

对象存储服务

图片处理特性指南（巴黎区域）

文档版本 01
发布日期 2020-03-31



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 简介	1
1.1 什么是图片处理?	1
1.2 应用场景.....	1
1.3 约束条件.....	2
1.4 常用概念.....	2
1.5 访问图片处理的方式.....	3
1.6 图片处理功能概述.....	3
2 开始图片处理（控制台）	5
2.1 使用流程.....	5
2.2 上传图片.....	6
2.3 创建图片样式.....	6
2.4 使用图片样式.....	7
3 开始图片处理（接口）	9
3.1 使用流程.....	9
3.2 上传图片.....	9
3.3 处理图片.....	10
4 典型案例	12
4.1 控制台图形界面模式.....	12
4.2 控制台代码编辑模式.....	13
5 获取图片信息	15
6 获取图片平均色值	16
7 设置图片效果	18
7.1 亮度.....	18
7.2 对比度.....	20
7.3 锐化.....	21
7.4 模糊.....	22
8 设置缩略	24
9 旋转图片	29
9.1 旋转设置.....	29
9.2 自适应方向.....	30

10 剪切图片	31
10.1 普通剪切	31
10.2 内切圆	33
10.3 索引剪切	34
10.4 圆角剪切	36
11 设置水印	39
11.1 公共参数	39
11.2 图片水印	41
11.3 文字水印	45
12 转化格式	51
12.1 格式转换	51
12.2 渐进显示	52
13 质量变换	54
14 设置图片瘦身	56
15 处理图片持久化	57
16 常见问题	60
16.1 什么是图片处理特性？	60
16.2 如何访问图片处理？	60
16.3 每个桶支持创建多少个图片样式？	60
16.4 图片处理支持哪些格式的图片？	60
16.5 如何通过 URL 访问图片处理服务？	61
A 修订记录	62

1 简介

1.1 什么是图片处理？

简介

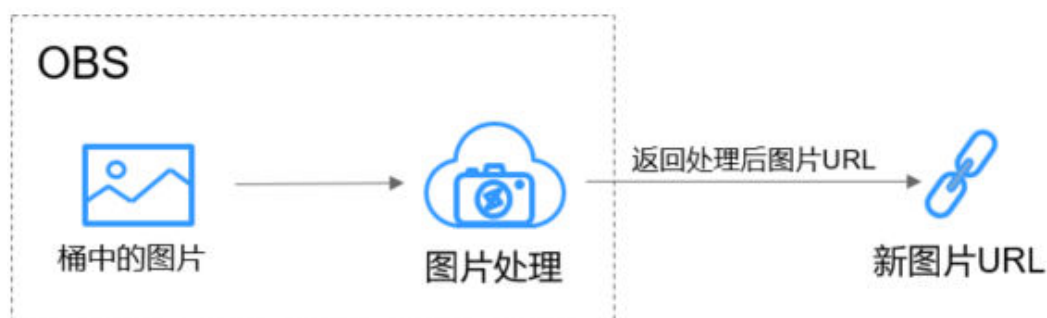
图片处理特性（Image Processing）是对象存储服务（Object Storage Service，OBS）为用户提供稳定、安全、高效、易用、低成本的图片处理服务，包括：图片瘦身、图片剪切、图片缩放、图片水印、格式转换等。

图片处理特性提供OBS管理控制台和RESTful API接口的使用方法，使用户可以在任何时间和地点对存储在对象存储中的图片进行处理，并且可以快速获取到处理后的图片。

架构

用户通过管理控制台、OBS客户端、RESTful API接口或者第三方客户端的方式将图片上传至OBS中进行存储。用户下载使用图片前，可以创建图片样式模板或传入图片处理参数对图片进行处理，例如剪切、压缩，并且可以获取到使用图片样式或经过参数处理的新图片URL。图片处理特性架构如图1-1所示。

图 1-1 图片处理架构



1.2 应用场景

图片处理支持用户在云端对图片进行简单的缩略、剪切、压缩等操作，不需下载占用空间巨大的本地图片处理软件即可直接在线对图片进行处理。

例如，可以随时随地在线对在云相册中的图片进行效果、缩略设置后，即刻在线将图片分享给好友、同事等。

1.3 约束条件

操作限制

- 所有的图片处理操作均不会修改原图。
- 冷存储不支持图片处理。
- 使用SSE-KMS服务端加密的OBS桶不能用做图片处理。
- 当前只有桶版本号为3.0的桶支持图片处理。桶版本号可以在对象存储服务控制台上，进入桶概览页后，在“基本信息”中查看。
- 只有图片设置为匿名可访问，图片处理请求无需携带签名通过浏览器即可访问。例如：

```
https://bucketname.obs.region.example.com/example.jpg?x-image-process=style/stylename
```

图片限制

- 支持的原图格式：jpg、jpeg、png、bmp、webp、gif、tiff。
- 支持的格式处理后的目标图格式：jpg、png、bmp、webp。
- 支持处理前的图片最大为25MB，且处理后的图片宽度限制为4096px，高度限制为5000px。
- 大于2M或者超过50帧的gif或webp等动态图片，直接返回原图，不作处理。
- 目前对CMYK颜色模式的图片进行处理可能出现颜色的变化。

命令限制

命令长度不超过512个字符，操作命令总数不超过20个。

1.4 常用概念

样式

样式是将图片处理对图片设置的参数或界面操作设置保存成的一个集合。在对多张图片进行重复相同设置时，可以先创建一个图片样式，在不同的图片套用图片样式，就不需要进行重复的命令操作。每个桶最多支持创建100个样式。

Exif 信息

Exif (Exchangeable Image File) 是“可交换图像文件”的缩写，在一些相机或手机拍摄的图片中会含有Exif信息。Exif信息是嵌在JPEG或TIFF格式的图片内的一组参数，主要包括图片在拍摄时的参数如相机型号、拍摄时间、拍摄模式等，还有缩略图和其他属性信息。

1.5 访问图片处理的方式

您可以使用两种方式访问图片处理。

- 通过管理控制台查看OBS中图片使用相应图片样式模板的图片预览效果。
用户可以在管理控制台图片处理界面通过按钮或编辑代码操作创建图片样式模板，在图片预览区域预览图片使用相应样式模板的效果，通过复制链接按钮获取使用样式后的图片URL进行使用。
控制台访问方式快速入门请参考[开始图片处理（控制台）](#)。
- 通过应用程序调用简单的RESTful API接口来访问图片处理服务。
OBS提供了REST接口。REST从资源的角度来观察整个网络，分布在各处的资源由URI（Uniform Resource Identifier）确定，而客户端的应用通过URL（Unified Resource Locator）来获取资源。URL的一般格式为：`https://Endpoint/uri`。用户只需在浏览器地址栏中输入符合图片处理命令规则的URL，即可获取到处理后的图片。更多接口访问信息请参见《对象存储服务API参考》。
API接口访问方式快速入门请参考[开始图片处理（接口）](#)。

1.6 图片处理功能概述

图片处理为存储在OBS上的图片提供如[表1-1](#)所示功能。

表 1-1 图片处理功能一览表

功能	说明	使用方式
获取图片信息	获取图片的基本信息，包括：图片格式、大小、平均色值等。	接口调用
设置图片效果	支持对图片进行基础效果的设置，包括设置图片亮度、对比度，对图片进行整体锐化及模糊等处理。	控制台图形界面 控制台代码编辑 接口调用
图片缩略	支持对图片按照指定宽高和比例进行缩放。	控制台代码编辑 接口调用
设置图片水印	支持在图片指定位置添加文字、图片、图文混合的水印，并支持设置文字的颜色、字体和大小，对水印图片进行缩略、旋转和剪切等预处理。	控制台图形界面 控制台代码编辑 接口调用
转换图片格式	支持将图片转换为支持的图片格式，并支持对转换格式后的图片设置渐进显示。	控制台图形界面 控制台代码编辑 接口调用
图片旋转	支持将图片按照顺时针方向进行旋转，以及根据相机或手机的旋转信息进行自动旋转。	控制台代码编辑 接口调用

