Alibaba Cloud
Object Storage Service

Data Processing

MORE THAN JUST CLOUD | C-D Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
⛔	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⛔ Danger: Resetting will result in the loss of user configuration data.
⚠️	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	⚠️ Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
📋	This indicates warning information, supplementary instructions, and other content that the user must understand.	ⓘ Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	📋 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
Courier font	It is used for commands.	Run the `cd  / d   C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae   log   list  --instanceid Instance_ID`
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	**It indicates that it is a required value, and only one item can be selected.**	`swich` *{stand | slave}*

Contents

1 Image Processing

1.1 Image processing

This topic describes the basic features and versions of Alibaba Cloud OSS Image Processing (IMG) service and how to use this service. Featuring high capacity and cost efficiency, IMG provides two image processing APIs that allow you to build image -related services.

> **Note:**
>
> IMG is activated automatically when you activate OSS. For information about how to activate OSS, see *Sign up for OSS*.

Basic features

- Query image metadata
- Convert image formats
- Scale, crop, and rotate images
- Attach images, text, and text-and-image watermarks to images
- Customize image processing styles
- Call multiple image processing features by using pipelines

Versions

There are two versions of IMG.

> **Note:**
>
> This topic describes the features of the latest version of IMG. The API of the previous version will no longer receive updates. For details related to compatibility, see *FAQ on using old and new versions of APIs and domain names*.

Quick start

· Create an image style by completing the following steps:

1. Log on to the *OSS console*.

2. Click your bucket name to go to the Overview page of the bucket.

3. Choose the Image Processing tab, and then click Create Rule.

4. On the Create Rule page, configure the image style by performing graphical operations on the Basic Edit tab page or by using an SDK or parameters on the Advanced Edit tab page.

   The parameters on the Basic Edit tab page are as follows:

   - `Rule   Name` : the name of the image style. The name must be 1 to 64 characters in length, and can contain numbers, letters, underscores (_), hyphens (-), and periods (.).

   - `Format` : the format of images. Values: `Original | jpg | jpeg | png | bmp | gif | webp | tiff` .

   - `Fade    In` : Specifies whether to enable the Fade In function.

   - `Adaptive   Orientatio  n` : Specifies whether to enable the Adaptive Orientation function.

     We recommend that you enable the Adaptive Orientation function. When this function is enabled, an image is rotated according to its EXIF data before it is resized.

   - `Image    Quality` : the quality of images. Values: `Relative | Absolute | Uncompress  ed` .

   - `Resize    Type` : the method of resizing images. Values: `Thumbnail Disabled | Proportion  al   Scale   Down | Proportion  al Scale   Up | Fixed   Width   and / or   Height` .

     📋 Note:
     The "long side" refers the side with a larger source size to target size ratio. The same applies to the "short side." For example, for an original image that is scaled from 400x200 to 800x100, the original-to-target ratios are 0.5

> (400/800) and 2 (200/100). Given that 0.5 is less than 2, the 200 side is the longer side and the 400 side the shorter one.

- `Image   Brightness` : the image brightness.

- `Image   Contrast` : the level of contrast in an image.

- `Image   Sharpening` : Specifies whether to sharpen an image. If this function is enabled, the level of sharpening applied to an image.

- `Image   Blurring` : Specifies whether to blur an image. If this function is enabled, the level of blur effect applied to an image.

  When the Image Blurring function is enabled, you can set `Blur   Radius` and `Blur   Sigma` .

- `Image   Rotation` : the angle of rotation of an image.

- `Watermark` : Specifies whether to enable the Watermark function. If this function is enabled, the type of watermark added to an image.

The parameters on the Advanced Edit tab page are as follows:

- `Rule   Name` : the same as the `Rule   Name` parameter on the Basic Edit tab page.

- `Code` : You can enter API code to edit an image.

  Examples:

  ■ `image / resize , w_200`

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

  > 📋 Note:
  > You can edit images only by using the API of the latest version.

5. Click OK.

   After the new image style is created, you can apply it to your images through OSS
   .

- Apply an image style.

  1. Log on to the *OSS console*.

  2. Click your bucket name to go to the Overview page of the bucket.

  3. Choose the Files tab, select an existing image or upload a new image to open the Preview page.

     For information about how to upload an image, see *Upload objects*.

  4. Select an image style from the Image Style drop-down list.

     You can view the effect of the image that is processed according to the selected style. Additionally, an address that carries the image style is generated accordingly. You can click Copy File URL to obtain the URL.

1.2 Image processing access rules

In Image Service, URLs are accessed with standard HTTP GET requests, and all processing parameters are in the QueyString of the URL.

Request for thumbnails through processing parameters

If you want to have a source image processed and then returned, the following two formats are available:

- URL

  Access through a third-level domain name: `http :// bucket .< endpoint >/ object ? x - oss - process = image / action , parame_val  ue`

  - `Bucket`: your Image Service channel.
  - `endpoint`: the access domain name for a Bucket's data center.
  - `Object`: In Image Service, an Object is the basic data unit for operating images. It is the same as the Object specified for the OSS instance. The maximum size of a single Object (that is, each image) is 20 MB.
  - `action`: the operation to be performed on the image.
  - `parame`: the parameter which indicates the operation to be performed on the image.

- Combination of multiple actions

  Multiple actions are executed in sequence. For example, `image / resize`, `w_200 / rotate`, `90` has the effect of scaling down an image to 200 in width and then rotating the image 90 degrees.

Example

Assume that the requested Bucket is *image-demo* and located in *China East 1 (Hangzhou)*, with the domain name *oss-cn-hangzhou.aliyuncs.com*, and the requested image is *example.jpg*. The URL format for scaling down the image to 200 in width is:

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

The URL format for HTTPS access is:

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

The URL format for access through a custom domain name is:

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

Request for thumbnails through styles

Style

Image Service allows you to save image processing operations and parameters as an alias, that is, a style. With styles, a series of operations can be achieved through a very short URL.

