# 对象存储 OSS

## 图片处理指南

为了无法计算的价值 | 阿里云

# 图片处理指南

# 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：**当您开通OSS服务后，图片服务也随之开通，无需您另外单独开通图片服务。

## 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

## 历史版本说明

图片处理目前有两版API接口，本文档介绍的是新版接口的功能使用说明，老版接口的功能今后不会再更新。使用兼容详细说明参考这里。

## 快速使用图片服务

### 新建图片样式

1. 进入 OSS 管理控制台 。

单击您的Bucket名字，进入Bucket**概览**页。

在**概览**页中单击**图片处理**，然后单击**新建样式**。

在**图片样式**窗口中，新建您需要的图片样式，如下图所示：

图片样式窗口说明：

规则名：新建图片样式的名称，建议取一个有意义的名称，方便您记住，比如XX水印图旋转。

编辑类型：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。

缩略方式：设置图片的缩略方式。

> **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。

缩略尺寸：设置图片的缩略尺寸。

缩略限制：设置是否限制图片放大。

适应方向：设置图片的适应方向。

图片锐化：设置图片是否需要锐化。

图片质量：设置图片质量。

水印：设置图片水印方式。

编辑好图片样式后单击**确定**保存样式。

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

进入 OSS 管理控制台 。

单击您的Bucket名字，进入Bucket**概览**页。

单击**文件管理**，选择一张已经存在的图片或者上传新图片，单击图片名称打开图片**预览**页。

从**图片样式**下拉列表中选择一个图片样式。
您即刻在文件预览窗口中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，您只需单击**复制文件URL**就可获取到文件访问地址。

## 相关文档

- 开通OSS服务
- 获取图片信息
- 图片格式转换
- 图片缩放
- 图片水印
- 图片样式

# 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。

**parame**：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，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区域在华东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,parame_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）

开通的数据中心及域名见**访问域名**。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区域在华东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 |
| iOS SDK | 图片处理 | AliyunOSSiOSTests.m testGetImage |
| Android SDK | 图片处理 | OSSGetObjectTest.testAsyncGetImageWithXOssProcess |

# 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4096 x 4096， 且单边的长度不能超过4096 x 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。

gif，webp动图由于资源消耗原因目前只支持一级处理。如不能够“缩略操作”后面直接再进行“图片裁剪”。

# 图片缩放

将图片按照要求生成缩略图，或者进行特定的缩放。

说明：图片处理支持的格式：jpg、png、bmp、gif、webp、tiff。

## 参数

操作名称：resize

- 指定宽高缩放

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| m | 指定缩略的模式：<br>- lfit：等比缩放，限制在设定在指定w与h的矩形内的最大图片。<br>- mfit：等比缩放，延伸出指定w与h的矩形框外的最小图片。<br>- fill：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。<br>- pad：固定宽高，缩略填充。 | [lfit,mfit,fill,pad,fixed]，默认为lfit。 |

| | | |
|---|---|---|
| | - fixed：固定宽高，强制缩略 | |
| w | 指定目标缩略图的宽度。 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |
| limit | 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 |
| color | 当缩放模式选择为pad（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00FF00（绿色）。 | [000000-FFFFFF] |

- 按比例缩放

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

# 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。
- 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4096 x 4096，且单边长度不能超过4096 x 4。
- 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。即如果请求的图片对原图大，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）

# 示例

## 单边缩略

- 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

## 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, 剩余的部分以单色填充。

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

## 相关文档

- 质量变换
- 格式转换
- 图片裁剪
- 图片旋转
- 图片水印
- 图片原图保护

# 图片裁剪

# 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 |
| --- | --- | --- |
| r | 从图片取出的圆形区域的半径 | 半径 r 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jpeg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

# 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,ne] |

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

# 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 |
| --- | --- | --- |
| x | 进行水平切割，每块图片的长度。x 参数与 y 参数只能任选其一。 | [1,图片宽度] |
| 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。 |

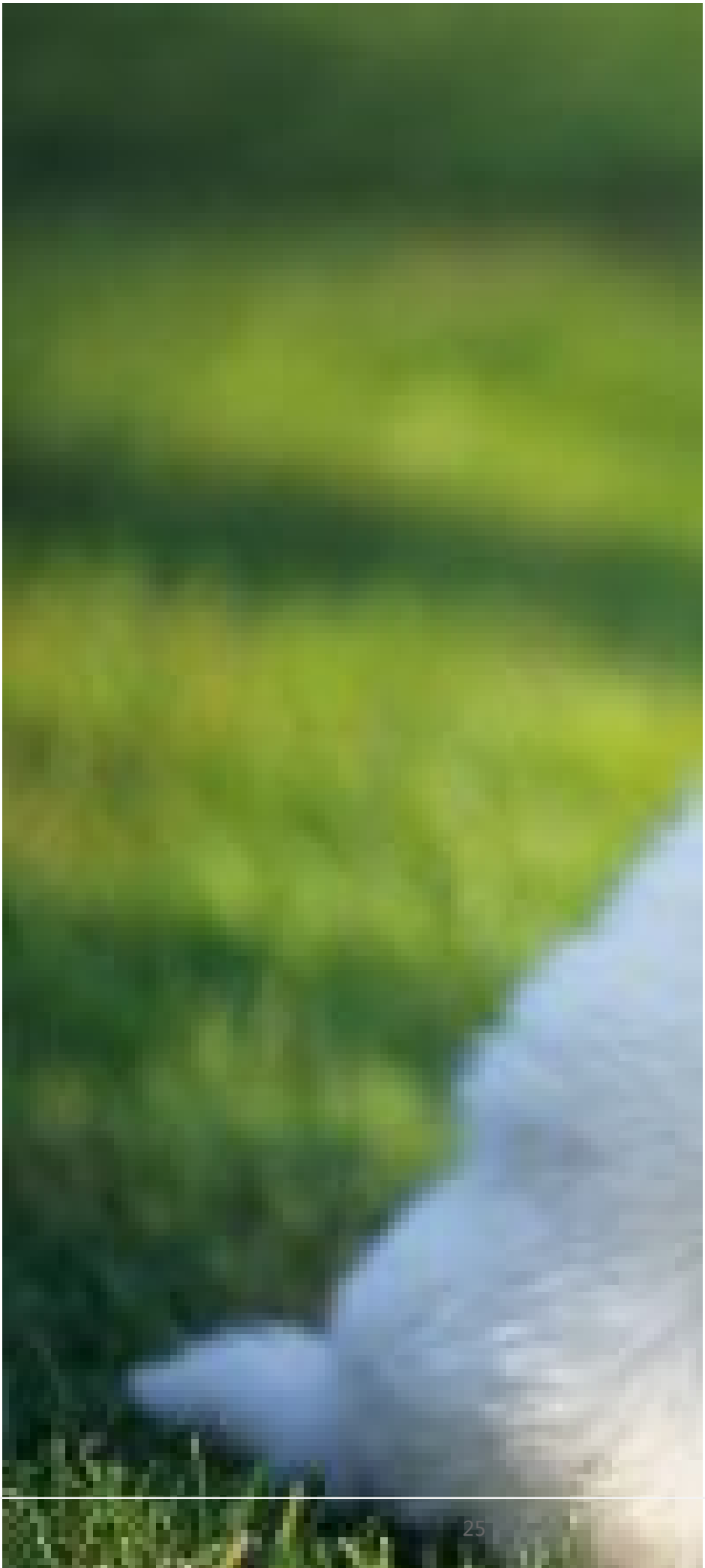
# 注意事项

- 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。

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

# 圆角矩形

可以把图片保存成圆角矩形，并可以指定圆角的大小 。

## 参数

操作名称：rounded-corners

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| r | 将图片切出圆角，指定圆角的半径。 | [1, 4096]<br>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

# 图片旋转

## 自适应方向

某些手机拍摄出来的照片可能带有旋转参数（存放在照片exif信息里面）。可以设置是否对这些图片进行旋转。默认是设置自适应方向。

### 参数

操作名称：auto-orient

| 参数 | 描述 | 取值范围 |
| --- | --- | --- |
| value | 进行自动旋转 | [0, 1] |

| | | |
|---|---|---|
| | 0：表示按原图默认方向，不进行自动旋转。<br>1：先进行图片进行旋转，然后再进行缩略 | |

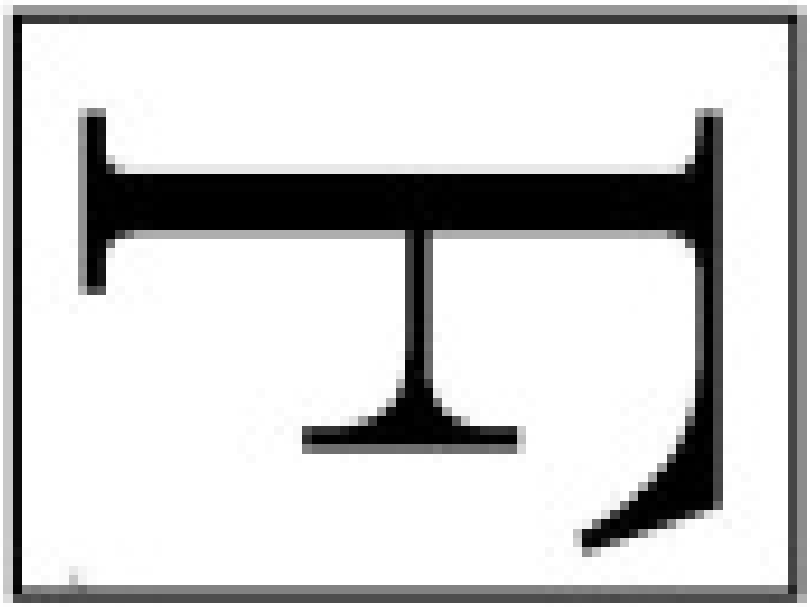
# 注意事项

- 进行自适应方向旋转，要求原图的宽度和高度必须小于 4096。
- 如果原图没有旋转参数，加上auto-orient参数不会对图有影响。

# 示例

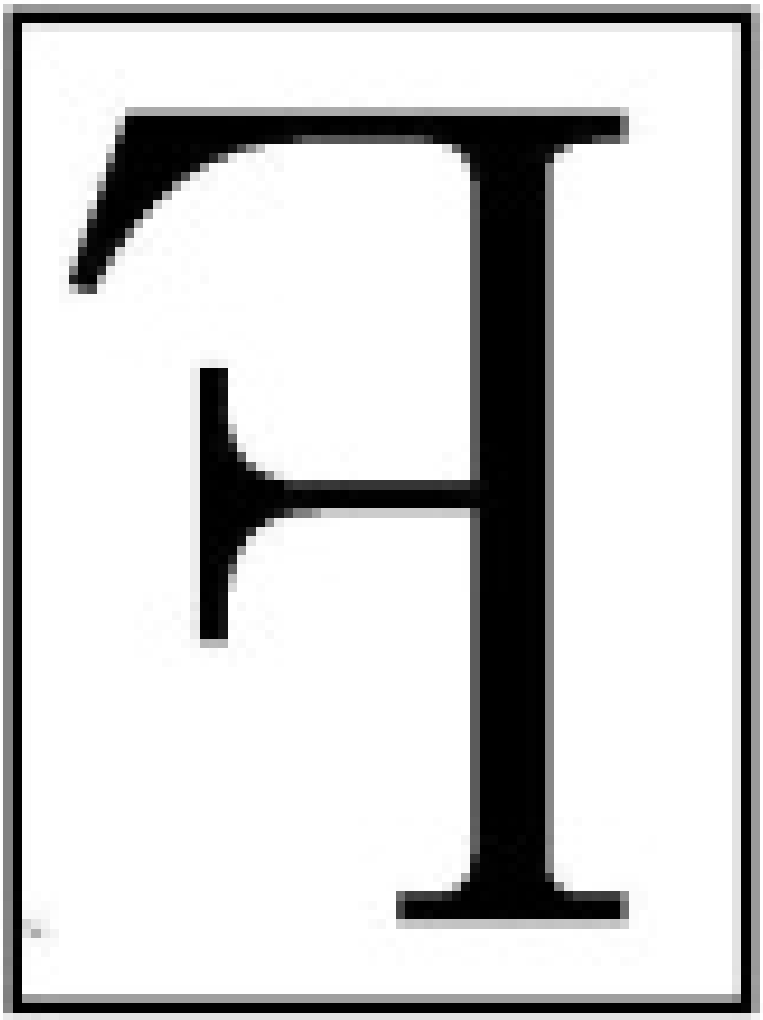
- 将图缩略成宽度为 100，对图片不做自动旋转处理。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,0



