用OSS图片服务.....	3
2.2 图片处理访问规则.....	5
2.3 图片缩放.....	8
2.4 图片裁剪.....	13
2.4.1 内切圆.....	13
2.4.2 裁剪.....	14
2.4.3 索引切割.....	16
2.4.4 圆角矩形.....	18
2.5 格式转换.....	19
2.5.1 演进显示.....	19
2.5.2 质量变换.....	21
2.5.3 格式转换.....	22
2.6 图片水印.....	24
2.7 错误响应.....	35
2.8 图片样式.....	36
2.9 图片原图保护.....	40
2.10 数据处理持久化.....	42
2.11 新老版本API和域名使用FAQ.....	44
2.12 老版图片服务手册.....	45
2.12.1 介绍.....	45
2.12.2 基本概念.....	46
2.12.3 访问域名.....	48
2.12.4 图片上传.....	49
2.12.5 错误响应.....	49
2.12.6 图片处理报错<Code>NoSuchKey</Code>.....	51
3 视频截帧.....	54
4 智能媒体管理 (IMM)	56
4.1 快速开始.....	56
4.2 文档预览.....	58
4.3 人脸识别.....	59
4.4 图片识别.....	61

1 数据处理介绍

阿里云OSS为在云上的数据打通了一个处理与使用的快速通道。通过简单的 RESTful 接口，可以在任何时间、任何地点、任何互联网设备上对存储在OSS中的数据进行分析处理。

数据处理包含以下两种：

- 阿里云OSS原生处理服务

阿里云OSS原生处理服务包括图片处理和视频截帧，其中图片处理包括图片的缩略、剪裁、参数调节等，OSS原生处理服务无需开通，默认集成在OSS中，创建完Bucket后即可使用。产生的数据处理费用直接在OSS上结算。

- 智能媒体管理服务

阿里云OSS与智能媒体管理（IMM）深度结合，支持文档预览、文档格式转换、人脸识别、图片分析、二维码识别等丰富的数据分析处理操作。



说明：

智能媒体管理服务需要先开通 IMM，并在OSS控制台中一键绑定。产生的数据处理费用在 IMM 上结算。

触发规则

数据处理有以下两种触发方式：

- GET方式触发

当操作用于即时处理返回结果时，可以采用GET方式触发。参数在QueryString中传递。

GET方式触发：`http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100`



：

针对图片、视频处理，如果原始数据权限为Private，需要添加URL签名。如果是 IMM 触发的计算请求，都需要添加签名。

- POST方式触发

当操作需要做处理结果写回OSS时，可以采用POST方式触发。参数放置在Body中。

POST方式触发

```
POST /ObjectName?x-oss-process HTTP/1.1
Content-Length: ContentLength
Content-Type: ContentType
```

```
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
x-oss-process=image/resize,w_100|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

现将一张图裁切成圆形，示例如下：

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-
process=image/circle,r_100
```

参数x-oss-process=module/action,parame_value说明如下：

- x-oss-process: 数据处理操作的固定参数名。当携带该参数时，可以做相应的数据处理操作。
- module: 操作模块，这里OSS原生的操作模块有图片处理模块image，视频处理模块video。
 - 。IMM 提供统一的处理模块即imm。
- action: 数据处理，如上面示例中为circle圆形裁切。
- parame: 数据处理操作所对应的参数，采用key_value形式，通过下划线（_）分割。当某些操作参数唯一时，可以单独输出value，具体以各个操作中的定义为准。

数据处理操作

模块	功能	操作参数	参考文档
image	图片缩放	resize	点击查看
	图片裁剪	crop	点击查看
	图片旋转	rotate	点击查看
	图片锐化调节	sharpen	点击查看
	图片格式转换	format	点击查看
	图片质量调节	quality	点击查看
	图片水印	watermark	点击查看
video	视频截帧	snapshot	点击查看
imm (需单独开通)	文档预览	docprivew	点击查看
	图片场景识别	imagetag	点击查看
	图片二维码识别	imageqrcode	点击查看
	人脸识别	facedetect	点击查看

2 图片处理指南

2.1 快速使用OSS图片服务

阿里云OSS图片处理服务 (Image Processing, 简称 IMG) , 是阿里云OSS对外提供的海量、安全、低成本、高可靠的图片处理服务。您可以将原始图片上传保存在 OSS 上，通过简单的 RESTful 接口，在任何时间、任何地点、任何互联网设备上对图片进行处理。图片处理服务提供图片处理接口，图片上传请使用 OSS 上传接口。基于 IMG，您可以搭建出跟图片相关的服务。



说明:

开通OSS服务后，图片服务也随之开通，无需单独开通。

图片服务基础功能

图片处理提供以下功能：

- 获取图片信息
- 图片格式转换
- 图片缩放、裁剪、旋转
- 图片添加图片、文字、图文混合水印
- 自定义图片处理样式
- 通过管道顺序调用多种图片处理功能



:

图片服务使用限制请参考[使用限制](#)。

历史版本说明

图片处理目前提供两版API接口：新版接口和老版接口。



说明:

本文档介绍的是新版接口的功能使用说明，老版接口的功能今后不再更新。使用兼容详细说明，参考[新老版本API和域名使用FAQ](#)。

快速使用图片服务

- 新建图片样式
1. 登录[OSS管理控制台](#)。

2. 单击您的Bucket名字，进入Bucket概览页。
3. 在概览页中单击图片处理，然后单击新建样式。
4. 在新建样式窗口中，可以选择基本编辑，使用图形化的操作来编辑图片样式。也可以选择高级编辑，使用SDK或者处理参数的方式编辑图片样式。

基本编辑页面的各项参数详细说明如下：

- 规则名称：新建图片样式的名称。

命名规则如下：

- 长度为 1-64 个字符。
- 只能包含数字、大小写字母、下划线（_）、短横线（-）以及小数点（.）。
- 格式转换：有原图格式、jpg、png等图片格式供选择。
- 渐进显示：可以选择是否开启渐进显示。
- 自适应方向：可以选择是否开启自适应方向。

建议默认开启，开启后会根据图片中 EXIF 旋转信息先旋转后进行缩略。

- 图片质量：有相对质量、绝对质量和不压缩三种方式供选择。
- 缩略方式：有不使用缩略、等比例缩小、等比例放大和指定宽高缩放四种方式供选择。



说明：

“长边”是指原尺寸与目标尺寸的比值大的那条边，“短边”同理。例如，原图400x200，缩放为800x100，由于 $400/800=0.5$ ， $200/100=2$ ， $0.5 < 2$ ，所以在这个缩放中200那条是长边，400那条是短边。

- 图片亮度：设置图片亮度。
- 图片对比度：设置图片对比度。
- 图片锐化：设置图片是否需要锐化。
- 图片模糊：设置图片是否需要模糊。

开启图片模糊选项后，您可以设置图片的模糊半径以及模糊标准差。

- 图片旋转：设置图片的旋转角度。
- 水印：有不使用水印、图片水印和文字水印三种方式供选择。

5. 编辑好图片样式后单击确定保存样式。

高级编辑页面的各项参数详细说明如下：

- 规则名称：与基本编辑中的名称命名规则相同。
- 代码：您可以输入 API 代码对图片进行高级编辑。

API 代码举例如下：

- `image/resize,w_200`
- `image/crop,w_100,h_100/rounded-corners,r_10/format,png`



说明：

只能使用新版 API 编辑，详情请参考本文档的[历史版本说明](#)。

图片样式新建好之后您就可以通过OSS将您新建的图片样式应用到您的图片上了。

- 使用图片样式

1. 进入[OSS 管理控制台](#)。
2. 单击您的Bucket名称，进入Bucket概览页。
3. 单击文件管理，选择一张已上传的图片或者上传新图片，然后单击图片名称打开图片预览页。
上传新图片，请参考[上传文件](#)。
4. 从图片样式下拉列表中选择一个图片样式。

在文件预览窗口中即刻看到经过图片处理后的图片效果。带有图片样式的公网访问地址也会同步生成，只需单击复制文件URL就可获取到文件访问地址。

- 视频演示

以下视频详细演示了图片样式处理操作：<https://cloud.video.taobao.com/play/u/2955313663/p/1/e/6/t/1/50079046055.mp4>

2.2 图片处理访问规则

图片服务都是使用标准的 HTTP 的 GET 请求来访问的，所有的处理参数也在 URL 的 `QueryString` 中。

通过处理参数来请求缩略图

如果用户对原图进行一定的处理再返回，可有两种形式：

- 通过URL的形式

通过三级域名访问：`http://bucket.<endpoint>/object?x-oss-process=image/action,parame_value`

- `Bucket`：用户的 IMG 频道。
- `endpoint`：用户的 Bucket 所在数据中心的访问域名。

- **Object**: 在 IMG 中, 用户操作图片的基本数据单元是 Object。即 OSS 对应的 Object, 单个Object (即每张图片) 允许的最大大小是 20MB。
- **action**: 用户对图片做的操作。
- **param**: 用户对图片做的操作所对应的参数。

- 多个 action 之间组合形式

多个 action 之间效果顺序执行, 格式为image/action1,param_value1/action2,param_value2, 例如: image/resize,w_200/rotate,90表示图片先进行宽为 200 的缩放, 再进行 90 度的旋转。

示例

假如请求 Bucket 是 *image-demo*, 该 Bucket 区域在 #1, 对应的域名是 *oss-cn-hangzhou.aliyuncs.com*, 请求图片是 *example.jpg*, 对图片进行按目标宽度 200 进行缩略。

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200
```

如果是https方式, 访问形式如下:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200
```

通过用户自定义域名访问链接如下:

```
http://userdomain/object?x-oss-process=image/action,param_value
```

通过样式来请求缩略图

- **Style (样式)**

图片处理可将用户的图片处理操作和参数保存成一个别名, 即样式。利用样式功能后, 原本一系列操作, 只需要用一个很短的URL就能实现相同的效果。

- 一个 Channel 下面有多个样式, 目前一个 Channel 允许最多有 50 个样式。
- 样式适应于 Channel 下面的 Object 图像变化操作。假如在 A Channel 下面有样式, 名称为 abc, 样式内容是 100w.jpg (按宽缩略成 100, 保存成 jpg 格式) 那么 A Channel 下面所有的 Object 都能使用样式 abc, 实现缩略成 100w.jpg 的效果。
- 样式的作用范围只在一个 Channel 下, 即 A Channel 不能使用 B Channel 的样式。

Style 命名规范:

- 长度为 1-63 个字符。

- 只能包含数字、大小写字母、下划线（_）、短横线（-）以及小数点（.）。
- Channel（频道）

Channel 是 IMG 上的命名空间，也是计费、权限控制、日志记录等高级功能的管理实体。

IMG 名称在整个图片处理服务中具有全局唯一性，且不能修改。一个用户最多可创建10个 Channel，但每个 Channel 中存放的 Object 的数量没有限制。

