Image transformation

June 27, 2022 • Martina Farkasova • 20 min read

Image transformation is a real-time image manipulation and optimization feature of the Delivery API that speeds up image delivery. Because the Delivery API caches the results of image transformations, you can retrieve the transformed images even faster the next time you need them.

Overview

Input and output formats

To transform images from Kontent.ai, the **source images** need to be in one of the following image formats: jpeg, png, gif, and webp. You can also set the limitations in content types to only accept these adjustable image formats.

The optimized **output image** can be any of the following image formats: jpg, pjpg, png, png8, gif, and webp. The format parameter enables you to convert the image by choosing the output format.

By adding parameters to your image URL, you can enhance, resize, and crop images, compress them, or change their format for better performance. These operations can be applied programmatically in real-time without the need to batch process or maintain multiple copies of an image to match device breakpoints.

Name	Parameter	Notes
Size		
Width		Resize the width of the image.
Height		Resize the height of the image.
Device pixel ratio		Serve correctly sized images for devices that expose a device pixel ratio.
Resize fit mode		Set how the image will fit within the size bounds provided.
Source rectangle region		Select a sub-region of the source image to use for processing.
Focal point crop		
Focal point crop	, ,	Choose the point of interest of your image when cropping the image.
Background color		
Background color		Fill transparent areas of the image with the specified color.
Format		
Output format		Specify the output format to convert the image to.
Quality		Optimize the image to the given compression level for lossy file formatted images.
Lossless		Enable delivery of lossless images in formats that support lossless compression.
Automatic format selection		Automatic delivery of WebP format images.

Working with the image URLs

Image transformation works by applying the query parameters of the transformation methods to the absolute URLs of images. You can get the image URL directly from the asset details in the UI.

Transform images outside content items

The image doesn't need to be added to a content item to be used for transformation.

Once the image is added to a content item, the image URL can also be retrieved via the Delivery API. You can get the image URLs by either listing the items or retrieving a single content item containing the images.

Adjustable images can be used in the Asset elements and in the Rich text elements of the content items.

In Asset elements, the absolute URL for the image can be found in the property of the asset object returned in the JSON response.

JSON

In Rich Text elements, the absolute URL for the image can be found in the property of the object and in the formatted text of the property.

JSON

Not sure where to start?

- Check out our image optimization guide that explains the most common methods used for optimizing images.
- Try our hands-on tutorial where you'll go from setting the limitations in Kontent.ai to displaying the images on your website.

Size

You can use the size parameters to resize, crop, and change the aspect ratio of your images. In general, when using an invalid combination of parameters for the image, the Delivery API returns the original image without applying any changes. The maximum output dimensions are 8192 × 8192 pixels. All output images will be sized down to accommodate this limit.

Image width

Image width represents the width of the output image. You can use the parameter to dynamically resize the image based on pixels and percent values. If the width is represented by an integer greater than or equal to 1, the value is interpreted as a pixel width. If the width is a float greater than 0 but less than 1, it's interpreted as a percentage width. The maximum output width can be 8192 px. If both width and height parameters are omitted, the input image dimensions are used.

Value	Units	Description	Example
Integer	Pixels	A whole number between 1 and 12000.	changes image width to 300 px.
Float	Percentage	A floating-point number greater than 0 but less than 1.	changes the width to 10 % of the input image.

Upscaling is not possible

When using the width parameter with a value greater than the input image width, the Delivery API returns the original image without any changes to its width.

Image height

Image height represents the height of the output image. You can use the parameter to dynamically resize the image based on pixels and percent values. If the height is represented by an integer greater than or equal to 1, the value is interpreted as a pixel height. If the height is a float greater than 0 but less than 1, it's interpreted as a percentage height. The maximum output height can be 8192 px. If both width and height are omitted, the input image dimensions are used.

Value	Units	Description	Example
Integer	Pixels	A whole number between 1 and 12000.	changes image height to 300 px.
Float	Percentage	A floating-point number greater than 0 but less than 1.	changes image height to 10 % of the input image.

