Alibaba Cloud

Object Storage Service

Data Processing

C-⊃ 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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice 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.
🔔 Warning	A warning notice 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 restart an instance.
🔊 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔊 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings> Network> Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
Courier font	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Imag
1.1. Overview

You can add Image Processing (IMG) parameters to GetObject requests to process image objects stored in Object Storage Service (OSS). For example, you can add image watermarks to images or convert image formats.

Parameters

OSS allows you to directly use one or more parameters to process images. You can also encapsulate multiple IMG parameters in a style to batch process images. For more information about image styles, see 图片样式.

When multiple IMG parameters are specified, OSS processes the image in the order of the parameters. The following table describes the parameters that you can configure to process images.

IMG operation	Parameter	Description
Resize images	resize	Resizes images to a specified size.
Incircle	circle	Crops images based on the center point of images to ellipses of the specified size.
Custom crop	crop	Crops rectangular images of the specified size.
Indexed cut	indexcrop	Cuts images along the specified horizontal or vertical axis and selects one of the images.
Rounded rectangle	rounded-corners	Crops images to rounded rectangles based on the specified rounded corner size.
Auto-rotate	auto-orient	Auto-rotates images for which the auto-orient parameter is configured.
Rotate	rotate	Rotates images clockwise based on the specified angle.
Blur	blur	Blurs images.
Brightness	bright	Adjusts the brightness of images.
Sharpen	sharpen	Sharpens images.
Contrast	contrast	Adjusts the contrast of images.
Gradual display	interlace	Configures gradual display for the JPG images.
Adjust image quality	quality	Adjusts the quality of images in the JPG and WebP formats.
Format conversion	format	Converts image formats.
Add watermarks	watermark	Adds image or text watermarks to images.
Query the average tone	average-hue	Queries the average tone of images.
Query the EXIF data of an image	info	Queries image information, including basic information and EXIF information.

For example, after you configure the `resize` and `quality` parameters for the `example.jpg` source image, the URL for an image named exmaple.jpg is `https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/quality,q_90` . You can configure Content Delivery Network (CDN) back-to-origin rules to filter or retain the IMG parameters contained in the URLs of images that you want to retrieve. This way, you can retrieve source images or images processed by specifying IMG parameters from OSS.

- Retrieve source images

  You can enable Parameter Filtering for CDN to remove all IMG parameters that follow the question mark (?) in the URL of the image you want to retrieve.In this case, the source image `example.jpg` is retrieved.

- Retrieve processed images

  You can enable Retain Specified Parameters for CDN to retain all IMG parameters that follows the question mark (?) in the URL of the image you want to retrieve.In this case, the processed image is retrieved.

For more information about how to configure CDN back-to-origin rules, see Enable Alibaba Cloud CDN to retain only specified URL parameters and ignore other URL parameters.

Implementation methods

You can use object URLs, API operations, and SDKs to process images. For more information, see IMG implementation modes.

Usage notes

When you use IMG, take note of the following items:

- Limits on source images
  - Only JPG, PNG, BMP, GIF, WebP, TIFF, and images are supported.
  - The size of the source image cannot exceed 20 MB.
  - For the rotate operation, the height or width of the source image cannot exceed 4,096 pixels. For other operations, the width or height of the source image cannot exceed 30,000 pixels, and the total pixel number of the source image cannot exceed 250 million.

    The pixel number of a dynamic image, such as a GIF image, is calculated by using the following formula: `Width × Height × Number of image frames` . The pixel number of a static image, such as a PNG image, is calculated by using the following formula: `Width × Height` .

- Limits on resized images

  The width or height of the resized image cannot exceed 16,384 pixels. The total pixel number of the resized image cannot exceed 16,777,216 pixels.

- Limits on image styles

You can create up to 50 image styles for each bucket. To create more than 50 styles for a bucket, contact the technical support .

Billing

When you use IMG, you are charged the following fees:

- Image processing fees

  You are charged for IMG based on the size of the source images only after the free quota is exceeded. For more information about image processing fees, see Data processing fees.

- Request fees

  A GetObject request is generated each time when you use IMG to process an image. You are charged based on the number of generated requests. For more information about request fees, see API operation calling fees.

- Traffic fees

  You are charged for the outbound traffic over the Internet based on the size of the source images. For more information about traffic fees, see Traffic fees.

Release notes

IMG provides two API versions: the later API version and the earlier API version. This topic describes the APIs of the later version. APIs of the earlier version are not updated. For more information about the compatibility between the later and earlier versions of APIs, see FAQ on using old and new versions of APIs and domain names.

1.2. IMG implementation modes

You can use object URLs, API operations, or SDKs to process images in Object Storage Service (OSS). This topic describes how to use these methods to process images.

Add parameters to the URL of an image object

To process an image, you can add Image Processing (IMG) parameters or image style parameters to the end of the URL of the image object.

> **Notice** By default, when you use a URL to access an image object in a bucket, the image is downloaded. To ensure that an image object is previewed when you access the image object, you must map a custom domain name to your bucket and add a CNAME record. For more information, see Map custom domain names.

Image processing mode	Add IMG parameters	Add image style parameters
IMG URL	`https://bucketname.endpoint/objectname?x-oss-process=image/action,parame_value`	`https://bucketname.endpoint/objectname?x-oss-process=style/stylename`
Parameter description	- `https://bucketname.endpoint/objectname` : the URL of the image object. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?. - `x-oss-process=image/` : the fixed parameter, which indicates that the image object is processed by adding IMG parameters. - `action, param_value` : the action, parameter, and value of an IMG operation. These parameters determine the IMG operation that is used to process the image object. Separate multiple operations with forward slashes (/). OSS processes images in the order of IMG parameters. For example, `image/resize,w_200/rotate,90` indicates that OSS resizes the image to a width of 200 pixels, and rotates the image 90°. For more information about the supported IMG parameters, see Parameters.	- `https://bucketname.endpoint/objectname` : the URL of the image object. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?. - `x-oss-process=style/` : the fixed parameter, which indicates that the image object is processed by adding image style parameters. - `stylename` : the name of the style that you set in the OSS console. For more information, see 图片样式. If you specify a custom delimiter, you can use the delimiter to replace `?x-oss-process=style/` to simplify the IMG URL. For example, if you set the delimiter to an exclamation point (!), the URL of the processed image object is `<https://bucketname.endpoint/objectname!stylename` . For more information about how to configure custom delimiters, see Set delimiters.
Examples	https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/quality,q_90	https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/panda_style

The preceding description applies only to objects that can be anonymously accessed. If your object does not allow anonymous access, you must add the IMG operation to the signature by using SDKs. For more information about OSS SDKs, see the following topics:

- Java SDK
- Python SDK
- PHP SDK
- Go SDK
- C SDK
- C++ SDK
- .NET SDK
- Android SDK
- iOS SDK
- Node.js SDK
- browser.js SDK

Use OSS SDKs to process images

You can configure IMG or image style parameters in SDKs to process images. The following code provides an example on how to use OSS SDK for Java to process images.

- Add IMG parameters

  When you use IMG parameters to process images, separate multiple IMG operations with forward slashes (/).

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) reg
ion, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API
operations. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O&M. To create a RAM user,
log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket in which the image you want to process is stored.
String bucketName = "examplebucket";
// Specify the full path of the image object. The full path of the object cannot contain the bucket name.
String objectName = "exampleobject.jpg";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Resize the image to the height and width of 100 pixels.
String style = "image/resize,m_fixed,w_100,h_100";
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
// Name the processed image example-resize.jpg and save the image to your local computer.
// Specify the local path to which you want to save the image. Example: D:\\localpath\\example-resize.jpg. If the specified local file exists, it i
s overwritten by the processed image. Otherwise, the local file is created.
// By default, if you set this parameter to the name of a local file such as example-resize.jpg without specifying the local path of the file, the
processed object is saved to the local path of the project to which the sample program belongs.
ossClient.getObject(request, new File("D:\\localpath\\example-resize.jpg"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about the naming conventions for objects and buckets, see Bucket naming conventions and Object naming conventions.

- Add image style parameters

```
// Set yourEndpoint to the endpoint of the region in which the bucket is located. For example, if the bucket is located in the China (Hangzhou) reg
ion, set yourEndpoint to https://oss-cn-hangzhou.aliyuncs.com.
String endpoint = "yourEndpoint";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to access OSS because the account has permissions on all API
operations. We recommend that you use a RAM user to call API operations or perform routine O&M. To create a RAM user, log on to the RAM console.
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// Specify the name of the bucket in which the image you want to process is stored.
String bucketName = "examplebucket";
// Specify the full path of the image object. The full path of the object cannot contain the bucket name.
String objectName = "exampleobject.jpg";
// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Use the custom style to process the image.
// Set yourCustomStyleName to the name of the image style you create in the OSS console.
String style = "style/yourCustomStyleName";
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
// Name the processed image example-new.jpg and save the image to your local computer.
// Specify the local path to which you want to save the image. Example: D:\\localpath\\example-new.jpg. If the specified local file exists, it is o
verwritten by the processed image. Otherwise, the local file is created.
// By default, if you set this parameter to the name of a local file such as example-new.jpg without specifying the local path of the file, the pro
cessed object is saved to the local path of the project to which the sample program belongs.
ossClient.getObject(request, new File("D:\\localpath\\example-new.jpg"));
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about SDK demos for other programming languages, see the following topics:

- Java SDK
- Python SDK
- PHP SDK
- Go SDK
- C SDK
- C++ SDK
- .NET SDK
- Android SDK
- iOS SDK
- Node.js SDK
- browser.js SDK

Call API operations to process images

You can add IMG or image style parameters to GetObject requests to process images.

- Add IMG parameters

  The following code provides a sample request:

  ```
  GET /oss.jpg?x-oss-process=image/resize,w_100 HTTP/1.1
  Host: oss-example.oss-cn-hangzhou.aliyuncs.com
  Date: Fri, 24 Feb 2012 06:38:30 GMT
  Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJkcde6OhZ9J*****
  ```

- Add image style parameters

The following code provides a sample request:

```
GET /oss.jpg?x-oss-process=style/styleexample HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:40:10 GMT
Authorization: OSS qn6qrrawuk53oqxo2otfjbyc:UNapEgQDb7GJkcde6OhZ9J*****
```

Save processed images

By default, IMG does not save processed images. However, OSS allows you to add the saveas parameter in an IMG request to save the processed image as an object to a specified bucket. For more information, see Save processed images.

1.3. IMG parameters

1.3.1. Resize images

You can use the resize parameters to adjust the size of images stored in Object Storage Service (OSS). This topic describes the parameters used to resize images and provides examples on how to resize images.

Parameters

Operation: `resize`

The following table describes the parameters that you can configure when you resize images.

- Resize an image based on the specified height and width

| Parameter	Required	Description	Valid value
m	Yes	Specifies the type of the resize operation. Default value: lfit.	○ lfit: OSS resizes the source image proportionally as large as possible within a rectangle based on the specified width and height. ○ mfit: OSS resizes the source image proportionally as small as possible beyond a rectangle based on the specified width and height. ○ fill: OSS resizes the source image proportionally as small as possible beyond a rectangle, and then crops the resized image from the center based on the specified width and height. ○ pad: OSS resizes the source image as large as possible within a rectangle based on the specified width and height, and fills the empty space by using the specified color. ○ fixed: OSS forcibly resizes the source image based on the specified width and height. For more information, see the examples below the table. ⑦ **Note** When you set the parameter to lfit or mfit to proportionally resize an image, the ratio of w/h of the source image is rounded to numbers if the ratio is a decimal.
w	Yes	Specifies the width to which the image is to be resized.	[1,4096]
h	Yes	Specifies the height to which the image is to be resized.	[1,4096]
l	Yes	Specifies the length of the longer side to which the image is to be resized. ⑦ **Note** Longer side refers to the side with the larger ratio of the source length to the requested length. Shorter side refers to the side with the smaller ratio of the source length to the requested length. For example, if a source image is resized from 400 × 200 pixels to 800 × 100 pixels, the source-to-requested ratios are 0.5 (400/800) and 2 (200/100). Therefore, 0.5 is smaller than 2, the 200-pixel side is the longer side, and the 400-pixel side the shorter side.	[1,4096]
s	Yes	Specifies the length of the shorter side to which the image is to be resized.	[1,4096]
limit	No	Specifies whether to perform the resize operation when the resized image is larger than the source image.	0 and 1. Default value: 1. ○ 1: OSS returns the source image without resizing the image. ○ 0: OSS resizes the source image based on the specified parameter.
color	Yes (only when the value of `m` is pad)	When you set the resize type to pad, you can select a color to fill the empty space.	RGB color values. For example, 000000 indicates black, and FFFFFF indicates white. Default value: FFFFFF (white).

Example: The size of the source image is 200 × 100 pixels. The w parameter is set to 150 pixels, and the h parameter is set to 80 pixels. The source image is resized to different sizes when you specify different resize types.

- lfit
  - Proportional resizing: The value of w/h of the source image must be equal to that of the resized image. Therefore, if w is 150 pixels, h is 75 pixels. If h is 80 pixels, w is 160 pixels.
  - Maximum image within a rectangle based on the specified width and height: The value of w*h of the resized image cannot exceed 150 × 80 pixels.

 The size of the resized image is 150 × 75 pixels based on the preceding conditions.


lfit
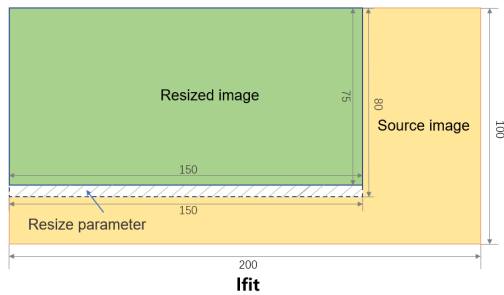
- mfit
  - Proportional resizing: The value of w/h of the source image must be equal to that of the resized image. Therefore, if w is 150 pixels, h is 75 pixels. If h is 80 pixels, w is 160 pixels.
  - Minimum image beyond a rectangle based on the specified width and height: The resized image must be a minimum rectangle whose size is greater than 150 × 80 pixels.

  The size of the resized image is 160 × 80 pixels based on the preceding conditions.

mfit
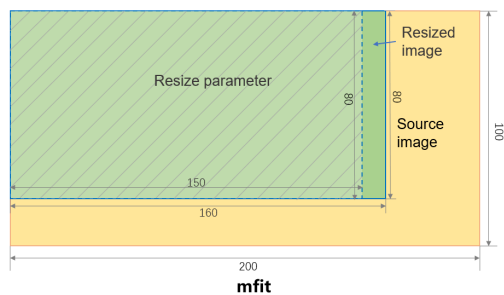
- fill

  The fill parameter is used to resize the source image proportionally as small as possible beyond a rectangle and crop the resized image based on the specified width and height. The source image is resized to 160 × 80 pixels, and w is cropped to 150 pixels from the center to obtain a resized image of 150 × 80 pixels.
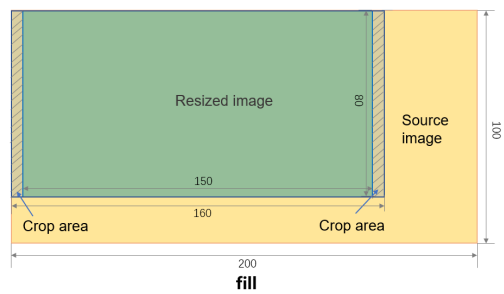

fill

- pad

  The pad parameter is used to resize the source image as large as possible within a rectangle and fill the empty space based on the specified width and height. The source image is resized to 150 × 75 pixels, and h is centered and filled to 80 pixels to obtain a resized image of 150 × 80 pixels.

pad
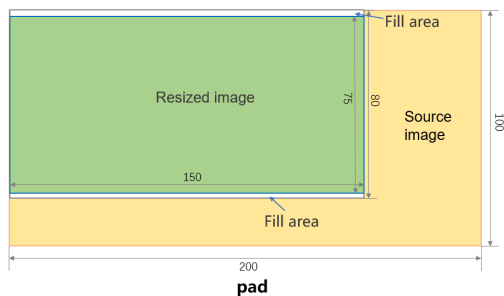
- fixed

  The fixed parameter is used to resize the image based on the specified width and height. If the width and height ratio of the image is different from that of the source image, the image is deformed.

  fixed
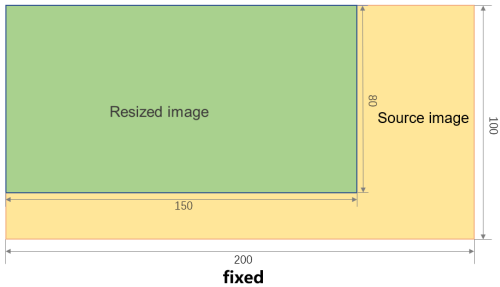
- Resize an image proportionally

Parameter	Required	Description	Valid value
p	Yes	OSS resizes an image by percentage.	[1,1000] A percentage smaller than 100 indicates that the image is scaled down. A percentage greater than 100 indicates that the image is scaled up.

Usage notes

- Limits on source images
  - Only JPG, PNG, BMP, GIF, WebP, and TIFF images are supported. GIF images can be resized based on the specified width and height but cannot be resized proportionally. GIF images become static images when you resize the images proportionally.
  - The size of the source image cannot exceed 20 MB.
  - The width or height of the source image cannot exceed 30,000 pixels. The total pixel number of the source image cannot exceed 250 million.

    The total pixel number of a dynamic image, such as a GIF image, is calculated by using the following formula: `Width × Height × Number of image frames`. The total pixel number of a static image, such as a PNG image, is calculated by using the following formula: `Width × Height`.

- Limits on resized images

  The width or height of a resized image cannot exceed 16,384 pixels. The total pixel number of the resized image cannot exceed 16,777,216.

- If the width or height of the resized image is specified:
  - The source image is resized proportionally when proportional resizing is performed. For example, if you resize the height of a source image from 200 × 100 pixels to 100 pixels, the width of the source image is resized to 50 pixels.
  - The source image is resized based on the specified width and height. For example, if you resize the height of a source image from 200 × 100 pixels to 100 pixels, the width of the source image is resized to 100 pixels.

- If you set the m parameter to mfit and specify a value for *l* or *s*, the longer side or the shorter side of the image is scaled based on the specified value of *l* or *s*.

- After you set the *m* parameter, the specified values of *l* and *s* do not take effect if you specify a value for *w* or *h*.

- By default, if the size of the resized image is larger than that of the source image, the source image is returned. You can add the `limit_0` parameter to enlarge the image. The URL used to enlarge the image is in the following format: `https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_500,limit_0`.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

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

- Resize the image proportionally

- Based on the width or height

  Configure parameters to resize the image:

  - Resize the source image to a height of 100 pixels: `resize,h_100`
  - Set the resize type to lfit: `m_lfit`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,h_100,m_lfit

- Based on the longer side

  Resize the source image based on the longer side of 100 pixels: `resize,l_100`

  The following URL is used to process the image: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,l_100

- Resize the image based on the specified width and height

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to fixed: `m_fixed`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fixed,h_100,w_100

- Crop the image based on the specified width and height

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to fill: `m_fill`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_fill,h_100,w_100

- Resize the source image based on the specified width and height and fill the empty space

  Configure parameters to resize the image:

  - Resize the source image to a width and a height of 100 pixels: `resize,h_100,w_100`
  - Set the resize type to pad: `m_pad`
  - Fill the empty space in red: `color_FF0000`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,m_pad,h_100,w_100,color_FF0000
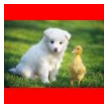
- Resize an image proportionally

  Configure parameters to resize the image:

  Resize the source image by 50%: `resize,p_50`

  The following URL is used to process the image: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,p_50

FAQ

How do I access a resized image if the access control list (ACL) of the image is private?

To access the image, you must add signature information to the URL of the resized image. For more information about how to obtain the URL of an object, see How do I obtain the URL of an uploaded object?.

1.3.2. Add watermarks

You can add text or image watermarks to an image stored in Object Storage Service (OSS) by using Image Processing (IMG) parameters. This topic describes and provides examples on how to use parameters to add watermarks to an image.

Usage notes

- Only images stored in the current bucket can be used as watermarks. To use online or local images as watermarks, you must first upload the images to the current bucket.
- Only JPG, PNG, BMP, WebP, and TIFF images can be used as watermarks.
- You can add up to three different image watermarks to a single image, and the positions of each image watermark cannot be completely overlapped.
- Traditional Chinese characters cannot be used as text watermarks.

Parameters

Operation: watermark

The following table lists the parameters that you can configure when you add watermarks to images.

- Basic parameters

Parameter	Required	Description	Valid value
t	No	The opacity of the watermark.	[0,100] Default value: 100. A value of 100 indicates that the watermark is opaque.
g	No	The position of the watermark on the image.	○ nw: upper left ○ north: upper middle ○ ne: upper right ○ west: middle left ○ center: center ○ east: middle right ○ sw: lower left ○ south: lower middle ○ se: lower right For the precise position that each value indicates, see the figure in the following section.
x	No	The horizontal margin is the horizontal distance between the watermark and the image edge. This parameter takes effect only when the watermark is on the upper left, middle left, lower left, upper right, middle right, or lower right of the image.	[0,4096] Default value: 10 Unit: px
y	No	The vertical margin is the vertical distance between the watermark and the image edge. This parameter takes effect only when the watermark is on the upper left, upper middle, upper right, lower left, lower middle, or lower right of the image.	[0,4096] Default value: 10 Unit: px
voffset	No	The vertical offset from the middle line. 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 middle line.	[-1000,1000] Default value: 0. Unit: px

You can use the horizontal margin, vertical margin, and vertical offset from the middle line to adjust the position of a watermark on an image. You can also use these parameters to adjust the watermark layout when the image has multiple watermarks.

The following figure shows the positions of watermarks based on coordinates.

nw	north	ne
west	center	east
sw	south	se

- Image watermark parameters

Parameter	Required	Description	Valid value

Parameter	Required	Description	Valid value