图片服务的数据中心和 OSS 的数据中心相对应。用户在 OSS 的某个数据中心上创建一个 Bucket，再选择开通图片服务，对应的 Channel 也属于该数据中心。目前 Channel 跟 OSS 的 Bucket 相对应，即用户只能创建与自己在 OSS 上 Bucket 同名的 Channel。

Channel 命名规范：

- 只能包括小写字母，数字，短横线（-）。
- 必须以小写字母或者数字开头和结尾。
- 长度必须在 3-63 字节之间。

为简化使用，用户可以将特定的处理方法保存为样式，这样以后调用同样的处理方法只需指定某个样式即可。使用样式来进行图片处理的 URL 形式如下：

`http://userdomain/object?x-oss-process=style/name`

- **示例**

可以将上述处理参数保存成样式 `style-example`。假如请求 Bucket 为 `image-demo`，该 Bucket 区域在 ##1，对应的域名是 `oss-cn-hangzhou.aliyuncs.com`，请求图片是 `example.jpg`，访问图片的样式是 `style-example`，那么访问形式如下：

]

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/style-example
```

如果是 https 方式，访问形式如下：

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/style-example
```

使用 SDK 方式

除了 public bucket 可以直接通过url方式进访问，对于 private 的文件，我们一般采用 SDK 的方式来访问。因为图片处理服务都是 GET 操作，所以在 Get Object 基础上面添加 process 参数即可。

以 Python SDK 为例：

```
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret),  
endpoint, bucket_name)  
key = 'example.jpg'  
new_pic = 'new-example.jpg'  
process = "image/resize,m_fixed,w_100,h_100" //对于图片进行强制缩略  
bucket.get_object_to_file(key, new_pic, process=process)
```

OSS SDK 使用图片服务，请参看 SDK 文档中的图片服务部分。部分 SDK 图片处理的链接见下表：

SDK	图片处理文档	使用示例
Java SDK	图片处理	ImageSample.java
Python SDK	图片处理	image.py
C# SDK	图片处理	ImageProcessSample.cs
PHP SDK	图片处理	Image.php
JS SDK	图片处理	object.test.js
C SDK	图片处理	oss_image_sample.c

图片处理限制

- 图片处理支持的格式有：jpg、png、bmp、gif、webp、tiff。
- 指定缩略图宽度或者高度时，在等比缩放的情况下，都会默认进行单边的缩放。在固定宽高的模式下，会默认在宽高一样的情况下进行缩放。
- 对缩略后的图片大小有限制，目标缩略图的宽与高的乘积不能超过 4096x4096，且单边的长度不能超过 4096x4。
- 调用 `resize`，默认不允许放大。如果请求图片比原图大，那么返回的仍然是原图。如果想得到放大的图片，需要增加参数调用 `limit_0`。
- gif、webp 动图由于资源消耗原因目前只支持一级处理。如不能在缩略操作后直接进行图片裁剪。

2.3 图片缩放

上传到阿里云OSS的图片处理功能支持处理上传至OSS的图片文件，本文主要介绍如何进行图片缩放。



说明：

图片处理支持的格式：jpg、png、bmp、gif、webp、tiff。

图片缩放参数

操作名称: `resize`

· 指定宽高缩放

名称	描述	取值范围
<code>m</code>	指定缩略的模式: - <code>lfit</code> : 等比缩放, 限制在指定w与h的矩形内的最大图片。 - <code>mfit</code> : 等比缩放, 延伸出指定w与h的矩形框外的最小图片。 - <code>fill</code> : 固定宽高, 将延伸出指定w与h的矩形框外的最小图片进行居中裁剪。 - <code>pad</code> : 固定宽高, 缩略填充。 - <code>fixed</code> : 固定宽高, 强制缩略。	<code>lfit</code> 、 <code>mfit</code> 、 <code>fill</code> 、 <code>pad</code> 、 <code>fixed</code> , 默认为 <code>lfit</code> 。
<code>w</code>	指定目标缩略图的宽度。	1-4096
<code>h</code>	指定目标缩略图的高度。	1-4096
<code>l</code>	指定目标缩略图的最长边。	1-4096
<code>s</code>	指定目标缩略图的最短边。	1-4096
<code>limit</code>	指定当目标缩略图大于原图时是否处理。 值是 1 表示不处理; 值是 0 表示处理。	0/1, 默认是 1
<code>color</code>	当缩放模式选择为 <code>pad</code> (缩略填充) 时, 可以选择填充的颜色(默认是白色)参数的填写方式: 采用 16 进制颜色码表示, 如 <code>00FF00</code> (绿色)。	[000000-FFFFFF]

· 按比例缩放

名称	描述	取值范围
<code>p</code>	倍数百分比。 小于 100, 即是缩小, 大于 100 即是放大。	1-1000

图片缩放注意事项

· 对于原图:

- 图片格式只能是: `jpg`、`png`、`bmp`、`gif`、`webp`、`tiff`。
- 文件大小不能超过 20MB。
- 使用图片旋转时图片的宽或者高不能超过 4096。
- 原图单边大小不能超过30,000。

- 对于缩略图：对缩略后的图片大小有限制，目标缩略图宽与高的乘积不能超过 4096x4096，且单边长度不能超过 4096x4。
- 当只指定宽度或者高度时，在等比缩放的情况下，都会默认进行单边的缩放。在固定宽高的模式下，会默认宽高一样的情况下进行缩略。
- 如果只指定宽度或者高度，原图按原图格式返回。如果想保存成其他格式，详细可以查看[质量变换及格式转换](#)。
- 调用 `resize`，默认是不允许放大。即如果请求的图片比原图大，那么返回的仍然是原图。如果想取到放大的图片，即增加参数调用 `limit_0` (如：`https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_500,limit_0`)

图片缩放示例

- 单边缩略
 - 按宽高：将图缩略成高度为 100，宽度按比例处理。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,h_100



- 按长边短边：将图最长边限制在 100 像素，短边按比例处理。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,l_100



- 强制宽高缩略：将图强制缩略成宽度为 100，高度为 100。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fixed,h_100,w_100



- 等比缩放，限定在矩形框内
 - 将图缩略成宽度为 100，高度为 100，按长边优先。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_lfit,h_100,w_100



- 将图缩略成宽度为 100，高度为 100，按长边优先，将图片保存成 png 格式。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_lfit,h_100,w_100/format,png



- 等比缩放，限定在矩形框外
 - 将图缩略成宽度为 100，高度为 100，按短边优先。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_mfit,h_100,w_100



- 固定宽高，自动裁剪
 - 将图自动裁剪成宽度为 100，高度为 100 的效果图。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fill,h_100,w_100



- 固定宽高，缩略填充

- 将原图指定按短边缩略 100x100，剩余的部分以白色填充。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_pad,h_100,w_100



- 将图按短边缩略到 100x100，然后按红色填充。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_pad,h_100,w_100,color_FF0000



- 将图按比例缩略到原来的 1/2。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,p_50



SDK 示例

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)

- [Browser.js](#)

2.4 图片裁剪

2.4.1 内切圆

用户可以将图片只保存圆形图案，如果图片的最终格式是 png、webp、bmp 等支持透明通道的图片，那么图片非圆形区域的地方将会以透明填充。如果图片的最终格式是 jpg，那么非圆形区域是以白色进行填充。

参数

操作名称: `circle`

参数	描述	取值
<code>r</code>	从图片取出的圆形区域的半径	半径 <code>r</code> 不能超过原图的最小边的一半。如果超过，则圆的大小仍然是原圆的最大内切圆。

注意事项

- 如果图片的最终格式是 png、webp、bmp 等支持透明通道的图片，那么图片非圆形区域的地方将会以透明填充。如果图片的最终格式是 jpg，那么非圆形区域是以白色进行填充。推荐保存成 png 格式。
- 如果指定半径大于原图最大内切圆的半径，则圆的大小仍然是图片的最大内切圆。

示例

- 裁剪半径是 100，保存圆是原来大小。如果保存成 jpg 格式，外围是以白色填充。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100



- 裁剪半径是 100，保存圆是能包含圆的最小正方形。如果保存成 png 格式，外围是以透明色填充。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100/format,png



2.4.2 裁剪

可以裁剪图片，指定裁剪的起始点以及裁剪的宽高来决定裁剪的区域。

参数

操作名称: `crop`

名称	描述	取值范围
w	指定裁剪宽度	[0-图片宽度]
h	指定裁剪高度	[0-图片高度]
x	指定裁剪起点横坐标（默认左上角为原点）	[0-图片边界]
y	指定裁剪起点纵坐标（默认左上角为原点）	[0-图片边界]
g	设置裁剪的原点位置，由九宫格的格式，一共有九个地方可以设置，每个位置位于每个九宫格的左上角	[nw, north, ne, west, center, east, sw, south, se]

裁剪原点位置参数示意图：

nw	north	ne
west	center	east
sw	south	se

注意事项

- 如果指定的起始横纵坐标大于原图，将会返回错误：BadRequest，错误内容是：Advance cut 's position is out of image.
- 如果从起点开始指定的宽度和高度超过了原图，将会直接裁剪到原图结尾。

使用示例

- 裁剪图从起点(100, 50)到图的结束

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50



- 裁剪图从起点(100, 50)到裁剪100x100的大小

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50,w_100,h_100



- 裁剪图右下角200x200的大小

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_0,y_0,w_200,h_200,g_se



- 裁剪图右下角200x200的大小，再往下10x10

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_10,y_10,w_200,h_200,g_se



2.4.3 索引切割

将图片分成 x, y 轴，按指定长度 (length) 切割，指定索引 (index)，取出指定的区域。

参数

操作名称： `indexcrop`

参数	描述	取值
<code>x</code>	进行水平切割，每块图片的长度。 <code>x</code> 参数与 <code>y</code> 参数只能任选其一。	[1,图片宽度]

参数	描述	取值
y	进行垂直切割，每块图片的长度。x 参数与 y 参数只能任选其一。	[1,图片高度]
i	选择切割后第几个块。（0表示第一块）	[0,最大块数)。如果超出最大块数，返回原图。

注意事项

如果指定的索引大于切割后范围，将返回原图。

示例

对图片 x 轴按 100 平分，取出第一块。

- 对图片 x 轴按 100 平分，取出第一块。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,x_100,i_0



- 对图片 x 轴按 100 平分，取出第一百块，仍然是原图。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,x_100,i_100



2.4.4 圆角矩形

您可以把图片保存成圆角矩形，并指定圆角的大小。

参数

操作名称: `rounded-corners`

参数	描述	取值
<code>r</code>	将图片切出圆角，指定圆角的半径。	[1, 4096] 生成的最大圆角的半径不能超过原图的最小边的一半。

注意事项

- 如果图片的最终格式是 png、webp、bmp 等支持透明通道的图片，那么图片非圆形区域的地方将会以透明填充。如果图片的最终格式是 jpg，那么非圆形区域是以白色进行填充。推荐保存成 png 格式。
- 如果指定半径大于原图最大内切圆的半径，则圆角的大小仍然是图片的最大内切圆。