功能	说明	使用方式
图片剪切	支持将图片按照指定宽高、内切圆半径、索引方式和圆角矩形大小进行剪切。	控制台代码编辑 接口调用
图片压缩	支持按照相对质量和绝对质量对jpg格式的图片进行压缩，压缩后的图片画质降低，占用的空间变小，在对图片质量要求不高的场景下可以节省流量、减少加载时间。	控制台代码编辑 接口调用
图片瘦身	支持在图片画质基本不变的情况下对图片进行瘦身，减小图片占用的空间，此功能可以在不影响图片质量的情况，有助于节省获取图片导致的流量开销，提升终端加载速度。	控制台代码编辑 接口调用
图片处理持久化	支持将处理后的图片异步存储至指定的OBS桶中，方便下次直接访问处理后的图片，减少图片处理花费的时间，提升用户体验。	控制台代码编辑 接口调用
按顺序对图片进行多种处理	支持在图片URL后输入多组图片处理命令，并通过指定的分割符连接，最终按照从左至右的顺序依次对图片进行处理。	控制台代码编辑 接口调用
自定义图片样式	支持通过创建图片样式对图片进行集中处理。对于需要相同处理操作的多张图片，可以通过创建样式避免重复的设置操作。	控制台图形界面 控制台代码编辑

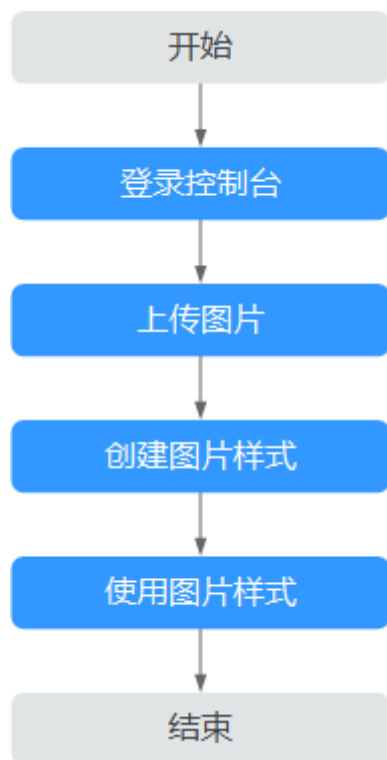
2 开始图片处理（控制台）

2.1 使用流程

用户可以在OBS管理控制台图片处理界面通过按钮或编辑代码操作创建图片样式，在图片预览区域预览图片使用相应样式模板的效果，通过复制链接按钮获取使用样式后的图片URL进行使用。

本章简要展示使用控制台方式访问和使用图片处理的流程，具体如[图2-1](#)所示。

图 2-1 控制台场景使用流程



2.2 上传图片

您可以通过OBS管理控制台、OBS客户端、RESTful API方式上传图片。

本章节介绍如何使用OBS管理控制台上传图片。如果您的桶中已经存在需要处理的图片，请跳过此章节。

📖 说明

关于上传图片的格式和大小的约束与限制请参见[约束条件](#)。

前提条件

至少已创建了一个桶。

操作步骤

步骤1 在管理控制台首页，选择“服务列表 > 存储 > 对象存储服务”。

步骤2 在桶列表中单击待操作的桶，进入“对象”页面。

步骤3 单击“上传对象”，进入“上传对象”页面。

步骤4 选择待上传的图片，单击“打开”。

步骤5 单击“上传”。

----结束

2.3 创建图片样式

背景信息

可以通过创建图片样式对图片进行剪切、压缩、添加水印等处理。对于需要相同处理操作的多张图片，可以通过创建样式避免重复的设置操作。创建成功的样式可以被桶中的多张图片使用。

在创建样式过程中，可以在右侧的样例图片中实时看到设置的样式效果。

使用RESTful API接口访问时，可以在URL中直接调用样式名称对图片进行处理，避免输入复杂的处理命令。接口访问的域名规则请参见[样式访问方式](#)。

每个桶您最多可以创建100个样式。

您可以新建图片处理样式，也可以从其他桶复制已有图片处理样式。

新建图片处理样式

步骤1 在管理控制台首页，选择“服务列表 > 存储 > 对象存储服务”。

步骤2 单击目标桶名称，在左侧导航栏中选择“图片处理”。

步骤3 单击“创建”，进入样式编辑界面。

步骤4 在编辑页面，可以进行编辑样式名称和基本属性，设置缩略方式、旋转/剪切、图片水印、图片输出等操作。

- 样式名称

输入方便记忆的样式名称，只能由大写或小写的英文字母、数字、英文句号（.）、下划线（_）和连字符（-）组成。长度为1~256个字符。例如：
rotate_0001。

- 编辑模式

可以选择图形界面模式进行可视化样式编辑，也可以选择代码编辑模式进行样式编辑。

代码编辑样例如下：

```
image/sharpen,100/blur,r_1,s_1/resize,m_lfit,h_400,w_400,limit_1
```

- 参数设置

根据需要设置图片效果、缩略、水印和输出的参数值。

步骤5 完成样式的编辑后，单击“确定”保存样式。新建成功的样式将会显示在样式列表中。

----结束

2.4 使用图片样式

当桶中存在已经创建好的图片样式时，您可以通过以下两种方法使用图片样式。

- **复制链接**：在OBS控制台上通过图片对象详情页面的图片预览，获取访问链接，在浏览器中打开即可获得到使用图片样式处理后的图片。其具体操作步骤如[复制链接](#)所示。

- **拼写域名**：参照以下域名拼写规则，将拼写的域名在浏览器中打开以快速获取使用图片样式处理后的图片。

```
<图片链接>?x-image-process=style/<样式名称>
```

图片链接从对象详情页面获取，具体操作步骤请参见[拼写域名](#)。样式名称为创建样式时用户自定义的样式名，只能由大写或小写的英文字母、数字、英文句号（.）、下划线（_）和连字符（-）组成。长度为1~256个字符。例如：
rotate_0001。

复制链接

步骤1 在管理控制台首页，选择“服务列表 > 存储 > 对象存储服务”。

步骤2 选中样式所在桶，进入“对象”页面。


步骤3 单击已存在的图片文件名称或者上传新图片后，进入文件详情页面。

步骤4 单击“图片预览”页签，您可以在此页面预览使用图片样式的图片效果。

步骤5 单击“复制链接”，按钮提示“复制成功”时，即可获得到使用样式的图片文件访问地址在浏览器中使用。

----结束

拼写域名

- 步骤1** 在OBS对象列表页面，单击待处理图片名称，进入图片详情页面。
- 步骤2** 在“对象ACL > 公共访问权限 > 匿名用户”中，单击“编辑”为匿名用户授予对象读取权限后，单击“保存”。
- 步骤3** 单击“链接”后的  图标，复制图片链接。
- 步骤4** 在复制成功的链接后加上?x-image-process=style/<样式名称>，即可在浏览器中打开并获取到使用样式处理后的图片。

示例：

`https://bucketname.obs.region.example.com/example.jpg?x-image-process=style/
stylename`

说明

以上图片链接以及样式名称均为示例，使用过程中请根据实际情况修改。

----结束

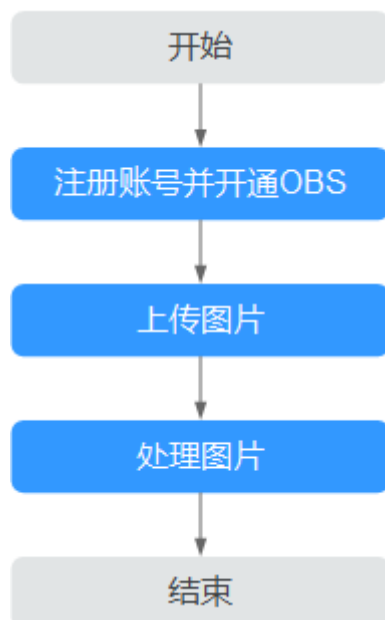
3 开始图片处理（接口）

3.1 使用流程

OBS提供了REST接口。REST从资源的角度来观察整个网络，分布在各处的资源由URI（Uniform Resource Identifier）确定，而客户端的应用通过URL（Unified Resource Locator）来获取资源。URL的一般格式为：`https://Endpoint/uri`。用户只需在浏览器地址栏中输入符合图片处理命令规则的URL，即可获取到处理后的图片。