- 将图缩略成宽度为 100，对图片进行自动旋转 1。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,1



得到的目标效果图宽度是 100，高度是 127。

# 旋转

可以将图片按顺时针旋转。

## 参数

操作名称：rotate

| 参数 | 描述 | 取值范围 |
| --- | --- | --- |

| | | |
|---|---|---|
| value | 图片按顺时针旋转的角度 | [0, 360]<br>默认值为 0，表示不旋转。 |

# 注意事项

- 旋转图片可能会导致图片的尺寸变大。
- 旋转对图片的尺寸有限制，图片的宽或者高不能超过 4096。

# 示例

- 将原图按顺时针旋转 90 度。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rotate,90

- 将原图缩略成宽度为 200，高度为 200，并按顺时针旋转 90 度。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200,h_200/rotate,90

## 图片效果

## 模糊效果

可以对图片进行模糊操作。

# 参数

操作名称：blur

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| r | 模糊半径 | [1,50]<br>r 越大图片越模糊。 |
| s | 正态分布的标准差 | [1,50]<br>s 越大图片越模糊。 |

# 示例

- 对图片进行模糊半径是 3， 标准差是 2 的处理。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/blur,r_3,s_2

- 对图片进行缩略成宽度是200, 并且进行模糊半径是 3， 标准差是 2 的处理。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/blur,r_3,s_2

# 亮度

可以对处理后的图片进行亮度调节。

## 参数

操作名称：bright

| 参数 | 描述 | 取值范围 |
| --- | --- | --- |
| value | 亮度调整。0 表示原图亮度，小于 0 表示低于原图亮度，大于 0 表示高于原图亮度。 | [-100, 100] |

## 示例

- 将原图只进行亮度调整。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/bright,50

- 对图片进行缩略成宽度是200, 并且进行亮度调整

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/bright,50

# 对比度

可以对处理后的图片进行对比度调节。

# 参数

操作名称：contrast

| 参数 | 描述 | 取值范围 |
| --- | --- | --- |
| value | 对比度调整。0 表示原图对比度，小于 0 表示低于原图对比度，大于 0 表示高于原图对比度。 | [-100, 100] |

# 示例

- 将原图只进行对比度调整。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/contrast,-50

- 对图片进行缩略成宽度是200, 并且进行对比度调整。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/contrast,-50

# 锐化

可以对处理后的图片进行锐化，使图片变得清晰。

## 参数

操作名称：sharpen

| 参数 | 描述 | 取值范围 |
| --- | --- | --- |
| value | 表示进行锐化处理。取值为锐化参数，参数越大，越清晰。 | [50, 399]<br>为达到较优效果，推荐取值为100。 |

## 示例

- 对原图进行锐化处理，锐化参数为 100。

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/sharpen,100

- 对图片进行缩略成宽度是200, 并且进行锐化参数是100

http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/sharpen,100

# 格式转换

# 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

操作名称：format

| 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

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

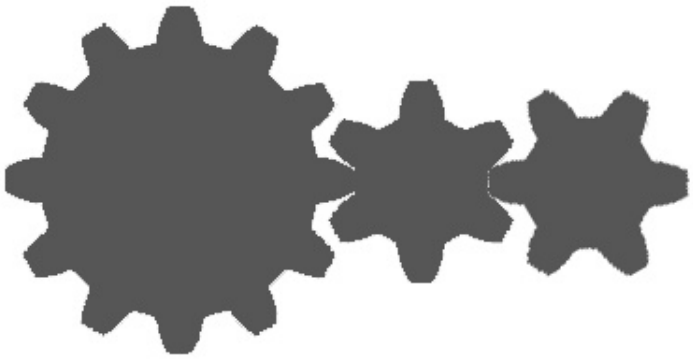
将png保存成jpg格式。

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jpeg格式是支持渐变显示

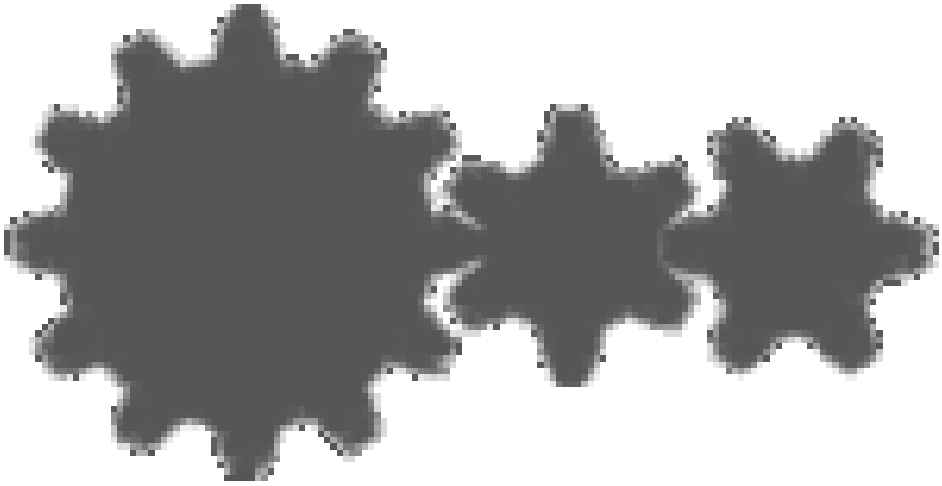
请求URL：http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/interlace,1/format,jpg

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

# 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 |
| --- | --- | --- |
| [value] | 1 表示保存成渐进显示的 jpg 格式<br>0 表示保存成普通的 jpg 格式 | [0, 1] |

注意：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

# 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 |
| --- | --- | --- |
| q | 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。<br>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 |
| Q | 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。 | 1-100 |

| | | |
|---|---|---|
| | 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q和Q，按Q来处理。 | |

## 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

# 图片水印

水印操作可以在图片上设置另外一张图片或者文字做为水印。

## 参数

操作名称：watermark

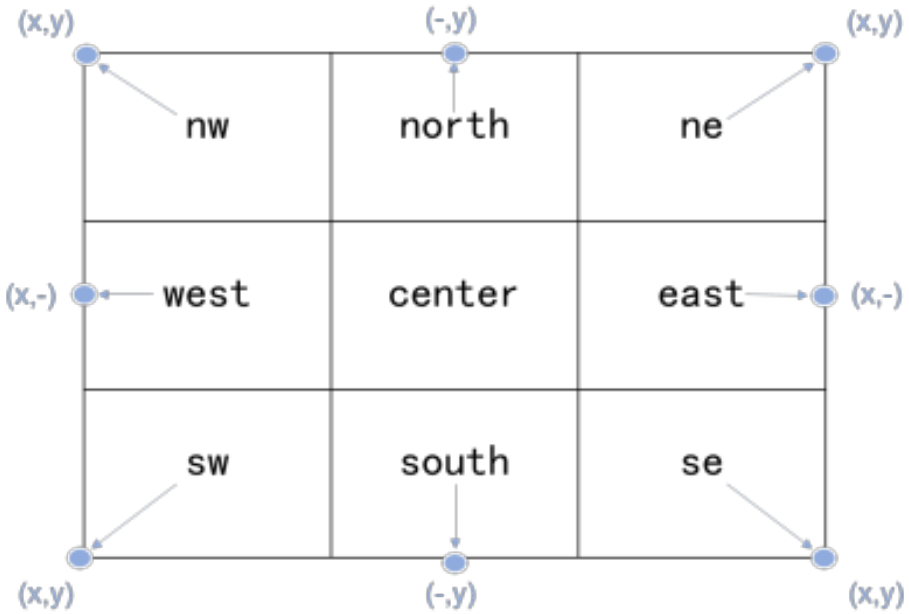
## 基础参数

| 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。<br>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。<br>取值范围：[nw,north,ne,west,center,east,sw,south,se] | 可选参数 |
| 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 | 可选参数 |

| | | |
|---|---|---|
| | 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<br>默认值：10<br>取值范围：[0 – 4096]<br>单位：像素（px） | |
| 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<br>默认值：10<br>取值范围：[0 – 4096]<br>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。<br>默认值：0<br>取值范围：[-1000, 1000]<br>单位：像素(px) | 可选参数 |

## 注意事项

- 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 | 参数意义： 水印图片的object名字(必须编码)<br>注意：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” | 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。

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=cGFuZGEucG5nQDMwUA&t=90&p=9&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 | 参数意义：表示文字水印的文字内容(必须编码)<br>注意：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) | 必选参数 |
| type | 参数意义：表示文字水印的文字类型(必须编码)<br>注意：必须是URL安全base64编码 encodeText = url_safe_base64_encode(fontType)<br>取值范围：见下表（文字类型编码对应表）<br>默认值：wqy-zenhei（编码后的值：d3F5LXplbmhlaQ) | 可选参数 |
| color | 参数意义：文字水印文字的颜色参数的构成必须是：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<br>默认值：000000黑色 | 可选参数 |
| size | 参数意义：文字水印文字大小 | 可选参数 |

| | | |
|---|---|---|
| | (px)<br>取值范围：(0，1000]<br>默认值：40 | |
| shadow | 参数意义：文字水印的阴影透明度<br>取值范围：(0,100] | 可选参数 |
| rotate | 参数意义：文字顺时针旋转角度<br>取值范围：[0,360] | 可选参数 |
| fill | 参数意义：进行水印铺满的效果；<br>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 | 可选参数 |

| | | |
|---|---|---|
| | 取值范围：[0, 1] order = 0 图片在前(默认值)； order = 1 文字在前。 | |
| 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加上水印文件是 :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_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjU,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ,color_FFFFFF,shadow_50,order_0,align_2,interval_10,t_100,g_se,x_10,y_10

# 获取图片信息

# 获取图片主色调

本接口是为了获取图片的平均色调。

## 请求操作

操作名称：average-hue

## 返回格式

0xRRGGBB （RR GG BB都是十六进制表示红 绿 蓝三个颜色）

# 示例

可以在浏览器访问

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

得到结果

```
{"RGB": "0x5c783b"}
```

原图是：

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

0x5c783b对应的颜色RGB(92,120,59)是：

# 获取信息

# 获取基本信息和exif信息

可以通过 info操作 来获取文件的基本信息，包括宽度、长度、文件大小、格式。如果文件有exif信息，就返回exif信息；如果没有exif信息，就只返回基本信息。返回结果是json格式。exif详情可以参考EXIF2.31。

## 请求格式

操作名称：info

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| - | - | - |

## 返回格式

json格式

## 示例

- 没有exif 的例子

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

```
{
"FileSize": {"value": "21839"},
"Format": {"value": "jpg"},
"ImageHeight": {"value": "267"},
"ImageWidth": {"value": "400"}
}
```

- 有exif 的例子

http://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/info

```
{
"Compression": {"value": "6"},
"DateTime": {"value": "2015:02:11 15:38:27"},
"ExifTag": {"value": "2212"},
"FileSize": {"value": "23471"},
```

```
"Format": {"value": "jpg"},
"GPSLatitude": {"value": "0deg "},
"GPSLatitudeRef": {"value": "North"},
"GPSLongitude": {"value": "0deg "},
"GPSLongitudeRef": {"value": "East"},
"GPSMapDatum": {"value": "WGS-84"},
"GPSTag": {"value": "4292"},
"GPSVersionID": {"value": "2 2 0 0"},
"ImageHeight": {"value": "333"},
"ImageWidth": {"value": "424"},
"JPEGInterchangeFormat": {"value": "4518"},
"JPEGInterchangeFormatLength": {"value": "3232"},
"Orientation": {"value": "7"},
"ResolutionUnit": {"value": "2"},
"Software": {"value": "Microsoft Windows Photo Viewer 6.1.7600.16385"},
"XResolution": {"value": "96/1"},
"YResolution": {"value": "96/1"}}
```