示例

- 裁剪圆角半径是 30，格式是 jpg。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rounded-corners,r_30



- 图片先自动裁剪成 100x100，然后保存成圆角半径是 10，格式是 png。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,w_100,h_100,rounded-corners,r_10/format,png



SDK 示例

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)

2.5 格式转换

2.5.1 渐进显示

图片格式为 jpg 时有两种呈现方式：

- 自上而下的扫描式
- 先模糊后逐渐清晰（在网络环境比较差时明显）

默认保存为第一种，如果要指定先模糊后清晰的呈现方式，请使用渐进显示参数。

参数

操作名称: `interlace`

名称	描述	取值范围
<code>[value]</code>	1 表示保存成渐进显示的 jpg 格式0 表示保存成普通的 jpg 格式	[0, 1]



说明:

此参数只有当效果图是 jpg 格式时才有意义。

示例

- 将 png 格式的图片保存成渐进显示的 jpg 格式。

<http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/format,jpg/interlace,1>



- 将图片缩略成宽度是200, 并且保存成渐进显示的 jpg 格式。

<http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/format,jpg/interlace,1>



2.5.2 质量变换

如果图片保存成 jpg 或 webp, 可以支持质量变换。

参数

操作名称: `quality`

名称	描述	取值范围
<code>q</code>	决定图片的相对质量, 对原图按照 $q\%$ 进行质量压缩。如果原图质量是 100%, 使用 $90q$ 会得到质量为 90% 的图片; 如果原图质量是 80%, 使用 $90q$ 会得到质量 72% 的图片。 只能在原图是 jpg 格式的图片上使用, 才有相对压缩的概念。如果原图为 webp, 那么相对质量就相当于绝对质量。	1-100
<code>Q</code>	决定图片的绝对质量, 把原图质量压到 $Q\%$, 如果原图质量小于指定数字, 则不压缩。如果原图质量是 100%, 使用 “ $90Q$ ” 会得到质量 90% 的图片; 如果原图质量是 95%, 使用 “ $90Q$ ” 还会得到质量 90% 的图片; 如果原图质量是 80%, 使用 “ $90Q$ ” 不会压缩, 返回质量 80% 的原图。 只能在保存格式为 jpg/webp 效果上使用, 其他格式无效果。 如果同时指定了 <code>q</code> 和 <code>Q</code> , 按 <code>Q</code> 来处理。	1-100

注意事项

如果不填 `Q` 或者 `q` 这两个参数, 有可能会导致图片占用空间变大。如明确想得到一个质量固定的图片, 请采用 `Q` 参数。

示例

- 将原图缩略成 `100w_100h`, 相对原图质量的 80% 的 jpg 图。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100,h_100/quality,q_80



- 将原图缩略成 100w_100h，绝对质量为80的 jpg 图。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100,h_100/quality,Q_80



2.5.3 格式转换

可以将图片转换成对应格式，包括 jpg、png、bmp、webp、gif、tiff。默认不填格式，是按原图格式返回。

参数

名称	描述
jpg	将原图保存成 jpg 格式，如果原图是 png、webp、bmp 存在透明通道，默认会把透明填充成白色。
png	将原图保存成 png 格式。
webp	将原图保存成 webp 格式。
bmp	将原图保存成 bmp 格式。
gif	将 gif 格式保存成 gif 格式，非 gif 格式是按原图格式保存。
tiff	将原图保存成 tiff 格式。

注意事项

- 对于普通缩略请求，建议 format 参数放到处理参数串最后，例如：image/resize,w_100/format,jpg。
- 对于缩略+水印的请求，建议 format 参数跟缩略参数放在一起，例如：image/resize,w_100/format,jpg/watermark,...
- 保存成 jpg 格式时，默认是保存成标准型的 jpg (Baseline JPEG)，如果想指定是渐进式 JPEG (Progressive JPEG)，可以指定参数 interlace，详见[渐进显示](#)。

示例

- 将 png 保存成 jpg 格式。

请求 URL: <http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/format,jpg>



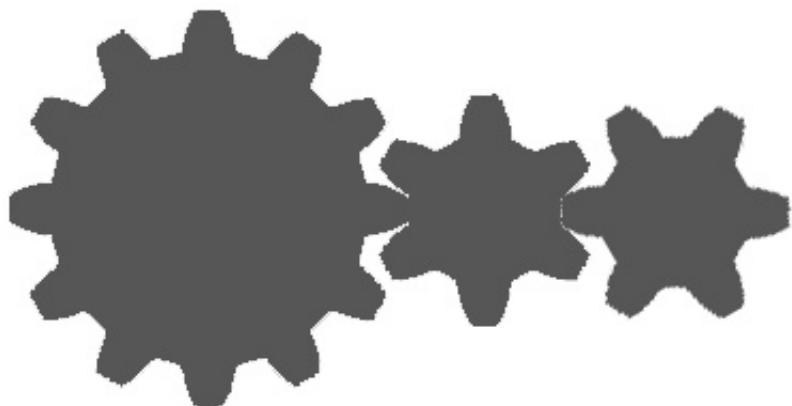
- 将 png 保存成 jpg 格式，并且该 jpg 格式是支持渐变显示。

请求 URL: <http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/interlace,1/format,jpg>



- 将一张 gif 图片保存后成 jpeg 格式。

请求 URL: <http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/format,jpg>



缩略成宽度为200的缩略图。

请求url: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/resize,w_200/format,gif



- 将一张gif图片保存后成动态webp格式。

请求URL: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/resize,w_200/format,webp

2.6 图片水印

您可以在图片上设置另外一张图片或者文字做为水印。



说明:

图片水印的水印图只能使用当前存储空间内的图片，网络图片需上传至当前存储空间内方可使用。

参数

操作名称: `watermark`

基础参数

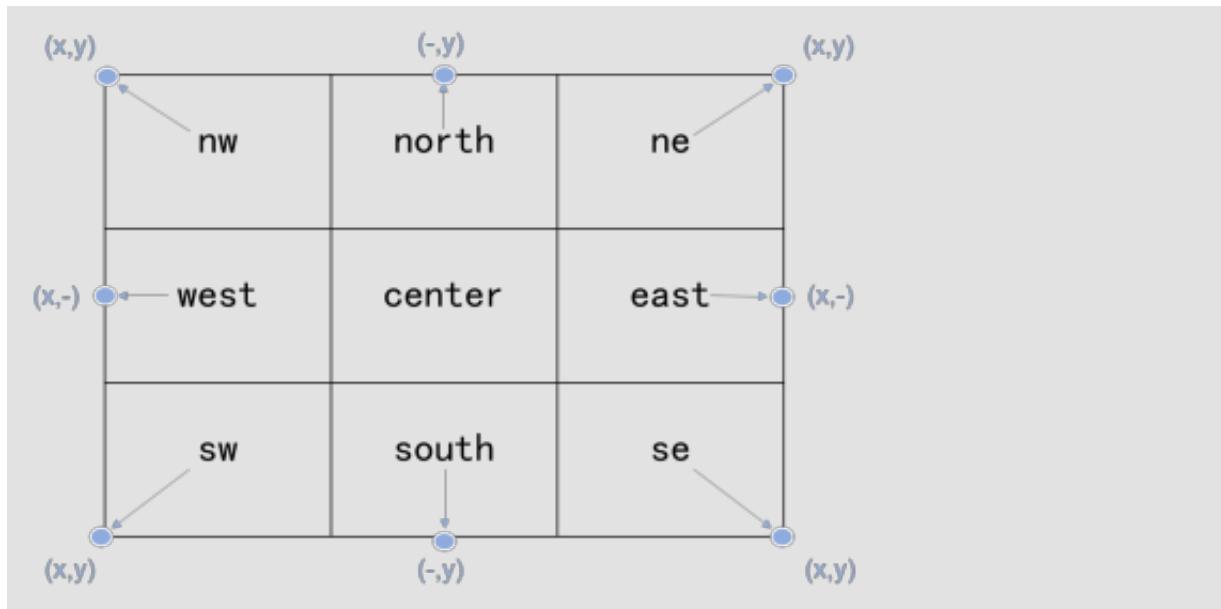
名称	描述	参数类型
t	参数意义: 透明度, 如果是图片水印, 就是让图片变得透明, 如果是文字水印, 就是让水印变透明。 默认值: 100, 表示 100 % (不透明) 取值范围: [0-100]	可选参数
g	参数意义: 位置, 水印打在图的位置, 详情参考下方区域数值对应图。 取值范围: [nw,north,ne,west,center,east,sw,south,se]	可选参数

名称	描述	参数类型
x	参数意义：水平边距，就是距离图片边缘的水平距离，这个参数只有当水印位置是左上，左中，左下，右上，右中，右下才有意义。 默认值：10 取值范围：[0 – 4096] 单位：像素 (px)	可选参数
y	参数意义：垂直边距，就是距离图片边缘的垂直距离，这个参数只有当水印位置是左上，中上，右上，左下，中下，右下才有意义 默认值：10 取值范围：[0 – 4096] 单位：像素(px)	可选参数
voffset	参数意义：中线垂直偏移，当水印位置在左中，中部，右中时，可以指定水印位置根据中线往上或者往下偏移 默认值：0 取值范围：[-1000, 1000] 单位：像素(px)	可选参数



说明：

- 水平边距、垂直边距、中线垂直偏移不仅可以调节水印在图片中的位置，而且当图片存在多重水印时，也可以调节两张水印在图中的布局。
- 用到的URL安全的Base64位编码可以参考文档下方的解释。
- 区域数值以及每个区域对应的基准点如下图。



图片水印参数

名称	描述	参数类型
image	<p>参数意义：水印图片为当前的Bucket下Object，直接针对Object名称进行base64编码。</p> <p> 说明： 内容必须是URL安全base64编码 encodedObject = url_safe_base64_encode(object) 如object为”panda.png”，编码过后的内容为”cGFuZGEucG5n”</p>	必选参数

水印图片预处理

用户在打水印时，可以对水印图片进行预处理，支持的预处理操作有：图片缩放，图片裁剪（不支持内切圆），图片旋转（具体内容请直接查看文档相关章节）。在“resize”操作下还额外支持一个参数：P（大写P），表示水印图片按主图的比例进行处理，取值范围为[1, 100]，表示百分比。

预处理示例

设置了P_10，当主图是100x100，水印图片大小就为10x10，当主图变成了200x200，水印图片大小就为20x20。如果生成的图片大小不一样，而使用相同的水印处理参数，就会导致一些小图，水印图片过大。或者一些大图，水印图片过小。增加P参数，就可以解决这个问题。采用P参数，IMG会根据主图的大小来动态调整水印图片的大小。