本章简要展示使用RESTful API接口方式访问和使用图片处理的流程，具体如[图3-1](#)所示。

图 3-1 接口场景使用流程



3.2 上传图片

您可以使用OBS管理控制台、OBS客户端和RESTful API方式上传图片。

如果您的桶中已经存在需要处理的图片，请跳过此章节。

说明

如果您需要通过自定义域名使用图片处理，需要为待处理的图片配置匿名用户读取权限。配置请参考《权限配置指南》中配置对象ACL进行配置。

3.3 处理图片

此章节介绍通过RESTful API方式进行图片处理操作的URL构成。当已成功开通OBS后，在浏览器地址中输入符合规则的URL即可通过RESTful API方式使用图片处理。

URL 组成形式

URL由OBS域名、OBS上的桶名、原图片名称、处理命令或样式名组成。

命令访问方式

URL形式为：`https://bucketName.endpoint/objectName?x-image-process=image/commands`

- *endpoint*为桶所在区域对应的终端节点地址，可以在桶的基本信息处获取。有关OBS区域和终端节点的更多信息，请参考[地区和终端节点](#)。
- *bucketName*为OBS中待处理图片存放的桶名。
- *objectName*为存储在OBS上*bucketName*桶下的原图片名称，后缀名必须为图片处理支持的图片格式。
- *commands*为处理命令。处理命令之间和命令参数之间会使用三种分隔符，参见[分隔符](#)。如果不输入处理命令，将返回原图。

例如：`https://image-demo.obs.region.example.com/example.jpg?x-image-process=image/crop,x_100,y_50`

- **分隔符**

分隔符是指URL中用来区分命令访问方式一些关键字段的分隔符号。具体说明如[表3-1](#)所示。

表 3-1 分隔符

名称	样式	顺序	说明
参数分隔符	_	固定顺序	命令参数与参数值之间的分隔符。
命令分隔符	,	无关	多项处理命令参数之间的分隔符。
管道分隔符	/	有关	两组处理命令之间分隔符，参见 管道 。

- **管道**

如果一个图片，需要进行多重处理，如先剪切、后缩略等，则多组操作命令之间，需要以管道分隔符“/”进行连接，执行顺序按管道指定顺序从左至右执行。

例如：`https://image-demo.obs.region.example.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80`为两个管道，执行顺序为从左到右，左侧命令处理结果将会作为右侧命令的输入。

样式访问方式

URL形式为：`https://bucketName.endpoint/objectName?x-image-process=style/styleName`

- *endpoint*为桶所在区域对应的终端节点地址，可以在桶的基本信息处获取。有关OBS区域和终端节点的更多信息，请参考[地区和终端节点](#)。
- *bucketName*为OBS上的桶名。
- *objectName*为存储在OBS上*bucketName*桶下的原图片名称，后缀名必须为图片处理支持的图片格式。
- *styleName*为在控制台*bucketName*桶下创建好的样式名称。RESTful API方式现暂不支持其他样式相关等操作，如创建样式、更改样式和删除样式。



样例：`https://image-demo.obs.region.example.com/example.jpg?x-image-process=style/styleName`

4 典型案例

4.1 控制台图形界面模式

本章节提供控制台图形界面模式创建一个在图片左上角的位置添加方正书宋的文字水印的样式并进行使用的操作案例。

操作步骤

- 步骤1** 在管理控制台首页，选择“服务列表 > 存储 > 对象存储服务”。
- 步骤2** 在桶列表中单击待操作的桶，进入“对象”页面。
- 步骤3** 单击“上传对象”，进入“上传对象”页面。
- 步骤4** 选择待上传的图片，单击“打开”。
- 步骤5** 单击“上传”，开始上传图片。上传完成后的图片会显示在对象列表。
- 步骤6** 在左侧导航栏，选择“图片处理”。
- 步骤7** 单击“创建”，进入样式编辑界面。
- 步骤8** 输入样式名称为“obs-111”。在编辑页面，选中“图形界面模式”。
- 步骤9** 单击“水印设置”页签。
 - 在水印类型下拉列表中选择“文字水印”。
 - 在文字内容输入框中输入“Hello”。
 - 文字字体选择“方正书宋”。
 - 文字大小输入“600”。
 - 水印透明度保持默认值“100”。
- 水印位置选择左上。
- 文字阴影保持不开启。
- 垂直边距和水平边距保持默认值“10”。

右侧会实时显示设置的样式效果。最终的样式效果如图4-1所示。

图 4-1 样式 obs-111



- 步骤10** 完成样式的编辑后，单击“确定”保存样式。新建成功的“obs-111”样式就会显示在样式列表中。
- 步骤11** 选中左侧导航栏中的“对象”。单击对象列表中的“mountain.jpg”，进入文件详情页面。
- 步骤12** 单击“图片预览”页签，您可以在此页面预览使用该图片样式的图片效果。
- 步骤13** 单击“复制链接”，按钮提示“复制成功”时，即可获取到图片文件的访问地址来使用图片。

----结束

4.2 控制台代码编辑模式

本章节提供控制台代码编辑模式创建一个缩略的样式并进行使用的操作案例。

操作步骤

- 步骤1** 在管理控制台首页，选择“服务列表 > 存储 > 对象存储服务”。
- 步骤2** 在桶列表中单击待操作的桶，进入“对象”页面。
- 步骤3** 单击“上传对象”，进入“上传对象”页面。
- 步骤4** 选择待上传的图片，单击“打开”。

步骤5 单击“上传”，开始上传图片。上传完成后的图片会显示在对象列表。

步骤6 在左侧导航栏，选择“图片处理”。

步骤7 单击“创建”，进入样式编辑界面。

步骤8 输入样式名称为“style002”。在编辑页面，选中“代码编辑模式”。

步骤9 在代码输入框中输入如下处理缩略命令和参数。

指定一个w和h为100的矩形，将图片进行等比缩放，取在这个100*100的矩形延伸区域的最小图片。

```
image/resize,m_mfit,h_100,w_100
```

右侧会实时显示设置的样式效果。最终的样式效果如图4-2所示。

图 4-2 样式 style002



步骤10 完成样式的编辑后，单击“确定”保存样式。新建成功的“style002”样式就会显示在样式列表中。

步骤11 选中左侧导航栏中的“对象”。单击对象列表中的“mountain.jpg”，进入文件详情页面。

步骤12 单击“图片预览”页签，您可以在此页面预览使用该图片样式的图片效果。

步骤13 单击“复制链接”，按钮提示“复制成功”时，即可获取到图片文件的访问地址来使用图片。

----结束

5 获取图片信息

此功能仅支持通过接口调用模式使用。

通过此操作可以获取到图片的基本信息：长、宽、以及文件的大小和格式。如果图片包含 **Exif信息**，将按照json格式返回完整的Exif信息。

操作名称：info

示例

查询example.jpg信息。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/info>

```
{  
  "format": "JPEG",  
  "height": 2000,  
  "size": 1046583,  
  "width": 2668  
}
```

6 获取图片平均色值

此功能仅支持通过接口调用模式使用。

通过此操作可以获取图片的平均色值，平均色值将以json格式返回一个十六进制值。

操作名称：average-hue

示例

获取图片example.jpg的平均色值，可在浏览器直接访问以下地址。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/average-hue>

