对象存储

（Object-Oriented Storage,OOS）

图片处理用户手册

中国电信股份有限公司

云计算分公司

目录

1　产品介绍

OOS 为用户提供图片处理的功能，用户将原始图片上传并保存在 OOS 上，通过简单的 RESTful 接口，可以对图片进行处理并下载。

2　主要概念

2.1　参数说明

图片处理服务通过 URL 来处理图片，需要以下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。如果 URL 中的图片处理参数非法，会返回客户端原图。

分隔符名称	分隔符	含义
处理分隔符	@	区分 object 名称和处理字符串，后面加 oosImage，表示进行图片处理。
管道分隔符	|	区分多种操作，相关描述详见 3.3。

2.2　URL 构成规则

图片处理服务通过 URL 来处理图片，以下定义访问方式的规范。

1.　直接显示原图

形式为：

- http://endpoint/bucket/object
- http://bucket.endpoint/object
- http://website/object　，即支持 website 的方式访问对象

2.　通过处理参数访问

形式为：

- http://endpoint/bucket/object@oosImage|100w_100h_90q.jpg
- http://bucket.endpoint/object@oosImage|100w_100h_90q.jpg
- http://website/object@oosImage|100w_100h_90q.jpg，即支持 website 的方式访问对象

Object 为用户 Bucket 上存储的原图片。100w_100h_90q 为转换字符串，用来转换处理图片的一段参数。通过指定转换字符串，可以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、质量（q）90%、jpg 格式的图片。

@oosImage|120w_120h_90q.jpg

初始操作 转换参数 转换格式

转换字符串分为 3 部分：初始操作、转换参数、转换格式：

- 初始操作是一个“@”符号+“oosImage”+ “|”管道符号，后面都为转换字符串。
- 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。
- 转换格式是指定图片转换的输出格式，通过指定转换格式，可以对原图处理并返回指定的图片格式。支持的转换格式为： jpg、webp、png、bmp。

2.3 规则说明

- **顺序无关**

转换参数中键值对是与顺序无关的，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定义的顺序重新排序，然后再进行处理。（由于参数的顺序不同，有时会表达不同的语义。如“100w_100h_200p”表达的是“先缩放到 100*100，再放大 2 倍”，即得到 200*200 的图片；而"200p_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，OOS 会对参数按照文档中出现的顺序排序后处理。上例中的"200p_100w_100h"会被理解为“100w_100h_200p”，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”。

- **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”是指原尺寸与目标尺寸的比值小的那条边。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。

- **图片格式**
  1. 只能转换图片文件格式的对象，支持的原图片格式为：jpg、png、bmp、webp。
  2. 转换后的图片格式支持 jpg、png、bmp 和 webp。其中：
     - jpg、 png、 bmp、 webp 的图片可以保存为 jpg、png、bmp 和 webp 中的任意一种格式；
     - 图片如果不指定保存格式，默认保存为原格式。例如：格式为 png 的图片，默认保存为 png 格式的图片。
  3. 原图片的大小不能超过 20MB。
  4. 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。
  5. 一次转换请求最多支持 4 个管道处理。
- **容器属性限制**

  处理的图片所在 bucket 的**容器属性-访问权限**不可以设置为**私有**。

2.4 用户签名

图片处理服务的签名验证与 OOS 签名验证方法使用相同的验证逻辑。其中 CanonicalizedResource 由三部分构成：

- Bucket 名称；
- 对象名称+@oosImage；
- 转换字符串。

如：用户的 bucket 名称为 image-demo，object 名字为 example.jpg，转换字符串为：100w.jpg。在图片处理服务中，CanonicalizedResource 为 /image-demo/example.jpg@oosImage|100.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。

3 图片处理接口

3.1 图像处理接口

可以通过 HTTP GET 操作，对 object 进行图片处理，并下载处理后的图片。

3.1.1 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。

1. 参数

名称	描述	取值范围
w	指定目标缩略图的宽度。	整数形式，取值范围是：1~4096，单位是像素(px)。
h	指定目标缩略图的高度。	整数形式，取值范围是：1~4096，单位是像素(px)。
l	目标缩略图大于原图是否处理。	整数形式,取值范围是 0、1，默认是 0。 ● 1 表示不处理； ● 0 表示处理。