**说明:**

对panda.png按30%缩放。那么水印文件是: panda.png?x-oss-process=image/resize,P_30 (经过URL安全base64编码后是:
cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA)

如果水印操作是: watermark=1&object=cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10 (右下角打水印) 原图按宽度是400, 需要缩略, 再打上述水印的示例:

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_400/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10



如果原图按宽度300缩略, 再打上上述水印的示例:

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10



文字水印参数

名称	描述	参数类型
text	<p>参数意义：表示文字水印的文字内容(必须编码)</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  说明： 必须是URL安全base64编码 encodeText = url_safe_base64_encode(fontText) 最大长度为64个字符(即支持汉字最多20个左右) </div>	必选参数
type	<p>参数意义：表示文字水印的文字类型(必须编码)</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  说明： 必须是URL安全base64编码 encodeText = url_safe_base64_encode(fontType) 取值范围：见下表（文字类型编码对应表） 默认值：wqy-zenhei (编码后的值：d3F5LXplbmhlaQ) </div>	可选参数
color	<p>参数意义：文字水印文字的颜色</p> <p>参数的构成必须是：六个十六进制数，如：000000表示黑色。000000每两位构成RGB颜色，FFFFFF表示的是白色</p> <p>默认值：000000黑色</p>	可选参数
size	<p>参数意义：文字水印文字大小(px)</p> <p>取值范围：(0, 1000]</p> <p>默认值：40</p>	可选参数
shadow	<p>参数意义：文字水印的阴影透明度</p> <p>取值范围：(0,100]</p>	可选参数
rotate	<p>参数意义：文字顺时针旋转角度</p> <p>取值范围：[0,360]</p>	可选参数

名称	描述	参数类型
fill	参数意义：进行水印铺满的效果 取值范围：[0,1]， 1表示铺满， 0表示效果无效	可选参数

文字类型编码对应表

参数值	中文意思	URL安全base64编码后的值	备注
wqy-zenhei	文泉驿正黑	d3F5LXplbmhlaQ==	根据RFC，可省略填充符=变为d3F5LXplbmhlaQ
wqy-microhei	文泉微米黑	d3F5LW1pY3JvaGVp	
fangzhengshusong	方正书宋	ZmFuZ3poZW5nc2h1c29uZw==	根据RFC，可省略填充符=变为ZmFuZ3poZW5nc2h1c29uZw
fangzhengkaiti	方正楷体	ZmFuZ3poZW5na2FpdGk=	根据RFC，可省略填充符=变为ZmFuZ3poZW5na2FpdGk
fangzhengheiti	方正黑体	ZmFuZ3poZW5naGVpdGk=	根据RFC，可省略填充符=变为ZmFuZ3poZW5naGVpdGk
fangzhengfangsong	方正仿宋	ZmFuZ3poZW5nZmFuZ3Nvbmc=	根据RFC，可省略填充符=变为ZmFuZ3poZW5nZmFuZ3Nvbmc
droidsansfallback	DroidSansFallback	ZHJvaWRzYW5zZmFsbGJhY2s=	根据RFC，可省略填充符=变为ZHJvaWRzYW5zZmFsbGJhY2s

文图混合

名称	描述	参数类型
order	参数意义：文字，图片水印前后顺序 取值范围：[0, 1] order = 0 图片在前(默认值); order = 1 文字在前	可选参数
align	参数意义：文字、图片对齐方式 取值范围：[0, 1, 2] align = 0 上对齐(默认值) align = 1 中对齐 align = 2 下对齐	可选参数
interval	参数意义：文字和图片间的间距 取值范围: [0, 1000]	可选参数

URL安全的Base64位编码

在图片处理服务里会有很多参数需要变成Base64位编码，参考[RFC4648](#)。注意这里的URL 安全Base64位编码只是用在水印操作某些特定参数(文字水印的文字内容，文字颜色，文字字体及图片水印的水印object)里，不要将其用来签名字符串(Signature)的内容。编码的格式是：

- 先将内容编码成Base64结果；
- 将结果中的加号“+”替换成中划线“-”；
- 将结果中的斜杠“/”替换成下划线“_”；
- 将结果中尾部的“=”号全部保留；

以Python为例子

```
import base64
input='wqy-microhei'
print(base64.urlsafe_b64encode(input))
```

示例

- 下面URL的含义，是example.jpg加上水印文件为:panda.png (panda.png 经过URL安全base64编码后是：cGFuZGEucG5n)

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/auto-orient,1/quality,q_90/format,jpg/watermark,image_cGFuZGEucG5n,t_90,g_se,x_10,y_10



- 对panda.png按宽度是50缩放。那么水印文件是: panda.png?x-oss-process=image/resize,w_50 (经过URL安全base64编码后是:cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=)

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/auto-orient,1/quality,q_90/format,jpg/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA_=,t_90,g_se,x_10,y_10



- 最简单水印: 文字内容是: Hello, 图片服务

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/watermark,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ



- 字体是文泉驿正黑，字体大小是40，颜色是白色(#FFFFFF)，文字阴影是50，文字水印内容是：Hello, 图片服务！，水印位置是：右下，水平边距是：10，中线垂直偏移是：10

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/watermark,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ,color_FFFFFF,shadow_50,t_100,g_se,x_10,y_10



- 文图混合水印，文字内容是：Hello, 图片服务！阴影是50，位置在右下角，图片object是panda.png。水平边距和垂直边距都是10，水印透明是100，排版方式是图片前，对齐方式是下对齐，间距是10

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/auto-orient,1/quality,q_90/format,jpg/watermark,image_cGFuZGEucG5nP3gtb3NzLXBByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjU,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ,color_FFFF,shadow_50,order_0,align_2,interval_10,t_100,g_se,x_10,y_10



- 如果需要对原始图片加上多个水印，可以通过多个处理action组合（即image/action1,param_value1/action2, param_value2）来实现。示例：

```
x-oss-process=image/watermark,image_cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_nw,x_10,y_10/watermark,image_cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2VzczIpbWFnZS9yZXNpemUsUF8yMA,g_se,x_10,y_10/watermark,image_cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2VzczIpbWFnZS9yZXNpemUsUF8yMA,g_se,x_100,y_100/watermark,image_cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_west,x_10,y_10/watermark,image_cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2VzczIpbWFnZS9yZXNpemUsUF8yMA,g_se,x_100,y_100/watermark,image_cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_west,x_100,y_100/watermark,text_b3Nz5paH5a2X5rC05Y2wMQ,size_20,x_10,y_200/watermark,text_b3Nz5paH5a2X5rC05Y2wMg,color_0000b7,size_20,g_sw,x_100,y_100/watermark,text_b3Nz5paH5a2X5rC05Y2wMw,color_b3b324,size_20,g_ne,x_10,y_10/watermark,text_b3Nz5paH5a2X5rC05Y2wNA,color_b3b324,size_20,g_east,x_10,y_10
```



**说明:**

上图水印的内容为：在原始图片上增加了9个水印。包含：4个不同的文本水印和5个图片水印。5个图片水印中，用到的水印图有3张。



：

- 一张图片上，最多支持3张不同的水印图做水印。
- 同一张水印图，可以重复使用，但必需放在不同位置。
- 各个水印位置，不能完全重叠。

SDK 示例

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)

2.7 错误响应

当用户访问图片处理服务出现错误的时候，图片处理服务会返回给用户相应的错误码和错误信息，以帮助用户定位与处理问题。

图片处理服务错误的响应格式

错误响应的消息体例子：

```
<Error>
  <Code>BadRequest</Code>
  <Message>Input is not base64 decoding.</Message>
  <RequestId>52B155D2D8BD99A15D0005FF</RequestId>
  <HostId>userdomain</HostId>
</Error>
```

错误包含以下元素：

- **Code**：图片处理服务返回给用户的错误码。
- **Message**：图片处理服务给出的详细错误信息。

- RequestId: 用以标识错误请求的唯一UUID，在无法解决问题时候，可以使用此错误ID发送给图片处理服务的工程师去定位错误的原因。
- HostId: 用来标识访问的图片处理服务集群。

图片处理服务的错误码

错误码	描述	HTTP 状态码
InvalidArgument	参数错误	400
BadRequest	错误请求	400
MissingArgument	缺少参数	400
ImageTooLarge	图片大小超过限制	400
WatermarkError	水印错误	400
AccessDenied	拒绝访问	403
SignatureDoesNotMatch	签名不匹配	403
NoSuchFile	图片不存在	404
NoSuchStyle	样式不存在	404
InternalError	服务内部错误	500
NotImplemented	方法未实现	501

处理参数限制说明

目前图片处理有如下默认限制：

- 处理源文件的大小限制在20MB。
- 缩略操作：对缩略后的图片的大小有限制，目标缩略图的宽与高的乘积不能超过 $4096 * 4096$ ，而且单边的长度不能超过 $4096 * 4$ 。
- 旋转操作：旋转对图的尺寸有限制，图片的宽或者高不能超过4096。
- 管道目前限制在4个。

2.8 图片样式

所有对图片的变换都会加在URL后面，这样会导致URL变得冗长，不方便管理与阅读。OSS的图片处理服务允许您将常见的操作保存成一个别名，即样式（Style）。一个复杂操作，利用样式功能后，只需一个很短的URL就能实现相同的效果。

一个Bucket下有多个样式，样式的作用范围只在一个Bucket下，目前一个Bucket允许最多有50个样式。

样式访问规则

· URL参数方式访问

```
<文件URL>?x-oss-process=style/<StyleName>
```

示例: `bucket.aliyuncs.com/sample.jpg?x-oss-process=style/stylename`

这种方式是图片处理默认支持的样式访问方式。

· 分隔符方式访问

```
<文件URL><分隔符><StyleName>
```

示例: `bucket.aliyuncs.com/sample.jpg{分隔符}stylename`

在URL带了分隔符, 图片处理服务会把该分隔符后面的内容当成样式的名称。分隔符只有在控制台配置成功后才能生效, 支持-、_、/、!等多种样式分隔符。该方式为图片处理可选的使用方式。

- `StyleName`是样式的名称。
- 创建样式、删除样式和修改样式都必须在控制台操作。
- 当访问的样式在指定Bucket中不存在时, 将返回`NotSuchStyle`错误。



说明:

如果使用IMG域名, 必须开启OSS同步功能才能使用分隔符方式访问。

设置分隔符

1. 在[OSS管理控制台](#)左侧存储空间列表中, 单击您要向其设置分隔符的存储空间名称。
2. 单击图片处理页签, 找到访问设置按钮。



The screenshot shows the OSS Management Console interface for the 'aliyun-best1' storage space. The 'Pictures Processing' tab is selected. Under the 'Pictures Processing' section, there is a 'Default Rule' (域名 /sample.jpg?x-oss-process=style/stylename) and a 'Custom Rule' (域名 /sample.jpg 自定义分隔符 stylename). Below these are buttons for 'Create Style' (新建样式), 'Import Style' (导入样式), 'Export Style' (导出样式), 'Access Settings' (访问设置), 'Delete' (删除), and 'Refresh' (刷新). A note indicates 'Selected: 1 / 1'. Below the buttons, there is a table with two rows: 'rule1' (image/format.jpg/auto-orient,1/quality.q_90) and 'rule2' (image/format.jpg/auto-orient,1/quality.q_100).