image	Yes	The complete name of the image object that you want to use as a watermark. The object name must be Base64-encoded. For more information, see Encode watermarks. For example, if you want to use an image object named *panda.png* in the *image* directory of the current bucket as a watermark, the object name to encode is *image/panda.png* and the encoded object name is `aW1hZ2UvcGFuZGEucG5n`. ⑦ **Note** Only objects stored in the current bucket can be used as watermarks.	Base64-encoded strings.

- Parameters for watermark image preprocessing

  You can preprocess a watermark image by calling the Resize images, Custom crop, Indexed cut, Rounded rectangle, and Rotate operations. In addition, the P parameter is supported when you resize a watermark image.

Parameter	Description	Valid value
P	The size of the watermark image relative to the source image. The value of this parameter specifies the size of the watermark as a percentage of the source image. For example, if you set this parameter to 10 for a source image of 100 × 100 pixels, the size of the watermark image is 10 × 10 pixels. If the source image is 200 × 200 pixels, the size of the watermark image is 20 × 20 pixels.	[1,100]

- Text watermark parameters

Parameter	Required	Description	Valid value
text	Yes	The content of the text watermark. The text content must be Base64-encoded. For more information, see Encode watermarks.	Base64-encoded strings. The string can be up to 64 characters in length.
type	No	The font of the text watermark. The font name must be Base64-encoded.	For more information about the supported fonts and the encoded strings for the fonts, see Font types and encoded strings. Default value: wqy-zenhei (encoded value: d3F5LXplbmhlaQ)
color	No	The color of the text watermark. The valid values for this parameter are RGB color values.	RGB color values. For example, 000000 indicates black, and FFFFFF indicates white. Default value: 000000. A value of 000000 indicates that the color of the text is black.
size	No	The size of the text watermark.	(0,1000] Default value: 40. Unit: px
shadow	No	The opacity of the shadow for the text watermark.	[0,100] Default value: 0. A value of 0 indicates no shadows are added to the text.
rotate	No	The degree by which the text is rotated clockwise.	[0,360] Default value: 0. A value of 0 indicates that the text is not rotated.
fill	No	Specifies whether to tile the source image with the text watermarks.	0 and 1. Default value: 0. ○ *1*: The source image is tiled with the text watermarks. ○ *0*: The source image is not tiled with the text watermarks.

The following table describes the valid values of the type parameter and the encoded strings of these values.

Parameter value	Font name	Encoded value
wqy-zenhei	WenQuanYi Zen Hei	d3F5LXplbmhlaQ
wqy-microhei	WenQuanYi Micro Hei	d3F5LW1pY3JvaGVp
fangzhengshusong	Fangzheng Shusong	ZmFuZ3poZW5nc2h1c29uZw
fangzhengkaiti	Fangzheng Kaiti	ZmFuZ3poZW5na2FpdGk
fangzhengheiti	Fangzheng Heiti	ZmFuZ3poZW5naGVpdGk

Parameter value	Font name	Encoded value
fangzhengfangsong	Fangzheng Fangsong	ZmFuZ3poZW5nZmFuZ3Nvbmc
droidsansfallback	DroidSansFallback	ZHJvaWRzYW5zZmFsbGJhY2s

- Text-and-image watermark parameters

Parameter	Required	Description	Valid value
order	No	The order of the text watermark and image watermark.	0 and 1. Default value: 0. ○ 0: The image watermark is on the top of the text watermark. ○ 1: The text watermark is on the top of the image watermark.
align	No	The alignment of the text watermark and image watermark.	0, 1, and 2. Default value: 2. ○ 0: The text watermark and the image watermark are aligned based on top alignment. ○ 1: The text watermark and the image watermark are aligned based on center alignment. ○ 2: The text watermark and the image watermark are aligned based on bottom alignment.
interval	No	The spacing between the text watermark and image watermark.	[0,1000] Default value: 0. Unit: px

Encode watermarks

The content, color, and font of a text watermark and the image name of an image watermark must be a URL-safe string that is Base64-encoded. Perform the following steps to encode watermarks:

1. Encode the content by using Base64.
2. Replace part of the encoded string.
   - Replace the plus signs (+) in the encoded string with hyphens (-).
   - Replace the forward slashes (/) in the encoded string with underscores (_).
   - Omit the equal signs (=) that are at the end of the encoded string.

We recommend that you use URL-safe Baes64 encoding tools to encode the content, color, and font of a text watermark and the image name of an image watermark.

🔊 **Notice** Encoded strings can be used only as parameters in specific watermark operations. Do not use encoded strings in signature strings.

Example 1: Add a text watermark to an image.

In the following examples, the source image named example.jpg is stored in a bucket named image-demo that is located in the China (Zhangjiakou) region. The URL used to access the image is https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg.

The following examples show how to add a text watermark to example.jpg:

- Add the string "Hello World" to the image as a text watermark

Base64-encode the string "Hello World" and convert the encoded result to a URL-safe string. For more information, see Encode watermarks. The encoded URL-safe string is `SGVsbG8gV29ybGQ` . Therefore, you can use the following URL to add the text watermark to example.jpg: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,text_SGVsbG8gV29ybGQ.

- Configure multiple IMG parameters when you add a text watermark to the image

  In this example, IMG parameters are configured to perform the following operations on the source image and text watermark "Hello World" that you want to add to the source image:

  - Resize the source image *example.jpg* to 300 × 300 pixels. IMG parameter: `resize,w_300,h_300`
  - Set the text font of the text watermark to WenQuanYi Zen Hei. IMG parameter: `type_d3F5LXplbmhlaQ` (d3F5LXplbmhlaQ is the Base64-encoded value for WenQuanYi Zen Hei.)
  - Add the string "Hello World" to the source image as a text watermark. IMG parameter: `text_SGVsbG8gV29ybGQ`
  - Set the color of the text watermark to white and the size of the text to 30 pixels. IMG parameter: `color_FFFFFF,size_30`
  - Set the opacity of the shadow of the text watermark to 50%. IMG parameter: `shadow_50`
  - Set the position of the text watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `g_se,x_10,y_10`

  You can use the following URL to configure multiple IMG parameters when you add the text watermark to the image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/watermark,type_d3F5LXplbmhlaQ,size_30,text_SGVsbG8gV29ybGQ,color_FFFFFF,shadow_50,t_100,g_se,x_10,y_10

Example 2: Add an image watermark to the source image.

The following examples show how to add image watermarks to the source image named example.jpg:

- Add an image named panda.png to the source image as an image watermark

  Base64-encode the image name panda.png and convert the encoded result to a URL-safe string. The encoded URL-safe string is `cGFuZGEucG5n` . Therefore, you can use the following URL to add panda.png to example.jpg as an image watermark: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGFuZGEucG5n.

- Configure multiple IMG parameters when you add an image watermark to the source image

  In this example, IMG parameters are configured to perform the following operations on the source image and the image named panda.png that you want to add to the source image as an image watermark:

  - Resize the source image example.jpg to 300 × 300 pixels. IMG parameter `resize,w_300,h_300`
  - Set the quality of the source image example.jpg to 90%. IMG parameter: `quality,q_90`
  - Add the image watermark *panda.png*. IMG parameter: `watermark,image_cGFuZGEucG5n` (cGFuZGEucG5n is the Base64-encoded value for panda.png.)
  - Set the opacity of the image watermark to 90%: `t_90`
  - Set the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `g_se,x_10,y_10`

You can use the following URL to configure multiple IMG parameters when you add the image watermark to the source image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300,h_300/quality,q_90/watermark,image_cGFuZGEucG5n,t_90,g_se,x_10,y_10

- Preprocess the image watermark and add it to the source image

In this example, IMG parameters are configured to perform the following operations on the source image and the image named panda.png that you want to add to the source image as an image watermark:

  ◦ Resize the width of the source image *example.jpg* to 300 pixels. IMG parameter: `resize,w_300`
  ◦ Resize the image watermark *panda.png* to 30% of the original size. IMG parameter: `image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA` (`cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA` is the Base64-encoded value for `panda.png?x-oss-process=image/resize,P_30` .)
  ◦ Set the opacity of the image watermark to 90%, the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. IMG parameter: `t_90,g_se,x_10,y_10`

You can use the following URL to configure multiple IMG parameters when you add the image watermark to the source image: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_300/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,t_90,g_se,x_10,y_10

Example 3: Add text and image watermarks to the source image.

The following examples show how to add text and image watermarks to the source image named example.jpg:

- Add an image named panda.png to example.jpg as an image watermark and a string "Hello World" to example.jpg as a text watermark

Based on the encoded results of the image name panda.png and the string "Hello World", you can use the following URL to add panda.png and "Hello World" to example.jpg as an image watermark and a text watermark: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGFuZGEucG5nP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA,text_SGVsbG8gV29ybGQ.

- Add multiple image and text watermarks to example.jpg

In this example, two text watermarks (Watermark 1 and Watermark 2) and three image watermarks (Jellyfish.jpg, Koala.jpg, and Tulips.jpg) are added to the source image example.jpg. When you add multiple watermarks to the source image, use forward slashes (/) to separate the operations performed to configure each watermark.

  ◦ Add the image Jellyfish.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to upper left, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_nw,x_10,y_10` . `cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA` is the Base64-encoded value of the image name Jellyfish.jpg.
  ◦ Add the image Koala.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to lower right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF8yMA,g_se,x_10,y_10` . `cGljcy9Lb2FsYS5qcGc_eC1vc3MtcHJvY2Vzcz1pbWFnZS9yZXNpemUsUF8yMA` is the Base64-encoded value of the image name Koala.jpg.
  ◦ Add the image Tulips.jpg to the source image as an image watermark. Resize the image watermark to 20% of the original size. Set the position of the image watermark to middle left, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 10 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,image_cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_west,x_10,y_10` . `cGljcy9UdWxpcHMuanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA` is the Base64-encoded value of the image name Tulips.jpg.
  ◦ Add the text watermark "Watermark 1" to the source image. Set the size of the text to 20 pixels. Set the position of the text watermark to upper right, the horizontal margin to 10 pixels, and the vertical offset from the middle line to 200 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,text_V2F0ZXJtYXJrIDE,g_ne,size_20,x_10,y_200` . `V2F0ZXJtYXJrIDE` is the Base64-encoded value of the string "Watermark 1".

- Add the text watermark "Watermark 2" to the source image. Set the size of the text to 20 pixels and the color of the text to dark blue. Set the position of the text watermark to lower left, the horizontal margin to 100 pixels, and the vertical offset from the middle line to 50 pixels. You can configure the following IMG parameters to perform the preceding operations: `watermark,text_V2F0ZXJtYXJrIDI,color_0000b7,size_20,g_sw,x_100,y_50`. `V2F0ZXJtYXJrIDI` is the Base64-encoded value of the string "Watermark 2".

Based on the preceding IMG parameters, you can use the following URL to add two text watermarks (Watermark 1 and Watermark 2) and three image watermarks (Jellyfish.jpg, Koala.jpg, and Tulips.jpg) to the source image example.jpg: https://image-demo-oss-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/example.jpg?x-oss-process=image/watermark,image_cGljcy9KZWxseWZpc2guanBnP3gtb3NzLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfMjA,g_nw,x_10,y_10/watermark,image_cGljcy9Lb2FsY

FAQ

How do I use online or local images as watermark images?