# 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。

# 图片样式

所有对图片的变换都会加在URL后面，会导致URL变得冗长，不方便管理与阅读。图片处理服务提供用户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，只要用一个很短的URL就能实现相同的效果。

一个Bucket下面有多个样式，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@!stylename。其中@!是样式的分隔符，在URL带了这个分隔符，图片处理服务会把该分隔符后面的内容当成样式的名称。该方式为图片处理可选的使用方式，您也可以在控制台设置分隔符。同时还能支持-、_、/、!等多种可选样式分隔符。

- StyleName表示的是样式的名字。
- 创建样式、删除样式和修改样式都在前端控制台实现。
- 当访问的样式在指定Bucket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

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分隔符的存储空间名称。

单击**图片处理**页签，找到**访问设置**按钮。如下图所示：

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。

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_sytle | 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。

# 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>

细则说明：

- 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

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。

单击**图片处理**页签，找到**访问设置**按钮。如下图所示：

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。

设置好您需要的选项之后单击**确定**完成图片保护的设置。

# 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的operation来支持另存为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，如果不指定的话默认保存到当前bucket。 |

# 细节分析

- saveas操作需要调用方有写目标bucket和object的权限，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**

```
PUT /?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:KU5h8YMUC78M30dXqf3JxrTZHiA=

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
```

# 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 |
| --- | --- | --- |
| t | 截图时间 | 单位ms，[0,视频时长] |
| w | 截图宽度，如果指定为0则自动计算 | 像素值：[0,视频宽度] |
| h | 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] |
| m | 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的最近的一个关键帧 | 枚举值：fast |
| f | 输出图片格式 | 枚举值：jpg、png |

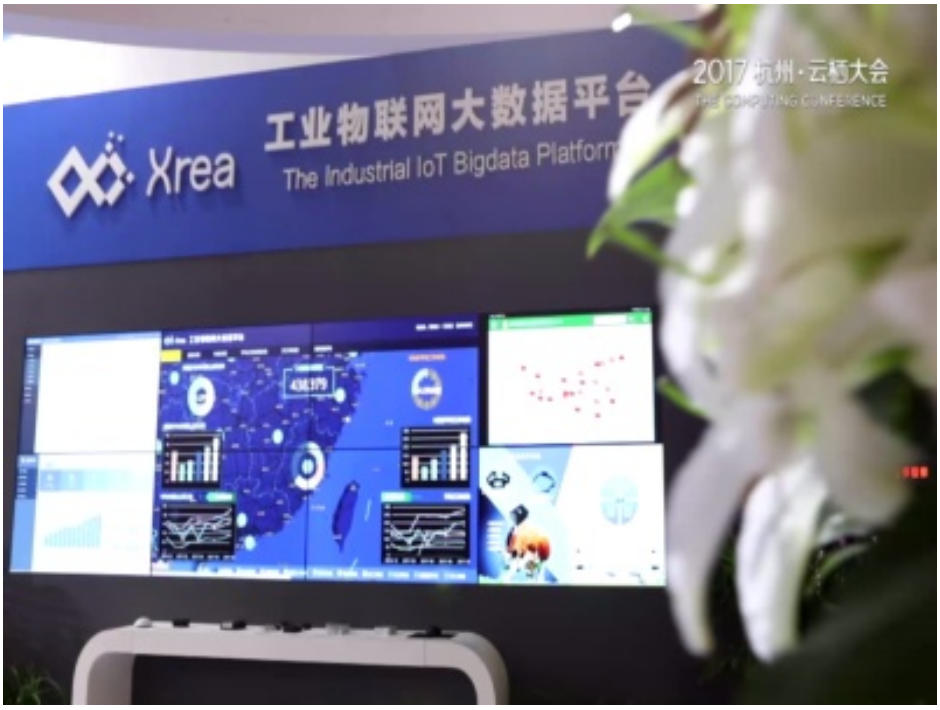
## 示例

- 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




# 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,parame_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。

# 老版图片服务手册

# 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，新版接口详情参考这里。使用兼容详细说明参考这里。

# 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(.)。

样式访问示例

# 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。 |
| 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
以图片访问的URL为例子
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 x 4096， 且单边的长度不能超过4096 x 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。

# 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 |
| 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OSS数据中心及访问域名

# 接入图片服务

# 图片URL规则

图片服务都是使用标准的HTTP的GET请求来访问的，所有的处理参数也是编码在URL中的。

## 直接获取原图

用户有两种方式访问图片，分别是：

- 三级域名
- 自定义域名

### 三级域名访问的URL

```
http://channel.<endpoint>/object
```

这里的endpoint指的是用户图片所在的Region的图片服务的访问域名，关于访问域名请参考访问域名，object为用户所关联的channel上存储的原图片。

## 自定义域名访问的URL

另外一种方式是通过用户绑定的图片服务域名，也就是自定义域名来访问，形式如下：

```
http://userdomain/object
```

其中userdomain为用户开通图片服务绑定的自定义域名，这个域名会关联到一个channel，这里假设用户自定义域名userdomain已经CNAME到channel.endpoint这个三级域名上。 如何绑定图片服务的自定义域名可以参考开通图片服务。object为用户所关联channel上存储的原图片。

# 通过处理参数访问原图

如果用户对原图进行一定的处理然后返回，同样有两种形式，URL的格式如下：

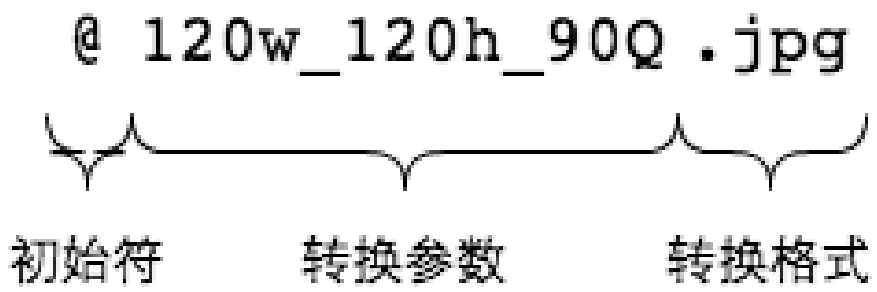
```
通过三级域名访问
http://channel.<endpoint>/object@action.format

channel：用户的IMG频道
endpoint：用户的Channel所在数据中心的访问域名
object：用户上传在OSS上的图片文件
action：用户对图片做的处理
format：用户指定处理后的图片格式



通过用户自定义域名访问
http://userdomain/object@action.format
```

一个典型的URL的例子如下：

```
http://bucket.endpoint/object@100w_100h_90Q.jpg 三级域名
http://userdomain/object@100w_100h_90Q.jpg 自定义域名
```

其中的@100w_100h_90Q为转换的具体action，jpg为转换的format，合在一起的100w_100h_90Q.jpg称为转换字符串，用来指定对于目标图片的操作，通过指定转换字符串，生成并返回另一张转换处理后的图片。

一个典型的转换字符串，如“@100w_100h_90Q.jpg”，代表需要一张宽（w）100px、高（h）100px、绝对质量（Q）90%、jpg格式的图片。




## 通过样式访问原图

为了简化使用，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要指定某个样式即可。使用样式来进行图片处理的URL形式如下：

```
http://userdomain/object@!style
```

@!是目前支持的样式分隔符, style 是样式的名称

如：http://image-demo.img.aliyuncs.com/example.jpg@!pipe1

其中pipe1 是样式名称。

# 关键词

## 顺序无关

转换参数中键值对是循序无关的，即"120w_120h_90Q"和"90Q_120w_120h"都能取到想要的图片，系统会对参数按照本规范以下定义的顺序重新排序后再处理。（由于参数的顺序不同有时会表达不同的语义，如“100w_100h_2x”表达的是“先缩放到100*100，再放大2倍”，即得到200*200的图片；而"2x_100w_100h"按照字面顺序理解是“先放大2倍再缩放到100*100”，即得到100*100的图片，为了避免这样的理解误差，同时简化处理方式，IMG会对参数按照文档中出现的顺序排序后处理。上例中的"2x_100w_100h"会被理解为“100w_100h_2x”，得到200*200的图片。）

## 覆盖处理

如果转换参数中出现多个相同“键”，后面定义的覆盖前面定义。如“120w_120h_240w”等同于“120h_240w”。

## 冲突处理

见每个参数中关于冲突的说明。

## 长边与短边

关于“长边”和“短边”的定义需要特别注意，它们表达的是在缩放中相对比例的长或短。“长边”是指原尺寸与目标尺寸的比值大的那条边；“短边”同理。如原图400 * 200，缩放为800 * 100，（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是短边。

## 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

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

# 快速开始

## 基于控制台快速开始：

1. 登录OSS控制台，开通OSS。
2. 创建一个Bucket。
3. 上传和下载图片。

具体见控制台快速开始文档。

## 快速了解图片的上传下载

开始使用SDK之前，请先参考开发人员指南关于上传下载文件的功能介绍。

IMG是使用RESTful API来操作的，所有的请求都是标准的HTTP请求。

- 图片服务中的图片的上传等同于OSS的上传。OSS提供了多种类型的上传文件的方法，如使用单次PUT请求完成的简单上传、使用网页表单直接上传的表单上传、以及用于大文件上传的分片上传。
- 图片服务的下载可以通过浏览器等发送HTTP GET请求图片的URL即可获得图片。

## 快速体验图片服务的功能

图片服务功能体验请参考http://oss-demo.aliyuncs.com/image/index.html

# 用户鉴权

如果用户需要不经过任何授权，通过浏览器即可匿名访问图片服务来处理图片，需要在创建Bucket的时候将Bucket的权限设置为公共读。

创建Bucket

- 控制台：创建Bucket
- SDK：Java SDK-Bucket中新建Bucket
- API：Put Bucket

设置Bucket权限

- 控制台: 创建Bucket权限设置
- SDK：Java SDK-Bucket中**设置Bucket ACL**
- API：Put BucketACL

默认创建的Bucket权限是私有读写，Object默认继承Bucket的权限。 如果用户需要通过图片服务访问私有的Object，需要进行身份鉴权。

图片服务的用户鉴权方式和OSS是一致的，有两种鉴权方式：

- 在Header中包含签名
- 在URL中包含签名

具体可参考在Header中包含签名和在URL中包含签名。

# 在URL中包含签名

假定用户绑定的域名www.test.com对应的频道名字为image-demo，object名字为example.jpg，转换字符串为100w.jpg。

首先需要计算Signature字段，计算方法如下：

```
Signature = base64(hmac-sha1(AccessKeySecret,
        VERB + "\n"
        + Content-MD5 + "\n"
        + Content-Type + "\n"
        + Expires + "\n"
        + CanonicalizedOSSHeaders
        + CanonicalizedResource))
```

- AccessKeySecret表示签名所需的秘钥
- VERB表示HTTP请求方法，例如PUT，GET，DELETE等
- Content-MD5表示请求内容数据的MD5值，对于图片处理服务这里为空字符串
- Content-Type表示请求内容的类型，对于图片处理服务这里为空字符串
- Expires表示授权给用户URL签名过期时间
- CanonicalizedOSSHeaders表示HTTP中的ObjectMeta组合，对于图片处理服务这里为空字符串
- CanonicalizedResource 表示用户想要访问的OSS资源，在图片处理服务中这项的组成，格式为 /channelname/object@处理参数。
- 构建CanonicalizedResource的方法如下：
    i. 将CanonicalizedResource置成空字符串("")
    ii. 放入要访问的图片服务资源：“`/channelname/object”（无Object则不填）
    iii. 在结尾添加处理参数：@处理参数，（无处理参数则不填）。此时CanonicalizedResource例子如：/channelname/object@100w.jpg。
    iv. 如果涉及样式管理操作，那么将这些查询字符串及其请求值按照字典序，以&分割，添加到CanonicalizedResource中。此时的CanonicalizedResource例子：/channelname?style&styleName=YourStyleName