返回如下类似结果，表示成功获取到图片的平均色值。

```
{"RGB": "#28577D"}
```

example.jpg原图如下：



获取到的平均色值（#28577D）对应的颜色为：



7 设置图片效果

7.1 亮度

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。

具体的参数说明如表7-1所示。

操作名称：bright

表 7-1 亮度参数说明

参数	取值说明	代码样例
value	调整图片亮度，取值范围为[-100, 100]。 原始的亮度值为0。亮度从-100至100递增。	image/bright,50

示例

- 将图片样式的亮度设置为50。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/bright,50>



- 将图片样式的亮度设置为-50。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/bright,-50>



7.2 对比度

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。

具体的参数说明如表7-2所示。

操作名称：contrast

表 7-2 对比度参数说明

参数	取值说明	代码样例
value	调整图片对比度，取值范围为[-100, 100]。 原始的对比值0。对比度从-100至100递增。	image/contrast,-50

示例

- 将图片样式的对比度设置为-50。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/contrast,-50>



7.3 锐化

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。

具体的参数说明如表7-3所示。

操作名称：sharpen

表 7-3 锐化参数说明

参数	取值范围	代码样例
value	调整图片锐化程度，取值范围为[50, 399]。 50为最低的锐化效果，推荐最优取值为100。数值越大清晰度越高。但数值过高容易使图片不真实。	image/sharpen,100

示例

- 将图片样式的锐化设置为100。

<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/sharpen,100>



7.4 模糊

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。

具体的参数说明如表7-4所示。

操作名称：blur

表 7-4 模糊参数说明

参数	取值范围	代码样例
r	模糊的半径，取值范围为 [1, 50]。 数值越大模糊的范围越大。	image/blur,r_3,s_2
s	正态分布标准差，取值范围为 [1, 50]。 数值越大图片越模糊。	

说明

在图形界面模式中，对数值进行加减时两个参数为同增同减。

示例

- 将图片样式的模糊设置r为3，s为2。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/blur,r_3,s_2



8 设置缩略

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。通过resize操作能够使图片按照一定规则进行缩放，支持按照指定宽高和比例进行缩放。

📖 说明

- 关于长短边：长边是指原尺寸与目标尺寸的比值大的那条边，短边同理。例如原图为400px*200px，缩放为100px*100px，由于 $400/100=4$ ， $200/100=2$ ， $4>2$ ，所以在这个缩放中400那条是长边，200那条是短边。
- 目标缩略图最长边不能超过9999，宽和高乘积不能超过24999999。
- 仅指定高或宽进行缩略时，缩略后宽高比例为原图宽高比例，且按照原图格式返回。
- resize操作默认限制原图的放大。如果请求的目标缩略图比原图大时，需要调用参数limit,0才可以取到放大后的图片，否则将返回原图。调用格式如下：

```
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/  
resize,w_500,limit_0
```

具体参数说明如表8-1所示。

操作名称：resize

表 8-1 缩放设置

参数	取值说明	代码样例
m	<p>设置缩略的类型。取值为lfit、mfit、fill、pad和fixed，默认值为lfit。</p> <ul style="list-style-type: none">• lfit：指定一个w和h的矩形，将图片进行等比缩放，取在矩形内最大的图片。• mfit：指定一个w和h的矩形，将图片进行等比缩放，取在矩形延伸区域的最小图片。• fill：指定一个w和h的矩形，将图片进行等比缩放，取在延伸区域的最小图片，并进行居中剪切。即将mfit缩略类型的图片进行居中剪裁。• pad：指定一个w和h的矩形，将图片进行等比缩放，取在矩形内最大的图片，并在矩形空白处进行颜色填充。即lfit缩略类型的图片在矩形空白处进行颜色填充。• fixed：强制按照固定的宽高进行缩略。• ratio：指定一个w和h的目标比例，取原图内满足目标比例的最大的图片。比例范围为1-1000。	image/ resize,m_lfit,h_100,w_100
p	<p>等比例缩放的倍数百分比。取值范围为[1, 1000]。当取值为：</p> <ul style="list-style-type: none">• <100：缩小。• =100：保持原图大小。• >100：放大。	image/resize,p_50
h	<p>目标缩略图的高度。取值为[1, 9999]，且输出宽高乘积不超过24999999。</p>	image/ resize,m_lfit,h_100
w	<p>目标缩略图的宽度。取值为[1, 9999]，且输出宽高乘积不超过24999999。</p>	image/ resize,m_fixed,h_100,w_100
l	<p>指定目标缩略图的最长边。取值为[1, 4096]，且输出宽高乘积不超过24999999。</p> <p>长边为指定的值，短边按照比例缩放。</p>	image/resize,l_100

参数	取值说明	代码样例
s	指定目标缩略图的最短边。取值为[1, 4096]，且输出宽高乘积不超过24999999。 短边为指定的值，长边按照比例缩放。	image/resize,s_100
color	填充的颜色。选择pad（缩略后填充）模式时可以使用。 格式为十六进制颜色码，取值为000000到FFFFFF，默认值为白色。	image/ resize,m_pad,h_100,w_100,color_FF0000
limit	是否在目标缩略图比原图大时进行限制放大。取值为0或1，默认值为1。 <ul style="list-style-type: none"> 0：不进行限制。 1：进行限制。 	image/ resize,p_150,limit_0

注意

如果缩略后的图片出现颗粒感或锯齿感，可以通过在图片处理的URL后加“/marker,u_plus”来进行优化。

如：图片处理URL为https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0，添加该参数后的链接为https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0/marker,u_plus。

示例

- 指定高为100，宽度按比例处理（默认为lfit）。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_lfit,h_100

- 等比缩放，按短边将图片样式缩略为100*100。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_lfit,h_100,w_100

- 指定长边为100，短边按照比例缩放。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,l_100


- 固定宽高，居中自动剪切，将图片样式缩略为100*100。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_fill,h_100,w_100

- 强制将缩略图的宽和高固定为100。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_fixed,h_100,w_100

- 固定宽高，按短边将图片样式缩略为100*100，填充白色。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100

- 固定宽高，按短边将图片样式缩略为100*100，填充红色。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100,color_FF0000

- 等比例放大150%，调用参数“limit,0”取到放大图片。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,p_150,limit_0



- 调用p参数将图等比例缩小为30%。

https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,p_30



- 指定宽高比例，将原图宽高比例缩放至3:2。

https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,p_30



9 旋转图片

9.1 旋转设置

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

对图片进行旋转设置后，图片将会按顺时针方向进行旋转。具体说明如表9-1所示。

操作名称：rotate

表 9-1 图片旋转

参数	取值说明	代码样例
value	按照顺时针旋转的角度，取值范围为[0, 360]。 默认值为0，0表示不旋转。数值越大，图片按顺时针方向旋转的角度越大。	image/rotate,90

说明

- 旋转后的图片可能会比原图尺寸大。

示例

将图片宽度设置为100且旋转角度设置为90。

```
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,w_100/rotate,90
```



将图片宽度设置为100且旋转角度设置为220。

https://obs.example.region.com/image-demo/example.jpg?x-image-process=image/resize,w_100/rotate,220



9.2 自适应方向

此功能支持控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。

在一些相机或手机拍摄的图片中会含有**Exif信息**，其中可能包含例如Orientation的方向参数。图片拍摄时相机或手机的旋转信息会记录在方向参数中，浏览器可以根据这个参数信息将图片自动旋转到正确的方向。

设置自适应方向，带有方向参数的图片会先根据方向参数信息进行自动旋转。具体说明如**表9-2**所示。

操作名称：auto-orient

表 9-2 自适应方向说明

参数	取值说明	代码样例
value	取值为0或1。默认值为1。 0：不设置自适应方向。图片将不会进行自动旋转，保持默认方向。 1：设置自适应方向。图片会先自动旋转后再进行缩略。	image/resize,w_100/ auto-orient,0

说明

- 图片的高度和宽度须均小于4096才能设置自适应方向。
- 如果图片的Exif信息不含有旋转参数或者不带有Exif信息，则使用auto-orient参数操作无效且对图片无任何影响。

示例

- 将图片宽度设置为100，不设置自适应方向。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,w_100/auto-orient,0