When IMG is used to add image watermarks to a source image, make sure that the watermark images and the source image are stored in the same bucket. To use online or local images to a source image as watermarks, you must first upload the images to the bucket in which the source image is stored.

1.3.3. Custom crop

You can use custom crop parameters to crop a rectangular image based on a specified size from a source image stored in OSS. This topic describes the parameters and examples to crop an image based on a specified dimension.

Parameters

Operation name: crop

The following table lists the parameters.

Parameter	Description	Valid value
w	The width of the cropped area.	[0, image width]. Default value: the maximum value.
h	The height of the cropped area.	[0, image height]. Default value: the maximum value.
x	The abscissa of the starting point. By default, the origin is located in the upper-left corner.	[0, image bound]
y	The ordinate of the starting point. By default, the origin is located in the upper-left corner.	[0, image bound]
g	The location of the origin for cropping. The origin is located in the upper-left corner of any of the nine-cell matrix.	• nw • north • ne • west • center • east • sw • south • se The following schematic view shows the available locations for the origin.

The following schematic view shows the available locations for the origin.

nw	north	ne
west	center	east
sw	south	se

Usage notes

Before you crop an image, take note of the following items:

- If the specified starting abscissa or ordinate values exceed those of the source image, the system returns `BadRequest` and the following error message: Advance cut's position is out of image.

- If the width and height specified from the starting point exceed those of the source image, the source image is cropped to its boundaries.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

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

- Crop an image from the starting point (100, 50) to the bounds

  Requirements and parameters:

  - From the starting point (100, 50): `crop,x_100,y_50`

  - To the bounds: By default, the maximum values of w and h are used for cropping. Therefore, the w and h parameters can be omitted.

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50

- Crop an area of 100 × 100 pixels from the starting point (100, 50)

  Requirements and parameters:

  - From the starting point (100, 50): `crop,x_100,y_50`

  - An area of 100 × 100 pixels: `w_100,h_100`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_100,y_50,w_100,h_100

- Crop an area of 200 × 200 pixels in the lower-right corner of the source image

  Requirements and parameters:

  - From the starting point in the lower-right corner of the source image: `crop,g_se`

  - An area of 200 × 200 pixels: `w_200,h_200`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,w_200,h_200,g_se

- Crop an area of 200 × 200 pixels in the lower-right corner of an image and stretch the cropped area downward by (10, 10)

  Requirements and parameters:

  - From the starting point in the lower-right corner of an image and stretch the cropped area downward by (10, 10): `crop,g_se,x_10,y_10`

  - An area of 200 × 200 pixels: `w_200,h_200`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,x_10,y_10,w_200,h_200,g_se

1.3.4. Adjust image quality

The quality adjustment operation uses the format of a source image to compress the image. You can use the quality adjustment parameters to modify the quality of source images stored in Object Storage Service (OSS). This topic describes the parameters and examples for image quality adjustment.

Quality adjustment applies only to JPG and WebP images.

Parameters

Operation name: quality

The following table describes the parameters that you can configure when you adjust the quality of an image.

Parameter	Description	Valid value
q	Specifies the relative quality of the image and compresses the source image based on percentage. If the source image quality is 100%, you can obtain an image whose quality value is 90% after you add the `quality,q_90` parameter. If the quality value of the source image is 80%, you can obtain an image whose quality value is 72% after you add the `quality,q_90` parameter. ⑦ **Note** The q parameter applies only to source images in the JPG format to specify the relative quality of the images. If a source image is in the WebP format, this parameter works the same as Q. The absolute quality is specified for the image.	[1,100]
Q	Specifies the absolute quality of the image and compresses the source image based on Q%. If the quality value of the source image is smaller than the specified Q value, the quality value of the compressed image is the quality value of the source image. For example, if the quality value of the source image is 95%, you can obtain an image that has a quality value of 90% after you add `quality,Q_90`. If the quality value of the source image is 80%, you can obtain an image that has a quality value of 80% after you add `quality,Q_90`. ⑦ **Note** The Q parameter applies only to JPG and WebP images.	[1,100]

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

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

- Adjust the relative quality of an image

  Configure the parameters based on the following requirements:

  - Resize the image to a width of 100 pixels: `resize,w_100`
  - Set the relative quality value of the image to 80%: `quality,q_80`

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/quality,q_80

- Adjust the absolute quality of an image

  Configure the parameters based on the following requirements:

  ○ Resize the image to a width of 100 pixels: `resize,w_100`

  ○ Set the absolute quality value of the image to 80%: `quality,Q_80`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/quality,Q_80

1.3.5. Format conversion

You can use format conversion parameters to convert the format of a source image stored in OSS. This topic describes the parameters and examples to convert the format of an image.

Parameters

Operation name: format

The following table lists the parameters.

Valid value	Description
jpg	Saves the source image in the JPG format. By default, OSS fills the transparent area in white if the source image is in the PNG, WebP, or BMP format that supports alpha channels.
png	Saves the source image in the PNG format.
webp	Saves the source image in the WebP format.
bmp	Saves the source image in the BMP format.
gif	Saves the source image in the GIF format. If the source image is not in the GIF format, it is saved in the format of the source image.
tiff	Saves the source image in the TIFF format.

Usage notes

- When you perform a standard resize operation on an image, we recommend that you add the format parameter to the end of the last Image Processing (IMG) parameter.

  Example: `image/resize,w_100/format,jpg`

- When you resize and watermark an image, we recommend that you add the format parameter to the end of the resize parameter.

  Example: `image/reisze,w_100/format,jpg/watermark,...`

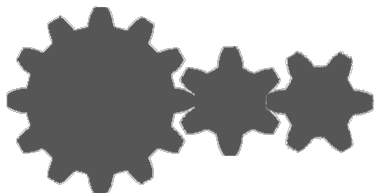### Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif

- Convert the format of the source image to PNG

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/format,png
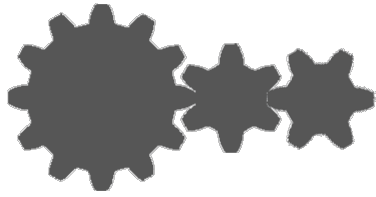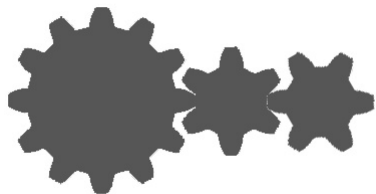

- Convert the format of the source image to JPG that supports gradual display

  Requirements and parameters:

  - Set the image to gradual display: `interlace,1`
  - Convert the format of the image to JPG: `format,jpg`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/interlace,1/format,jpg

- Resize the image to a height of 200 pixels and convert the format of the image to WebP

  Requirements and parameters:

  - Resize the image to a height of 200 pixels: `resize,w_200`
  - Convert the format of the image to WebP: `format,webp`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.gif?x-oss-process=image/resize,w_200/format,webp

1.3.6. Query the EXIF data of an image

Some images may contain Exchangeable Image File Format (EXIF) data that includes the attribute information and photographic information of the images. If you want to obtain the EXIF data of an image, add the info parameter to the URL of the image.

> ⓘ **Note** The EXIF data of an image includes the image information such as the compression ratio, orientation, horizontal resolution, and vertical resolution. For more information about EXIF data, see EXIF2.31.

Parameters

Operation name: info

The image information is returned in the JSON format.

Examples

- Query an image that does not contain EXIF data

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

  If you add the info parameter to the URL of an image that does not contain EXIF data, only the basic information about the image is returned, such as the size, format, height, and width of the image.

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

- Query an image that contains EXIF data

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

  If you add the info parameter to the URL of an image that contains EXIF data, the basic information about the image and the EXIF data of the image are returned.

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

1.3.7. Auto-rotate

You can use the auto-orient parameter to specify whether source images stored in Object Storage Service (OSS) are rotated based on auto-rotate configurations. This topic describes the parameters and examples to rotate images when you configure auto-rotate.

Parameters

Operation name: auto-orient

The following table describes the parameters that you can configure when you configure auto-rotate.

Parameter	Description	Example
[value]	Specifies whether to perform auto-rotate.	0 and 1. Default value: 1. • 0: The orientation of the source image is retained. • 1: OSS performs auto-rotate on the image.

Usage notes

- If the source image does not have rotation parameters, the operation that you perform to set the auto-orient parameter does not affect the image.
- Most tools can be used to perform auto-rotate on images that have rotation parameters. Therefore, the images you view may be automatically rotated.
- Images that are processed by using auto-orient are re-compressed, which results in size differences between the processed images and the source images.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg

- Resize the image and retain the orientation

  Configure parameters to resize the image:

  - Resize the image to a width of 100 pixels: `resize,w_100`

  - Disable auto-rotate: `auto-orient,0`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,0

- Resize and automatically rotate the image

  Configure parameters to resize the image:

  - Resize the image to a width of 100 pixels: `resize,w_100`

  - Automatically rotate the image: `auto-orient,1`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/f.jpg?x-oss-process=image/resize,w_100/auto-orient,1

1.3.8. Incircle

You can use the incircle parameter to process an image stored in OSS as an incircle. This topic describes the parameters and examples to save an image in an ellipse.

Parameters

Operation name: circle

The following table lists the parameters.

Parameter	Description	Valid value
r	Specifies the radius of an incircle.	[1,4096]

Usage notes

- If the final format of the image is PNG, WebP, or BMP that supports alpha channels, the areas of the image outside the ellipse become transparent. If the final format of the image is JPG, the areas of the image outside the ellipse become white. We recommend that you save the processed image in PNG.
- If the value of r is greater than half of the shortest side of the source image, the incircle is returned based on the value of the half of the shortest side of the source image. Value of r = The shortest side of the source image/2.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

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

- Crop an image that has a crop radius of 100 pixels. If the image is saved in the JPG format, the areas of the image outside the ellipse become white.

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100

- Crop an image that has a crop radius of 100 pixels. If the image is saved in the PNG format, the areas of the image outside the ellipse become transparent.

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/circle,r_100/format,png

1.3.9. Indexed cut

You can use indexed cut parameters to cut a source image stored in OSS based on the specified size and retrieve the required image. This topic describes the parameters and examples to perform indexed cut.

Parameters

Operation name: indexcrop

The following table lists the parameters.

Parameter	Description	Valid value
x	The length of each image partition during horizontal cutting. One of the x and y parameters must be used.	[1, image width].