2. **注意事项**

- 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。

3. **示例**

- 将图缩略成高度为 100，宽度按比例处理。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h
- 将图缩略成宽度为 100，高度按比例处理。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|500w_1l

3.1.2 指定宽高缩略

可以对图片指定宽或高，按照长边或短边进行调整。

1. **参数**

名称	描述	取值范围
w	指定目标缩略图的宽度。	整数形式，取值范围是：1~4096，单位是像素(px)。
h	指定目标缩略图的高度。	整数形式，取值范围是：1~4096，单位是像素(px)。
e	缩放优先边， 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	取值范围是 0 和 1，默认值为 0。 ● 0 表示按长边优先；

名称	描述	取值范围
	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（e=0），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。	● 1 表示按短边优先。
1	目标缩略图大于原图是否处理。	取值范围是 0 和 1，默认值为 0。 ● 0 表示处理； ● 1 表示不处理。

2. 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。

3. 示例

- 将图缩略成宽度为 100，高度为 100，按长边优先
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w_0e
- 将图缩略成宽度为 100，高度为 100，按短边优先
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w_1e

3.1.3 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。

1. 参数

名称	描述	取值范围
w	指定目标缩略图的宽度。	整数形式，取值范围是：1~4096，单位是像素(px)。
h	指定目标缩略图的高度。	整数形式，取值范围是：1~4096，单位是像素(px)。
e	缩放优先边。如果是强制缩略，值是 2。	强制缩略，取值为 2。
l	目标缩略图大于原图是否处理。	整数形式，取值范围是 0 和 1，默认是 0。 ● 0 表示处理； ● 1 表示不处理。

2. 注意事项

如果压缩后的图与原图比例不一致，压缩后的图会变形。

3. 示例

将图强制缩略成宽度为 100，高度为 100：

http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w_2e

3.1.4 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。

1. 参数

名称	描述	取值范围
w	指定目标缩略图的宽度。	整数形式，取值范围是：1~4096，单位是像素(px)。
h	指定目标缩略图的高度。	整数形式，取值范围是：1~4096，单位是像素(px)。

名称	描述	取值范围
e	缩放优先边，这里指定按短边优化	整数形式，取值是 1。
c	是否对图形进行自动裁剪。	整数形式，取值是：0 和 1，默认值是 0。 ● 0 表示不对图进行自动裁剪； ● 1 表示对图进行自动裁剪。
l	如果目标缩略图大于原图是否处理。	整数形式，取值范围是：0 和 1，默认是 0。 ● 0 表示处理； ● 1 表示不处理。

2. **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。

3. **示例**

- 将图自动裁剪成宽度为 100，高度为 100 的效果图
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w_1e_1c
- 将图片按短边裁剪然后，裁剪出左半部分。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w_1e|0-0-100-100a

3.1.5 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。

1. 参数

名称	描述	取值范围
p	倍数百分比。	整数形式，取值范围是：1~1000。 ● 小于 100，表示缩小； ● 等于 100，表示不缩放； ● 大于 100，表示放大。

1.注意:

- 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 。

2. 示例

- 将图按比例放大两倍。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|200p
- 将图按比例缩略到原来的 1/2。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|50p

3.1.6 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。

1. 参数

名称	描述	取值范围
a	参数的类型：x-y-width-length 如：100-50-200-150a 一共四个参数，每个参数之间以“-”隔开。 ● 第一个参数表示起始点 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。 **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。	width, height 的值为整数形式，取值范围是：1~4096，单位是像素(px)。

2. 注意事项

- 如果指定的起始横纵坐标大于原图，将会返回原图。
- 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。

3. 使用示例

- 裁剪图从起点(100, 50)到图的结束。
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100-50-0-0a

- 裁剪图从起点(100, 50)到裁剪 100x100 的大小。
http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100-50-100-100a