10 剪切图片

10.1 普通剪切

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

可以设置图片上的任意一点为起始点，根据指定宽高进行图片剪切，剪切后的图片为矩形。具体说明如[表10-1](#)所示。

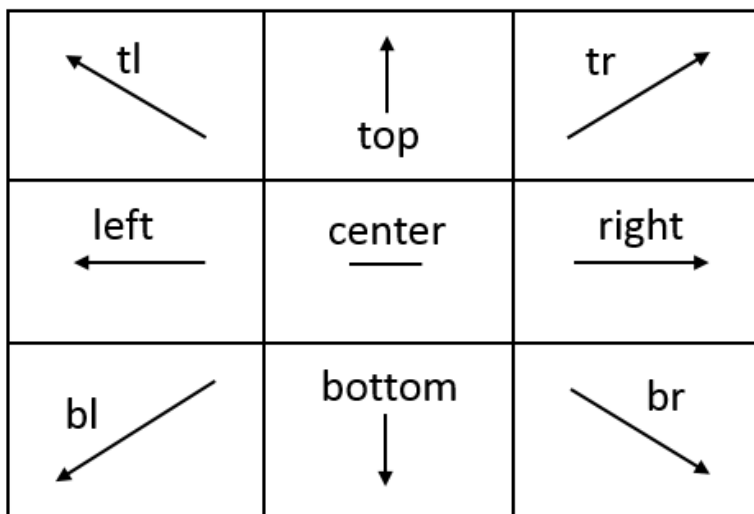
操作说明：crop

表 10-1 普通剪切

参数	取值说明	代码样例
g	表示剪切起始原点位置。取值为tl、top、tr、left、center、right、bl、bottom和br，共9个取值。取值的九宫格如 图10-1 所示，每个取值位置均位于每一小格的左上角顶点。	image/ crop,x_10,y_10,w_200,h_ 200,g_br
h	剪切的高度，取值范围为[0，图片高度]。	
w	剪切的宽度，取值范围为[0，图片宽度]。	
x	表示剪切起始点的横坐标，默认左上角为原点。取值范围为[0，图片边界]。	
y	表示剪切起始点的纵坐标，默认左上角为原点。取值范围为[0，图片边界]。	

剪切原点位置九宫格如[图10-1](#)所示。

图 10-1 剪切原点位置九宫格



📖 说明

- 如果从剪切原点开始设置的x和y坐标大于原图，则剪切无法正常执行，将返回错误。
- 如果从剪切原点开始设置的h和w均大于原图，则会直接剪切至图片边界。

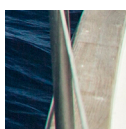
示例

- 剪切起点坐标设置为(1000, 500)，剪切的宽和高均设置为1000。
`https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/crop,x_1000,y_500,w_1000,h_1000`



- 剪切起点设置在九宫格右下角br区，起点坐标为(10, 10)，剪切的宽和高均设置为200。

https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/crop,x_10,y_10,w_200,h_200,g_br



10.2 内切圆

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

以图片的中心为圆心，根据指定的半径进行图片剪切，剪切后的图片为圆形。具体说明如表10-2所示。

操作名称：circle

表 10-2 内切圆说明

参数	取值说明	代码样例
r	图片剪切的圆形半径，取值范围为[0，图片最短边的一半]。	image/circle,r_100

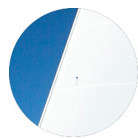
 说明

- 如果图片的输出格式是jpg，则内切圆以外区域为白色。如果图片的输出格式是png、webp、bmp等矢量图格式，则内切圆以外区域为透明。建议输出为png格式。
- 如果r取值超过图片最短边的一半，则输出的内切圆仍然是图片的最大内切圆。

示例

设置图片剪切的圆形半径为100，输出格式为jpg，内切圆以外区域为白色。

https://obs.example.region.com/image-demo/example.jpg?x-image-process=image/circle,r_100



10.3 索引剪切

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

以图片左上角顶点为起始点，设宽为x轴，高为y轴。根据指定长度进行等长剪切，根据指定索引取出剪切后区域。具体说明如表10-3所示。

操作名称：indexcrop

表 10-3 索引剪切说明

参数	取值范围	代码样例
x	水平剪切的每块图片长度。取值范围为[1，图片宽度]。x和y参数只能任选其一。	image/ indexcrop,x_1000,i_0
y	垂直剪切的每块图片长度。取值范围为[1，图片高度]。 x和y参数只能任选其一。	

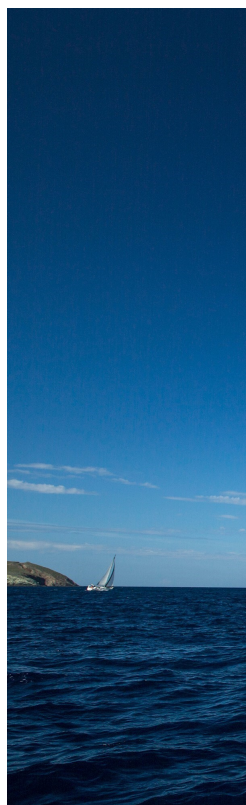
参数	取值范围	代码样例
i	如果剪切后一共有n块，则i的取值范围为[0, n-1]。0表示第一块，超过最大的块数则返回原图。	

示例

- 按照水平切割，每块图片的长度为1000，选择第一块。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/indexcrop,x_1000,i_0



- 按照水平切割，每块图片的长度为600，选择第一块。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/indexcrop,x_600,i_0



10.4 圆角剪切

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

指定圆角大小将图片剪切为圆角矩形。支持通过圆角半径大小和水平垂直大小两种方式设置，具体说明如表10-4所示。

操作名称：rounded-corners

表 10-4 圆角剪切参数

参数	取值说明	代码样例
r	将图片切出圆角，指定圆角的半径，水平和垂直的值相同，可以使用像素数（如200）或百分比（如25p）。 像素数取值范围为[1, 4096]，当像素数大于原图最小边的一半时，取最小边的二分之一。 百分比取值范围为[1p, 50p]。 不能与rx和ry参数同时使用。	image/rounded-corners,r_100

参数	取值说明	代码样例
rx	圆角水平大小的参数，可以使用像素数（如200）或百分比（如25p）。 像素数取值范围为[1, 4096]，当像素数大于原图最小边的一半时，取最小边的二分之一。 百分比取值范围为[1p, 50p]。 需要与ry同时使用。	image/rounded-corners,rx_100,ry_200
ry	圆角垂直大小的参数，可以使用像素数（如200）或百分比（如25p）。 像素数取值范围为[1, 4096]，当像素数大于原图最小边的一半时，取最小边的二分之一。 百分比取值范围为[1p, 50p]。 需要与rx同时使用。	

说明

如果图片的输出格式是jpg，则非圆角区域为白色。如果图片的输出格式是png、webp、bmp等矢量图格式，则非圆角区域为透明。建议在圆角剪切后保存为png格式。

示例

- 将example.jpg图片设置为圆角大小为100、格式为png的圆角图片。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/rounded-corners,r_100/format,png



- 将example.jpg图片设置为水平圆角大小为100、垂直圆角大小为200的圆角图片。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/rounded-corners,rx_100,ry_200



11 设置水印

11.1 公共参数

此功能在控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。可以将文字内容或另一张图片设置为原图的水印。

添加水印中的文字内容和字体或图片水印路径均需使用用于URL传输的Base64编码。标准的Base64编码并不适合直接放在URL里传输。用于URL传输的Base64编码的编码方式基本过程是：先用标准Base64编码将内容编码为字符串，然后检查该字符串，将字符串中的加号（+）换成中划线（-），并且将斜杠（/）换成下划线（_）。详细编码规范请参考RFC4648标准中的相关描述。