- A Channel can have multiple styles. Currently, a Channel is allowed to have up to 50 styles.
- A style can be applied to change all Objects in a Channel. For example, if style abc is in Channel A and the style content is 100w.jpg (scaled to 100 in width and saved as a .jpg file), style abc can be applied to all the Objects in Channel A to scale them to 100 in width and saved them as .jpg files.
- A style is only effective within a Channel, that is, the Objects in Channel A cannot use any style in Channel B.

Style naming conventions:

- A name can be 1 to 63 characters in length.

- Only numbers, upper-case or lower-case letters, underscores (_), hyphens (-), and periods (.) are permitted.

Channel

A channel is a namespace of image processing, and the management entity for billing , permission control, logging, and other advanced functions. An image name is globally unique in Image Service and cannot be modified. You can create up to 10 Channels, but the number of Objects in each Channel is not limited.

Image processing data centers correspond to the OSS data centers. If you create a Bucket in an OSS data center and then activate Image Service, the corresponding Channel belongs to this data center. Currently, a Channel corresponds to a Bucket in the OSS instance, that is, you can only create a Channel of the same name as a Bucket that you have created on the OSS instance.

Channel naming conventions:

- Only lower-case letters, numbers, and hyphens (-) are permitted.
- It must start and end with a lower-case letter or number.
- The length must be 3–63 bytes.

To simplify the process, you can save a specific processing method as a style. Later, you just need to specify a style to call the same processing method. The URL format for image processing by style is as follows:

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

Example

The preceding processing parameters can be saved as the style `style-example`. Assume that the requested Bucket is `image-demo` and located in `China East 1 (Hangzhou)`, with the domain name `oss-cn-hangzhou.aliyuncs.com`, the requested image is `example.jpg`, and the image access style is `style-example`, the URL format is constructed as follows:

]

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

The URL format for HTTPS access is:

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

Access through SDK

Public buckets can be accessed using URLs, whereas private files are typically accessed using SDKs. Because in Image Service, URLs are accessed with standard GET operations, only the process parameter needs to be added to the Get Object.

The Python SDK is used as an example:

```
bucket  =  oss2 .  Bucket ( oss2 .  Auth ( access_key  _id ,
access_key  _secret ),  endpoint ,  bucket_nam  e )
key  = ' example . jpg '
new_pic  = ' new - example . jpg '
process  = " image / resize , m_fixed , w_100 , h_100 " // Scale
  down   the   image   based   on   the   target   width   and
height
bucket . get_object  _to_file ( key ,  new_pic ,  process = process )
```

For more information about Image Service used for OSS SDKs, see Image Processing in the SDK documentation. The following table lists links of image processing used for some SDKs.

SDK	Image processing document	Example
Java SDK	*Image processing*	*ImageSample.java*
Python SDK	*Image processing*	*image.py*
C# SDK	*Image processing*	*ImageProcessSample.cs*
PHP SDK	*Image processing*	*Image.php*
JS SDK	*Image processing*	*object.test.js*
C SDK	*Image processing*	*oss_image_sample.c*

Image processing restrictions

· The supported formats include JPG, PNG, BMP, GIF, WEBP, and TIFF.

· When the width or height of a thumbnail is specified, the image is scaled by a single side by default in the case of proportional scaling. With fixed width and height, the image is scaled down by assuming equal width and height.

· The scaled image size is restricted. The product of the width and height of the target thumbnail cannot exceed 4096 x 4096, and the length of a single side cannot exceed 4096 x 4.

· When resize is called, the image cannot be enlarged by default. That is, if the requested image is larger than the source image, the source image is returned. If you want to enlarge the image, add the parameter `limit_0` .

· Currently, GIF and WEBP images can be processed once at a time to reduce resource consumption. For example, you cannot crop a GIF or WEBP image immediately after resizing it.

1.3 Resize images

Generate a thumbnail of the image as required or make the specified scaling.

> Note:
> The supported formats include jpg, png, bmp, gif, webp, and tiff.

Parameters

Operation name: `resize`

· **Scale up and down with specified width and height**

Name	Description	Value range
m	Specify the scaling mode: - `lfit` : proportional scaling. It refers to the maximum image that is limited in the rectangle of the specified w and h. - `mfit` : proportional scaling. It refers to the minimum image extending out of the rectangle of the specified w and h. - `fill` : fixed width and height. It refers to the cropped and centered minimum image extending out of the rectangle of the specified w and h. - `pad` : fixed width and height, scaling down and filling. - `fixed` : fixed width and height, enforced scaling down.	[lfit, mfit, fill, pad, fixed], the default value is lfit.
w	Specify the target width.	1-4096
h	Specify the target height.	1-4096
l	Specify the longer side of the target.	1-4096
s	Specify the shorter side of the target.	1-4096

Name	Description	Value range
limit	Specify whether to process the target thumbnail when it is larger than the original image. 1 indicates not to process, and 0 indicates to process.	0/1. The default value is 1
color	When you set the scaling mode as pad (scaling down and filling), you can select the filling color (The default is white). Filling format of parameters: use hexadecimal color codes, for example 00FF00 (green).	[000000-FFFFFF]

· Proportional scaling

·
Name	Description	Value range
p	Percentage. If it is smaller than 100, it means to scale down; if it is bigger than 100, it means to scale up.	1-1000

Note

· For the original image:

  - Formats supported: jpg, png, bmp, gif, webp, and tiff.

  - File size cannot exceed 20 MB.

  - When using the image rotation, the width or height of the image cannot exceed 4096.

  - The size of a single side cannot exceed 30,000.

· For the thumbnail: The scaled image size is restricted. The product of the width and height of the target thumbnail cannot exceed 4096 x 4096, and the length of a single side cannot exceed 4096.