3.1.7 质量变换

如果图片保存成 jpg, 可以支持质量变换。

1. **参数**

名称	描述	取值范围
q	决定 jpg 图片的 quality，对原图按照 q%进行 quality 压缩。	整数形式，取值范围是：1~100，默认值为 75。

2. **注意事项**

如果不填 q 这个参数，这样有可能会导致图片占用大小变大。

3. **示例**

将原图缩略成 100w_100h,图片质量 80%的 jpg 图：
http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100w_100h_80q

3.1.8 格式转换

可以将图片转换成对应格式（jpg, png, bmp, webp）。

1. **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。

2. **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)。

3. **示例**

- 将 png 保存成 jpg 格式：

  http://bucket.oos.ctyunapi.cn/panda.png@oosImage|.jpg
- 将 png 保存成 jpg 格式，透明的地方填充成白色：

  http://bucket.oos.ctyunapi.cn/panda.png@oosImage|1wh.jpg
- 将 jpg 保存成高度为 100，宽度为 100 的 png 格式：

  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100h_100w.png

3.1.9 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。

1. **示例**

- 没有 exif 的例子

http://bucket.oos.ctyunapi.cn /example.jpg@oosImage|infoexif


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


- 有 exif 的例子

http://bucket.oos.ctyunapi.cn/f.jpg@oosImage|infoexif


```
{
    "DateTime": {"value": "2015:02:11 15:38:27"},
    "FileSize": {"value": "23471"},
    "Format": {"value": "jpg"},
    "GPSLatitude": {"value": "0deg "},
    "GPSLatitudeRef": {"value": "North"},
    "GPSLongitude": {"value": "0deg "},
    "GPSLongitudeRef": {"value": "East"},
    "ImageHeight": {"value": "333"},
```

```
    "ImageWidth": {"value": "424"},
    "Orientation": {"value": "7"}
}
```

3.2 图片水印接口

可以通过 HTTP GET 操作，对 object 进行图片水印处理，并下载处理后的图片。

3.2.1 概要

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。

3.2.2 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	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。 整数形式,取值范围是:0~100。默认值:100，表示 100%（不透明）。	可选参数
p	水印打在图的位置，位置如区域数值对应表所示。 整数形式，取值范围是：1~9。默认值：9，表示在右下角打水印 。	可选参数
x	水印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。 整数形式，取值范围是：0~4096。默认值：10 。单位是像素（px）。	可选参数
y	水印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。 整数形式，取值范围是：0~4096。默认值：10。单位是像素(px)。	可选参数
voffset	水印中线垂直偏移，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。	可选参数

名称	描述	参数类型
	整数形式，取值范围是：-1000~1000，默认值：0 。单位是像素(px)。	

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

1. **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。

2. **使用示例**

- 右下角打上文字水印
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=2&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&t=50&p=6&x=10&voffset=20

3.2.3 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

1. **访问类型**

```
@oosImage|watermark=1&bucket=bucketName&object=<encodedobject>&t=<transparency>&x=<distanceX>&y=<distanceY>&p=<position>…
```

其中 watermark 与 object 两个参数为必填项。

2. **参数**

名称	描述	参数类型
object	水印图片的 object 名字(必须编码)。 **注意** 内容必须是url 安全Base64编码 EncodedObject = url_safe_base64_encode(object) 如object 为"panda.png"，编码过后的内容就是"cGFuZGEucG5n"。	必选参数
bucket	水印图片所在的 bucket，必须是和要加图片水印的 object 是同一个用户，否则返回 403 Access Denied。	必选参数

3. **水印图片预处理**

水印图片支持预处理，目前支持的处理有调节亮度(b)和对比度(d)，按比例缩略(p)，指定宽度缩略(w,h) 这几个参数，其他参数暂未支持。

参数	描述	取值范围
p	对当前水印图片进行按比例缩略或放大。	整数形式，取值范围是：1~1000。如 10p 表示基于水印图片的 10%进行处理。
P	水印图片按主图的比例进行处理。	整数形式,取值范围是:1~100。如果设置了 10P，主图是 100x100，那么水印图片此时的大小就是 10x10，当主图变成了 200x200，那么水印图片就变成了 20x20。
w	按宽度缩略。指定水印图的宽度，	整数形式，取值范围是：