例子中的CanonicalizedResource为/image-demo/example.jpg@100w.jpg。

注意：上例中的转换字符串可以是简单缩略，文字水印，图片水印、管道和样式（样式的分隔符是@!）

这里需要注意的是，使用URL签名中Expires和CanonicalizedResource不能为空。

最后生成在URL签名，必须在参数后包含OSSAccessKeyId、Expires、Signature这三项，具体生成方法可以参考OSS的API文档中的在URL中包含签名，上文的例子生成的URL签名如下：

```
http://www.test.com/example.jpg%40100w.jpg?OSSAccessKeyId=j4y55h3z88ihxxhlr9nhjjis&Expires=1392949804&Signature=IDBJ09e8Ow4GaPRM1yIf7plH/CI%3D
```

# 在Header中包含签名

除了在URL中包含签名之外，还可以在HTTP请求的Header中包含签名，签名是由Authorization这个Header指定的，具体的构成规则如下：

```
"Authorization: OSS " + AccessKeyId + ":" + Signature
Signature = base64(hmac-sha1(AccessKeySecret,
        VERB + "\n"
        + Content-MD5 + "\n"
        + Content-Type + "\n"
        + Date + "\n"
        + CanonicalizedOSSHeaders
        + CanonicalizedResource))
```

- AccessKeySecret表示签名所需的秘钥
- VERB表示HTTP请求方法，例如PUT，GET，DELETE等
- Content-MD5表示请求内容数据的MD5值，对于图片处理服务这里为空字符串
- Content-Type表示请求内容的类型，对于图片处理服务这里为空字符串
- Date表示此次操作的时间，且必须为HTTP1.1中支持的GMT格式
- CanonicalizedOSSHeaders表示 http中的object meta组合，对于图片处理服务这里为空字符串
- CanonicalizedResource构造方法请参考上文URL签名中的CanonicalizedResource的生成方法。

需要注意的

- Date和CanonicalizedResource不能为空
- 如果请求中的Date时间和OSS服务器的时间差正负15分钟以上，OSS图片处理服务将拒绝该服务，并返回HTTP 403错误。

# 使用SDK处理图片

本节主要介绍如何使用 OSS 的 Python SDK 去获取 private bucket 的图片处理服务，因为图片处理服务都是 GET 操作，所以使用 OSS Python SDK 主要以 Get Object 为主。

以 Python SDK 为例，在 Python SDK里有一个 get_object 操作，传入的参数一般是 bucket，object。

## OSS

获取bucket: image-demo，object：example.jpg

```
bucket = 'image-demo'
object = 'example.jpg'
```

```
self.oss.get_object(bucket，object)
```

# 图片服务

## 简单缩略

获取bucket: image-demo，object: example.jpg ，

转换字符是： 100w_100h.jpg

```
bucket = 'image-demo'
object = 'example.jpg'
query = '100w_100h.jpg'
object = object + '@' + query
self.oss.get_object(bucket， object)
```

## 图片水印

获取bucket: image-demo，object: example.jpg ，

转换字符是：watermark=1&object=cGFuZGEucG5n&t=90&p=5

```
bucket = 'image-demo'
object = 'example.jpg'
query = 'watermark=1&object=cGFuZGEucG5n&t=90&p=5'
object = object + '@' + query
self.oss.get_object(bucket， object)
```

## 文字水印

获取bucket: image-demo， object:example.jpg

转换字符是：watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ

```
bucket = 'image-demo'
object = 'example.jpg'
query = ' watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ '
object = object + '@' + query
self.oss.get_object(bucket， object)
```

## 样式

获取bucket: image-demo， object:example.jpg

样式名：pipe1

```
bucket = 'image-demo'
object = 'example.jpg'
style = ' pipe1 '
object = object + '@!' + style
self.oss.get_object(bucket， object)
```

## 管道

获取bucket: image-demo， object:example.jpg

管道操作: 200w.jpg|watermark=1&object=cGFuZGEucG5n&t=90&p=5

```
bucket = 'image-demo'
object = 'example.jpg'
query = ' 200w.jpg|watermark=1&object=cGFuZGEucG5n&t=90&p=5'
object = object + '@' + query
self.oss.get_object(bucket， object)
```

# 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 开发人员指南中的上传文件

假如用户需要使用杭州的图片服务，域名为img-cn-hangzhou.aliyuncs.com **上传前提条件**：

- 1 同区域OSS存储空间(Bucket)。例如叫oss-sample，杭州的OSS访问域名为oss-cn-hangzhou.aliyuncs.com。
- 2 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

# 图片缩放

# 单边固定缩略

可以对图片某一边（宽或高）进行固定到一个长度，另外一边按照比例进行调整。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| w | 指定目标缩略图的宽度 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |
| l | 目标缩略图大于原图是否处理。值是1, 即不处理，是0，表示处理 | 0/1, 默认是0 |

## 注意事项

- 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。
- 如果只指定宽度或者高度，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。

## 示例

- 将图缩略成高度为100，宽度按比例处理。

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

- 将图缩略成宽度为100，高度按比例处理。

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

- 将图缩略成宽度为500，高度按比例处理，如果目标缩略图大于原图不处理。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@500w_1l

# 指定宽高缩略

可以对图片指定宽或高原图，按照长边短边进行调整。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| w | 指定目标缩略图的宽度 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |
| e | 缩放优先边， 默认值：0：长边（默认值） 由于图片缩放过程中，原图尺寸与缩放尺寸不一定是相同比例，需要指定以长边还是短边优先进行缩放，如原图200 * 400（比例1:2），需要缩放为100 * 100（比例1:1）.长边优先时，缩放为50 *100；短边优先时(e=1)，缩放为`100* 200`，若不特别指定，则代表长边优先 | 0表示按长边优先，默认值<br>1表示按短边优先 |
| l | 目标缩略图大于原图是否处理。如果值是1, 即不处理，是0，表示处理 | 0/1, 默认是0 |

# 注意事项

- 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。
- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。

# 示例

- 将图缩略成宽度为100，高度为100，按长边优先

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w_0e

- 将图缩略成宽度为100，高度为100，按短边优先

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w_1e

# 强制宽高缩略

可以强制指定目标缩略图的高度和宽度，忽略原图的宽高比。注意这可能会导致图片变形。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| w | 指定目标缩略图的宽度 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |
| e | 缩放优先边，如果是强制缩略，值是:2 | 2（强制缩略） |
| l | 目标缩略图大于原图是否处理。如果值是1, 即不处理，是0，表示处理 | 0/1, 默认是0 |

## 注意事项

- 此操作会导致图变形。
- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。

## 示例

- 将图强制缩略成宽度为100，高度为100

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w_2e

# 按比例缩放

可以通过指定一个比例百分比参数，让图片按照指定的比例进行缩略或者放大。

## 参数

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

注意:

- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。
- 如果参数p跟w, h 合用时，p将直接作用于w, h (乘以p%) 得到新的w,h， 如100w_100h_200p 的作用跟200w_200h 的效果是一样的。
- 如果对图片进行倍数放大，单边的最大长度不能超过4096 * 4 。

# 示例

- 将图按比例放大两倍。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@200p

- 将图按比例缩略到原来的1/2

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@50p

# 缩略后填充

先把图按短边优先缩略，然后再用指定颜色填充剩余区域

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| w | 指定目标缩略图的宽度 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |
| e | 缩略优先边，这里必须指定值为4 | 4 |
| bgc | 指定填充的背景颜色。默认不指定，为白色填充。参数格式：<red>-<green>-<blue>bgc<br>如：100-100-100bgc<br>red,green,blue指定一个颜色 | Red, green, blue[0-255] |

## 注意事项

- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。

- bgc 由红绿蓝三原色这三个参数指定生成对应颜色。

# 示例

- 将图按短边缩略到100x100, 然后按白色填充

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_4e

- 将图按短边缩略到100x100, 然后按红色填充

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_4e_100-0-0bgc

# 图片裁剪

# 自动裁剪

自动裁剪表示图片先按短边缩略，然后从缩略的目标图片裁剪出中间部分得到对应指定高度和宽度的目标缩略图。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| w | 指定目标缩略图的宽度 | 1-4096 |
| h | 指定目标缩略图的高度。 | 1-4096 |

| | | |
|---|---|---|
| e | 缩放优先边，这里指定按短边优化 | 1 |
| c | 是否进行裁剪。如果是想对图进行自动裁剪，必须指定为1 | 0，1 |
| l | 如果目标缩略图大于原图是否处理，值是1, 即不处理，是0，表示处理 | 0/1, 默认是0 |

# 注意事项

- 自动裁剪从按短边优先缩略的图中间进行裁剪，如果想从裁剪出图的左边部分或者右边部分。即不指定裁剪参数C, 然后再利用管道实现
- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。

# 示例

- 将图自动裁剪成宽度为100，高度为100的效果图

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w_1e_1c

- 将图片按短边裁剪然后，裁剪出左半部分。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w_1e|0-0-100-100a

# 区域裁剪

可以将图片分成多个区域，按照区域进行裁剪。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| rc | 用户可以指定对某一个区域进行裁剪。在这里把图片分成 9 个区域。<br>参数格式：<width>x<height>-<pos>rc.jpg<br>width 指的是裁剪的宽度[0, 4096]<br>Height 指的是裁剪的高度[0, 4096]<br>pos指的是裁剪区域,取值范围是 | width, height的范围是[1,4096] |

| | | |
|---|---|---|
| | [1,9],默认是左上角,区域数值对应表见下图<br>如果想裁剪左上角，宽度是100,高度是200的区域，参数是：100x200-1rc<br>如果想裁剪左上角，宽度是100,高度是图片的原高度参数是：100x0-1rc 或者100x-1rc<br>如果高度或者宽度不填，或者参数是0，或者参数大于原图。默认是按原图的高度或宽度返回。 | |

**区域数值对应表**

| | | |
|---|---|---|
| 1 左上 | 2 中上 | 3 右上 |
| 4 左中 | 5 中部 | 6 右中 |
| 7 左下 | 8 中下 | 9 右下 |

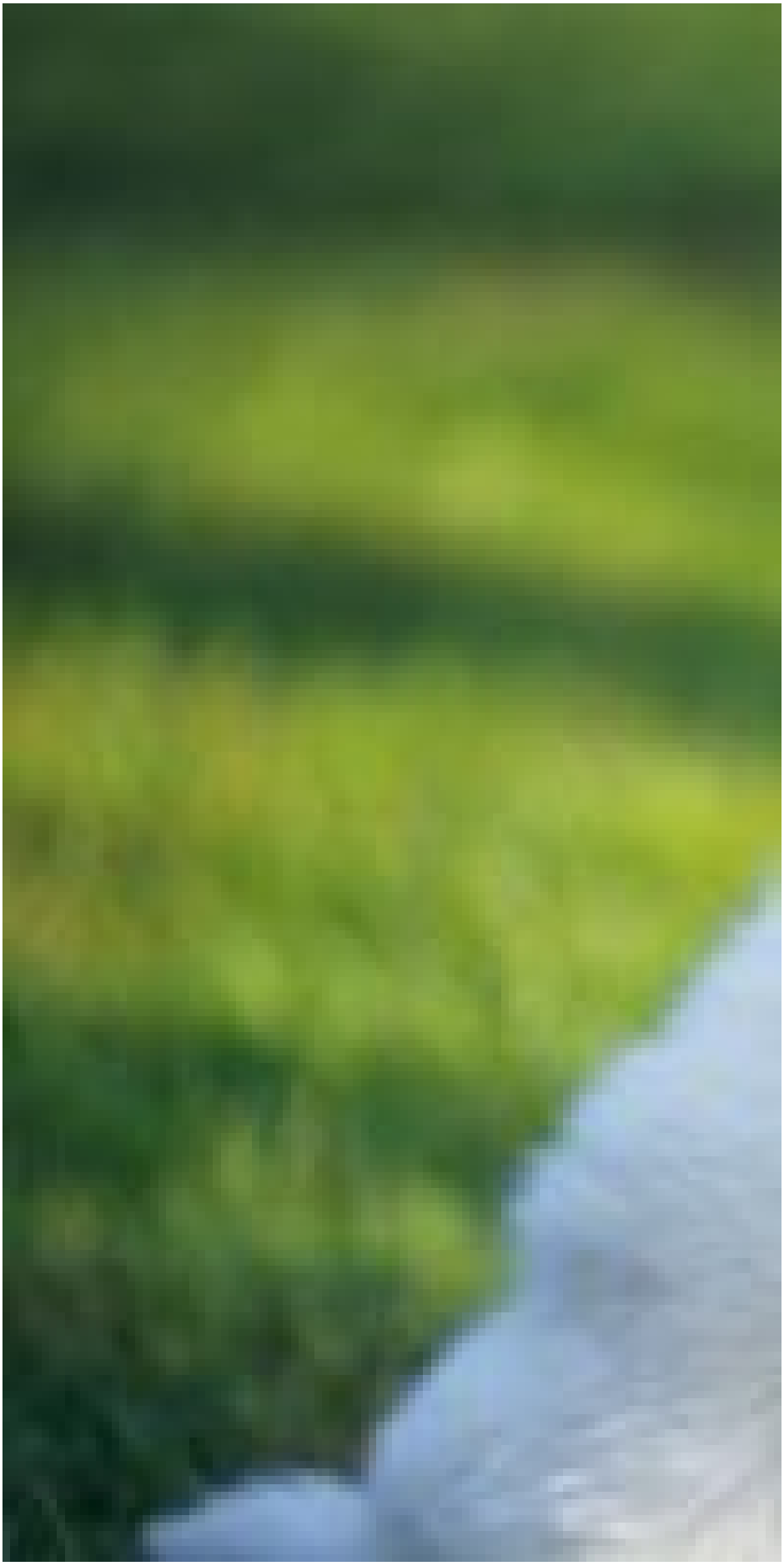
# 注意事项

- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。
- 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。