- When the width or height of a thumbnail is specified, the image is scaled by a single side by default in the case of proportional scaling. With fixed width and height, the image is scaled down by assuming equal width and height.

- When only the width or height of a thumbnail is specified, the image is returned in the same format as the original image. If you want to save the image into other formats, see *Quality Transformation* and *Format conversion*.

- When `resize` is called, the image cannot be enlarged by default. That is, if the requested image is larger than the original image, the original image is returned. If you want to enlarge the image, add the parameter `limit , 0` to be called (for example: `https :// image - demo . oss - cn - hangzhou . aliyuncs . com / example . jpg ? x - oss - process = image / resize , w_500 , limit_0`)

Example

Scaling-down by a single side (by width and height)

- Scale down an image to 100 in height, and the width is adjusted proportionally.

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

Scaling-down by a single side (by the longer side and shorter side)

- Limit the longer side of an image to 100, and the shorter side is adjusted proportionally.

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

Scaling-down based on target width or height

· **Scale down an image to 100 x 100 (w x h).**

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

  resize,m_fixed,h_100,w_100

Proportional scaling, restricted in a rectangle frame

· **Scale down an image by the longer side to 100 x 100 (w x h).**

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

  resize,m_lfit,h_100,w_100

· **Scale down an image by the longer side to 100 x 100 (w x h) and save it as png.**

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

  resize,m_lfit,h_100,w_100/format,png

Proportional scaling, restricted out of a rectangle frame

·   **Scale down an image by the shorter side to 100 x 100 (w x h)**

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

resize,m_mfit,h_100,w_100

Fixed width and height, automatic cropping

·   **Automatically crop an image to 100 x 100 (w x h)**

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

resize,m_fill,h_100,w_100

Fixed width and height, scaling down and filling

·   **Scale down an image by the shorter side to 100 x 100, and then fill the remaining
    area with a solid color.**

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

resize,m_pad,h_100,w_100

· Scale down an image by the shorter side to 100 x 100, and then fill the remaining area with red.

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

resize,m_pad,h_100,w_100,color_FF0000

· Scale down an image to 1/2 of the original size.

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

resize,p_50

1.4 Convert formats

1.4.1 Gradual display

There are two ways to present a picture in JPG format:

· Top-down Scanning
· Fuzzy first and then gradually clear (when the network environment is poor)

The default is saved as the first, if you want to specify a presentation that is vague and clear, use the progressive display parameters.

Parameter

Operation name: `interlace`

Name	Description	Value range
[value]	1 For JPG format saved as a progressive display 0 for a normal JPG format	[0, 1]

> **Note:**
>
> This parameter only makes sense if the effect graph is in JPG format.

Examples

- Saves the photos in PNG format to a JPG format that is displayed incrementally.

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

- The image is reduced to a width of 200, and saved in a gradually displayed JPG format.

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

1.4.2 Quality Transformation

If the image is saved as JPG or webp, the quality transformation can be supported.

Parameter

Operation name: `quality`

Name	Description	Value range
q	Determine the relative quality of the picture and compress the original graph according to Q. If the original graph has a quality of 100%, use 90 Q to get an image of quality of 90% If the original image quality is 80%, the use of 90 Q will get a picture of the quality of 72. The concept of relative compression can only be used on images in the original image that are in JPG format. If the original diagram is webp, then the relative mass is equivalent to absolute mass.	1-100

Name	Description	Value range
Q	To determine the absolute quality of the picture, press the original graph quality to Q %, if the original graph quality is less than the specified number, not compressed . If the original graph quality is 100%, the use of "90q" will get a picture of the quality of 90; if the original graph quality is 95%, the use of "90 Q" will also get a picture of the quality of 90; if the original graph quality is 80%, using "90q" does not compress and returns the original graph of quality 80 . It can be used only on the SAVE as JPG/webp effects, and the other formats have no effect. If both Q and q are specified, press Q to process.	1-100

Note:

If you don't fill in Q parameter or q parameter, it may cause the image to take up more space. If you really want to get a picture of fixed quality, use the Q parameter.

Examples

- The original graph is reduced to a JPG graph of 80% relative to the original graph quality.

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

- The original graph is reduced to a JPG graph with an absolute quality of 80.

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

1.4.3 Format conversion

You can convert an image to different formats, such as JPG, PNG, BMP, WebP, and GIF
. By default, no format is specified and images are returned in the source format.

Parameter

Name	Description
jpg	Saves the source image in JPG format. If the source image is in PNG, WebP, or BMP format that supports transparent channels, OSS fills the transparent section in white by default.
png	Saves the source image in PNG format.
webp	Saves the source image in WebP format.
bmp	Saves the source image in BMP format.
gif	Saves the source image in GIF format. If the source image is not in GIF format, it is saved in the source format.

Name	Description
`tiff`	Saves the source image in TIFF format.

Precautions

· For a standard image resize action, we recommend that you add the format parameter to the end of the image processing parameter string, such as `image / resize , w_100 / format , jpg` .

· For a resize image action that includes adding a watermark to the image, we recommend that you add the format parameter next to the resize parameter, for example: `image / reisze , w_100 / format , jpg / watermark ,....`

· When an image is saved in JPG format, it is saved in baseline JPEG format by default. To save it in progressive JPEG format, you can set the interlace parameter. For more information, see *Gradual display*.

Example

· Save a PNG image in JPG format.