3. 单击访问设置, 打开设置窗口。



在设置窗口中您可以进行以下设置：

- 原图保护

开启原图保护后，您需要了解以下信息：

- 图片文件只能通过传入stylename和带签名的方式访问。
- 将禁止直接访问OSS原文档或传入图片参数修改图片样式的访问。
- 您需要选择原图保护后缀，如png、jpg等。

- 自定义分隔符。

设置好您需要的选项之后单击确定完成分隔符的设置。

示例

假如对image-demo这个bucket创建一个样式：

样式名	样式内容
panda_style	image/resize,m_fill,w_300,h_300,limit_0/auto-orient,0/quality,q_90/watermark,image_cGFuZGEucG5n,t_61,g_se,y_10,x_10

- 直接参数访问

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fill,w_300,h_300,limit_0/auto-orient,0/quality,q_90/watermark,image_cGFuZGEucG5n,t_61,g_se,y_10,x_10



- 通过URL参数，使用样式访问

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/panda_style



- 通过样式分隔符，使用样式方式访问

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg@!panda_style



三者可以达到同样的效果。

SDK 示例

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)

2.9 图片原图保护

对于公共读文件，为了在业务中避免被盗图的风险，需要限制向外暴露的图片URL，匿名访问者只能获取到经过缩略处理或者添加水印后的图片。这一需求可以通过开启原图保护的方式实现。

规则说明

开启原图保护以后，无法通过下面两种方式访问：

- OSS地址直接访问：`http://bucket.<endpoint>/object`
- 处理参数请求缩略图：`http://bucket.<endpoint>/object?x-oss-process=image/`
`action,parame_value`

只能通过样式方式访问图片：

- URL参数方式: `http://bucket.<endpoint>/object?x-oss-process=style/<StyleName>`
- 分隔符方式: `http://bucket.<endpoint>/object<分隔符><StyleName>`



说明:

- 上述规则仅针对于匿名访问公共读文件有效。配置原图保护后，可以通过签名的访问获取到图片原图。
- 原图保护主要针对于图片文件，必须配置被保护图片的后缀。例如配置限制了`.jpg`后缀文件，仍然可以直接访问`.png`后缀的原图。

您可以在控制台对应Bucket中图片处理模块中配置访问规则。

配置访问规则

1. 在[OSS管理控制台](#)左侧存储空间列表中，单击您要向其中设置原图保护的存储空间名称。
2. 单击图片处理页签，找到访问设置按钮。如下图所示：

图片处理域名规则 ②

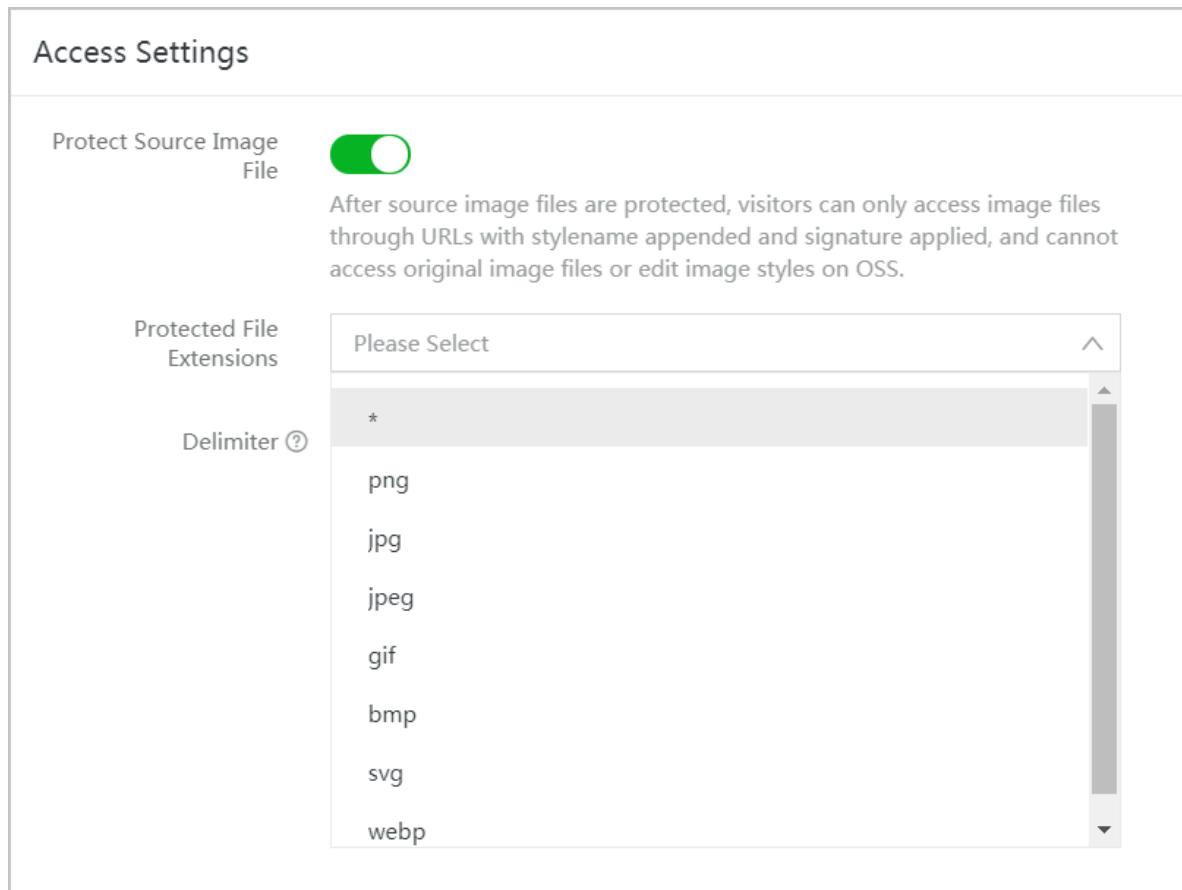
默认规则 域名/sample.jpg?x-oss-process=style/stylename 自定义规则 域名/sample.jpg自定义分隔符stylename

新建样式 导入样式 导出样式 **访问设置** 刷新

② 图片处理域名规则 ② 通过 SDK 实现图片处理

操作
规则名称 代码
test114 image/auto-orient,1/interlace,1/quality,q_90

3. 单击访问设置，打开设置窗口，如下图所示：



在设置窗口中您可以进行以下设置：

- **开启原图保护：**开启原图保护后，图片文件只能通过传入`stylename`和带签名的方式访问；将禁止直接访问OSS原文件或传入图片参数修改图片样式的访问。
 - **设置原图保护的后缀。**
 - **自定义分隔符。**
4. 设置好您需要的选项之后单击确定完成图片保护的设置。

2.10 数据处理持久化

我们提供图片转存的数据处理操作，将处理结果作为资源保存到指定存储空间（Bucket）内，并赋以指定Key。保存成功后，下一次可直接通过指定Bucket来访问该资源，以达到提升下载速度的效果。比较适用于超大图的切割或者其他高延时操作。

请求语法

```
POST /ObjectName?x-oss-process HTTP/1.1
Content-Length: ContentLength
Content-Type: ContentType
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

```
x-oss-process=image/resize,w_100|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

使用Post接口来调用图片服务，参数在body中传递，增加saveas参数支持另存为OSS文件。其中x-oss-process后面跟的参数和使用queryString来调用图片处理功能的参数一致。

saveas的参数列表

参数	含义
o	目标Object名称，参数需经过urlsafe的base64编码。
b	目标Bucket名称，参数需经过urlsafe的base64编码。如果不指定目标Bucket，则默认保存到当前Bucket。

细节分析

- saveas操作需要调用方有目标Bucket和Object的写权限，否则返回403。
- saveas参数中的Bucket和Object名称需满足OSS的Bucket和Object名称的规范，否则返回400。
- saveas指定的Bucket需和当前Bucket在同一个region，否则返回400。
- saveas操作只能在Post操作中使用，不能在Get操作中指定，否则返回400。