# 示例

- 裁剪原图左上区域宽度100高度200的区域

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100x200-1rc



# 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

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| ci | 从图片取出圆形区域<br>参数格式：[radius]-[type]ci | radius : [1, 4096]<br>如果radius能指定圆的半径。但是圆的的半径不能超过原图的最小边的一半。如果半径超过。圆的大小仍然是原圆的最大内切圆。<br>type: [0, 1]<br>0:表示图片最终大小仍然是原图大小<br>1: 表示图片最终大小是能包含这个圆的最小正方形 |

## 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。推荐使用保存成png格式
- 指定半径大于原图最大内切圆的半径。 圆的大小仍然是图片的最大内切圆。

## 示例

- 裁剪半径是100, 保持圆是原来大小

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100-0ci

- 裁剪半径是100, 保存圆是能包含圆的最小正方形，格式是png

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100-1ci.png

- 裁剪半径是1000, 保存圆是能包含圆的最小正方形，格式是png

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@1000-1ci.png

# 圆角矩形

可以保存图片是圆角矩形，并可以指定圆角的大小 。

## 参数

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| 2ci | 从图片取出圆形区域<br>参数格式：[radius]-2ci | radius : [1, 4096]<br>radius指定圆角的半径。<br>但是生成的最大圆角的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 。推荐使用保存成png格式
- 指定半径大于原图最大内切圆的半径。 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.img-cn-hangzhou.aliyuncs.com/example.jpg@30-2ci

- 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.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_1e_1c_10-2ci.png

# 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

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| ic | 参数格式：[length]x-[index]ic<br>或[length]y-[index]ic<br>length是切割长度<br>index 是表示块数。**（0表示第一块）**<br>其中x表示按x轴，水平线切割。<br>y 表示按y轴，垂直线切割 | length:[1,切割边边长]，单位px。如果超出切割边的大小，返回原图。<br>index:[0,最大块数)。如果超出最大块数，返回原图。 |

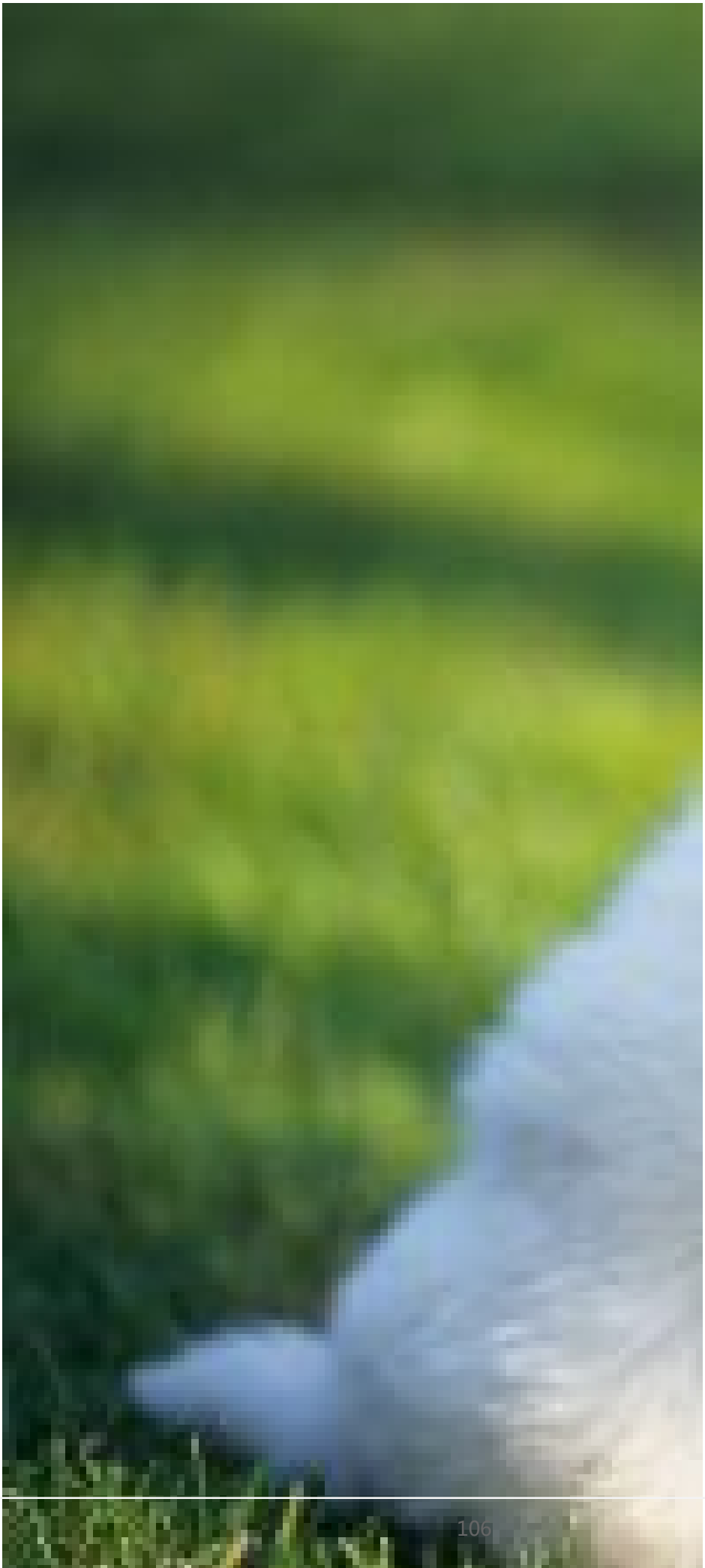
# 注意事项

- 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。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100x-0ic



对图片y轴按100平分，取出第一块。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100y-0ic

对图片x轴按100平分，取出第一百块，仍然是原图

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

# 高级裁剪

可以通过指定起始横坐标，纵坐标及裁剪的宽度和裁剪的高度对图进行高级裁剪。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| a | 参数的类型：x-y-width-length<br>如：100-50-200-150a<br>一共四个参数，每个参数之间以”-“隔开。第一个参数表示起 | width, height的范围是1-4096 |

| | | |
|---|---|---|
| | 始点x坐标（以左上角为原点）,第二个参数表示起始点y坐标,第三个参数表示要裁剪的宽度，第四个参数表示要裁剪的高度。如100-50-200-150a 表示从点(100, 50) 裁剪大小为(200, 150)的图片。<br>注意：可以将第三个参数，第四个参数置为0, 表示裁剪到图片的边缘。如100-50-0-0a 表示从点(100, 50) 裁剪到图片的最后 | |

# 注意事项

- 如果不指定格式，原图将默认转换成jpg格式，如果原图是png, webp, bmp可能会导致图出现变形。详细可以查看质量变换及格式转换。
- 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.img-cn-hangzhou.aliyuncs.com/example.jpg@100-50-0-0a

- 裁剪图从起点(100, 50)到裁剪100x100的大小

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100-50-100-100a

# 图片旋转

## 旋转

可以对处理后的图片进行按顺时针旋转

### 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| r | 默认值 ：0(表示不旋转) | [0, 360] |

# 注意事项

- 旋转后的图可能会导致图的尺寸变大。
- 旋转对图的尺寸有限制，图片的宽或者高不能超过4096。

# 示例

- 将原图缩略成宽度为100,高度为100,按顺时针旋转90度

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_90r

- 将原图按顺时针旋转270度

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@270r

# 自适应方向

某些手机拍摄出来的照片可能带有旋转参数（存放在照片exif信息里面）

# 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| o | 进行自动旋转<br>0：表示按原图默认方向，不进行自动旋转<br>1：表示根据图片的旋转参数，对图片进行自动旋转，如果存在缩略参数，是先进行缩略，再进行旋转。<br>2: 表示根据图片的旋转参数，对图片进行自动旋转，如果存在缩略参数，先进行旋转，再进行缩略 | [0, 2] |

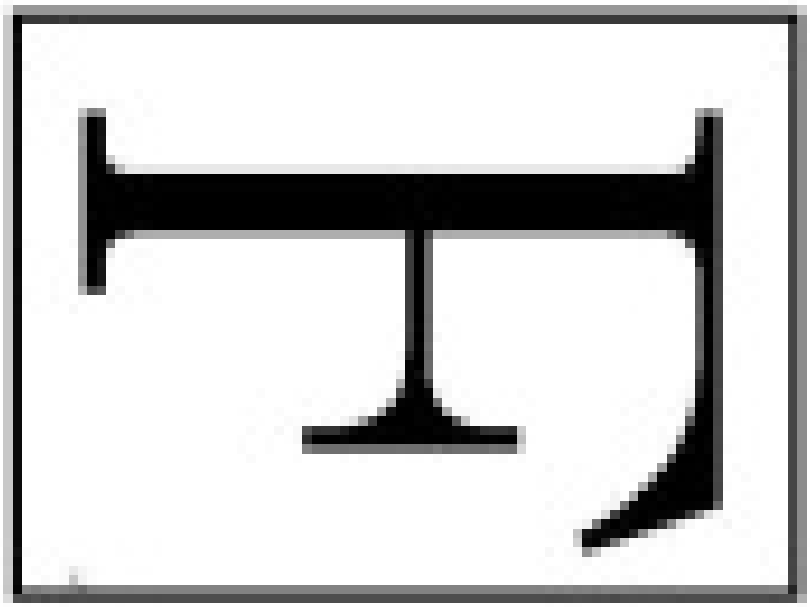
# 注意事项

- 如果采用缩略旋转1,可能会导致图片最终的宽度和高度跟指定的参数不符。
- 进行自适应方向旋转，必须要求原图的宽度和高度必须小于4096.
- 如果原图是没有旋转参数，加上1o, 2o参数不会对图有影响。