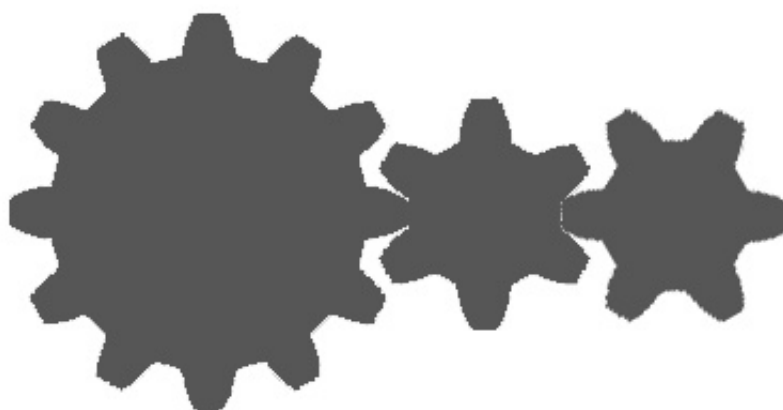Request URL:*http://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/format,jpg*

· Save a PNG image in JPG format that supports progressive JPEG display.

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

· Save a GIF image in JPEG format.

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

Scale down the image to 200 in width.

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

· Save a GIF image in WebP format.

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

1.5 Add watermarks

This feature allows you to add an image or text as a watermark to another image.

Parameters

This table provides a description of the basic parameters and their values, which can be used with the `watermark` operation.

Basic parameters

Name	Description	Parameter type
t	It indicates the transparency. This parameter makes the added image watermark or text watermark transparent Default value: 100 (in the unit of %), indicating no transparency; value range: [0–100]	Optional
g	It indicates the position of a watermark on the target image. The position is shown in the following figure. Value range: [nw,north,ne,west,center,east,sw,south,se]	Optional

Name	Description	Parameter type
x	It indicates the horizontal margin, that is, the horizontal distance between the watermark and the image edge. This parameter is meaningful only when the watermark is in the upper left, middle left, lower-left corner, upper right, middle right, or lower-right corner of the image. Default value: 10 Value range: [0 - 4,096] Unit: pixel (px)	Optional
y	It indicates the vertical margin, that is, the vertical distance between the watermark and the image edge. This parameter is meaningful only when the watermark is in the upper left, top center, upper right, lower-left corner, bottom center, or lower-right corner corner of the image. Default value: 10 Value range: [0 - 4,096] Unit: pixel (px)	Optional

Name	Description	Parameter type
voffset	It indicates the midline vertical offset. When the watermark is in the middle left, center, or middle right of the image, you can designate the vertical offset of the watermark along the midline. Default value: 0 Value range: [- 1, 000, 1,000] Unit: pixel (px)	Optional

📋 **Note:**

· In addition to the position of a watermark on the image, the horizontal margin, vertical margin, and the midline vertical offset can regulate the watermark layout when the image has multiple watermarks.

· The URL-safe Base64 encoding can be used during image processing. For more information, see RFC4648 or the URL-safe Base64 encoding section.

· The Parameter-Position Mapping Table for the g parameter, is provided as follows :

nw	north	ne
west	center	east
sw	south	se

Image watermark parameters

Name	Description	Parameter type
`image`	It indicates the object name of an image watermark (which must be encoded). Note: The URL-safe base64 encoding is required: encodedObject = url_safe_base64_encode(object). For example, if the object name is `panda.png`, the encoded name is `cGFuZGEucG5n`.	Required parameter

Watermark image preprocessing

When a user applies a watermark, the watermark image can be preprocessed. Supported preprocessing operations include: image scaling, image cropping (incircle not supported), and image rotation. Additionally, another parameter is supported for the resize operation: P . P indicates the watermark image scale relative to the master image. The value range is [1-100], indicating the scale percentage.

Preprocessing examples

For example, if P_10 is set, for a master image of 100x100, the size of the watermark is 10x10. If the same watermark processing parameters are applied to images of different sizes, the watermark image may be too large or too small. The P parameter solves this problem. Using the p parameter, IMG dynamically adjusts the size of the watermark image according to the size of the main image.

If you scale panda.png to 30% in width, then the watermark file is: `panda.png?x-oss -process=image/resize,P_30` After adding URL-safe Base64 encoding this watermark file is: `cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA`

If the watermark is placed in the lower-right corner corner and the source image width is reduced to 400, the watermark operation is: `watermark=1&object= cGFuZGEucG5nQDMwUA&t=90&p=9&x=10&y=10`. This is applied to the image as follows:

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

If the source image is reduced to 300 in width, the watermark operation is:

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

Text watermark

Name	Description	Parameter type
text	It indicates the text of the text watermark (encode required). **Note:** The URL-safe base64 encoding is required: encodeText = url_safe_base64_encode(fontText). Maximum length: 64 characters	Required
type	It indicates the literal type of a text watermark (encoding required) **Note:** NOTE: The URL-safe base64 encoding is required: encodeText = url_safe_base64_encode(fontType). Value range: See the Literal Type Encoding Table as follows. Default value: wqy-zenhei (encoded value: d3F5LXplbmhlaQ)	Optional

Name	Description	Parameter type
color	It indicates the color of the textual content of a text watermark (encoding required) The URL-safe base64 encoding is required : EncodeFontColor = url_safe_base64_encode(fontColor). The parameter format must be # + six hexadecimal numbers . For example, #000000 indicates black. "#" indicates the prefix. Every two digits of 000000 constitute an RGB color. # FFFFFF indicates the white color. Default value: #000000 (black)	Optional
size	It indicates the size (px) of the textual content of a text watermark. Value range: (0, 1,000] Default value: 40	Optional
shadow	It indicates the shadow transparency of a text watermark. Value range: (0, 100]	Optional

Name	Description	Parameter type
rotate	It indicates the clockwise rotation angle of the text. Value range: [0,360]	Optional
fill	It indicates the effect of filling the image with a watermark. Value range: [0,1]. 1 indicates that the image is filled with the watermark; 0 indicates no filling effect.	Optional