操作名称：watermark

公共参数为设置图片水印和文字水印均可以使用的参数，可以同时添加文字和图片水印。具体基础参数说明如表11-1所示。

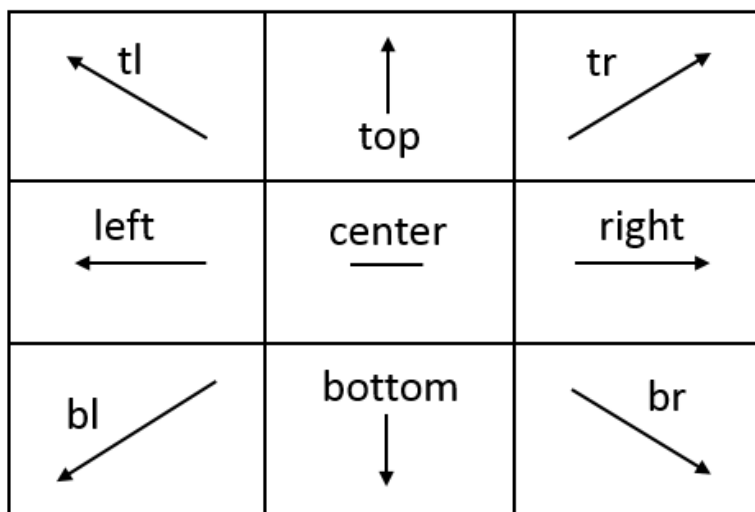
表 11-1 公共参数

参数	取值说明	代码样例
g	可选参数，表示水印处于图片的位置。取值为tl、top、tr、left、center、right、bl、bottom和br，共9个取值。默认值为tl。 取值的九宫格如图11-1所示，每个取值位置均位于每一小格的左上角顶点。	image/ watermark,image_aW1h Z2UtZGVtby 9sb2dvLnBuZw==,g_br,t 90,x_10,y_10
x	可选参数，表示距离图片边缘的水平距离，默认左上角为原点。取值范围为[0, 4096]。默认值为10。单位为像素（px）。	
y	可选参数，表示距离图片边缘的垂直距离，默认左上角为原点。取值范围为[0, 4096]。默认值为10。单位为像素（px）。	

参数	取值说明	代码样例
voffset	<p>可选参数，表示水印距离图片水平中线的垂直偏移方向。可以使水印根据中线往上或往下偏移。取值范围为[-1000, 1000]。默认值为0。单位为像素（px）。</p> <p>当g取值为left、center、right才有意义，即位置为左中、正中、右中才有意义。</p>	
align	<p>可选参数，水印文字和图片的对齐方式。取值为0、1或2。默认值为0。</p> <ul style="list-style-type: none"> • 0：表示上对齐。 • 1：表示中对齐。 • 2：表示下对齐。 	
order	<p>可选参数，水印文字和图片的前后顺序。取值为0或1。默认值为0。</p> <ul style="list-style-type: none"> • 0：表示图片在前面。 • 1：表示文字在前面。 	
t	<p>可选参数，文字或图片水印的透明度。取值范围为[0, 100]。默认值为100，100%表示不透明。</p>	
interval	<p>可选参数，表示文字和图片的间距。取值范围为[0, 1000]。</p>	

水印位置九宫格如图11-1所示。

图 11-1 水印位置九宫格



说明

如果为原图同时添加文字和图片水印时，水平距离x、垂直距离y和中线垂直偏移voffset可以用于调节水印位于原图中的位置，也可以调节两张水印的布局。

11.2 图片水印

添加图片水印时使用的参数。

可以在添加水印图片进行预处理操作，包括[设置缩略](#)、[旋转图片](#)和[剪切图片](#)，但不支持剪切为内切圆。进行缩略设置预处理时，还支持将水印图片按原图比例进行缩放。

具体图片水印参数说明如[表11-2](#)所示。

表 11-2 图片水印参数

参数	取值说明	代码样例
image	<p>水印图路径，添加图片水印时的必选参数。</p> <p>图片水印地址为：<i>bucketName/objectName</i>（必须编码）或 <i>bucketName/objectName?x-image-process=imageUrl/command</i>（必须编码）</p> <p>须知 内容必须是URL安全base64编码。 encodedObject = url_safe_base64_encode(object)。如object为“panda.png”，编码过后的内容为“cGFuZGEucG5n”。</p>	<pre>image/resize,w_400/ watermark,image_aW1h Z2UtZGVtby9sb2dvLnBuZz 94LWltYWdlLXByb 2Nlc3M9aW1hZ2UvcmluZS9 XplLFBfMzA=t_90, g_br,x_10,y_10</pre>
P	<p>水印图片尺寸，大写的P，表示将水印图片按原图（指被添加水印的图片）比例百分比P进行缩放。取值范围为[1，100]。</p> <p>须知 此处resize操作只支持大写P参数，不支持小写p参数。如需调整水印图片的大小，请参照resize参数（除小p参数之外）。</p>	<pre>image-demo/logo.png?x- image-process=image/ resize,P_50</pre>

接口调用示例

- 水印图片文件是：logo.png（logo图片地址：e-share/image-demo/logo.png），水印图片置于右下角，透明度为90，水平边距和垂直边距默认值都为10。

构造参数如下：

水印图片：e-share/image-demo/logo.png

对应安全Base64编码：ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5n

水印位置右下角：g_br

透明度：t_90

水平距离和垂直距离：x_10,y_10

URL请求如下：

```
https://obs.region.example.com/example.jpg?x-image-process=image/  
watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5n,g_br,t_90,x_1  
0,y_10
```

图 11-2 示例 1



```
https://obs.region.example.com/image-demo/example.jpg?x-image-  
process=image/  
watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5n,g_br,t_90,x_1  
0,y_10
```

- 水印图片文件是：logo.png（logo图片地址：e-share/image-demo/logo.png），对水印图片按宽度缩略为50进行预处理。其他参数同上一用例。

构造参数如下：

水印图片：e-share/image-demo/logo.png?x-image-process=image/resize,w_50

对应安全Base64编码：

```
ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2UtcHJvY2Vzcz1pbW  
FnZS9yZXNpemUsd181MA
```

水印图片放右下角：g_br

透明度：t_90

水平距离和垂直距离：x_10,y_10

URL请求如下：

```
https://obs.region.example.com/example.jpg?x-image-process=image/  
watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2  
UtcHJvY2Vzcz1pbWFnZS9yZXNpemUsd181MA,g_br,t_90,x_10,y_10
```

图 11-3 示例 2



https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2UtcHJvY2Vzcz1pbWFnZS9yZXNpemUsd181MA,g_br,t_90,x_10,y_10

- 水印图片文件是：logo.png（logo图片地址：e-share/image-demo/logo.png），对水印图片按原图比例缩小50%进行预处理。其他参数同上一用例。

构造参数如下：

水印图片：e-share/image-demo/logo.png?x-image-process=image/resize,P_50

对应安全Base64编码：

ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2UtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF81MA

水印图片放右下角：g_br

透明度：t_90

水平距离和垂直距离：x_10,y_10

URL请求如下：

https://obs.region.example.com/example.jpg?x-image-process=image/watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2UtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF81MA,g_br,t_90,x_10,y_10

图 11-4 示例 3



```
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5nP3gtaW1hZ2UtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF81MA,g_br,t_90,x_10,y_10
```

图形界面示例

图片水印可以通过OBS控制台新建图片样式设置。水印图片文件是：logo.png（logo图片地址：e-share/image-demo/logo.png），水印图片置于右下角，透明度为90，水平边距和垂直边距默认值都为10。在控制台上的具体操作步骤如下：

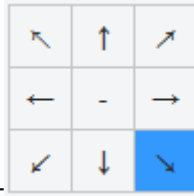
- 步骤1** 在OBS管理控制台首页，单击目标桶名称，在左侧导航栏中选择“图片处理”。
- 步骤2** 单击“创建”，进入样式编辑界面。
- 步骤3** 在编辑页面，输入“样式名称”，“编辑模式”选择“图形界面模式”，并选择“水印设置”。