# 示例

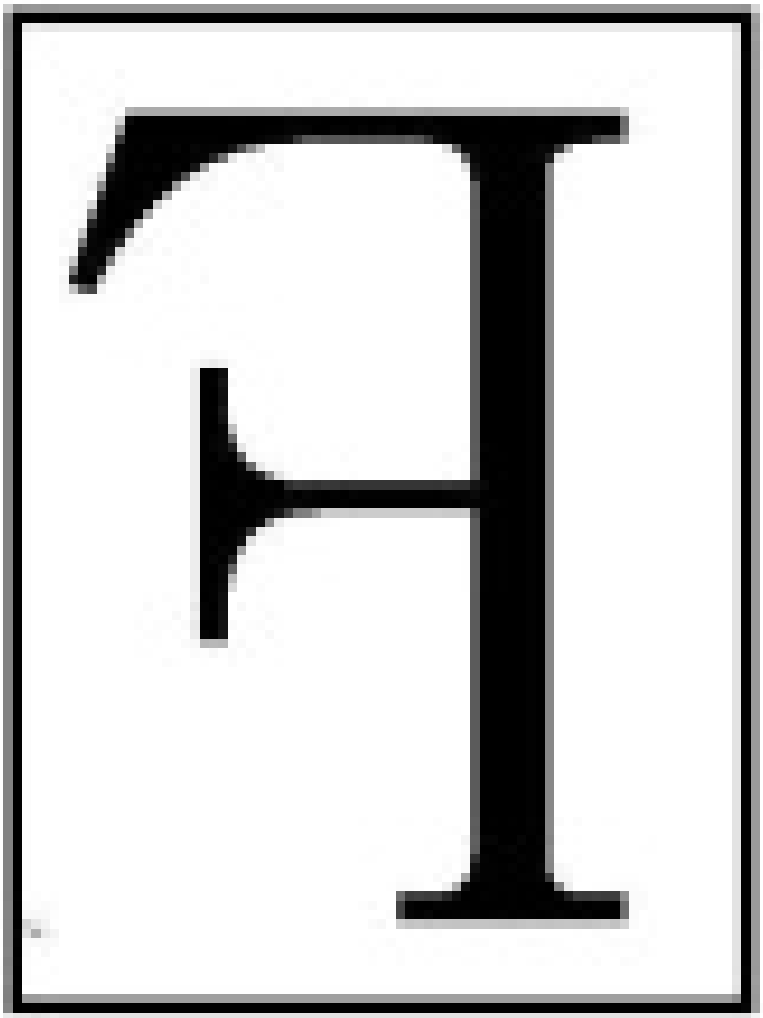
- 将图缩略成宽度为100，对图片不做自动旋转处理

http://image-demo.img-cn-hangzhou.aliyuncs.com/f.jpg@100w.jpg



- 将图缩略成宽度为100，对图片进行自动旋转1

http://image-demo.img-cn-hangzhou.aliyuncs.com/f.jpg@100w_2o.jpg



得到的目标效果图宽度是100，高度是127。

- 将图缩略成宽度为100，对图片进行自动旋转2

http://image-demo.img-cn-hangzhou.aliyuncs.com/f.jpg@100w_1o.jpg

得到的目标效果图宽度是79, 高度是100。

# 图片效果

# 锐化

可以对处理后的图片进行锐化处理，使图片变得清晰。

# 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| sh | 表示进行锐化处理。取值为锐化参数，参数越大，越清晰。 | [50, 399]<br>为达到较优效果，推荐取值为100 |

# 示例

- 对原图进行锐化处理，锐化参数为100。

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

- 对原图进行锐化处理，锐化参数为50。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@50sh

- 将原图缩略成高度100, 宽度100，并进行锐化操作，保存成png格式

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_100sh.png

# 模糊效果

可以对图片进行模糊操作。

# 参数

| 参数 | 描述 | 取值 |
| --- | --- | --- |
| bl | 参数格式：[radius]-[sigma]bl<br>radius是模糊半径<br>sigma是正态分布的标准差<br>如：3-2bl 模糊半径是3,标准差是2 | radius取值在 [1,50]，radius越大，越模糊<br>sigma取值 [1,50]，越大，越模糊 |

# 示例

- 对图片进行模糊半径是3, 标准差是2。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@3-2bl

- 图片先自动裁剪成100x100, 然后对图片进行模糊半径是3, 标准差是2。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_1e_1c_3-2bl

# 亮度和对比度

可以对处理后的图片进行亮度和对比度调节

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| b | 亮度调整<br>0表示原图亮度，小于0表示亮度越低，大于0表示亮度越高 | [-100, 100] |
| d | 对比度调整<br>0表示原图对比度，小于0表示对比越低，大于0表示对比越高 | [-100, 100] |

# 示例

- 将原图只进行亮度调整

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@50b

- 将原图只进行对比度调整

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@-50d

- 将原图进行亮度和对比度调整

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@-50d_50b

# 图片水印

## 概述

水印操作可以在图片上设置另外一张图片或者文字做为水印。

水印类型分成图片水印，文字水印，和文图混合水印。

## 基本参数

图片水印，文字水印，和文图混合水印都可以使用如下参数。

| 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。<br>默认值：100， 表示 100%（不透明） 取值范围: [0-100] | 可选参数 |

| | | |
|---|---|---|
| p | 参数意义：位置，水印打在图的位置，位置如成如下图。<br>默认值：9，表示在右下角打水印<br>取值范围：[1-9] | 可选参数 |
| 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<br>默认值：10<br>取值范围：[0 – 4096]<br>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<br>默认值：10<br>取值范围：[0 – 4096]<br>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。<br>默认值：0<br>取值范围：[-1000, 1000]<br>单位：像素(px) | 可选参数 |
| watermark | 参数意义：选择水印的类型 | 取值范围 1, 2, 3<br>1：图片水印 2：文字水印<br>3：文图混合水印 |

**区域数值对应表**

| | | |
|---|---|---|
| 1 左上 | 2 中上 | 3 右上 |
| 4 左中 | 5 中部 | 6 右中 |
| 7 左下 | 8 中下 | 9 右下 |

## 注意事项

- 水平边距、垂直边距、中线垂直偏移可以调节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位编码可以参考关键词。

## 示例

- 右下角打上文字水印

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ

- 右下角打上文字水印，水平边距是 10，垂直边距20

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&t=90&p=9&x=10&y=20

- 右中部分打上水印，水平边距为10， 垂直中线偏移为20,透明度为50

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&t=50&p=6&x=10&voffset=20

**注意**

文字水印参数请参考文字水印

# 图片水印

图片水印就是在原图的基础上加上一张水印图片

## 访问类型

```
@watermark=1&object=<encodedobject>&t=<transparency>&x=<distanceX>&y=<distanceY>&p=<position>...
```

其中watermark与object两个参数为必填项。文中出现的url_safe_base64_encode指的是URL安全base64编码，请参见关键词。

## 参数

| 名称 | 描述 | 参数类型 |
| --- | --- | --- |
| object | 参数意义： 水印图片的object名字(必须编码)<br>注意：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 | 必选参数 |

| | | |
|---|---|---|
| | ” cGFuZGEucG5n” | |

水印位置参数请参考基本参数

# 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），但不支持管道操作。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10P, 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。

如果水印操作是：watermark=1&object=cGFuZGEucG5nQDMwUA&t=90&p=9&x=10&y=10（右下角打水印。水印图片是：panda.png@30P, 表示水印的大小按主图的30%缩放。）

如果原图按宽度是400，需要缩略，再打上述水印的示例：

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@400w|watermark=1&object=cGFuZGEucG5nQDMwUA&t=90&p=9&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@300w|watermark=1&object=cGFuZGEucG5nQDMwUA&t=90&

p=9&x=10&y=10

# 访问形式：

参数中的object规则是：

- 1 图片水印原图名字（未经过URL安全base64编码的名字） + @ + Action
- 2 对第一步的字符串进行URL安全base64编码

如果要指定对水印图片进行预处理，处理参数带在水印object之后，以@符号连接。如：

- 对panda.png 不进行任何预处理: object = url_safe_base64_encode(“panda.png”)
- 对panda.png 进行放大2倍: object = url_safe_base64_encode(“panda.png@200p”)
- 对panda.png 进行缩小一倍，亮度调节成50, 对比度调节成40: object = url_safe_base64_encode(“panda.png@50p_50b_40d”)
- 对panda.png 增加按宽度50缩略，亮度调节成30: object = url_safe_base64_encode(“panda.png@50w_30b”)
- 对panda.png 增加按高度20缩略，对比度调节成10: object = url_safe_base64_encode(“panda.png@20h_10d”)
- 对panda.png 水印图的大小基于原图的20%进行处理，对比度调节成10: object = url_safe_base64_encode(“panda.png@20P_10d”)

# 示例

- 下面URL的含义，是example.jpg加上水印文件是 :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@watermark=1&object=cGFuZGEucG5n&t=90&p=9&x=10&y=10

- 对panda.png按宽度是50缩放。 那么水印文件是：panda.png@50w （ panda.png@50w 经过URL安全base64编码后是:cGFuZGEucG5nQDUwdw)

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=1&object=cGFuZGEucG5nQDUwdw&t=90&p=9&x=10&y=10

- 对panda.png按50%的比例缩小。那么水印文件是： panda.png@50p (panda.png@50p经过URL安全base64编码后是：cGFuZGEucG5nQDUwcA ）

http://image-demo.img-cn-

hangzhou.aliyuncs.com/example.jpg@watermark=1&object=cGFuZGEucG5nQDUwcA&t=90&p=9&x=10&y=10

- 对panda.png，自动裁剪成宽度是40，高度是30大小。那么水印文件是：panda.png@40w_20h_1e_1c(panda.png@40w_20h_1e_1c经过URL安全base64编码后是：cGFuZGEucG5nQDQwd18yMGhfMWVfMWM）

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=1&object=cGFuZGEucG5nQDQwd18yMGhfMWVfMWM&t=90&p=9&x=10&y=10

- 对panda.png. 高级裁剪，从起点(0, 0) 裁剪到(65, 65)的位置。那么水印文件是:panda.png@0-0-65-65a (panda.png@0-0-65-65a经过URL安全base64编码后是cGFuZGEucG5nQDAtMC02NS02NWE）

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=1&object=cGFuZGEucG5nQDAtMC02NS02NWE&t=90&p=9&x=10&y=10

# 文字水印

文字水印就是在原图的基础上加上一段文字内容做为水印。

## 访问类型

```
@watermark=2&text=<encodeText>&type=<encodeType>&size=<size>&color=<encodecolr>&t=<t>&p=<p>&x=<x>&voffset=<offset>&y=<y>
```

其中watermark与object两个参数为必填项。 文中出现的url_safe_base64_encode指的是URL安全base64编码，请参见关键词。

## 参数

| 名称 | 描述 | 参数类型 |
| --- | --- | --- |
| text | 参数意义：表示文字水印的文字内容(必须编码)<br>注意：必须是URL安全base64编码 encodeText = | 必选参数 |

| | | |
|---|---|---|
| | 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) | |
| type | 参数意义：表示文字水印的文字类型(必须编码)<br>注意：必须是URL安全base64编码 encodeText = url_safe_base64_encode(fontType)<br>取值范围：见下表（文字类型编码对应表）<br>默认值：wqy-zenhei（编码后的值：d3F5LXplbmhlaQ) | 可选参数 |
| color | 参数意义：文字水印文字的颜色(必须编码)<br>注意：参数必须是URL安全base64编码 EncodeFontColor = url_safe_base64_encode(fontColor) 参数的构成必须是：# + 六个十六进制数 如：#000000表示黑色。 #是表示前缀，000000每两位构成RGB颜色， #FFFFFF表示的是白色<br>默认值：#000000黑色base64编码后值：IzAwMDAwMA | 可选参数 |
| size | 参数意义：文字水印文字大小(px)<br>取值范围：(0，1000]<br>默认值：40 | 可选参数 |
| s | 参数意义：文字水印的阴影透明度<br>取值范围：(0,100] | 可选参数 |

水印位置参数请参考基本参数

**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 | ZmFuZ3poZW5na2F | 根据RFC，可省略填充 |

| | | | |
|---|---|---|---|
| | | pdGk= | 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 |

# 示例

- 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：右中，水平边距是：10， 中线垂直偏移是：20

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&s=50&t=90&p=6&x=10&voffset=20

- 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@watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ

# 文图混合水印

文图混合水印就是文字，图片并列一起做为水印打在图片上。

## 访问类型

```
@watermark=3&object=<encodeObject>&text=<encodeText>&type=<encodeType>&size=<size>&color=<encodecolor>&order=<order>&align=<align>&interval=<interval>&t=<t>&p=<p>&x=<x>&y=<y>
```