参数	描述	取值范围
	高度按比较缩略。	1~4096。单位是像素(px)。
h	按高度缩略。指定水印图的高度，宽度按比较缩略。	整数形式，取值范围是：1~4096。单位是像素(px)。

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@oosImage|连接。如：

- 对 panda.png 进行放大 2 倍: Object = url_safe_base64_encode(“panda.png@oosImage|200p”)

所以尽量不要让原 object 名字是带@。不然可能会导致访问异常。如不要让原始 object 名字是”panda@123.png”。

4. 使用示例

- 原图 example.jpg 加上水印图片是 panda.png，右下角，水平边距为 10， 垂直边距为 10, 透明度为 90:
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=1&bucket=bucketName&object=cGFuZGEucG5n&t=90&p=9&x=10&y=10
- 原图 example.jpg 加上水印图片是 panda.png。按宽度缩略成 100, 水印大小是主图的 40%, 此时水印图的是 40x40:
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|100w.jpg|watermark=1&bucket=bucketName&object=cGFuZGEucG5nQDQwUA&t=90&p=9&x=10&y=10
- 原图 example.jpg 加上水印图片是 panda.png。按宽度缩略成 200，水印大小是主图的 40%, 此时水印图是 80x80:
  http://bucket.oos.ctyunapi.cn/example.jpg@oosImage|200w.jpg|watermark=1&bucket=bucketName&object=cGFuZGEucG5nQDQwUA&t=90&p=9&x=10&y=10

3.2.4 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作为水印。

1. **访问类型**

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

其中 watermark 与 object 两个参数为必填项。

2. **参数**

名称	描述	参数类型
text	文字水印的文字内容 EncodefontText = 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	文字水印的文字类型(必须编码) 。 **注意** 必须是 Base64 编码 EncodeFontType = url_safe_base64_encode (fontType)。 取值范围见“文字类型编码对应表”。 默认值：wqy-zenhei (编码后的值：d3F5LXplbmhlaQ)。	可选参数
color	文字水印文字的颜色(必须编码) 。 **注意** 参数必须是 Base64 位编码 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。 默认值：#000000 黑色，base64 编码后值：IzAwMDAwMA。	可选参数
size	文字水印文字大小 。整数形式，取值范围是：(0，1000] ，默认值是：40。单位是像素(px)。	可选参数
s	文字水印的阴影透明度。整数形式，取值范围是：(0,100]。	可选参数

文字类型编码对应表

参数值	中文意思	编码后的值
wqy-zenhei	文泉驿正黑	d3F5LXplbmhlaQ

参数值	中文意思	编码后的值
wqy-microhei	文泉微米黑	d3F5LW1pY3JvaGVp
fangzhengshusong	方正书宋	ZmFuZ3poZW5nc2h1c29uZw
fangzhengkaiti	方正楷体	ZmFuZ3poZW5na2FpdGk
fangzhengheiti	方正黑体	ZmFuZ3poZW5naGVpdGk
fangzhengfangsong	方正仿宋	ZmFuZ3poZW5nZmFuZ3Nvbmc
droidsansfallback	DroidSansFallback	ZHJvaWRzYW5zZmFsbGJhY2s

3. **使用示例**

- 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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ

3.2.5 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。

1. **访问类型**

```
@oosImage|watermark=3&bucket=bucketName&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。

2. **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	3.2.6 描述	参数类型
order	文字，图片水印前后顺序。 取值范围是 0 和 1，默认值为 0。 ● order = 0 ，图片在前； ● order = 1 文字在前。	可选参数
align	文字、图片对齐方式 。 取值范围是 0、 1 和 2。默认值为 0。 ● align = 0 表示上对齐； ● align = 1 表示中对齐 ； ● align = 2 表示下对齐。	可选参数
interval	文字和图片间的间距。 整数形式，取值范围是：0~1000，单位是 px。	可选参数

3. 使用示例

- 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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=1&bucket=bucketName&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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=3&bucket=bucketName&object=cGFuZGEucG5n&type=d3F5LXplbmhlaQ&size=40&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&s=50&order=0&align=1&interval=10&t=90&p=9&x=10&y=10

3.3 管道

1. 访问规则