📖 说明

样式名称只能由大写或小写的英文字母、数字、英文句号（.）、下划线（_）和连字符（-）组成。长度为1~256个字符。例如：rotate_0001。

- 步骤4** 设置如下水印参数。
 - 水印类型：选择“图片水印”
 - 水印图路径：输入“e-share/image-demo/logo.png”
 - 图片尺寸(%)：根据是否缩小水印图标设置
 - 亮度：保持默认值0

- 对比度：保持默认值0
- 透明度：设置为90



- 水印位置：选择右下角的箭头
- 垂直边距：保持默认值10
- 水平边距：保持默认值10

步骤5 完成样式的编辑后，单击“确定”保存样式。新建成功的样式将会显示在样式列表中。

后续即可参考[使用图片样式](#)使用新建的水印样式处理图片。

----结束

11.3 文字水印

添加文字水印时使用的参数，包括字体大小、字体类型以及文字颜色等。具体文字水印参数如[表11-3](#)所示。

表 11-3 文字水印参数

参数	取值说明	代码样例
text	添加文字水印时的必选参数。 须知 必须是URL安全base64编码。encodeText = url_safe_base64_encode(fontText)，最大长度为64个字符（支持最多16个中文字符）。	image/ watermark,text_SGVsbG8 g5Zu- 54mH5pyN5YqhIQ,size_6 0,color_FF0000,type_ ZmFuZ3poZW5nc2h1c29 uZw==,g_center,rotate_3 0
size	可选参数，表示文字水印的文字大小。取值范围为(0, 1000]。默认值为40。	
type	可选参数，表示文字水印的文字类型。取值见下方 表11-4 所示。默认值为wqy-zenhei（文泉驿正黑，编码后的值：d3F5LXplbmhlaQ）。 须知 1、必须是URL安全base64编码。encodeText = url_safe_base64_encode(fontType)。 2、当前不支持换行。	
color	可选参数，表示文字水印的文字颜色。 格式为六位十六进制颜色码，取值为000000到FFFFFF，默认值为黑色。	
shadow	可选参数，表示文字水印的阴影透明度。取值范围为(0, 100]。	

参数	取值说明	代码样例
fill	可选参数，表示水印的铺满效果。取值为0或1。 ● 0：无效果。 ● 1：铺满。	
rotate	可选参数，表示文字水印的按顺时针旋转的角度。取值范围为(0, 360)。	

表 11-4 文字类型编码对照表

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

接口调用示例

- 在原图加上文字水印，文字内容为：Hello，文字大小设置为60，颜色设置为红色，字体为方正书宋。
 构造参数如下：

文字内容：对应安全Base64编码：SGVsbG8g5Zu，即text_SGVsbG8g5Zu

文字大小：size_60

文字颜色：color_FF0000

字体：type_ZmFuZ3poZW5nc2h1c29uZw==

URL请求如下：

```
https://obs.region.example.com/image-demo/example.jpg?x-image-  
process=image/  
watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h  
1c29uZw==
```



- 在原图加上文字水印，水印位置居中，文字顺时针旋转30度，其他参数同上。

```
https://obs.region.example.com/image-demo/example.jpg?x-image-  
process=image/  
watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h  
1c29uZw==,g_center,rotate_30
```



- 同时实现图文混合水印。文字内容是：Hello，文字大小为60，阴影设置为50，字体颜色为红色，位置在右下角。

水印图片是logo.png，水平边距和垂直边距都是10。

图文水印透明为50，排版方式是图片在前，对齐方式是下对齐。

```
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5n,text_SGVsbG8g5Zu,size_60,color_FF0000,shadow_50,type_ZmFuZ3poZW5nc2h1c29uZw==,g_br,x_10,y_10,align_2,order_0
```



- 分开实现图文混合水印。文字内容是：Hello，文字大小为60，字体颜色为红色，位置在右下角。

水印图片是logo.png，水平边距和垂直边距分别是40，10。

图片水印透明为90，先实现图片水印，再实现文字水印。

```
https://obs.region.example.com/example.jpg?x-image-process=image/  
watermark,image_ZS1zaGFyZS9pbWFnZS1kZW1vL2xvZ28ucG5n,g_br,t_90,x_4  
5,y_10/  
watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h  
1c29uZw==,g_br,x_0,y_330
```



12 转化格式

12.1 格式转换

此功能在控制台图形界面模式和代码编辑模式使用，也支持通过接口调用模式使用。可以将原图转换为支持的图片格式，具体说明如表12-1所示。

- 支持的原图格式：jpg、jpeg、png、bmp、webp、gif、tiff。
- 支持输出的目标图格式：jpg、png、bmp、webp。

操作名称：format

表 12-1 格式转换

参数	取值说明	代码样例
jpg	保存为jpg格式。如果原图是webp、bmp、png等矢量图格式，则原图透明部分会填充为白色。	image/format,jpg
webp	保存为webp格式。	image/format,webp
bmp	保存为bmp格式。	image/format,bmp
png	保存为png格式。	image/format,png

示例

- 将格式为jpg的图片保存成png的格式。
`https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/format,png`



12.2 渐进显示

此功能仅支持通过控制台代码编辑模式和接口调用模式使用。

使用格式转换功能时，默认输出的jpg格式是标准型的jpg（Baseline JPEG）。如果需要输出能够渐进显示的jpeg（Progressive JPEG），使用此参数。具体参数说明如表12-2所示。

- 标准型jpg格式图片的呈现方式：自上而下显示。
- 渐进显示jpeg格式图片的呈现方式：由模糊逐渐清晰。

操作名称：interlace

表 12-2 渐进显示

参数	取值说明	代码样例
value	取值为0或1。 0：表示输出为自上而下显示的jpg格式图片。 1：表示输出为可以渐进显示的jpeg格式图片。	image/format.jpg/ interlace,1

示例

- 将图片输出为可以渐进显示的jpg格式图片。
`https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/format,jpg/interlace,1`



- 将图片输出为自上而下显示的jpg格式图片。
<https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/format,jpg/interlace,0>



13 质量变换

此功能支持在控制台代码编辑模式和接口调用模式使用。

可以对输出格式为jpg的图片进行图片压缩，不使用压缩则可能会使图片占用的空间变大。具体参数说明如表13-1所示。

操作名称：quality

表 13-1 图片压缩

参数	取值说明	代码样例
q	<p>图片的相对质量，即将图片按照原图的q%进行压缩。取值范围为[1, 100]。</p> <p>压缩公式：目标图质量=原图质量 * q%</p> <p>例如：如果原图质量为100%，将图片压缩至80%的相对质量，则目标图片的质量为80%。如果原图的质量为80%，将图片压缩至80%的相对质量，则目标图片的质量为64%。</p>	<pre>image/ resize,w_100,h_100/ quality,q_80</pre>
Q	<p>图片的绝对质量，即直接将图片压缩为Q%，与原图不存在相对关系，不依赖于原图。取值范围为[1, 100]。</p> <p>压缩公式：</p> <ul style="list-style-type: none">• 原图质量 > Q%，目标图质量 = Q%• 原图质量 = Q%，目标图质量 = 原图质量 = Q%• 原图质量 < Q%，目标图质量 = 原图质量 <p>例如：如果原图质量为100%，将图片压缩至80%的绝对质量，则目标图片的质量为80%。如果原图质量为70%，将图片压缩至80%的绝对质量，则目标图片的质量为70%。</p>	

📖 说明

- q参数仅对输出格式为jpg的图片有效。
- 如果同时使用了q和Q参数，则以Q为输出结果。
- 输出格式为jpg格式以外的图片使用此参数则不会产生效果，也不会产生影响。

示例

- 先将图片缩略为高和宽均为100，输出相对质量80%格式为jpg的目标图。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80



- 先将图片缩略为高和宽均为100，输出绝对质量80%格式为jpg的目标图。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,w_100,h_100/quality,Q_80