Literal type encoding table

Parameter value	Meaning	URL-safe Base64 encoded value	Remarks
wqy – zenhei	Wen quan yi zheng hei ti, a type of Chinese font	D3F5LXplbmhlaQ ==	According to the RFC, the padding characters == can be omitted, that is, d3F5LXplbmhlaQ
wqy – microhei	Micro Hei font of the WenQuanYi Chinese font project	d3F5LW1pY3JvaGVp	
fangzhengs husong	Founder ShuSong, a Chinese Simplified font	ZmFuZ3poZW 5nc2h1c29uZw==	According to the RFC, the padding characters == can be omitted, that is, ZmFuZ3poZW 5nc2h1c29uZw
fangzhengk aiti	Founder Kai, a Chinese Simplified font	ZmFuZ3poZW 5na2FpdGk=	According to the RFC, the padding character = can be omitted, that is, ZmFuZ3poZW 5na2FpdGk

Parameter value	Meaning	URL-safe Base64 encoded value	Remarks
fangzhengh eiti	Founder Hei, a Chinese Simplified font	ZmFuZ3poZW 5naGVpdGk=	According to the RFC, the padding character = can be omitted, that is, ZmFuZ3poZW 5naGVpdGk
fangzhengf angsong	Founder FangSong , a Chinese Simplified font	ZmFuZ3poZW 5nZmFuZ3Nvbmc=	According to the RFC, the padding character = can be omitted, that is, ZmFuZ3poZW 5nZmFuZ3Nvbmc
droidsansf allback	DroidSansFallback	ZHJvaWRzYW 5zZmFsbGJhY2s=	According to the RFC, the padding character = can be omitted, that is, ZHJvaWRzYW 5zZmFsbGJhY2s

Text & image watermark

Name	Description	Parameter type
order	It indicates the order of the text watermark and image watermark of a text & image watermark. Value range: [0, 1]. 0 (default) indicates that the image watermark is before the text watermark ; 1 indicates that the text watermark is before the image watermark.	Optional

Name	Description	Parameter type
align	It indicates the alignment of the text watermark and image watermark of a text & image watermark. Value range: [0, 1, 2]. 0 (default): top alignment; 1: center alignment; 2: bottom alignment	Optional
interval	It indicates the spacing between the text watermark and image watermark of a text & image watermark. Value range: [0, 1000]	Optional

URL-safe Base64 encoding

Many parameters must be Base64 encoded during image processing. For more information, see *RFC4648*. The URL-safe Base64 encoding is only applicable to some specific watermark parameters (text content, color, and font of a text watermark, and object of an image watermark). Do not use it in a signature. The encoding format is:

· Encode the content to produce a base64 result.

· Replace the plus sign (+) in the result with a minus sign (-).

· Replace the slash sign (/) in the result with an underscore (_).

· Keep all equal signs (=) at the end of the result;

An example in Python is shown as follows:

```
import   base64
input =' wqy - microhei '
```

```
print ( base64 . urlsafe_b6  4encode ( input ))
```

Example

- The following URL watermarks the file example.jpg with panda.png (after URL-safe base64 encoded: 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

- Scale panda.png to 50 in width. Then the watermark
  file is panda.png?x-oss-process=image/resize,w_50, and
  cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=) after URL-safe Base64 encoding.

  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_cGFuZGEucG5nP3
  gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=,t_90,g_se,x_10,y_10

1.6 Response to errors

If an error occurs while accessing the Image Service, the Image Service returns an error code and error message. This enables you to locate and correct the error.

Image Service error response format

An example of an error response message is given as follows:

```
< Error >
  < Code > BadRequest </ Code >
  < Message > Input   is   not   base64   decoding .</ Message >
  < RequestId > 52B155D2D8  BD99A15D00  05FF </ RequestId >
  < HostId > userdomain </ HostId >
</ Error >
```

This error response message contains the following elements:

· Code: An error code that the Image Service returns to the user.

· Message: Detailed error information provided by the Image Service.

· RequestId: A unique UUID used to identify an error request. When a problem cannot be solved, this ID can be sent to the Image Service engineers to help locate the cause of the error.

· HostId: Used to identify the accessed Image Service cluster.

Image Service error codes

Error code	Description	HTTP status code
InvalidArgument	Parameter error	400
Badrequest	Incorrect request	400
Missingargument	A parameter is missing	400
Imagetoolarge	Image size exceeds the limit	400
Watermarkerror	Watermark error	400
AccessDenied	Access is denied	403
Signaturedoesnotmatch	Signature does not match	403
Nosuchfile	Image does not exist	404
Nosuchstyle	Style does not exist	404
InternalError	Internal service error	500
NotImplemented	Method not implemented	501

Processing parameter restrictions

Image Service has the following restrictions:

- The source file size cannot exceed 20 MB.
- Scaling operation: The scaled image size is limited. The product of its width and height cannot exceed 4096 x 4096, and the length of a single side cannot exceed 4096 x 4.
- Rotation operation: The size of the image to be rotated is limited. The image width or height cannot exceed 4,096.
- A maximum of 4 channels are allowed.

1.7 Image style

Adding all the changes to the image after the URL makes the URL too long and inconvenient for management and reading. IMG allows you to save common operations as an alias, that is, a style. With the style, a complicated operation can be performed through a short URL.

Multiple styles (50 at most) are grouped under a bucket. Each style is effective only within the bucket.

Style access rules

URL parameters

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

Example:

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

This is the default style access method supported by IMG.