y	The length of each image partition during vertical cutting. One of the x and y parameters must be used.	[1, image height].
i	The image partition selected after cutting.	[0, maximum number of partitions). By default, the value is 0, which indicates the first partition.

Usage notes

- If the specified index exceeds that of the cut range, the system returns the source image.
- If both x and y are specified and their values are valid, the value of y takes effect.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

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

- Cut an image along the horizontal axis

    Requirements and parameters:

    - Cut the image by 100 pixels along the horizontal axis: `indexcrop,x_100`
    - Retrieve the first partition: `i_0`

    The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,x_100,i_0

- Cut an image along the vertical axis

    Requirements and parameters:

    - Cut the image by 100 pixels along the vertical axis: `indexcrop,y_100`
    - Retrieve the 11th partition: `i_10`

    The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/indexcrop,y_100,i_10

    The source image is returned because the maximum number of partitions exceeds that of the cut range.

1.3.10. Rounded rectangle

You can round the corners of a rectangle image stored in OSS by adding rounded-corners parameters. This topic describes the parameters used to round the corners of a rectangle image and provides examples on how to round the corners of a rectangle image.

Parameters

Operation name: rounded-corners

The following table describes the parameters you can configure.

Parameter	Description	Valid value
r	The radius at which the corners are rounded.	[1,4096]

Usage notes

- If the final format of an image is PNG, WebP, or BMP that supports alpha channels, the areas of the image outside the rounded rectangle become transparent. If the final format of the image is JPG, the areas of the image outside the rounded rectangle become white. We recommend that you save the processed image in the PNG format.
- If the specified radius for the rounded corners is greater than the radius of the largest incircle of the source image, the radius of the largest incircle of the source image is used as the radius to round the corners. In this case, the radius at which the corners are rounded is equal to half of the smallest edge of the source image.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

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

- Round the corners of the source image.

  Take the following steps to configure parameters:

  ○ Set the radius at which the corners are rounded to 30 pixels to perform the rounded-corners operation: `rounded-corners,r_30` .

  ○ Save the processed image in the PNG format: `format,jpg` . If the format of the source image is JPG, you can leave this parameter unspecified.

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rounded-corners,r_30

- Crop the source image before you round the image corners. Save the processed image in the PNG format.

  Take the following steps to configure parameters:

  ○ Crop the source image to 100 × 100 pixels from the default start position: `crop,w_100,h_100` .

  ○ Set the radius at which the corners are rounded to 10 pixels to perform the rounded-corners operation: `rounded-corners,r_10` .

  ○ Save the processed image in the PNG format: `format,png` .

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/crop,w_100,h_100/rounded-corners,r_10/format,png

1.3.11. Blur

You can use blur parameters to blur a source image stored in Object Storage Service (OSS). This topic describes the parameters and examples to blur an image.

Parameters

Operation name: blur

The following table describes the parameters that you can configure to blur an image.

Parameter	Required	Description	Valid value
r	Yes	Specifies the blur radius.	[1,50] A greater value indicates a blurrier image.
s	Yes	Specifies the standard deviation of a normal distribution.	[1,50] A greater value indicates a blurrier image.

Examples

An image in the bucket named image-demo in the China (Hangzhou) region is used in the following examples. The following URL is used to access the image over the Internet:

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

If you set the r parameter to 3 and the s parameter to 2, the URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/blur,r_3,s_2

1.3.12. Rotate

You can rotate an image stored in OSS clockwise by adding rotate parameters. This topic describes the parameters used to rotate an image and provides examples on how to rotate an image.

Parameters

Operation name: rotate

The following table describes the parameters you can configure.

Parameter	Description	Valid value
[value]	The degree by which the image is rotated clockwise.	[0,360] Default value: 0. A value of 0 indicates that the image is not rotated.

Usage notes

- If an image is not rotated by 90°, 180°, 270°, or 360°, the size of the processed image increases.
- An image you want to rotate cannot exceeds 4096 × 4096 pixels.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

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

- Rotate the source image by 90 degrees clockwise.

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rotate,90

- Rotate the source image by 270 degrees clockwise

  The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/rotate,70

1.3.13. Gradual display

You can use gradual display parameters to configure gradual display for source images stored in OSS. This topic describes the parameters and examples to configure gradual display.

When the network environment is poor or the image size is large, the image can be displayed in two ways on the web page:

- Standard display: loads and displays images row by row from top to bottom.
- Gradual display: displays the fuzzy outline of the image, and then loads the image gradually until the complete image is displayed.

The gradual display operation applies only to the source images in the JPG format. If the source image is not in the JPG format, you must add the `format,jpg` parameter to convert the format of the image to JPG.

Parameters

Operation name: interlace

The following table lists the parameters.

Parameter	Description	Valid value
[value]	Specifies whether to set the image to gradual display.	0 and 1 • 1: indicates that the source image is set to gradual display. • 0: indicates that the source image is set to standard display.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. The public endpoint of the image is `https://image-demo.oss-cn-hangzhou.aliyuncs.com` . The images used are example.jpg and panda.png in the root folder.

- Resize the image to a width of 200 pixels and set the image to gradual display

  Requirements and parameters:

  ○ Resize the image to a width of 200 pixels: `resize,w_200`
  ○ Set the image to gradual display: `interlace,1`

  The URL used to process the image is in the following format: http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_200/interlace,1

- Save the PNG image as a JPG image and set the image to gradual display

  Requirements and parameters:

  - Convert the format of the image to JPG: `format,jpg`
  - Set the image to gradual display: `interlace,1`

  The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/panda.png?x-oss-process=image/format,jpg/interlace,1

1.3.14. Query the average tone

This topic describes the parameters and examples to query the average tone of an image.

Parameters

Operation name: average-hue

The returned average tone information is in the following format: 0xRRGGBB. RR, GG, and BB use two hexadecimal digits. RR indicates red. GG indicates green. BB indicates blue.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

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

The URL used to query the average tone of the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/average-hue

The average color information returned in the browser is 0x5c783b. 0x5c783b indicates the RGB color value (92,120,59).

1.3.15. Brightness

You can adjust the brightness of an image stored in OSS by adding bright parameters. This topic describes the parameters used to adjust the brightness of an image and provides examples on how to adjust the brightness of an image.

Parameters

Operation name: bright

The following table describes the parameters you can configure.

Parameter	Description	Valid value
[value]	The percentage by which to adjust the image brightness.	[-100, 100] • A value smaller than 0 indicates that the brightness of the image is decreased. • A value of 0 indicates that the brightness of the image is not changed. • A value greater than 0 indicates that the brightness of the image is increased.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

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

- Increase the brightness of the image by 50 percent

    The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/bright,50

- Decrease the brightness of the image by 50 percent

    The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/bright,-50

1.3.16. Sharpen

You can sharpen an image stored in OSS by adding sharpen parameters. This topic describes the parameters used to sharpen an image and provides examples on how to sharpen an image.

Parameters

Operation name: sharpen

The following table describes the parameters you can configure.

Parameter	Description	Valid value
[value]	The degree of sharpness.	[50,399] A greater value indicates a clearer image. However, an overlarge value may result in image artifacts. We recommend that you set this parameter to 100 for optimal effects.

Examples

An image in the bucket named image-dem in the China (Hangzhou) region is used in this example. The following URL is used to access the image over the Internet:

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

Sharpen the source image. The degree of sharpness is set to 100. The following URL is used to process the image: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/sharpen,100

1.3.17. Contrast

Contrast refers to the measurement of different brightness levels between the brightest white and the darkest black of an image, that is, the grayscale contrast of an image. You can use contrast parameter to adjust the contrast of the source images stored in OSS. This topic describes the parameters and examples to adjust the contrast for an image.

Parameters

Operation name: contrast

Parameter	Description	Valid value
[value]	The contrast of the image.	[-100,100] ● A value smaller than 0: reduces the contrast. ● A value of 0: maintains the contrast. ● A value greater than 0: increases the contrast.

Examples

The image-demo bucket that is located in the China (Hangzhou) region is used as an example. Public endpoint of the image:

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

● Reduce the contrast by 50

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/contrast,-50

● Increase the contrast by 50

The URL used to process the image is in the following format: https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/contrast,50

1.4. Save processed images

By default, Image Processing (IMG) does not save processed images. You must add the saveas parameter to an IMG request to save a processed image as an object to a specified bucket.

Usage notes

- Permission

    To save a processed image, you must have the `oss:PostProcessTask` permission on the source bucket in which the source image is stored, the `oss:PutBucket` permission on the destination bucket in which you want to store the processed image, and the `oss:PutObject` permission on the object as which you want to store the processed image.

- Region

    You can save the processed image to the same bucket where the source image is stored or to a different bucket. However, the source bucket and the destination bucket must belong to the same Alibaba Cloud account and must be in the same region.

- Storage method

    Images that are processed by using object URLs cannot be directly saved to a specified bucket. You can save the processed images to your local device and then upload them to the specified bucket.

- ACL

    The access control list (ACL) of the processed image is the same as that of the bucket in which the image is saved and cannot be customized.

- Storage duration

    If you want to store the processed image for the specific duration, configure a lifecycle rule for the image object to specify the time when the object expires. For more information, see Lifecycle rules.

SDK

If you use Object Storage Service (OSS) SDKs to process images, you can use the ImgSaveAs operation to save the processed images to a specified bucket. The following code provides an example on how to use OSS SDK for Java to save a processed image to a specified bucket:

```
// Specify the endpoint of the region in which the specified bucket is located. In this example, the endpoint of the China (Hangzhou) region is used.

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS because the account has permissions on all API operations. We recommend that you use a RAM user to call API operations or perform routine operations and maintenance. To create a RAM user, log on to the RAM console.
String accessKeyId = "YourAccessKeyId";
String accessKeySecret = "YourAccessKeySecret";
// Specify the name of the bucket where the source image is stored.
String bucketName = "SourceBucketName";
// Specify the name of the source image. The source image name must be the full path of the source image that excludes the bucket name. Example: example/image.png.
String sourceImage = "SourceObjectName";
// Create an OSSClient instance.
    OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
try {
    // Resize the image to 100 × 100 pixels and save the processed image to the specified bucket.
    StringBuilder sbStyle = new StringBuilder();
    Formatter styleFormatter = new Formatter(sbStyle);
    String styleType = "image/resize,m_fixed,w_100,h_100";
    String targetImage = "TargetImageName";
    styleFormatter.format("%s|sys/saveas,o_%s,b_%s", styleType,
            BinaryUtil.toBase64String(targetImage.getBytes()),
            BinaryUtil.toBase64String(bucketName.getBytes()));
    System.out.println(sbStyle.toString());
    ProcessObjectRequest request = new ProcessObjectRequest(bucketName, sourceImage, sbStyle.toString());
    GenericResult processResult = ossClient.processObject(request);
    String json = IOUtils.readStreamAsString(processResult.getResponse().getContent(), "UTF-8");
    processResult.getResponse().getContent().close();
    System.out.println(json);
} catch (Exception e) {
    e.printStackTrace();
}
// Shut down the OSSClient instance.
ossClient.shutdown();
```

For more information about the naming conventions for buckets and objects, see Terms.

For more information about SDK demos for other programming languages, see the following topics:

- OSS SDK for Python
- OSS SDK for PHP
- OSS SDK for Go
- OSS SDK for C++
- OSS SDK for Android
- OSS SDK for iOS
- OSS SDK for Node.js
- OSS SDK for Browser.js

API

When you use the PostObject operation to call IMG, x-oss-process is passed by the request body. You can add saveas to the IMG request to save the processed image to a specified bucket.

You must specify the parameters described in the following table when you add saveas to the request.

Parameter	Description
o	The name of the object as which the processed image is stored. This parameter must be URL-safe Base64-encoded. For more information, see Encode watermarks.
b	The name of the bucket in which the processed image is stored. This parameter must be URL-safe Base64-encoded. By default, the processed image is saved to the current bucket if this parameter is not specified.

You can use the following two methods to process an image and save the processed image to the specified bucket:

- The following code provides an example on how to configure IMG parameters to process an image and save the processed image to a specified bucket:

```
POST /ObjectName?x-oss-process HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT****=
// Proportionally scale up the source image named test.jpg to a width of 100 pixels and save the processed image to a bucket named test.
x-oss-process=image/resize,w_100|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

- The following code provides an example on how to use an image style to process an image and save the processed image to a specified bucket:

```
POST /ObjectName?x-oss-process HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:22:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT****=
// Use an image style named examplestyle to process the source image named test.jpg and save the processed image to a bucket named test.
x-oss-process=style/examplestyle|sys/saveas,o_dGVzdC5qcGc,b_dGVzdA
```

1.5. Common errors

If an error occurs when you access Image Processing (IMG), IMG returns an error code and an error message. This helps you locate and fix the error.

Error response format

The following example shows an error response:

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

The error response contains the following elements:

- `Code` : the error code that IMG returns to the user.
- `Message` : the detailed error information provided by IMG.
- `RequestId` : the UUID used to identify an error request. If the problem persists, you can send this UUID to technical support to help locate the cause of the error.
- `HostId` : the ID used to identify the accessed IMG cluster.

Error codes

The following table describes the error codes contained in responses to IMG requests.

Error code	Description	Solution
InvalidArgument	The error message returned because the parameter is invalid.	HTTP 400 status code
BadRequest	The error message returned because a request error occurs.	
MissingArgument	The error message returned because a required parameter is not specified.	
ImageTooLarge	The error message returned because the image size exceeds the limit.	
WatermarkError	The error message returned because a watermark error occurs.	
NotImplemented	The error message returned because the access is denied.	
AccessDenied	The error message returned because access is denied.	HTTP 403 status code
SignatureDoesNotMatch	The error message returned because the signature calculated by OSS does not match the signature provided in the request.	

Error code	Description	Solution
NoSuchKey	The error message returned because the specified image does not exist.	HTTP 404 status code
NoSuchStyle	The error message returned because the specified style does not exist.	
InternalError	The error message returned because an internal error occurs.	HTTP 500 status code

SDK demos

- OSS SDK for Java
- OSS SDK for Python
- OSS SDK for PHP
- OSS SDK for Go
- OSS SDK for C++
- OSS SDK for C
- OSS SDK for .NET
- OSS SDK for Node.js
- OSS SDK for Browser.js
- OSS SDK for Android

1.6. FAQ

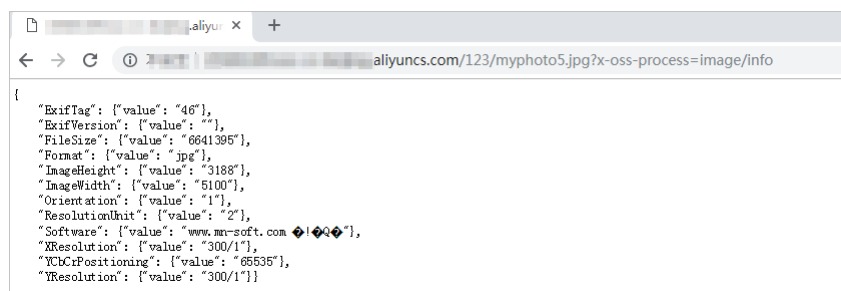This topic provides answers to frequently asked questions about OSS Image Processing (IMG).

If you encounter problems such as invalid parameter values, you can use the `? x-oss-process=image/info` parameter to check whether parameter values of the source image exceed thresholds. Each side of the image cannot exceed 4,096 pixels. The product of dimensions cannot exceed 4,096 pixels × 4,096 pixels.

What do I do if "Picture exceed the maximum allowable rotation range" is reported when I rotate an image?

Cause: The length of a single side of the source image exceeds 4,096 pixels, or the product of dimensions of four sides exceed 4,096 pixels × 4,096 pixels.

Troubleshooting:

1. Use the `? x-oss-process=image/info` parameter to obtain information about the image and determine whether the length of a single side or product of dimensions exceed limits.

```
{
  "ExifTag": {"value": "46"},
  "ExifVersion": {"value": ""},
  "FileSize": {"value": "6641395"},
  "Format": {"value": "jpg"},
  "ImageHeight": {"value": "3188"},
  "ImageWidth": {"value": "5100"},
  "Orientation": {"value": "1"},
  "ResolutionUnit": {"value": "2"},
  "Software": {"value": "www.mn-soft.com �!�Q��"},
  "XResolution": {"value": "300/1"},
  "YCbCrPositioning": {"value": "65535"},
  "YResolution": {"value": "300/1"}}
```

2. The `ImageWidth` value is 5100, which exceeds 4096.

Solution: Use the `auto-orient,0` parameter to disable automatic rotation, use the `resize` parameter to adjust the image size, and rotate the image. Example: *http://test.oss-cn-beijing.aliyuncs.com/123/myphoto5.jpg?x-oss-process=image/auto-orient,0/resize,m_lfit,h_2000,w_2000,limit_1/rotate,90*.

What do I do if the average tone queried by OSS does not match that of the image?

Cause: The average tone is not obtained by calculating the ratios of colors, but by the prominent colors of the image. Logic of calculating the average tone:

1. Calculate the average colors of the image.
2. Traverse each pixel. Calculate the difference between the pixel color and each average color. If the difference is greater than a threshold, the pixel is added to the list of prominent pixels.
3. Calculate the average value of the prominent pixels. The average value is the average tone.

Solution: Use the `? x-oss-process=image/average-hue` parameter to query average tone-related parameters for the OSS image.

What do I do if black outlines appear when I add a watermark to an image?
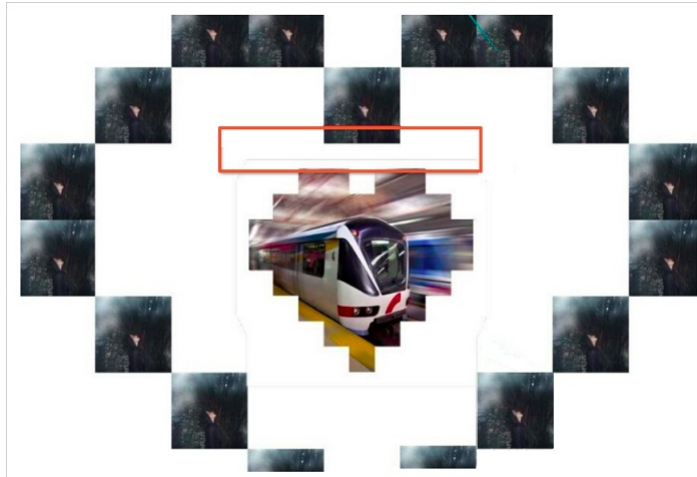
Cause: The black outlines are not caused by IMG. When you add a watermark to a source image, the source image is overlaid with the watermark. If the watermark and source image differ in RGB values, black outlines may appear after the source image is overlaid. This phenomenon is common when you use image processing tools.

Use the `? x-oss-process=image/average-hue` parameter to query the RGB parameters of the two images. You can add this parameter to the end of the source image URL to determine whether the RGB parameter values of the two images are the same.

Solution: The RGB parameter value of the source image is 0x0e0e0e. The RGB parameter value of the watermark is 0xffffff. When you add the watermark to the image, the black outline appears. You can use the `t` parameter to adjust the transparency and make the outline invisible. Valid values of t: 1 to 100. Example:

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,t_50.

What do I do if my image style fails to be accessed over CDN back-to-origin?

Use OSS endpoints to test whether the image style can be accessed. Use the following URLs to analyze the cause:

http://test.oss-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg@30-30bl

http://test.img-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg@30-30bl

- *img-cn-region.aliyuncs.com* indicates the OSS endpoint of the earlier version. OSS endpoints of the earlier version and the current version differ in the delimiter and syntax used for IMG.
- *oss-cn-region.aliyuncs.com* indicates the endpoint used after 2017. This endpoint does not support the syntax of IMG or the at sign (@) as those used in the earlier endpoint. To access an image whose endpoint contains img, synchronize the endpoint with the latest OSS endpoint.

The preceding endpoint of the earlier version uses Gaussian Blur-related parameters. After the earlier endpoint is synchronized with the latest OSS endpoint, you can access the image through the latest OSS endpoint.

http://test.oss-cn-beijing.aliyuncs.com/MomClass/ChuXin/3_2_336_462.jpg?x-oss-process=image/blur,r_3,s_30

What do I do if my image color is brightened after the image is resized?

Solution: Use tools such as Photoshop to query the color model of the source image. If the source image uses the RGB color model, the colors remain unchanged after the image is resized. If the source image uses the CMYK color model, the colors slightly change after the image is resized. The support for CMYK is in progress. Colors change after the image is resized.

What do I do if my image can be opened from my local computer, but the system shows that the image has been corrupted when I process the image?

Problem description: My image can be opened from my local computer, but the system shows that the image has been corrupted when I upload the image to OSS and process the image.

Troubleshooting:

1. Obtain the URL of the source image. Use `? x-oss-process=image/info` to view information of the source image. If the source image is unavailable, and an error is reported, the source image is corrupted.

2. To troubleshoot this problem, use the ImageMagick tool to adjust the image. If an error is reported, the image is corrupted. An example of resizing the image:

```
convert -resize 1024x768 1123331261_15353307414801n.jpg
```

The corrupted image can be opened from the local computer because the local image viewer has repaired the image. However, OSS does not repair the corrupted image. Therefore, the image cannot be displayed by the browser.

What do I do if an image stored in OSS is rotated 90 degrees?

Problem description: This image can be accessed through an OSS endpoint.
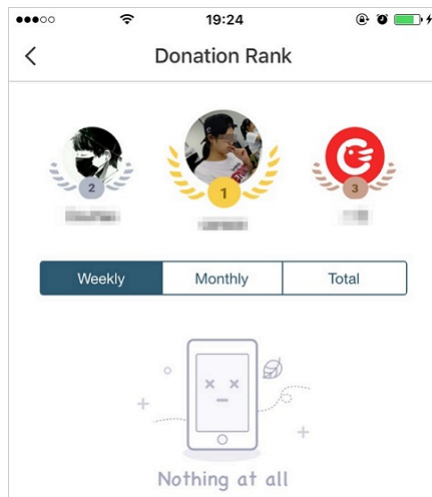
When the image is accessed over CDN, the image is rotated 90 degrees.
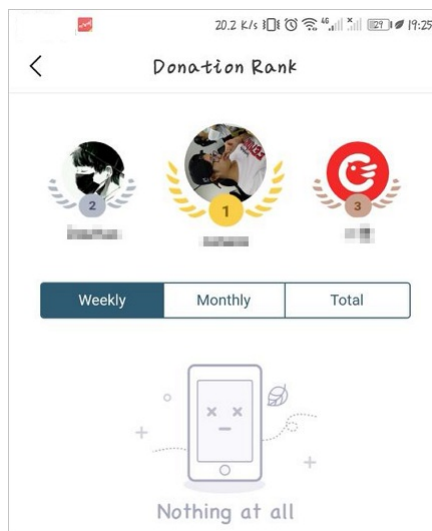
Cause: OSS works properly because the image can be accessed through an OSS endpoint. The possible cause lies in how the browser processes the image because the image is rotated when the image is accessed over CDN. Use the `? x-oss-process=image/info` parameter to query information about the source image. The `rotation 90` parameter is included.