示例

- 请求示例

```
POST /ObjectName?x-oss-process HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
x-oss-process=image/resize,w_100|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

示例中的参数含义为将目标图片缩放之后保存到名称为test的Bucket，Object名称为test.jpg。

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

有关图片处理持久化SDK的具体实现，请参考以下文档：

- [Java SDK](#)
- [Python SDK](#)
- [Go SDK](#)

2.11 新老版本API和域名使用FAQ

新版API与老版API的方式主要区别：

- 新版API: `http://bucket.<endpoint>/object?x-oss-process=image/action, param_value`

所有的图片处理操作都通过`x-oss-process`进行传递。每个action之间顺序执行，无需通道进行管理。

- 老版本的API: `http://channel.<endpoint>/object@action.format`

操作通过`@`作为分隔符进行处理。

使用OSS域名来使用图片处理有什么优势以及区别？

对比项	采用IMG域名访问	直接使用OSS域名访问
使用方式	存储与处理两套域名系统	上传、管理、处理、分发，一站式处理
是否支持新版API	支持	支持
是否支持老版API	支持	默认不支持
是否支持https	不支持	支持
是否支持vpc网络	不支持	支持
是否支持多域名绑定	不支持	支持
是否支持源站更新自动刷新阿里CDN	不支持	支持



说明：

- 现在OSS域名已经全面支持图片处理服务，不过只能使用新版的API。而原有的IMG域名能够使用新老两个版本的API。
- 如果IMG域名期望能够进行多CDN加速，可以通过在CDN配置回源host的方式直接访问IMG域名，不需要进行域名绑定来完成CDN加速。

控制台这里对于两个API方式以及两种域名访问方式逻辑是怎样的？

之前开启老版图片处理的Bucket：

- 保持与原来基本一致的逻辑，用户看到的是老版IMG的域名，以及之前已经绑定的自定义域名。
- 用户在IMG域名上进行的原图保护等配置，对于OSS域名没有效果。当在跨区域复制中开启同步时，会将原图保护以及样式分隔符同步到OSS域名。
- 当用户关闭当前Bucket的图片处理服务时，会清空样式配置以及域名绑定，并自动跳转到新版的页面。

新创建的Bucket或者之前没有开启IMG服务的Bucket：

- 默认是能够使用图片处理服务的，无需开通，也没有关闭。
- 无需绑定域名，域名绑定操作直接同Bucket本身的域名管理一致。

使用老版API的用户如何切换到OSS域名上呢？

老版本的API暂时没有办法在OSS域名上面使用，如有特殊情况可以工单联系我们；但样式的方式两个域名都能兼容。所以如果您所有的图片都是通过样式方式访问，那么进行切换主要下面两个步骤：

1. 在当前图片服务配置里面，开启配置同步；这样样式分隔符以及原图保护能够同步到OSS域名。
2. 如果使用了自定义域名，将原有的自定义域名CNAME改到OSS域名即可。

样式配置IMG域名与OSS域名是否一致？

所有的样式配置在IMG域名与OSS域名都是共享的。不存在IMG配置的域名在OSS域名上面不能使用的问题。

2.12 老版图片服务手册

2.12.1 介绍

阿里云OSS图片处理服务（Image Service，简称IMG），是阿里云OSS对外提供的海量，安全，低成本高可靠的图片处理服务。用户将原始图片上传保存在OSS上，通过简单的 RESTful 接口，在任何时间、任何地点、任何互联网设备上对图片进行处理。图片处理服务提供图片处理接口，图片上传请使用OSS上传接口。基于IMG，用户可以搭建出跟图片相关的服务。

图片服务基础功能

图片服务提供以下功能：

- 获取图片信息
- 图片格式转换
- 图片缩放、裁剪、旋转

- 图片添加图片、文字、图文混合水印
- 自定义图片处理样式
- 通过管道顺序调用多种图片处理功能

历史版本说明

图片处理目前有两版API接口，本文档介绍的是老版接口的功能使用说明，今后老版接口的功能将不会再更新，新版接口详情请参考[快速使用OSS图片服务](#)。使用兼容详细说明请参考[新老版本API和域名使用FAQ](#)。

2.12.2 基本概念

图片服务基本概念

Object (对象、文件)

在IMG中，用户操作图片的基本数据单元是Object。即OSS对应的Object，单个Object（即每张图片）允许的最大大小是20MB。

Object 命名规范：

- 使用 UTF-8 编码。
- 长度必须在 1-1023 字节之间。
- 不能以“/”或者“\”字符开头。

Channel (频道)

Channel 是IMG上的命名空间，也是计费、权限控制、日志记录等高级功能的管理实体。IMG名称在整个图片处理服务中具有全局唯一性，且不能修改。一个用户最多可创建10个Channel，但每个Channel中存放的object的数量没有限制。目前Channel跟OSS的Bucket相对应，即用户只能创建与自己在OSS上Bucket同名的Channel。

Channel命名规范：

- 只能包括小写字母，数字，短横线(-)。
- 必须以小写字母或者数字开头和结尾。
- 长度必须在 3-63 字节之间。

Style (样式)

图片处理服务提供用户将图片的处理操作和参数保存成一个别名，即样式。一系列操作，利用样式功能后，只需要用一个很短的URL就能实现相同的效果。

- 一个Channel下面有多个样式，目前一个Channel允许最多有50个样式。

- 样式适应于Channel下面的object图像变化操作。假如在A Channel下面有样式，名称为abc，样式内容是100w.jpg（按宽缩略成100，保存成jpg格式）那么A Channel下面所有的object都能使用样式abc，实现缩略成100w.jpg的效果。
- 样式的作用范围只在一个Channel下，即A Channel不能使用B Channel的样式。

Style命名规范：

- 长度为1-63个字符。
- 只能包含数字、大小写字母、下划线(_)、短横线(-)以及小数点(.)。

处理字符串

图片服务定义了处理字符串，包含两个部分转换参数、转换格式：

- 转换参数由一个或多个键值对（以“_”连接）组成，“值”在前“键”在后，“值”为数字类型，“键”为一位字母。
- 转换格式是一种特殊的转换参数，用户指定转换格式，图片服务对原图处理并返回用户期望的图片文件格式。（支持格式是：jpg, jpeg, webp, png, bmp）。

分隔符

图片处理服务使用通过URL来访问处理的图片。所以需要分隔符来区分一些关键字段。不要在使用的图片文件名称中包含图片处理服务设定的分隔符。不然会导致解析出错的问题。

分隔符名称	分隔符	含义
处理分隔符	@	区分Object名称跟处理字符串。
样式分隔符	@!	区分Object跟样式内容，详见 样式访问 。
管道分隔符		区分多种操作，详见 管道 。

数据中心及访问域名

图片服务的数据中心和OSS的数据中心相对应。用户在OSS的某个数据中心上创建一个Bucket，然后选择开通图片服务，那么对应的Channel也属于该数据中心。开通的数据中心及域名见[访问域名](#)。

示例

下面以一个例子来说明上面提到的概念

`http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h.jpg`

这是一个图片访问的URL，其中各参数说明如下：

- `image-demo`: 用户的频道的名字, 即Channel
- `img-cn-hangzhou.aliyuncs.com`: 图片杭州地区访问域名, 即Endpoint
- `example.jpg`: 待处理的图片的原图名字, 即Object
- `@`: 处理分隔符
- `100w_100h.jpg`: 处理字符串
- `100w_100h`: 将原图进行处理的参数, 即转换参数
- `.jpg`: 将原图根据参数处理后的保持的格式, 即转换格式

限制

- 图片处理支持的格式有: jpg、png、bmp、gif、webp、tiff。
- 指定缩略图宽度或者高度时, 在等比缩放的情况下, 都会默认进行单边的缩放。在固定宽高的模式下, 会默认在宽高一样的情况下进行缩放。
- 对缩略后的图片大小有限制, 目标缩略图的宽与高的乘积不能超过4096 * 4096, 且单边的长度不能超过4096 * 4。
- 调用`resize`, 默认不允许放大。如果请求图片比原图大, 那么返回的仍然是原图。如果想得到放大的图片, 需要增加参数调用`limit,0`。
- 管道目前限制在4个。

其他限制可以参考文档其他参数的详细描述。

2.12.3 访问域名

目前, IMG 有以下几个数据中心 (Region) 对公众提供服务, 各区域的访问域名 (Endpoint) 设置如下:

Region 中文名称	Region 英文名称	Endpoint
杭州数据中心	oss-cn-hangzhou	img-cn-hangzhou.aliyuncs.com
青岛数据中心	oss-cn-qingdao	img-cn-qingdao.aliyuncs.com
北京数据中心	oss-cn-beijing	img-cn-beijing.aliyuncs.com
深圳数据中心	oss-cn-shenzhen	img-cn-shenzhen.aliyuncs.com
上海数据中心	oss-cn-shanghai	img-cn-shanghai.aliyuncs.com

Region 中文名称	Region 英文名称	Endpoint
香港数据中心	oss-cn-hongkong	img-cn-hongkong.aliyuncs.com
美国 (加利福尼亚州) 数据中心	oss-us-west-1	img-us-west-1.aliyuncs.com
亚洲 (新加坡) 数据中心	oss-ap-southeast-1	img-ap-southeast-1.aliyuncs.com

上传图片需要的 OSS 访问域名请参见[访问域名和数据中心](#)

2.12.4 图片上传

图片服务处理的图片来自于OSS，所以图片的上传实际是往OSS上同名Bucket上传的。所有的上传请参考 OSS 开发人员指南中的[简单上传](#)。

假如用户需要使用杭州的图片服务，域名为。

上传前提条件：

- 同区域OSS存储空间(Bucket)。例如叫oss-sample，杭州的OSS访问域名为。
- 通过控制台或者SDK上传图片。例如上传logo.png到oss-sample。

控制台上传图片

- 控制台：[上传文件](#)

SDK上传图片

- SDK：Java SDK-[Object](#)中PutObject

注意事项

- 必须是同区域的OSS和IMG
- 必须是同名的Bucket和Channel
- 必须使用OSS的域名上传

2.12.5 错误响应

当您访问图片处理服务出现错误时，图片处理服务会将相应的错误码和错误信息返回给您，以帮助您定位与处理问题。

图片处理服务错误的响应格式

错误响应的消息体例子：