Separators

```
< File   URL >< Separator >< StyleName >
```
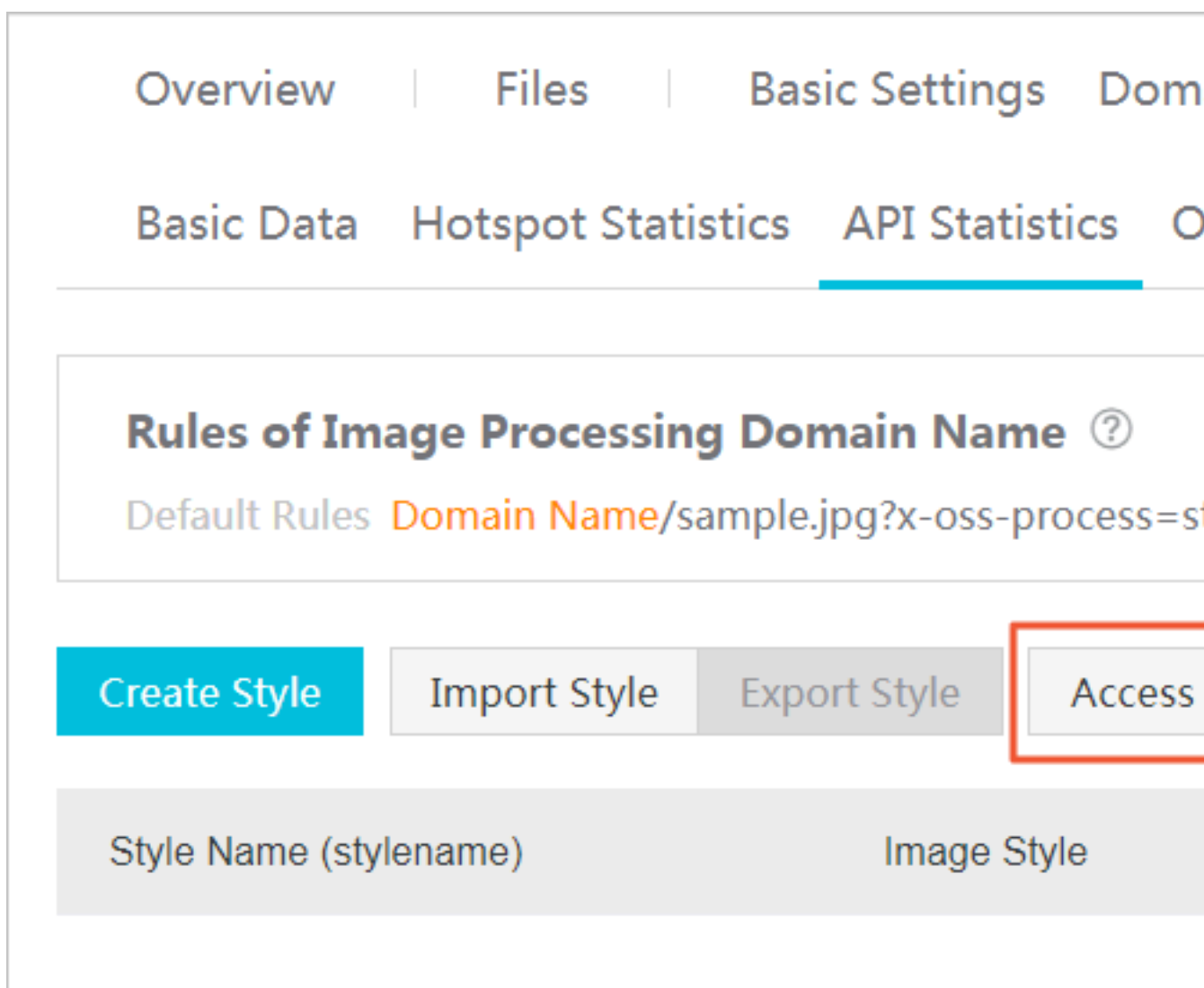
Example:

```
bucket . aliyuncs . com / sample . jpg { separator } stylename
```

IMG regards the content after the separator in a URL as the style name. This is an optional method provided by IMG. You must set separators in the console. Separators such as `-`, `_` , `/`, and `!` are supported.

- *StyleName*indicates the name of a style.
- Style creations, deletions, and modifications are all performed in the console.
- When the requested style does not exist in the specified bucket, the system returns the error NotSuchStyle.

Set separators

1. In the left-side bucket list of the *OSS console*, click the bucket to which you want to set separators.

2. Click the Image Processing tab, and then click Access Settings. As shown in the following figure:

3. **In the Access Settings dialog box, set the following parameters:**

Access Settings

Source Image
Protection

After Source Image Protection is enabl
accessed by adding stylename or signa
to access source files or add image pa
styles.

Custom Separator ⑦ ☐ -   ☐ _   ☐ /   ☐ !

- Source Image Protection: After enabling the original image protection, you can only access the image file by passing in the StyleName or using a signature-based method. Direct accesses to the OSS original file or accesses by passing in image parameters and modifying the image style are not allowed.
- Customize separator

Click OK.

Example

In this example, a style is created in the bucket image-demo.

Style name	Style content
panda_style	image/resize,m_fill,w_300,h_300,limit_0 /auto-orient,0/quality,q_90/watermark ,image_cGFuZGEucG5n,t_61,g_se,y_10, x_10

· **Access through parameters**

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

· **Access through parameters**

· **Access through URL parameters in style mode**

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

panda_style

· **Access through style separators in style mode**

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

These three methods bring the same result.

1.8 Source image protection

To avoid image piracy risks, the exposure to image URLs must be restricted so that only thumbnailed or watermarked images can be obtained. To do this, you can enable source image protection.

Rule description

After enabling the source image protection, you cannot access images in the following two ways:

- Access directly with an OSS address: `http :// bucket .< endpoint >/ object`
  `.`
- Request thumbnails with processing parameters: `http :// bucket .< endpoint`
  `>/ object ? x - oss - process = image / action , parame_val  ue`

You can only access images in style mode:

- Access through URL parameters `http :// bucket .< endpoint >/ object ? x`
  `- oss - process = style /< StyleName >`
- Access through separators `http :// bucket .< endpoint >/ object <`
  `separator >< StyleName >`

Note:

- The preceding rules only apply to anonymous accesses to public-read files. After enabling the source image protection, you can obtain source images using a signature-based method.
- The source image protection is designed for protecting image files, and the suffixes of the image files to be protected must be set. For example, if `. jpg` files are set for source image protection, you can still directly access the source images of `. png` files.