Upscaling is not possible

When using a height parameter greater than the input image height, the Delivery API returns the image with its original height value.

Device pixel ratio

Device pixel ratio is used to serve correctly sized images on different devices. The parameter value represents the ratio between physical and logical pixels. For example, when a mobile device has a device pixel ratio of 2, there are twice as many physical pixels on the screen.

Dependencies: You must specify width, height, or both for this parameter to work.

The maximum supported device pixel ratio value is 5. When the value is higher than 5, the device pixel ratio is calculated as if the value was equal to 5.

Value	Units	Description	Example
Float	DPR	A floating-point number greater than 0 but less than 5.	changes the width to 300 px for screens with a device pixel ratio of 2.

Resize fit mode

The parameter controls how the output image is constrained within the provided size values after resizing. Valid values are , , and . The default value is .

Dependencies: You must specify width ( ), height ( ), or both for this parameter to work. The resize fit mode set to crop ( ) is incompatible with the source rectangle region ( )

method.

There are differences in how the resize fit modes change the aspect ratio. The aspect ratio of an image describes the proportional relationship between its width and its height.

When using the default mode, the image is resized to match one of the constraining dimensions. For example, attempting to resize the image of 300 × 200 (aspect ratio of 3:2) to be 100 × 100 will result in an image that is 100 × 67 because the aspect ratio of the original image is maintained.

When using the fit mode, the image is scaled and distorted to fit dimensions. When resizing the image to wider than the original, the image is scaled horizontally to fit dimensions. Similarly, when resizing the image to taller than the original, the image is scaled vertically to fit dimensions.

The fit mode crops the excess image data while keeping the image centered. This means that the center of the output image corresponds with the center of the input image.

Crop works with pixels

With the resize fit mode set to crop ( ), you can only specify the width ( ) and height ( ) parameters in pixels, not percentages.

Fit mode	Description	Aspect ratio	Example
	Default mode. Resizes the image to fit within the width and height boundaries without cropping or distorting the image.	The resulting image will match one of the constraining dimensions, while the other dimension is altered to maintain the same aspect ratio as the input image.	tries to resize the image to 600 × 500 px with a fit mode of clip.
	Scales the image to fit the constraining dimensions exactly.	The resulting image will fill the dimensions, and will not maintain the aspect ratio of the input image.	scales the image to 1200 × 150 px, wider than the original.
	Resizes the image to fill the width and height dimensions and crops any excess image data.	The resulting image will match the width and height constraints without distorting the image.	crops the image to 300 px in width and 150 px in height.

Example: Resizing the original image to 300 × 100 using different resize fit modes.

Resize fit modes used to resize the image to 300x100.

Source rectangle region

By default, the entire image is used as the area for processing. The parameter selects a sub-region of the source image to use for processing.

Dependencies: The parameter is incompatible with the focal point crop ( ) method and resize fit mode set to crop ( ).

The value for is four relative values representing the starting coordinates of and and the width ( ) and height ( ), all separated by commas. All the values must be greater than or equal to 0.

If any of the values are represented by an integer greater than or equal to 1, they are interpreted in pixels. If the value is a float greater than 0 but less than 1, it's interpreted in percentages.

Value	Units	Description	Example
Integer	Pixels	A whole number between 0 and 8192.	selects a top-left region of the image for processing.
Float	Percentage	A floating-point number greater than 0 but less than 1.	selects a top-left region of the image for processing.

Example: Choosing a sub-region of the source image (604 × 403). Only the rectangle region of the image shown on the right will be used for processing.

Source rectangle region:

Focal point crop

You can use the focal point crop parameter to intentionally art-direct a point of interest when cropping an image. As the image is then resized and cropped, the focal point determines which areas are centered and within bounds of the image, and what gets cropped out.

Dependencies: The focal point crop parameter must always be used with the and parameters. Focal point crop is incompatible with the source rectangle region ( ) method.

Focal point crop consists of the horizontal ( ), vertical ( ), and zoom ( ) parameters.