14 设置图片瘦身

图片瘦身支持通过OBS控制台代码编辑模式和接口调用模式使用。

图片瘦身是对图片压缩功能的简化，操作原理为输出绝对质量为75%的图片。无需配置参数，通过操作命令即可完成图片瘦身。

操作名称：imageslim

📖 说明

- 仅支持对jpg格式的图片进行瘦身。
- 建议在其他图片处理操作后再进行图片瘦身。

示例

- 先将图片缩略为高和宽均为100，再对缩略后的图片进行瘦身。
https://obs.region.example.com/image-demo/example.jpg?x-image-process=image/resize,w_100,h_100/imageslim



15 处理图片持久化

图片处理持久化支持将处理后的图片异步存储至指定的OBS桶中，方便下次直接访问处理后的图片，减少图片处理花费的时间，提升用户体验。

图片处理持久化目前仅支持通过接口调用模式使用。在图片处理请求接口中，以**参数名=参数值**的格式，发送图片处理持久化请求，具体参数说明如表15-1所示。

表 15-1 图片处理持久化

参数名	参数值	说明
x-image-save-object	<i>objectName</i>	必填参数。 指定目标对象名称，即处理后的图片存储至桶中的对象名。 对象命名要求如下： <ul style="list-style-type: none">不能包含以下特殊字符：\:*? "<> 范围为1到1023位。
x-image-save-bucket	<i>bucketName</i>	选填参数。 指定目标存储桶，处理后的图片将存储至该桶。如果不指定的话将默认保存至当前桶。 桶名范围为1到64位，且必须是在OBS中已存在的桶。

Java示例代码：

```
ObsClient obsClient = null;
String endPoint = "obs-endpoint"; // 当前所在region
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
// 本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_ACCESS_KEY。
// 您可以登录访问管理控制台获取访问密钥AK/SK。
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY");
try {
    ObsConfiguration config = new ObsConfiguration();
    config.setEndPoint(endPoint);
    obsClient = new ObsClient(ak,sk ,config);
    TemporarySignatureRequest request = new TemporarySignatureRequest();
```

```
request.setObjectKey("test.jpeg"); // 处理前的原对象名
Map<String, Object> queryParams = new HashMap<>();
queryParams.put("x-image-process", "image/resize,w_100");
String objectName = "your saves objectName"; // 处理后的对象名
// 非必选参数
String bucketName = "your saves Bucket"; // 处理后保存的桶名
queryParams.put("x-image-save-object", ServiceUtils.toBase64(objectName.getBytes("UTF-8")));
queryParams.put("x-image-save-bucket", ServiceUtils.toBase64(bucketName.getBytes("UTF-8")));
request.setQueryParams(queryParams);
request.setBucketName("your bucket"); // 原对象的桶
TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
// 访问的url
response.getSignedUrl();
} catch (Exception e) {
    // 异常处理
} finally {
    if (obsClient != null) {
        obsClient.close();
    }
}
```

Python示例代码:

```
from obs import ObsClient
import os
import traceback
import requests
```

```
# 推荐通过环境变量获取AKSK，这里也可以使用其他外部引入方式传入，如果使用硬编码可能会存在泄露风险。
# 您可以登录访问管理控制台获取访问密钥AK/SK。
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# 【可选】如果使用临时AKSK和SecurityToken访问OBS，则同样推荐通过环境变量获取
security_token = os.getenv("SecurityToken")
# server填写Bucket对应的Endpoint，这里以region为例，其他地区请按实际情况填写。
server = "https://obs.region.com"
# 创建obsClient实例
# 如果使用临时AKSK和SecurityToken访问OBS，需要在创建实例时通过security_token参数指定securityToken值
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # 生成图片持久化处理的带授权信息的URL
    # 原对象所在的桶名
    bucketName = 'originBucketName';
    # 处理前的原对象名
    objectKey = 'test.png';

    # 处理后的对象名
    targetObjectName = "save.png"
    # 处理后保存的桶名, 可选参数
    targetBucketName = "saveBucketName"
    queryParams={}
    queryParams["x-image-process"]="image/resize,w_100"
    queryParams["x-image-save-object"]=base64.b64encode(targetObjectName.encode("utf-8")).decode()
    # 可选参数
    queryParams["x-image-save-bucket"]=base64.b64encode(targetBucketName.encode("utf-8")).decode()

    res = obsClient.createSignedUrl(method='GET', bucketName=bucketName, objectKey=objectKey,
    queryParams=queryParams, expires=3600)
    print('signedUrl:', res.signedUrl)
    print('actualSignedRequestHeaders:', res.actualSignedRequestHeaders)
    // 使用GET请求处理图片持久化
    r = requests.get(res.signedUrl)
    print(r)
except:
    print(traceback.format_exc())
```

nodejs示例代码:

```
// 引入obs库
const ObsClient = require('esdk-obs-nodejs');
```

```
const https = require('https');
const http = require('http');
const urlLib = require('url');

// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
// 本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_ACCESS_KEY。
// 您可以登录访问管理控制台获取访问密钥AK/SK。
const ak = process.env.ACCESS_KEY_ID;
const sk = process.env.SECRET_ACCESS_KEY;
const server = "obs-endpoint"; // 当前所在region

// 创建ObsClient实例
const obsClient = new ObsClient({
  access_key_id: ak,
  secret_access_key: sk,
  server: server
});

// 原对象所在的桶名
const bucketName = 'originBucketName';
// 处理前的原对象名
const objectKey = 'test.png';
const method = 'GET';

// 处理后的对象名
const targetObjectName = "save.png";
// 处理后保存的桶名，可选参数
const targetBucketName = 'saveBucketName';

const queryParams = {
  "x-image-process": "image/resize,w_100",
  "x-image-save-object": Buffer.from(targetObjectName, 'utf8').toString('base64'),
  // 可选参数
  "x-image-save-bucket": Buffer.from(targetBucketName, 'utf8').toString('base64')
}

const res = obsClient.createSignedUrlSync({
  Method: method,
  Bucket: bucketName,
  Key: objectKey,
  QueryParams: queryParams
});

// 使用GET请求处理图片持久化
const url = urlLib.parse(res.SignedUrl);
const request = server.startsWith('http://') ? http : https;
const req = request.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
```

📖 说明

- 对象名和桶名都必须是URL安全base64编码。encodedObject = url_safe_base64_encode(name)。例如对象名为“panda.png”，编码过后的内容为“cGFuZGEucG5n”。base64编码后如果含有“+”、“/”，需要分别替换为“-”、“_”。
- 如果报签名匹配错误，请检查AK/SK是否正确，以及访问的URL与代码生成的签名URL是否保持一致。
- 当前暂不支持带range头域的持久化。

16 常见问题

16.1 什么是图片处理特性？

图片处理特性（Image Processing）集成于对象存储服务（Object Storage Service, OBS），为用户提供稳定、安全、高效、易用、低成本的图片处理服务，包括：图片瘦身、图片剪切、图片缩放、图片水印、格式转换等。

图片处理特性提供控制台和RESTful API接口的使用方法，使用户可以在任何时间和地点对存储在对象存储中的图片进行处理，并且可以快速获取到处理后的图片。

16.2 如何访问图片处理？

- 通过管理控制台查看OBS中图片在相应图片样式模板下的图片预览效果。
- 通过应用程序调用简单的RESTful API接口来访问图片处理服务。

16.3 每个桶支持创建多少个图片样式？

每个桶最多支持创建100个样式。

OBS提供两种不同的方法创建图片样式：

- [通过控制台处理图片](#)
- [通过接口处理图片](#)

16.4 图片处理支持哪些格式的图片？

支持的原图格式有：jpg、jpeg、png、bmp、webp、gif、tiff。支持格式转化处理后的目标格式有：jpg、png、bmp、webp。

16.5 如何通过 URL 访问图片处理服务？

访问非公共可读的图片

针对非公共可读的图片，需要在计算签名时加上图片处理参数，最终通过生成带签名的临时URL来访问。

访问公共可读的图片

针对公共可读的图片，直接在访问图片的URL请求后添加图片处理参数即可。

A 修订记录

发布日期	修订记录
2020-03-31	第一次正式发布。