You can configure the access rules in the Image Processing module of the bucket in the console. Access rules
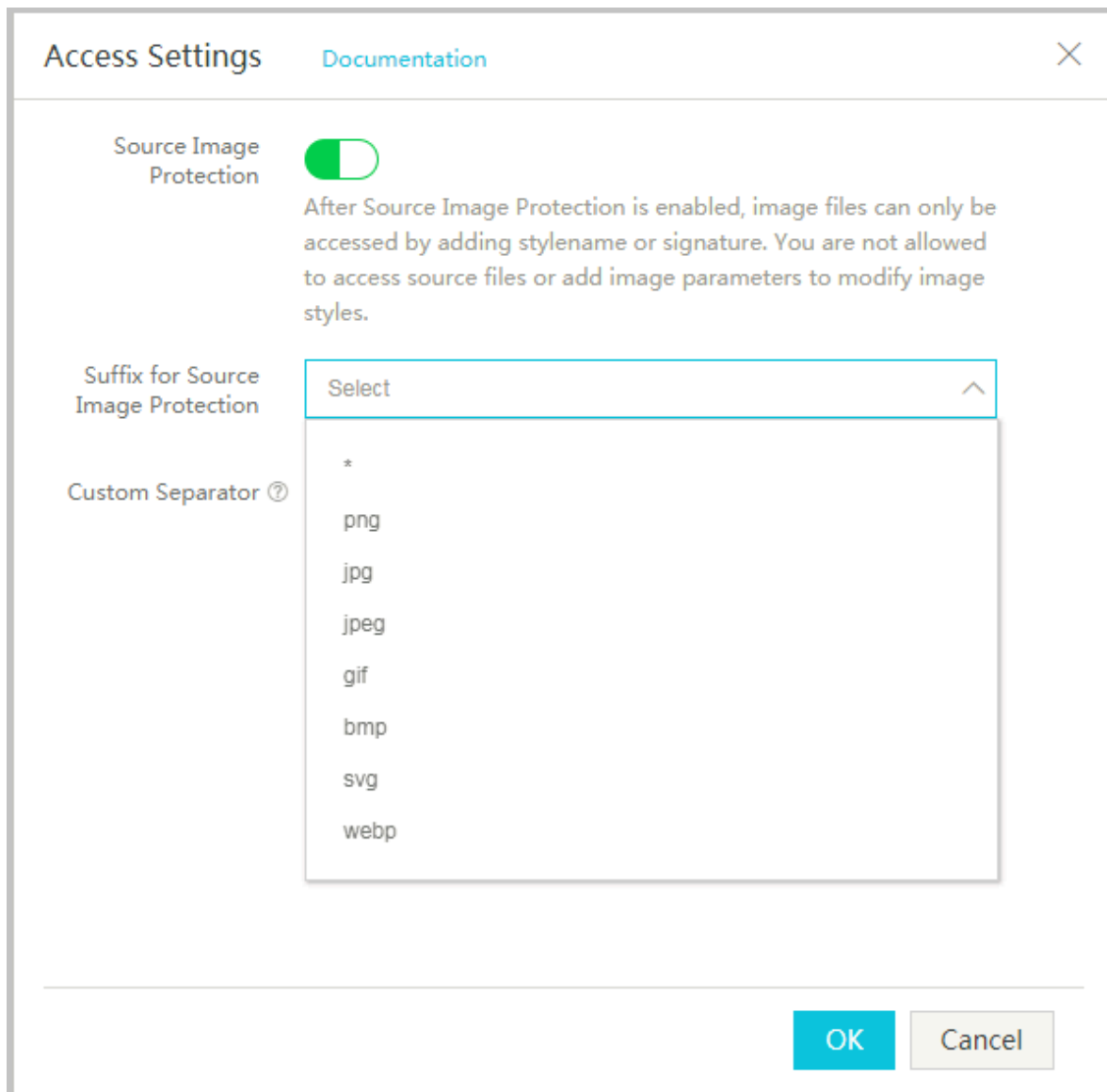
Configure access rules

1. In the left-side bucket list of the *OSS console*, click the bucket for which you want to set the source image protection.

2. **Click the Image Processing tab to locate the Access Settings button. See the following figure:**

3. Click Access Settings to open the Access Settings dialog box, as shown in the following figure:

In the Settings dialog box, perform the following settings:

· Enable source image protection: After enabling the source image protection, you can only access the image file by passing in the `stylename` or using a signature-based method. Direct accesses to the OSS source file or accesses by passing in image parameters and modifying the image style are not allowed.

· Set the suffixes of the image files for source image protection.

· Customize separators.

4. Once you set the needed options, click OK to finish setting the source image protection.

1.9 Save processing result

We provide the "saveas" operation for data processing. With this feature, you can
 save the processing result to a designated bucket as resources and assign it with
a specified key. After the resource is saved, you can visit the resource directly by
specifying the bucket to speed up resource download. This feature applies to ultra-
large image cropping or other high-latency operations.

Request syntax

```
POST  / ObjectName ?  x – oss – process   HTTP / 1 . 1
Content – Length :  ContentLen  gth
Content – Type :  ContentTyp  e
Host :  BucketName . oss – cn – hangzhou . aliyuncs . com
Date :  GMT   Date
Authorizat  ion :  SignatureV  alue
x – oss – process = image / resize , w_100 | sys / saveas ,
o_dGVzdC5q  cGc , b_dGVzdA
```

The Post interface is used to call the Image Processing Service. Parameters are passed
 in the body. The saveas operation is added to support saving the image as an OSS
object. Specifically, the parameters following "x-oss-process" are the same as those
for calling image processing features using queryString.