```
<Error>
  <Code>BadRequest</Code>
  <Message>Input is not base64 decoding.</Message>
  <RequestId>52B155D2D8BD99A15D0005FF</RequestId>
  <HostId>userdomain</HostId>
</Error>
```

错误包含以下元素：

- **Code**: 图片处理服务返回给用户的错误码。
- **Message**: 图片处理服务给出的详细错误信息。
- **RequestId**: 用以标识错误请求的唯一UUID。在无法解决问题时候，可以使用此错误ID发送给图片处理服务的工程师去定位错误的原因。
- **HostId**: 用来标识访问的图片处理服务集群。

图片处理服务的错误码

错误码	描述	HTTP 状态码
TooManyPipe	管道数目超过限制	400
InvalidArgument	参数错误	400
BadRequest	错误请求	400
MissingArgument	缺少参数	400
ImageTooLarge	图片大小超过限制	400
WatermarkError	水印错误	400
AccessDenied	拒绝访问	403
SignatureDoesNotMatch	签名不匹配	403
NoSuchFile	图片不存在	404
NoSuchStyle	样式不存在	404
NoSuchChannel	频道不存在	404
InternalError	服务内部错误	500
NotImplemented	方法未实现	501

处理参数限制说明

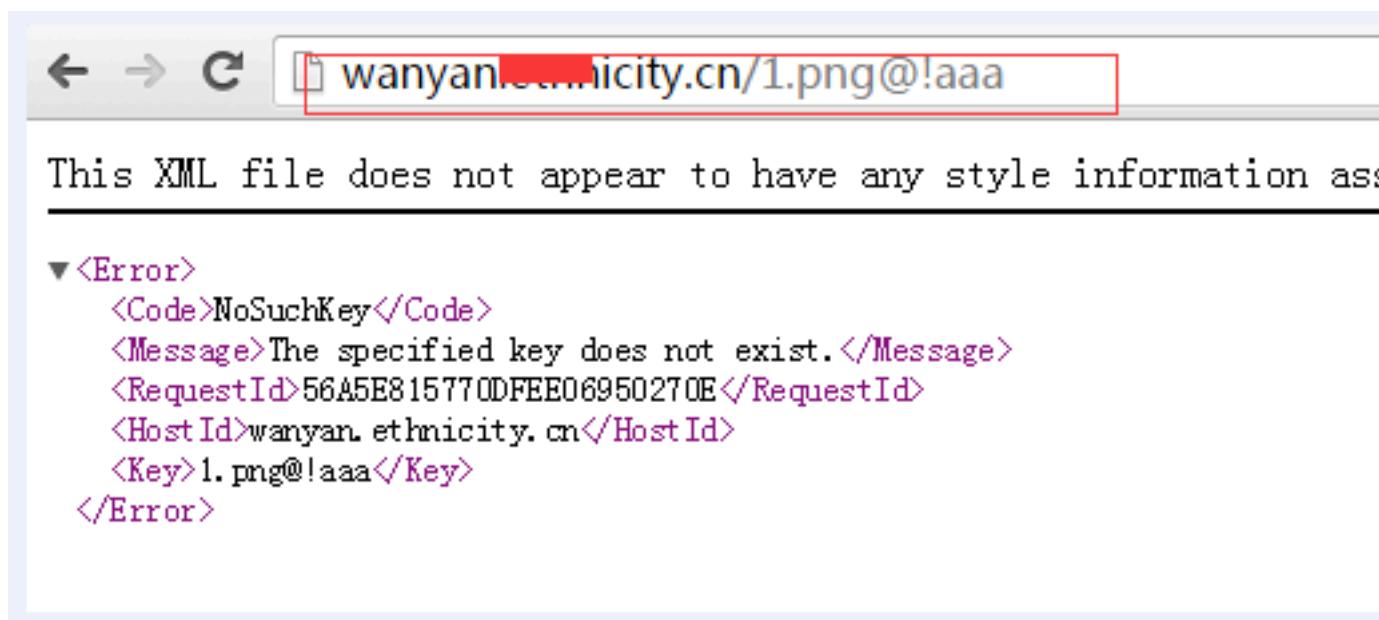
目前图片处理有如下默认限制：

- 处理源文件的大小限制在20MB。

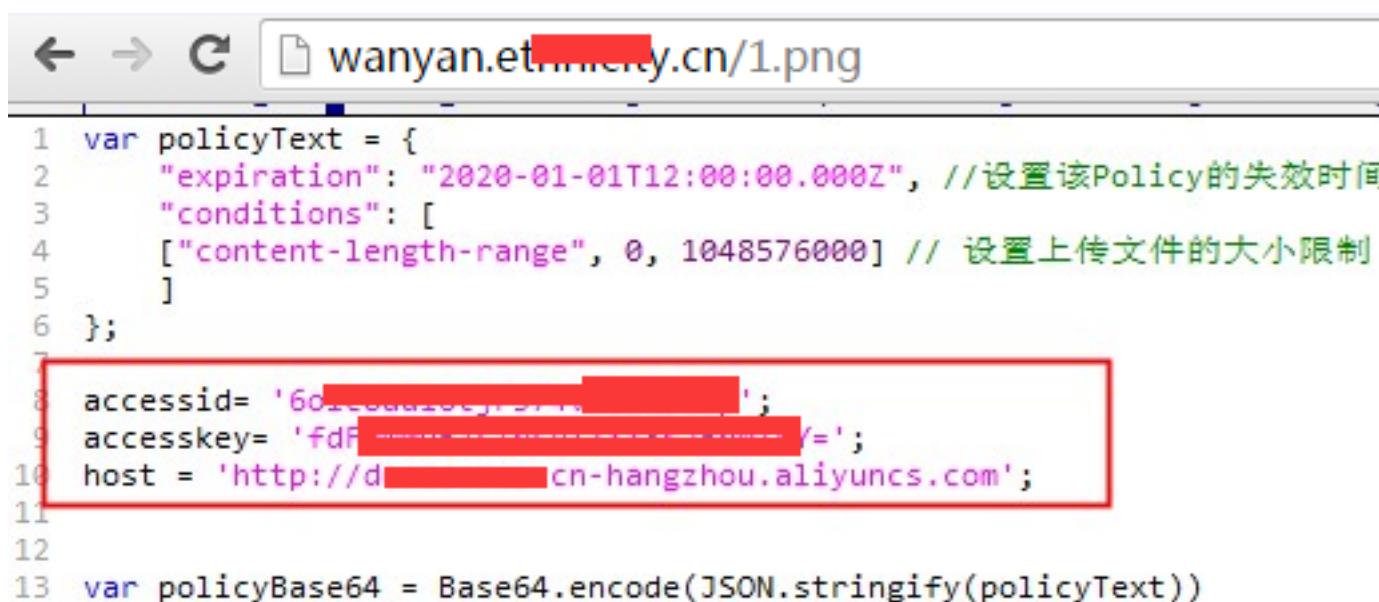
- 缩略操作：对缩略后的图片的大小有限制，目标缩略图的宽与高的乘积不能超过4096 * 4096，而且单边的长度不能超过4096 * 4。
- 旋转操作：旋转对图的尺寸有限制，图片的宽或者高不能超过4096。
- 管道目前限制在4个。

2.12.6 图片处理报错<Code>NoSuchKey</Code>

oss的图片处理服务，增加一个oss图片处理style，增加成功，在浏览器中输入bucket中某object的地址，能直接显示，但是加上stype (@aaa) 后，提示NoSuchKey。

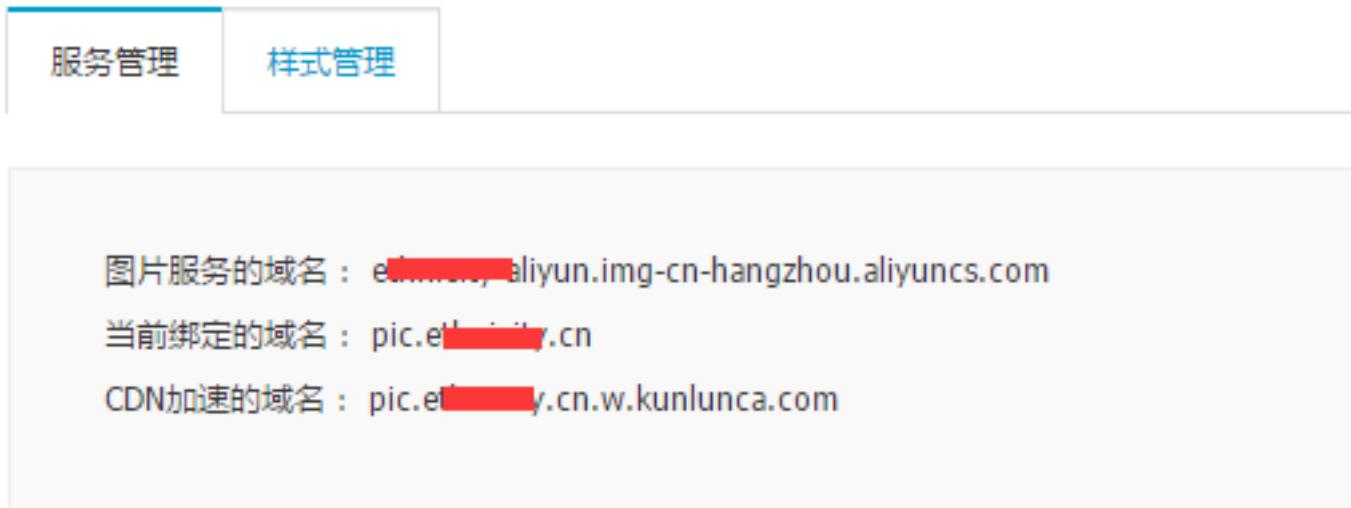


再次看下图片确实是存在的。



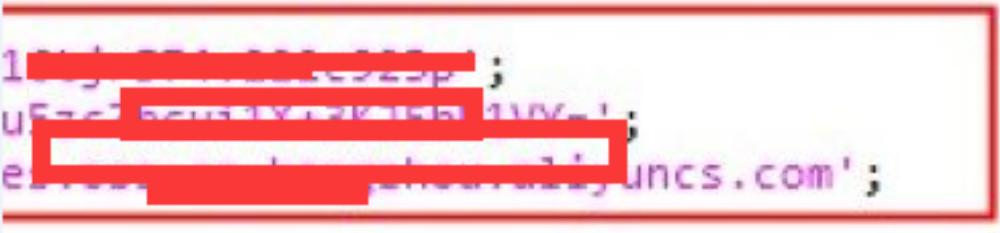
文件存在，样式也设置好了，图中为wanyan.ethnicity.cn绑定的域名。该域名是oss的buket属性的域名，而不是oss图片处理的域名。oss图片处理的域名和buket的域名是有区别的（不能相同），这里需要将源站填写为图片服务的域名，如下在图片处理服务的控制台可以看到这个域名：

图片处理



再次访问图片正常。

The screenshot shows a browser window with the URL `pic.ethnicity.cn/1.png@!aaa`. The page content is a JSON policy document:

```
"2020-01-01T12:00:00.000Z", // 设置该Policy的过期时间
[
  "Content-Type": "image/png",
  "Content-Length": 1048576000 // 设置上传文件的大小
]
1048576000-223px;
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.143 Safari/537.36;
Expires: 2020-01-01T12:00:00.000Z;

```

Base64.encode(JSON.stringify(policyText))

域名解析的验证方式如下（OSS的图片处理的域名是需要带img标签的，比如xxxxx-aliyun.img-cn-hangzhou.aliyuncs.com）：

```
[root@localhost ~]# nslookup pic.ethnicity.cn
Server:      10.65.0.1
Address:     10.65.0.1#53

Non-authoritative answer:
pic.ethnicity.cn      canonical name = ethnicity-aliyun.img-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
Name:  ethnicity-aliyun.img-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
Address: 112.124.219.97

[root@localhost ~]# nslookup wanyan.ethnicity.cn
Server:      10.65.0.1
Address:     10.65.0.1#53

Non-authoritative answer:
wanyan.ethnicity.cn      canonical name = ethnicity-aliyun.oss-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
Name:  ethnicity-aliyun.oss-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
Address: 112.124.219.93
Name:  ethnicity-aliyun.oss-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
Address: 112.124.219.82
```

如果问题还未能解决, 请联系[售后技术支持](#)。

3 视频截帧

图片处理除了能够处理已经存在的图片内容，还能够截取出视频中的指定位置产生图片，完成视频截帧。

参数

操作分类: `video`

操作名称: `snapshot`

参数	描述	取值范围
<code>t</code>	截图时间	单位ms, [0,视频时长]
<code>w</code>	截图宽度, 如果指定为0则自动计算	像素值: [0,视频宽度]
<code>h</code>	截图高度, 如果指定为0则自动计算, 如果w和h都为0则输出为原视频宽高	像素值: [0,视频高度]
<code>m</code>	截图模式, 不指定则为默认模式, 根据时间精确截图, 如果指定为fast则截取该时间点之前最近的一个关键帧	枚举值: <code>fast</code>
<code>f</code>	输出图片格式	枚举值: <code>jpg</code> 、 <code>png</code>

示例

- 找到视频7s处的内容，输出为jpg。

http://a-image-demo.oss-cn-qingdao.aliyuncs.com/demo.mp4?x-oss-process=video/snapshot,t_7000,f_jpg,w_800,h_600,m_fast



- 找到视频50s处的内容，输出为jpg，精确为对应的时间。

http://a-image-demo.oss-cn-qingdao.aliyuncs.com/demo.mp4?x-oss-process=video/snapshot,t_50000,f_jpg,w_800,h_600



4 智能媒体管理 (IMM)

4.1 快速开始

本文介绍智能媒体管理 (IMM) 的开通、OSS 绑定 IMM、以及如何在 OSS 中快速使用 IMM。

阿里云 OSS 与智能媒体管理 (IMM) 深度结合，支持文档预览、文档格式转换、人脸识别、图片分析、二维码识别等丰富的数据分析处理操作。

步骤一：开通 IMM

如果还没有开通 IMM，请先开通产品服务。

开通步骤如下：

1. 登录 [IMM 管理控制台](#)。
2. 在弹出的开通该服务提示对话框单击确认。
3. 在云产品开通页，勾选同意协议后，单击立即开通。
4. 单击新建项目，在弹出的授权提示对话框单击激活授权角色。

由于产品处理的数据来源为阿里云 OSS，需要激活授权角色 `AliyunIMMDefaultRole`，以授权 IMM 访问 OSS 的相关权限。

5. 单击同意授权。

步骤二：OSS 绑定 IMM

开通 IMM 后，执行以下步骤将 OSS 绑定 IMM：

1. 登录 [OSS 管理控制台](#)。
2. 选择任意一个 Bucket，单击智能媒体。
3. 单击批量创建。

批量创建将默认创建对应区域的 IMM Project，然后与该 Bucket 做绑定。

4. 在功能设置页面，勾选您要绑定的功能名称。

功能配置

功能名称	绑定项目	实例类型	付费方式	授权
文档功能	ossdocdefault	文档标准型	0.08/次	AliyunIMMDefaultRole
人脸识别	ossfacedefault	图片标准型	1QPS免费体验	AliyunIMMDefaultRole
图片识别	ossimagedefault	图片标准型	1QPS免费体验	AliyunIMMDefaultRole

确定 取消

5. 单击确定。



:

OSS 与 IMM 的绑定本质上是每个分析功能与 IMM Project 的关联调用。

步骤三：使用 IMM

完成上述功能绑定后，您可以通过 OSS 控制台或者 SDK 的方式使用 IMM。

· 在 OSS 管理控制台使用 IMM

在 OSS 管理控制台使用 IMM 的操作步骤如下：

1. 登录[OSS 管理控制台](#)。
2. 选择任意一个已绑定 IMM 的 Bucket，单击文件管理。
3. 单击该 Bucket 下的图片或者文档，在预览页面直接使用智能媒体相关服务。

· 通过 SDK 方式使用 IMM

您可以通过 OSS SDK 分析处理生产中的数据，示例如下：