```
<文件 URL>@oosImage|<action1>|<action2>
```

URL 通过@oosImage|后面的处理参数（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”|”

上述表示先做对文件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。

2. 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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|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://bucket.oos.ctyunapi.cn/example.jpg@oosImage|watermark=2&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&p=9|watermark=1&object=cGFuZGEucG5n&t=90&p=5

3.4 图片拼接成GIF接口

可以将多个图片拼接成gif文件，图片是原始图片，可以针对每个图片添加@oosImage参数,对图片进行单独的处理。原始图片格式支持jpg, png, bmp, webp。如果各个图片的大小不一致，那么以最小的宽高为标准，其他图片的宽高根据此进行缩放。

参数如下：

<objectURL>@oosImage|mergegif&delayTime=xx&loop=0&object=<encodeObject>&object=<encodeObject>&object=<encodeObject>...

参数说明：

参数名	描述	取值范围
object	object 和 objectURL 中指定的对象，需在同一个 bucket 中。object 是 url 安全 Base64 编码，EncodedObject = url_safe_base64_encode(objectName@oosImage|图片处理参数)。最多支持同时合并 20 个图片对象。例如：object=base64urlecnode(a.jpg@oosImage|20w)	无
objectURL	objectURL 是第一个要拼接图片的地址，如果要对第一个图片做处理，需要在 mergegif 参数前面增加管道参数。如果在 mergegif 后面加管道参数，说明是对拼接后的 gif 进行处理	无
delayTime	gif 文件中每帧的延迟时间。	整数形式，取值之范围是：0~5000。单位是毫秒。
loop	是否循环播放。	整数形式，取值范围是：0 和 1，默认值是 0。 ● 0 表示不循环； ● 1 表示是循环。

使用示例：

将名为 image 的 bucket 中的对象：test1.jpg, test2.png, test3.bmp 合并成 gif 图片。其中为 test1.jpg 设置宽 100，高 100；test2.png 也设置宽 100。url 如下：

http://oos.ctyunapi.cn/image/test1.jpg@oosImage|100h_100w|mergegif&object=base64urlecnode(test2.png@oosImage|100w)&object=base64urlecnode(test3.bmp)&delayTime=10

即

http://oos.ctyunapi.cn/bucket/test1.jpg@oosImage|100h_100w|mergegif&object=dGVzdDIucG5nQG9vc0ltYWdlfDEwMHc&object=dGVzdDMuYm1w&delayTime=10

4 示例

4.1 示例链接

原图 宽 400 高 300

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg

1. 获取单边固定宽度的缩略图片，宽度为 100，高度按比例处理

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|100h

2. 获取单边固定宽度的缩略图片，强制缩略成宽度为 100，高度为 100

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|100h_100w_2e

3. 按比例缩放图片，将图按比例缩小 1/2

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|50p

4. 格式转换，将 jpg 保存成高度为 100, 宽度为 100 的 png 格式

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|100h_100w.png

5. 获取图片的基本信息和 exif 信息

http://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|infoexif

6. 获取文字水印图片，字体是文泉驿正黑，字体大小是 20, 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://oos-hz.ctyunapi.cn/imagedemo/example.jpg@oosImage|watermark=2&type=d3F5LXplbmhlaQ&size=20&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&s=50&t=90&p=6&x=10&voffset=20

7. 图片拼接成 gif

将 test1.jpg, test2.jpg, test3.jpg 三张图片拼接成 gif 图片。其中为 test1.jpg 设置宽 100，高 100，test2.jpg 设置为宽 100， test3.jpg 也设置为宽 100.

http://oos-hz.ctyunapi.cn/imagedemo/test1.jpg@oosImage|100h_100w|mergegif&object=dGVzdDIuanBnQG9vc0ltYWdlfDEwMHc&object=dGVzdDMuanBnQG9vc0ltYWdlfDEwMHc&delayTime=1000

4.2 示例代码

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import java.util.TimeZone;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;

public class ImageSample {
private static String objectName = "your_object";
private static String ak = "your_ak";
private static String sk = "your_sk";