其中watermark与object两个参数为必填项。 文中出现的url_safe_base64_encode指的是URL安全base64编码，请参见关键词。

## 参数

文图混合水印，相当于文字水印跟图片水印的混合，并行在一行输出。所以文图混合水印支持文字水印和图片水印的参数。其中object, text是必选参数。

| 名称 | 描述 | 参数类型 |
| --- | --- | --- |
| order | 参数意义： 文字，图片水印前后顺序<br>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<br>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<br>取值范围: [0, 1000] | 可选参数 |

水印位置参数请参考基本参数

# 示例

- 单纯文字水印，文字内容是：Hello, 图片服务! 阴影是50, 位置在右下角， 水平边距和垂直边距都是10, 水印透明是:90

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&s=50&t=90&p=9&x=10&y=10

- 单纯图片水印，图片object 是panda.png，位置在右下角， 水平边距和垂直边距都是10, 水印透明是:90

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=1&object=cGFuZGEucG5n&t=90&p=9&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:90， 排版方式是图片前， 对齐方式是中对齐，间距是10

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=3&object=cGFuZGEucG5n&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&order=0&align=1&interval=10&t=90&p=9&x=10&y=10

# 格式转换

# 质量变换

如果图片保存成jpg、webp, 可以支持质量变换。

## 参数

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| q | - 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<br>- 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。<br>- 当取值为lossless时，webp格式图片会按照无损格式保存 | 1-100,lossless |
| Q | - 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。<br>- 只能在保存格式为jpg/webp效果上使用，其他格式无效果。 如果一个转换url里，同时指定了q和Q，按Q来处理<br>- 当取值为lossless时，webp格式图片会按照无损格式保存 | 1-100,lossless |

## 注意事项

- 如果不填Q或者q 这两个参数，这样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。如果想按原图质量来保存，指定成100q

# 示例

- 将原图缩略成100w_100h，相对原图质量的80%的jpg图

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_80q

- 将原图缩略成100w_100h,绝对质量的80的jpg图

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_100h_80Q

- 将png原图缩略成200w,保存成无损的webp格式

http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-

process=image/resize,w_200/format,webp/quality,lossless

# 格式转换

可以将图片转换成对应格式(jpg, png, bmp, webp)

## 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黑色。如果想把透明填充成白色可以指定1wh参数 |
| 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。 |
| src | 按原图格式返回，如果原图是gif, 此时返回gif格式第一帧,保存成jpg格式，而非gif格式，如果我想保存成gif格式，必须增加1an参数 |

## 注意事项

- wh只有当原图是四通道(即有透明背景)的png, webp, bmp转换成jpg格式时才有效果。即把原图当中的透明背景以白色填充，如果不指定wh, 那么上述图转换成jpg时，透明背景将会变成黑色。

  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1pr, 详见渐进显示。

## 示例

将png保存成jpg格式

http://image-demo.img-cn-hangzhou.aliyuncs.com/panda.png@.jpg

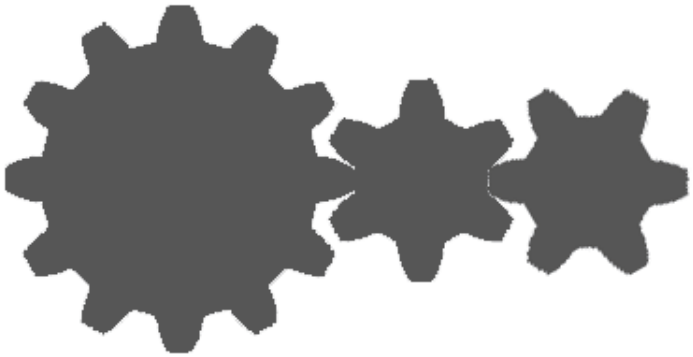
- 将png保存成jpg格式，透明的地方填充成白色

http://image-demo.img-cn-hangzhou.aliyuncs.com/panda.png@1wh.jpg

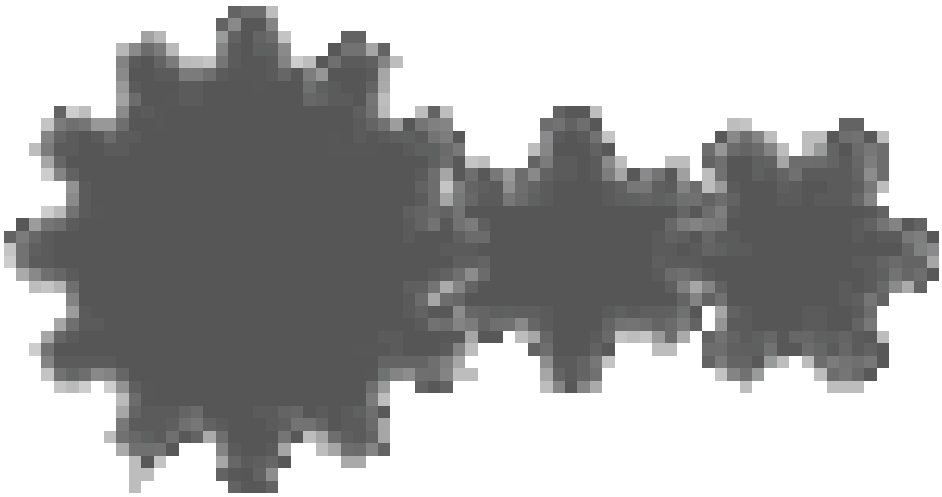
- 将jpg保存成高度为100, 宽度为100的png格式

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100h_100w.png

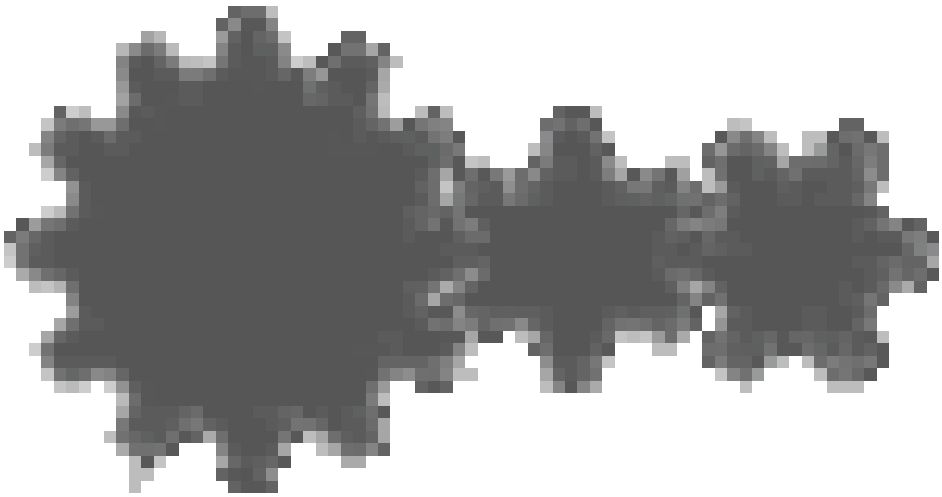
将一张gif图片保存后仍然是gif格式原图：https://gosspublic.alicdn.com/example.gif



缩略成宽度为100的缩略图:请求url:https://gosspublic.alicdn.com/example.gif@100w.gif



或者是：https://gosspublic.alicdn.com/example.gif@100w_1an.src



# 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
- 是先模糊后逐渐清晰（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

| 名称 | 描述 | 取值范围 |
| --- | --- | --- |
| pr | 1 表示保存成渐进显示的 jpg 格式<br>0 表示保存成普通的 jpg 格式 | [0, 1] |

注意：此参数只有当效果图是 jpg 格式时才有意义 。

## 示例

- 将图缩略成宽度 100，高度 100，并且保存成渐进显示的 jpg 格式

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@100w_200h_1pr.jpg

- 将 png 格式的图片保存成渐进显示的 jpg 格式

http://image-demo.img-cn-hangzhou.aliyuncs.com/panda.png@1pr

# 获取图片信息

# 获取基本信息

获取图片基本信息，是获取图片的宽度、高度和大小 http://image-demo.img.aliyuncs.com/example.jpg@info

## 请求格式

请求URL：@info

## 返回格式

```
{
    "height": 267,
    "size": 21839,
    "width": 400}
```

- height 表示图片的高度
- size 表示图片的大小
- width 表示图片的宽度

## 示例

http://image-demo.img.aliyuncs.com/f.jpg@exif

```
{
    "height": 267,
    "size": 21839,
    "width": 400}
```

# 获取exif信息

数码相机拍摄的照片文件中包含 exif 信息，用于记录数码照片的属性信息和拍摄数据。这些信息可以通过 @exif 来获取，返回格式是 Json 格式，目前支持返回的类型包括但不限于以下类型：

- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- DateTime
- DateTimeOriginal
- DateTimeDigitized
- Make
- Model
- Orientation

**注意：**

- 并非每一张图片都包含 exif 信息。如果原图没有 exif信息，当您请求 exif 信息时，会返回 400 错误。错误码为：**BadRequest**，错误内容是：**Image has no exif info**.
- 关于各参数的含义，请参见 Exif 标准。

# 示例

- 没有 exif 信息的图片示例

http://image-demo.img.aliyuncs.com/example.jpg@exif

返回信息

```
<Error>
<Code>BadRequest</Code>
<Message>Image has no exif info.</Message>
<RequestId>5502D98553F47BFAB7F95B8C</RequestId>
<HostId>image-demo.img.aliyuncs.com</HostId>
</Error>
```

- 包含 exif 信息的图片示例

http://image-demo.img.aliyuncs.com/f.jpg@exif

返回信息

```
{
"Compression": {"value": "6"},
"DateTime": {"value": "2015:02:11 15:38:27"},
```

```
"ExifTag": {"value": "2212"},
"FileSize": {"value": "23471"},
"GPSLatitude": {"value": "0deg "},
"GPSLatitudeRef": {"value": "North"},
"GPSLongitude": {"value": "0deg "},
"GPSLongitudeRef": {"value": "East"},
"GPSMapDatum": {"value": "WGS-84"},
"GPSTag": {"value": "4292"},
"GPSVersionID": {"value": "2 2 0 0"},
"ImageHeight": {"value": "333"},
"ImageWidth": {"value": "424"},
"JPEGInterchangeFormat": {"value": "4518"},
"JPEGInterchangeFormatLength": {"value": "3232"},
"Orientation": {"value": "7"},
"ResolutionUnit": {"value": "2"},
"Software": {"value": "Microsoft Windows Photo Viewer 6.1.7600.16385"},
"XResolution": {"value": "96/1"},
"YResolution": {"value": "96/1"}}
```


# 获取基本信息和exif信息

可以通过 @infoexif 来获取获取文件的基本信息包括宽度，长度，文件大小，格式。并且如果文件有exif信息，就返回exif信息，如果没有exif信息，就只返回基本信息。返回结果是json格式。

## 请求格式

请求URL + @infoexif

## 返回格式

json格式

## 该接口与exif接口的区别

本接口返回是exif信息的基本上增加 FileSize（文件大小），Format（图片格式），ImageHeight(图片高度), ImageWidth(图片宽度) 并且当图片没有exif信息时，只返回基本信息，当有图片有exif信息，将基本信息跟exif信息一起返回

## 示例

- 没有exif 的例子

http://image-demo.img.aliyuncs.com/example.jpg@infoexif

```
{
    "FileSize": {"value": "21839"},
    "Format": {"value": "jpg"},
    "ImageHeight": {"value": "267"},
    "ImageWidth": {"value": "400"}
}
```

- 有exif 的例子

http://image-demo.img.aliyuncs.com/f.jpg@infoexif

```
{
    "Compression": {"value": "6"},
    "DateTime": {"value": "2015:02:11 15:38:27"},
    "ExifTag": {"value": "2212"},
    "FileSize": {"value": "23471"},
    "Format": {"value": "jpg"},
    "GPSLatitude": {"value": "0deg "},
    "GPSLatitudeRef": {"value": "North"},
    "GPSLongitude": {"value": "0deg "},
    "GPSLongitudeRef": {"value": "East"},
    "GPSMapDatum": {"value": "WGS-84"},
    "GPSTag": {"value": "4292"},
    "GPSVersionID": {"value": "2 2 0 0"},
    "ImageHeight": {"value": "333"},
    "ImageWidth": {"value": "424"},
    "JPEGInterchangeFormat": {"value": "4518"},
    "JPEGInterchangeFormatLength": {"value": "3232"},
    "Orientation": {"value": "7"},
    "ResolutionUnit": {"value": "2"},
    "Software": {"value": "Microsoft Windows Photo Viewer 6.1.7600.16385"},
    "XResolution": {"value": "96/1"},
    "YResolution": {"value": "96/1"}}
```