List of saveas parameters

Parameter	Meaning
o	Name of the target object. The parameter must be encoded in URL Safe Base64.
b	Name of the target bucket. The parameter must be encoded in URL Safe Base64. If this parameter is not specified , the image is saved to the current bucket by default.

Detail analysis

· The saveas operation requires that the caller has the permission for writing data to
the target bucket and object. Otherwise, 403 is returned.

· The bucket and object names in the saveas parameter must conform to the bucket
and object naming conventions of the OSS. Otherwise, 400 is returned.

· The bucket specified for the saveas operation must be in the same region as the
current bucket. Otherwise, 400 is returned.

- The saveas operation is only valid in the Post operation and not in the Get operation. Otherwise, 400 is returned.

Examples

Request example

```
POST  / ObjectName ? x - oss - process   HTTP / 1 . 1
Host :  oss - example . oss . aliyuncs . com
Content - Length :  247
Date : Frey ,  04   2012   03 : 21 : 12   GMT
Authorizat ion :  OSS   qn6qrrqxo2  oawuk53otf  jbyc : KU5h8YMUC7
8M30dXqf3J  xrTZHiA =
x - oss - process = image / resize , w_100 | sys / saveas ,
o_dGVzdC5q  cGc , b_dGVzdA
```

In the sample, the parameters indicate to save the zoomed target image to the bucket named "test", and the object name is *test . jpg* .

Response sample

```
HTTP / 1 . 1   200   OK
x - oss - request - id :  534B371674  E88A4D8906  008B
Date : Fri ,  04   May   2012   03 : 21 : 12   GMT
Content - Length :  0
Connection :  keep - alive
Server :  AliyunOSS
```

1.10 FAQ on using old and new versions of APIs and domain names

There are major differences between new and old versions of APIs:

- New version API: `http :// bucket .< endpoint >/ object ? x - oss - process = image / action , parame_val  ue`

  All image manipulation operations are passed by `x - oss - process` . Each action is executed sequentially without any need for channel management.

- Old versionAPI: `http :// channel .< endpoint >/ object @ action . format`

  It can be processed as a separator by@.

What are the advantages of OSS domain names when used with the Image Service?

Item	Use IMG domain access	Direct use of OSS domain name access
Use	Store and process two Domain Name Systems	One-stop processing for upload, management, process, distribution.
Is new version of API supported?	Supported	Supported
Is old version of API supported?	Supported	Not supported by default
Is https supported?	Not supported.	Supported
Is VPC Network supported?	Not supported	Supported
Is multi-domain binding supported?	Not supported	Supported
Is source station update automatically refresh Alibaba CDN supported?	Not supported	Supported

Note:

· When OSS domain names are being used, only APIs for the new version of the ING service can be used. When IMG domain names are being used, APIs for the old and new versions of the IMG service can be used.

· If the IMG domain name is expected to be capable of multi-CDN acceleration, the IMG domain name can be directly accessed by configuring the CDN to go back to the source host, and domain name binding is not required to complete the CDN acceleration.

What is the logic here for the two API methods and the two domain name access methods on the console?

Bucket processed before enabling the old version of image

· To keep the logic consistent with the original, the user sees the Domain Name of the old version of IMG, and custom domain names that have previously been bound.

· The user's original graph protection configuration on the IMG domain name has no effect on the OSS domain name. When you start the same step in cross-region

replication, the original graph protection and style separator are synchronized to the OSS domain name.

· When the user closes the image processing service for the current bucket, the style configuration and domain name binding are cleared, and automatically jump to the new page.

Newly created bucket or a bucket that has not previously opened the IMG service:

· The default is to be able to use the image processing service, which does not need to be up or turned off.

· No need to bind domain names, the domain name binding operation is directly consistent with the domain name management of the bucket itself.

If I'm currently using APIs for the old version of the IMG service, how do I switch to OSS domain names?

Currently, APIs for the old version of the IMG service cannot be used with OSS domain names without a request being sent to Alibaba Cloud. To request use of APIs for the old version, submit a ticket to Alibaba Cloud asking for this service. For style -based access, both OSS and IMG domain names can be used. If all your images are accessed by style, follow these steps to switch to the use of OSS domain names:

1. Enable configuration synchronization in the current Image Service configurations, so that style separators and the source image protection feature can be synchroniz ed to OSS domain names.

2. If you use a custom domain name, direct its CNAME to the OSS domain name.

Are style configurations the same for IMG and OSS domain names?

All style configurations are shared by IMG and OSS domain names. Style configurat ions for IMG domain names can be applied to OSS domains.

2 Capture video frames

Alibaba Cloud OSS Image Processing (IMG) can not only process existing images but also capture the frame at a specified time in a video file as an image.

> ⓘ Notice:
> · Currently, OSS can only capture frames from video files in the H.264 format.
> · Captured images are not stored by default. You must manually download the captured images to the local device.

Parameters

Operation type: `video`

Operation name: `snapshot`

Parameter	Description	Value range
t	Specifies the time when the image needs to be captured.	Unit: ms Range: [0, video duration]
w	Specifies the width of the captured image. If this parameter is set to 0, the width of the image is automatically calculated.	Pixel value: [0, video width]
h	Specifies the height of the captured image. If this parameter is set to 0, the height of the image is automatically calculated. If `w` and `h` are both set to 0, the width and height of the captured image are the same as those of the video file.	Pixel value: [0, video height]

Parameter	Description	Value range
m	Specifies the image capturing mode. If this parameter is not specified , the image is captured in the default mode, that is, the image at the specified time in the video is captured. If it is set as fast, the most recent key frame before the specified time is captured.	Enumerated value: fast
f	Specifies the format of the captured image.	Enumerated value: jpg and png

Example

· **Capture the frame at the 7th second and output it as an image in jpg format.**

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

· **Capture the frame at the 50th second accurately and output it as an image in jpg format.**

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

t_50000,f_jpg,w_800,h_600