private static final int CONN_TIMEOUT = 10000;
private static final int READ_TIMEOUT = 30000;
private static final String DATE_STR = "EEE, d MMM yyyyHH:mm:ss 'GMT'";
private static String bucket = "your_bucket";
private static String host = "oos-hz.ctyunapi.cn";
private static final SimpleDateFormat DATE_FMT = newSimpleDateFormat(DATE_STR,
Locale.ENGLISH);
```

```
static {
    TimeZonegmt = TimeZone.getTimeZone("GMT");
    DATE_FMT.setTimeZone(gmt);
    }

public static void main(String[] args) throws Exception {
    ImageSample is = new ImageSample();
// 获取单边固定宽度的缩略图片，宽度为100，高度按比例处理
    is.getObject(bucket, objectName, "@oosImage|100h", new
    File("single_side_width.jpg"));
    // 获取单边固定宽度的缩略图片，强制缩略成宽度为100，高度为100
    is.getObject(bucket, objectName, "@oosImage|100h_100w_2e",
    new File("force_width_height.jpg"));
    // 按比例缩放图片，将图按比例缩小1/2
    is.getObject(bucket, objectName, "@oosImage|50p", new File("reduce.jpg"));
    //格式转换，将jpg保存成高度为100，宽度为100的png格式
    is.getObject(bucket, objectName, "@oosImage|100h_100w.png",
    new File("format_conversion.png"));
    // 获取图片的基本信息和exif信息
    is.getObject(bucket, objectName, "@oosImage|infoexif", new
File("info.txt"));
// 获取文字水印图片，字体是文泉驿正黑，字体大小是20，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
    is.getObject(bucket, objectName,
    "@oosImage|watermark=2&type=d3F5LXplbmhlaQ&size=20&text=SGVsbG8g5Zu-54mH5pyN5YqhIQ&color=I0ZGRkZGRg&s=50&t=90&p=6&x=10&voffset=20", new
File("wartermark.jpg"));
}

private String authorize(String httpVerb, String date, String bucket, String
objectName) throws Exception {
        String stringToSign = httpVerb + "\n\n\n" + date + "\n/" + bucket + "/"
+ objectName;
        Mac mac = Mac.getInstance("HmacSHA1");
mac.init(newSecretKeySpec(sk.getBytes("UTF-8"), "HmacSHA1"));
byte[] macResult = mac.doFinal(stringToSign.getBytes("UTF-8"));
        String signature = newString(Base64.encodeBase64(macResult), "UTF-8");
        String authorization = "AWS " + ak + ":" + signature;
        return authorization;
    }
```

```
public static void downloadInputStream(InputStreaminput, File file) throws
IOException {
 int c;
 FileOutputStream fos = newFileOutputStream(file);
 byte[] buff = newbyte[1024 * 4];
 try {
     while ((c = input.read(buff)) != -1) {
       fos.write(buff, 0, c);
       }
     input.close();
     fos.flush();
 } finally {
    fos.close();
        }
}

public void getObject(String bucketName, String objectName, String
imageParams, File destFile) throws Exception {
        String date = DATE_FMT.format(new Date());
        String authorization = authorize("GET", date, bucket, objectName +
imageParams);
        URL url = newURL("http", host, 80, "/" + bucket + "/" + objectName +
imageParams);
HttpURLConnectionconn = (HttpURLConnection) url.openConnection();
conn.setUseCaches(false);
conn.setRequestProperty("Date", date);
conn.setRequestProperty("Authorization", authorization);
conn.setConnectTimeout(CONN_TIMEOUT);
conn.setReadTimeout(READ_TIMEOUT);
conn.setDoInput(true);
conn.setRequestMethod("GET");
conn.connect();
downloadInputStream(conn.getInputStream(), destFile);
    }
}
```