To zoom in on an image, the value needs to be greater than 1, with representing the original size of the image. Every full step is then the equivalent of a 100% zoom, e.g., is the same as viewing the image at 200%.

Focal point crop consists of the following parameters:

- – The horizontal value of the focal point of an image. Must be a float between 0.0 and 1.0, inclusive. The default value is 0.5 – the center of the image.
- – The vertical value of the focal point of an image. Must be a float between 0.0 and 1.0, inclusive. The default value is 0.5 – the center of the image.
- – The zoom value of a focal point of an image. The default value is 1. The maximum value depends on the quality of the image, but we recommend using a float value between 1 and 100.

The top-left corner of an image is represented by and . When the and values are not set, the Delivery API uses the default value of 0.5 that represents the center of the image. For example, zooms-in on the center of the image by 300%.

Example: Choosing a focal point with the and values and then using a 200% zoom with the results in the image on the right.

Focal point crop:

Background color

You can use the background color ( ) parameter to fill in any transparent areas in your image with a color of your choice.

The bg parameter supports these color value formats:

- a 3-digit hexadecimal value (RGB)
- a 4-digit hexadecimal value (ARGB)
- a 6-digit hexadecimal value (RRGGBB)
- an 8-digit hexadecimal value (AARRGGBB)

"R", "G", and "B" represent red, green, and blue color values. The "A" in the 4- or 8-digit hex values represents the color's alpha transparency.

The 3- and 4-digit formats are shortened versions of the 6- and 8-digit formats. You can use them when the first and second (third and fourth, etc.) characters of the code are the same: can be shortened to while cannot.

For example, here's an image with a programmatically added light-blue semi-transparent background: .

Format

Convert an image to a specific format and set the output quality by choosing the compression strength.

Format parameter

You can use the format ( ) parameter to convert the source image from one format to another. Valid values for the parameter are , , , , , and . For example, when converting a jpg image to a png format, the request would look like this: .

The source image can be any of the following image formats: gif, png, jpeg, and webp.

GIF images aren't converted to WebP even when requested. When you use on a GIF image, you'll get the original GIF image.

Converting GIF to other formats works but if your GIF is animated, you'll lose the animation.

Value	Description	Quality parameter ( )
	Graphics Interchange Format	No
	Portable Network Graphics	No
	Portable Network Graphics palette variant with 8-bit transparency and 256 colors	No
	JPEG	Yes
	Progressive JPEG	Yes
	WebP	Yes

Quality parameter

The quality ( ) parameter enables control over the compression level for lossy file-formatted images (jpg, pjpg, webp). The value is an integer between 0 and 100. A lower number represents a lower-quality output image with smaller file size and vice-versa. For example, will lower the quality of the output image as well as its file size.

If no quality parameter is present, the output image will be returned with a default value of 85.

You can't set quality for lossless output

The quality parameter is ignored for all formats that are not lossy because lossless formats always retain the original image quality.

Similarly, the quality parameter is ignored for WebP output when the parameter is set to or .

Dependencies: Quality parameter only works with the JPG, PJPG, and WebP input image formats.

Lossless parameter

The parameter enables delivery of lossless images in the WebP format. Valid values are and or and . The default value is false, meaning a lossy compression. For example, converts the JPEG image to the WebP format using lossless compression.

When unset or set to an invalid value, lossless defaults to false.

When the parameter is set to or , the quality parameter ( ) is ignored.

Dependencies: Lossless parameter needs to be used with the parameter set to the WebP format.

Automatic format selection

Automatic delivery of WebP format images can be enabled by using the parameter. This overrides the format parameter in browsers that support WebP.

If is set and the browser does not support the WebP format, the Delivery API falls back to any format specified by the parameter, and finally the source image type. For example, the request is evaluated as follows: If the web browser supports the WebP format, the image is converted to webp. If the WebP format is not supported, the Delivery API tries to convert the image to png format. If this fails as well, the image stays in its original format which is jpg.