# 获取图片主色调

本接口是为了获取图片的平均色调

## 请求格式

<image-url>@imageAve

# 返回格式

0xRRGGBB （RR GG BB都是十六进制表示红 绿 蓝三个颜色）

# 示例

- 可以在浏览器访问

http://image-demo.img.aliyuncs.com/example.jpg@imageAve

得到结果

```
{"RGB": "0x5c783b"}
```

原图是：

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg
-

0x5c783b对应的颜色RGB(92,120,59)是：

-

# 管道

# 管道

管道是一种可以实现多种处理任务顺序执行的机制。用户可以通过管道在一次访问中按照顺序完成对图像的不同处理。

## 访问规则

```
<图片URL>@<action1>|<action2>
```

URL 通过@符号后面处理参数（action1, action2）来实现即时云处理，如果有多任务（比如先做缩略，再加上水印）可以用管道来实现，执行顺序按管道指定顺序执行，目前最多支持四级管道。

**管道的分隔符是”|”**

上述表示先做对图片URL做处理action1 然后再在上述的基础上做处理action2，然后输出结果。上述action1，action2可以是简单缩略，文字水印，图片水印任意一种。

## 使用示例

- 先对图片做按高度300缩略，然后再加上文字水印，水印内容是：Hello 图片服务！

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@300h|watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ

这个例子由action1( 300h : 按高度是300缩略) 操作完再执行action2(watermark=2&text=SGVsbG8g5Zu-a54mH5pyN5YqhIQ: 文字水印，水印内容是：Hello 图片服务！)处理时间， 先对图片执行Action1操作，再执行Action2操作。

- 先对图片做文字水印，水印内容是: Hello, 图片服务！ 水印位置在右下角，然后再对图片做图片水印，水印object是:panda.png, 水印位置在中间。

http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&p=9|watermark=1&object=cGFuZGEucG5n&t=90&p=5

# 样式

# 样式访问

因为所有对图片的变换都会加在URL后面，会导致URL变得冗长，不方便管理与阅读。图片处理服务提供用户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，只要用一个很短的URL就能实现相同的效果。

一个频道（Channel）下面有多个样式，样式的作用范围只在一个频道（Channel）下，目前一个频道（Channel）允许最多有50个样式。

## 样式访问规则

```
<文件URL>@!StyleName
```

文件URL即是由Channel+Object 组成的URL地址，参见图片URL规则

1. @!是样式的分隔符，在URL带了这个分隔符，图片处理服务会把该分隔符后面的内容当成样式的名称。
2. StyleName表示的是样式的名字。
3. 创建样式、删除样式和修改样式都在前端控制台实现。
4. 当访问的样式在指定频道（Channel）不存在时，将返回NotSuchStyle错误。

## 示例

假如对image-demo这个Channel创建三个样式。

| 样式名 | 样式内容 |
| --- | --- |
| pipe1 | 150w_150h_1e_1c_0i_100q_1x.jpg\|watermark=1&object=cGFuZGEucG5n&t=51&p=9&x=10&y=10 |
| pipe2 | 250w_250h_0e_0c_0i_90q_1x.jpg\|150w_150h_0e_1c_1i_90q_1x.jpg |
| pipe3 | 180w_180h_1e_1c_0i_0o_90q_1x.jpg\|watermark=2&type=d3F5LXplbmhlaQ&size=25&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=IzAwMDAwMA&t= |

| | |
|---|---|
| | 90&p=9&x=10&y=10 |

**对于样式1(pipe1):**

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@150w_150h_1e_1c_0i_100q_1x.jpg|watermark=1&object=cGFuZGEucG5n&t=51&p=9&x=10&y=10

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@!pipe1

两者是一样的。

**对于样式2(pipe2)：**

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@250w_250h_0e_0c_0i_90q_1x.jpg|150w_150h_0e_1c_1i_90q_1x.jpg

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@!pipe2

效果是一样的

**对于样式3 (pipe3)：**

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@180w_180h_1e_1c_0i_0o_90q_1x.jpg|watermark=2&type=d3F5LXplbmhlaQ&size=25&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=IzAwMDAwMA&t=90&p=9&x=10&y=10

- http://image-demo.img-cn-hangzhou.aliyuncs.com/example.jpg@!pipe3

效果是一样的

# 样式相关操作

图片服务提供了一些 Style 相关的操作，用于增删查改样式（Style）。

## Put Style

在某个频道（Channel）下，创建一个样式（Style），将复杂的图片服务的处理参数,保存成一个样式,创建样式后,通过样式实现与参数同样的图片处理效果。

### 请求语法

```
PUT /?style&styleName=YourStyleName  HTTP/1.1
```

```
Host: ChannelName.img-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue

<?xml version="1.0" encoding="UTF-8"?>
<Style>
<Content>100w_200h.jpg</Content>
</Style>
```

# 请求参数(Request Parameters)

| 参数名称 | 说明 | 是否必须 | 取值范围 |
| --- | --- | --- | --- |
| styleName | 待创建的Style的名称 | 是 | 参见style命名规则 |

# 请求元素(Request Elements)

| 参数名称 | 说明 | 是否必须 | 取值范围 |
| --- | --- | --- | --- |
| Content | Style对应的内容，表示处理图像所用的参数 | 是 | 参见图像处理参数 |

# 请求示例

```
PUT /?style&styleName=style-example HTTP/1.1
Host: channel-example.img-cn-hangzhou.aliyuncs.com
Date: Thu, 08 Jan 2015 06:17:55 GMT
Authorization: OSS 2onpuorvhikxergnrzmwkn0t:Hyc0UH+CXKQv6ExbZMi+HPn4Gxc=

<?xml version="1.0" encoding="UTF-8"?>
<Style>
<Content>100w_200h.jpg</Content>
</Style>
```

# 返回示例

```
HTTP/1.1 200 OK
x-img-request-id: 54AE211379B222C77F000016
Date: Thu, 08 Jan 2015 06:17:55 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

# 细节分析

1. 如果Channel不存在，返回404 Not Found错误，错误码：NoSuchChannel。
2. 每个Channel下最多能创建50个Style。如果超过50个，Create Style时返回403 Forbidden错误，错误码：AccessDenied，错误消息为：Your style count is exceeded 50.
3. 只有Channel的拥有者才能在该Channel下创建Style。如果试图在一个不属于自己的Channel下创建Style，返回403 Forbidden错误，错误码：AccessDenied。
4. 如果待创建的Style已经存在，则会更新原有的Style 。

# List Style

List Style可以获取某个频道（Channel）下的所有样式（Style）的信息。

## 请求语法

```
GET /?style HTTP/1.1
Host: ChannelName.img-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

## 响应元素(Response Elements)

| 参数名称 | 说明 |
| --- | --- |
| Name | Style名称 |
| Content | Style对应的内容 |
| CreateTime | Style创建时间 |
| LastModifyTime | Style最后修改的时间 |

## 请求示例

```
GET /?style HTTP/1.1
Host: channel-example.img-cn-hangzhou.aliyuncs.com
Date: Thu, 08 Jan 2015 06:28:02 GMT
Authorization: OSS 2onpuorvhikxergnrzmwkn0t:zDI6c1trJAGHbR8rreyzq6lMq9U=
```

## 返回示例

```
HTTP/1.1 200 OK
x-img-request-id: 54AE237279B222C77F000023
Date: Thu, 08 Jan 2015 06:28:02 GMT
```

```
Connection: close
Content-Type : application/xml
Content-Length: 568
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<StyleList>
  <Style>
    <Name>style-example1</Name>
    <Content>400w</Content>
    <CreateTime> Thu, 08 Jan 2015 06:28:02 GMT </CreateTime>
    <LastModifyTime> Thu, 08 Jan 2015 06:28:02 GMT</LastModifyTime>
  </Style>
  <Style>
    <Name>style-example2</Name>
    <Content>400w</Content>
    <CreateTime>Thu, 08 Jan 2015 06:28:02 GMT</CreateTime>
    <LastModifyTime> Thu, 08 Jan 2015 06:28:02 GMT</LastModifyTime>
  </Style>
</StyleList>
```

## 细节分析

1. 如果频道（Channel）不存在，返回404 Not Found错误，错误码：NoSuchChannel。
2. 只有频道（Channel）的拥有者才能List该频道（Channel）下的样式（Style）。如果试图在一个不属于自己的频道（Channel）下List Style，返回403 Forbidden错误，错误码：AccessDenied。

# Get Style

Get Style可以获取某个样式（Style）的属性信息，包括样式名称、内容，以及创建和最后修改时间。

## 请求语法：

```
GET /?style&styleName=YourStyleName HTTP/1.1
Host: ChannelName.img-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

## 请求参数(Request Parameters)

| 参数名称 | 说明 | 是否必须 | 取值范围 |
| --- | --- | --- | --- |
| styleName | 需要获取的Style的名称 | 是 | 参见style命名规则 |

## 响应元素(Response Elements)

| 参数名称 | 说明 |
| --- | --- |
| Name | Style名称 |
| Content | Style对应的内容 |
| CreateTime | Style创建时间 |
| LastModifyTime | Style最后修改的时间 |

## 请求示例

```
GET /?style&styleName=style-example HTTP/1.1
Host: channel-example.img-cn-hangzhou.aliyuncs.com
Date: Thu, 08 Jan 2015 06:20:12 GMT
Authorization: OSS 2onpuorvhikxergnrzmwkn0t:SQe/ZdW92fmFgLEiIwsH4f8YTA8=
```

## 返回示例

```
HTTP/1.1 200 OK
x-img-request-id: 54AB9937B703C78879000167
Date: Thu, 08 Jan 2015 06:20:12 GMT
Connection: close
Content-Type : application/xml
Content-Length: 236
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<Style>
  <Name>jujht9w0d4</Name>
  <Content>erp2g2twla</Content>
  <CreateTime>Thu, 08 Jan 2015 06:20:12 GMT</CreateTime>
  <LastModifyTime>Thu, 08 Jan 2015 06:20:12 GMT</LastModifyTime>
</Style>
```

## 细节分析

1. 如果频道（Channel）不存在，返回404 Not Found错误，错误码：NoSuchChannel。
2. 如果样式（Style）不存在，返回404 Not Found错误，错误码：NoSuchStyle。

# Delete Style

Delete Style用来删除某个样式（Style）。

## 请求语法

```
DELETE /?style&styleName=YourStyleName HTTP/1.1
Host: ChannelName.img-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

## 请求参数(Request Parameters)

| 参数名称 | 说明 | 是否必须 | 取值范围 |
| --- | --- | --- | --- |
| styleName | 待删除的Style的名称 | 是 | 参见style命名规则 |

## 请求示例

```
DELETE /?style&styleName=style-example HTTP/1.1
Host: channel-example.img-cn-hangzhou.aliyuncs.com
Date: Thu, 08 Jan 2015 06:30:20 GMT
Authorization: OSS 2onpuorvhikxergnrzmwkn0t:mK217ZMjVP30w4Q99vYwBEgddqw=
```

## 返回示例

```
HTTP/1.1 204 No Content
x-img-request-id: 54AE23FC79B222C77F000028
Date: Thu, 08 Jan 2015 06:30:20 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

## 细节分析

1. 如果频道（Channel）不存在，返回404 Not Found错误，错误码：NoSuchChannel。
2. 不管样式（Style）存不存在，只要删除操作合法，删除成功后都会返回204 No Content。
3. 只有频道（Channel）的拥有者才能删除该频道下的样式（Style）。如果试图删除一个不属于自己的频道下的样式，返回403 Forbidden错误，错误码：AccessDenied。

# 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 |
| --- | --- | --- |
| 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。