```
public class OssProcessPreview{
    public static void main(String[] args) {
        String ak = "";
        String sk = "";
        String bucketName = "imm-user-zzh";
        String objectKey = "test.jpg";
        URL url = getUrl("imm/detectface", ak, sk, bucketName,
        objectKey);
        System.out.println(url.toString());
        bucketName = "imm-user-zzh";
        objectKey = "a.xlsx";
```

```
        url = getUrl("imm/previewdoc", ak, sk, bucketName,
objectKey);
        System.out.println(url.toString());
    }
    private static URL getUrl(String process, String ak, String sk,
String bucketName, String objectKey) {
        OSSClient client = new OSSClient(ak, sk);
        client.setEndpoint("oss-cn-shanghai.aliyuncs.com");
        GetObjectRequest getObjectRequest = new GetObjectRequest(
bucketName, objectKey);
        getObjectRequest.setProcess(process);
        GeneratePresignedUrlRequest request = new GeneratePr
esignedUrlRequest(bucketName, objectKey);
        request.setProcess(process);
        request.setExpiration(new Date(new Date().getTime() + 3600
* 1000));
        return client.generatePresignedUrl(request);
    }
}
```

4.2 文档预览

文档预览功能支持 ppt、xls、doc、pdf 等格式文档进行在线预览，便于您进行文档内容管理与访问。



说明：

要使用文档预览功能，您需要先开通智能媒体管理 IMM，并在 OSS 中绑定 IMM，详情请参阅[智能媒体管理快速开始](#)。

参数

操作名称：imm/previewdoc

参数说明如下：

名称	取值	描述
copy	0/1	指定预览文件是否能够复制文档内容。 值为 1，表示支持复制。 值为 0，表示不支持复制。

流程介绍



说明：

无论源文件权限是否为公共读，出于安全性考虑此处建议使用 URL 签名方式。

文档预览流程如下：

1. 客户端向服务端发起预览请求，并提供要预览的文件。
2. 服务端根据请求文件进行 URL 签名，将签名完成的 URL 提供给客户端。
3. 客户端拿到签名后直接访问 OSS 查看文件。

示例

假如请求的 Bucket 为 doc-demo，该 Bucket 所在区域为华东 1，对应的域名为 oss-cn-hangzhou.aliyuncs.com，请求预览的文件为 example.docx。通过配置参数 copy_1 可以复制预览的文档内容。

```
http://doc-demo.oss-cn-hangzhou.aliyuncs.com/example.docx?x-oss-process=imm/previewdoc, copy_1
```

对于上面的 URL 采用 AK、SK 签名后得到 SignURL 进行访问。

通过 SDK 请求示例：

```
# 创建存储空间实例，所有文件相关的方法都需要通过存储空间实例来调用。
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret),
endpoint, bucket_name)
# 文档预览，获取signURL。
process = 'imm/previewdoc,copy_1'
params = {}
params.update({bucket.PROCESS: process})
url = bucket.sign_url("GET", objectKey, 3600, params=params)
#打印结果
print url
```

4.3 人脸识别

人脸识别功能基于图片 AI 技术，能够检测图片中的人脸矩形框和属性。如果图片有多张人脸，则会把多张人脸的矩形框和属性都检测出来。基于这些元数据，可以应用于年龄、性别的统计。



说明：

要使用人脸识别功能，您需要先开通智能媒体管理 IMM，并在 OSS 中绑定 IMM，详情请参阅[智能媒体管理快速开始](#)。

人脸矩形框包含 4 个值，分别为左上角纵坐标、左上角横坐标、宽度、高度。

人脸属性包含 6 个值，分别为性别、年龄、人脸头部姿势、眼睛状态、人脸模糊度、人脸质量。

参数

操作名称：imm/detectface

返回内容示例：

```
{  
  "Faces": [  
    {  
      "Age": 29,  
      "Attractive": 0.95,  
      "Emotion": "HAPPY",  
      "EmotionConfidence": 0.9875330924987793,  
      "EmotionDetails": {  
        "ANGRY": 0.000016857109585544094,  
        "CALM": 0.012278525158762932,  
        "DISGUSTED": 0.000012325451280048583,  
        "HAPPY": 0.9875330924987793,  
        "SAD": 0.0000388074986403808,  
        "SCARED": 0.000006888585176056949,  
        "SURPRISED": 0.000054363932576961815  
      },  
      "FaceAttributes": {  
        "Beard": "NONE",  
        "BeardConfidence": 1,  
        "FaceBoundary": {  
          "Height": 928,  
          "Left": 607,  
          "Top": 628,  
          "Width": 894  
        },  
        "Glasses": "NONE",  
        "GlassesConfidence": 1,  
        "Mask": "NONE",  
        "MaskConfidence": 0.9999999403953552,  
        "Race": "YELLOW",  
        "RaceConfidence": 0.598323404788971  
      },  
      "FaceConfidence": 0.9704222083091736,  
      "FaceId": "4199e1985b6d3bb075f0994c82e6d2fd82a274c1  
1ce183e1fdb222dd3aa8c7ce",  
      "Gender": "MALE",  
      "GenderConfidence": 1,  
    },  
    {  
      "ImageUri": "oss://image-demo/person.jpg",  
      "RequestId": "5C3D854A3243A93A275E9C99",  
      "httpStatusCode": 200,  
      "success": true  
    }  
  ]  
}
```

示例

假设请求 Bucket 是 imm-demo，该 Bucket 所在区域为华东 1，对应的域名为 oss-cn-hangzhou.aliyuncs.com，请求预览照片为 person.jpg，未签名的请求结构如下：

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/person.jpg?x-oss-  
process=imm/detecface
```

通过 Python SDK 实现接口调用如下：

```
# 创建存储空间实例，所有文件相关的方法都需要通过存储空间实例来调用。  
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret),  
endpoint, bucket_name)
```

```
# 人脸识别
style = 'imm/detectface'
result = bucket.get_object(objectKey, process=style)
#解析结果
buf = result.read(result.content_length)
print json.dumps(json.loads(buf), indent=4, sort_keys=True)
```

4.4 图片识别

图片识别功能基于图片 AI 技术，能够检测图片标签和置信度。



说明:

要使用图片识别功能，您需要先开通智能媒体管理 IMM，并在 OSS 中绑定 IMM，详情请参阅[智能媒体管理快速开始](#)。

标签采用分层体系结构，每个标签通常包含父标签（主标签），比如标签男人的父标签为人物，目前总共支持 25 个主标签，2131 个标签。

参数

操作名称: imm/tagimage

返回结果参数说明:

名称	类型	描述
TagId	String	标签 ID
TagLevel	String	标签级别，从 1 开始整数编码，1 为顶级，2 为次级，以此类推
TagName	String	标签名称
ParentTagId	String	上一级的 TagId，如果为顶级则 ParentTagId 为 0
ParentTagName	String	上一级的标签名称，如果为顶级则 ParentTagName 为空
TagScore	String	标签置信度得分，小于等于 1 的浮点数

返回结果示例

```
{
  "ImageUri": "oss://image-demo/example.jpg",
  "RequestId": "5C3D858E530E23D52CA0ED09",
  "Tags": [
    {
      "TagConfidence": 0.2999534606933594,
```

```
        "TagLevel":1,
        "TagName":"自然景观"
    },
    {
        "ParentTagName":"自然景观",
        "TagConfidence":0.2999534606933594,
        "TagLevel":2,
        "TagName":"夜晚"
    },
    {
        "TagConfidence":0.2677214741706848,
        "TagLevel":1,
        "TagName":"外部场景"
    },
    {
        "ParentTagName":"外部场景",
        "TagConfidence":0.2677214741706848,
        "TagLevel":2,
        "TagName":"城市全景"
    }
],
"statusCode":200,
"success":true
}
```

对于图片标签的格式解析, 请参见[内容识别](#)。

示例

假如请求 Bucket 是 imm-demo, 该 Bucket 所在区域为华东1, 对应的域名为 oss-cn-hangzhou.aliyuncs.com, 请求预览照片为 image.jpg, 未签名的请求结构如下:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/image.jpg?x-oss-process=imm/tagimage
```

通过 Python SDK 实现接口调用如下:

```
# 创建存储空间实例, 所有文件相关的方法都需要通过存储空间实例来调用。
bucket = oss2.Bucket(oss2.Auth(access_key_id, access_key_secret),
endpoint, bucket_name)

# 图像识别
style = 'imm/tagimage'
resp = bucket.get_object(objectKey, process=style)
#解析结果
data = resp.read(resp.content_length)
result = json.loads(data)
print "requestId: " + json.dumps(result["RequestId"], indent=4,
sort_keys=True)
print "SuccessDetails: " + json.dumps(result["SuccessDetails"], indent=4,
sort_keys=True)
print "FailDetails: " + json.dumps(result["FailDetails"], indent=4,
sort_keys=True)
```