Solution: Delete the rotation-related parameters.

What do I do if IMG does not take effect after CDN is used to access an image?

Troubleshooting: Check whether the **parameter filtering** feature is enabled for CDN. If this feature is enabled, parameters that follow the question mark (?) are removed from the URL when the image is accessed over CDN. For more information, see Configure parameter filtering.

What do I do if "Picture exceed the maximum allowable rotation range" is reported when I use an image?

Troubleshooting:

- Use the ImageMagick tool to check whether the automatic rotation feature is enabled for the source image.
- Use the `auto-orient,0` parameter to process the image. If the image can be processed, the source image has automatic rotation enabled.

If the source image has automatic rotation enabled, the image width and height cannot exceed 4,096 pixels.

What do I do if a blank image appears when I access an image configured with CDN from an iOS device through an URL that includes IMG parameters, but the image is accessible after I refresh the web page or when I access the image from a computer?

Cause: OSS works properly because the image can be accessed from the computer and mobile phone. Otherwise, the image cannot be accessed from a computer.

Troubleshooting:

1. Check whether access from a mobile phone to the image in OSS is normal.
   - If the image in OSS is accessible, whereas the access over CDN is abnormal, CDN node errors may cause loading failures or incorrect content is cached on CDN nodes.
   - If access to OSS and over CDN are both abnormal, but access over CDN becomes normal after the website is refreshed. One possible cause is that incorrect content is cached on CDN nodes.

2. Check whether access from a computer to the image in OSS is normal.
   - If the access is normal, the cause lies in the mobile phone.
   - If the access is abnormal, the cause lies in the image.

In this case, the image format is WebP. iOS does not support this format.

What do I do if I cannot open the source image and processed image that are stored in OSS?

Troubleshooting:

1. Download one of the images.

```
[root@edas02 aliyun-oss-php-sdk]# wget https://zh.mobi/test/123.jpg
--2018-11-2210:55:16--  https://zh.mobi/test/123.jpg
The IP address of the zh.mobi host is being resolved.
The HTTP request is sent. Waiting for the response. 200 OK
Image size: 4141232 (3.9M) [image/jpeg].
The image is saved as 123.jpg.
100%[============================================================================>] 4,141,232 12.5MB/s Time: 0.3s.
2018-11-2210:55:16 (12.5 MB/s) - 123.jpg [4141232/4141232]) saved.
```

2. Use ImageMagick to check the encoding methods used for the image.

```
[root@edas02 aliyun-oss-php-sdk]# identify 123.jpg
identify: Not a JPEG file: starts with 0x000x00 `123.jpg' @ error/jpeg.c/JPEGErrorHandler/316.
[root@edas02 aliyun-oss-php-sdk]#
```

If an encoding method error occurs, the cause does not lie in OSS. You can use this tool to analyze similar errors.

What do I do if a split line appears after I process my image?

Cause: No split lines exist, but a problem caused by pixel changes after the image is processed. The source image is an RGB (true color) image whose height is 2,560 pixels and width is 1,440 pixels. After the image is processed, the image height is cropped to 1,092 pixels, and the image width is cropped to 1,080 pixels. The image size in pixels is reduced, which causes abnormal display of the image.

Solution: Use the `quality,Q_100` parameter to increase the absolute quality value of the image to 100.

What do I do if "BadRequest" is reported when I use IMG?

```
<Error>
<Code>BadRequest</Code>
<Message>This image format is not supported.</Message>
<RequestId>5BA33754CBF4583BA2</RequestId>
<HostId>b.oss-cn-beijing.aliyuncs.com</HostId>
</Error>
```

Troubleshooting:

1. Use the convert command supported by ImageMagick to view the format of the source image.

2. Check whether the format of the source image is supported by IMG. For more information about image formats supported by OSS, see Usage notes.

Solution: Convert the format of the image to a format supported by IMG.

What do I do if "InvalidArgument" is reported when I use IMG?

```
<Code>InvalidArgument</Code>
<Message>The value: 0 of parameter: w is invalid.</Message>
<RequestId>5BA21FD8A642F41E6478</RequestId>
<HostId>luo.oss-cn-beijing.aliyuncs.com</HostId>
</Error>
```

Cause: Check the request parameters of the source image. Parameters similar to `20180899269957.jpg@0w_2e_1l_1an.src` are supported by the endpoints in the `img-cn-xx` format. After img-cn-xx is changed to `oss-cn-xxx`, endpoints in the oss-cn-xxx format does not support access that uses the request method of img-cn-xx-based endpoints. img-cn-xx-based endpoints do not support access over HTTPS.

Can OSS identify custom query parameters used for dynamically resizing an image in a URL?

No, this function is unavailable in OSS.

How do I display a text watermark in two lines and can I add multiple text watermarks to an image?

A text watermark cannot be displayed in multiple lines in OSS. However, you can add multiple text watermarks to an image. For more information, see Add watermarks.

1.7. 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_value`

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

- Old version API: `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

> ⑦ **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 synchronized 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 configurations for IMG domain names can be applied to OSS domains.

2.Video snapshots

This topic describes the parameters that you can configure to capture video snapshots and provides examples.

Usage notes

- When you capture video snapshots, you are charged for the number of captured images. For more information about billing, see Data processing fees.
- OSS can capture images from video objects only in the H.264 and H.265 formats.
- By default, OSS does not automatically store captured images. You must manually download the captured images to your local storage devices.

Parameters

Operation type: `video`

Operation name: snapshot

Parameter	Description	Valid value
t	The time when the image is to be captured.	[0, video duration] Unit: ms
w	The width based on which to capture the image. If this parameter is set to 0, the width based on which to capture the image is automatically calculated.	[0, video width] Unit: px
h	The height based on which to capture the image. If this parameter is set to 0, the height based on which to capture the image is automatically calculated. If w and h are set to 0, the width and height of the source image are used.	[0, video height] Unit: px
m	The mode used to capture the image. If this parameter is not specified, the image is captured in the default mode. In other words, the image at the specified point in time in the video is captured. If this parameter is set to fast, the most recent key image before the specified time is captured.	*fast*
f	The format of the captured image.	*jpg* and *png*
ar	Specifies whether to automatically rotate the image based on the video information. If this parameter is set to auto, the system automatically rotates the image based on the video information.	*auto*

Examples

- Use the fast mode to capture the image at the seventh second of the video. Export the captured image as a JPG image whose width is 800 pixels and height is 600 pixels.

  The URL of the processed image is in the following format: `<Video URL>?x-oss-process=video/snapshot,t_7000,f_jpg,w_800,h_600,m_fast`

- Capture the image at the fiftieth second of the video accurately. Export the captured image as a JPG image whose width is 800 pixels and height is 600 pixels.

  The URL of the processed the image is in the following format: `<Video URL>? x-oss-process=video/snapshot,t_50000,f_jpg,w_800,h_600`

Generate a signed URL for a captured snapshot

You can use OSS SDKs to generate a signed URL to capture a snapshot. The following code provides an example on how to use OSS SDK for Java to generate a signed URL to capture a snapshot:

```
// The endpoint of the China (Hangzhou) region is used in this example. Specify the actual endpoint.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you use the AccessKey pair of an Alibaba Cloud account to log on to OSS, because the account has permissions on all AP
I operations.  We recommend that you use your RAM user's credential to call API operations or perform routine operations and maintenance. To create a
RAM user, log on to the RAM console.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
// Specify the bucket in which the video object is stored.
String bucketName = "<yourBucketName>";
// Specify the full path of the video object. If the video object is not stored in the root directory of the bucket, the path must include the direct
ory in which the object is stored. Example: examplefolder/videotest.mp4.
String objectName = "<yourObjectName>";
// Create an OSSClient
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
// Capture the image at the fiftieth second of the video accurately. Export the captured image as a JPG image whose width is 800 pixels and height is
600 pixels.
String style = "video/snapshot,t_50000,f_jpg,w_800,h_600";
// Set the validity period of the URL to 10 minutes.
Date expiration = new Date(new Date().getTime() + 1000 * 60 * 10 );
GeneratePresignedUrlRequest req = new GeneratePresignedUrlRequest(bucketName, objectName, HttpMethod.GET);
req.setExpiration(expiration);
req.setProcess(style);
URL signedUrl = ossClient.generatePresignedUrl(req);
System.out.println(signedUrl);
// Shut down the OSSClient.
ossClient.shutdown();
```

The method used to generate signed URLs to capture snapshots is similar to that used to generated signed URLs to process images by using Image Processing (IMG). To use the following OSS SDKs to generate a signed URL to capture a snapshot, replace the IMG operations in the code with the snapshot operation.

- OSS SDK for Python
- OSS SDK for PHP
- OSS SDK for Go
- OSS SDK for C
- OSS SDK for C++
- OSS SDK for .NET
- OSS SDK for Android
- OSS SDK for iOS
- OSS SDK for Node.js
- OSS SDK for Browser.js