

Image Server REST API

Intro

Example image

This example image is used throughout the API documentation. It's located in the rbcom client (aka www.redbull.com), the source size is 1276x718 and the aspect ratio is approximately 1.777:1.

API documentation

Current API version: 2.1.

Be careful
The API is case-sensitive.

Get Image

GET IMAGE

`/images/0100/0/{path-segment-n}*/{image}`

Description

Return the requested image without any changes, except of default quality settings if possible.

Parameter

parameter	description
path-segment-n *	folder/path where image is located
image	filename of the image

Example

<http://image.redbull.com/rbcom/0100/0/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found and successefully send
400	malformed request, image data could not be supplied, image codec not supported
404	image could not be found

Scale Image

SCALE IMAGE MAXIMUM FIT

/images/0001/0/{width}/{height}/{path-segment-n}*/{image}

Be careful

The final image size is determined by the given ratio of the source image. For this reason the requested image size can be other than expected.

Description

The aspect ratio (aspect-ratio) of the (original) image is determined by width/height. The returned (target) image maximum fits into the box given by width and height with the same aspect ratio. That means either width or height is automatically adjusted to a smaller value (and never exceed the given one).

Parameter

parameter	description
width	(maximum) width of the image
height	(maximum)height of the image
path-segment-n *	folder/path where image is located
image	filename of the image

Examples

- request size 200x9999 (width = 200, height = 9999), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - possibility (A): adj.-height = width / aspect-ratio = 200px / 1.777 approx. 112.55px -> **113px**
 - possibility (B): adj.-width = height * aspect-ratio = 9999px * 1.777 approx. 17769.81px -> 17770px
 - (A) is valid: adj.height <= height
 - (B) is invalid due to constraint violation: adj.-width > width
 - the target image has size 200x113 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - consequently these parameter values ends up in the same image size: 200x1000, 200x500, 200x200
 - <http://image.redbull.com/rbcom/0001/0/200/9999/010/wiki/uci-wide,0.jpg>
- request size 300x168 (width = 300, height = 168), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - possibility (A): adj.-height = width / aspect-ratio = 300px / 1.777 approx. 168.81px -> 169px
 - possibility (B): adj.-width = height * aspect-ratio = 168px * 1.777 approx. 298.56px -> **299px**
 - (A) is invalid due to constraint violation: adj.height > height
 - (B) is valid: adj.-width <= width
 - the target image has size 299x168 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/0/300/168/010/wiki/uci-wide,0.jpg>
- request size 400x225 (width = 400, height = 225), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - possibility (A): adj.-height = width / aspect-ratio = 400px / 1.777 approx. 225.08px -> **225px**
 - possibility (B): adj.-width = height * aspect-ratio = 225px * 1.777 approx. 453.18px -> 453px
 - (A) is valid: adj.height <= height
 - (B) is invalid due to constraint violation: adj.-width > width
 - the target image has size 400x225 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/0/400/225/010/wiki/uci-wide,0.jpg>

- request size 300x169 (width = 300, height = 169), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - possibility (A): $\text{adj.-height} = \text{width} / \text{aspect-ratio} = 300\text{px} / 1.777 \text{ approx. } 168.81\text{px} \rightarrow \mathbf{169\text{px}}$
 - possibility (B): $\text{adj.-width} = \text{height} * \text{aspect-ratio} = 169\text{px} * 1.777 \text{ approx. } 300.34\text{px} \rightarrow \mathbf{300\text{px}}$
 - (A) is valid: $\text{adj.height} \leq \text{height}$
 - (B) is valid: $\text{adj.-width} \leq \text{width}$
 - the target image has size 300x169 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/0/300/169/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

SCALE IMAGE PROPORTIONATELY ON WIDTH

/images/0001/1/{width}/{path-segment-n}*/{image}

Description

Return the requested image where the width of the image is exactly the requested one. The height of the image is determined by the ratio of the original image.

Parameter

parameter	description
width	width of the image
path-segment-n*	folder/path where image is located
image	filename of the image

Examples

- request width 200px (width = 200), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - adj.-height = width / aspect-ratio = 200px / 1.777 approx. 112.54px -> **113px**
 - the target image has size 200x113 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/1/200/010/wiki/uci-wide,0.jpg>
- request width 400px (width = 400), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - adj.-height = width / aspect-ratio = 400px / 1.777 approx. 225.08px -> **225px**
 - the target image has size 400x225 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/1/400/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successfully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

SCALE IMAGE PROPORTIONATELY ON HEIGHT

/images/0001/2/{height}/{path-segment-n}*/{image}

Description

Return the requested image where the height of the image is exactly the requested on. The width of the image is determined by the ratio of the original image.

Parameter

parameter	description
height	height of the image
path-segment-n*	folder/path where image is located
image	filename of the image

Examples

- request height 600px (height = 600), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - adj.-width = height * aspect-ratio = 600px * 1.777 approx. 1066.3px -> **1066px**
 - the target image has size 1066x600 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/2/600/010/wiki/uci-wide,0.jpg>
- request height 500px (height = 500), aspect ratio of (original) image is 1276/718 (approx. 1.777:1)
 - adj.-width = height * aspect-ratio = 500px * 1.777 approx. 888.58px -> **889px**
 - the target image has size 889x500 and nearly same aspect ratio of approx. 1.777:1 (the *same* aspect ratio is not possible due to discrete pixel values)
 - <http://image.redbull.com/rbcom/0001/2/500/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

Crop Image

MAXIMUM FIT IMAGE

/images/0010/1/{width}/{height}/{path-segment-n}*/{image}

Description

Return the requested image exactly with the requested size by (width, height). First, the original image is scaled to match width or height exactly. If the geometry of the intermediate image does not agree with the requested one then the border is uniformly deleted. This is guaranteed by centering the intermediate image.

Parameter

parameter	description
width	(exact) width of the image
height	(exact)height of the image
path-segment-n*	folder/path where image is located
image	filename of the image

Examples

- request size 200x112
 - means by method definition the exact width of 200 pixel and a height of 112px (in case of 'SCALE IMAGE PROPORTIONATELY ON WIDTH' where the resulting image has an geometry of 200x113 pixels, due to the aspect ratio of the image)
 - <http://image.redbull.com/rbcom/0010/1/200/112/010/wiki/uci-wide,0.jpg>
- request size 888x500
 - means by method definition the exact width of 888 pixel and a height of 500px (in case of 'SCALE IMAGE PROPORTIONATELY ON HEIGHT' where the resulting image has an geometry of 889x500 pixels, due to the aspect ratio of the image)
 - <http://image.redbull.com/rbcom/0010/1/888/500/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

CROP IMAGE BY CENTERED BOX

/images/0010/2/{width}/{height}/{path-segment-n}*/{image}

Description

Return the requested image exactly with the requested size by (width, height). The original image is cropped to the requested size of (width, height). The point of origin is the center of the image.

Parameter

parameter	description
width	(exact) width of the image
height	(exact) height of the image
path-segment-n *	folder/path where image is located
image	filename of the image

Example

- get the picture detail of size 700x550, with the identically center of the picture detail and of the original one
 - the picture detail is the highlighted one
 - <http://image.redbull.com/rbcom/0010/2/700/550/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

CROP IMAGE BY VIEWPORT

/images/0010/3/{width}/{height}/{top}/{left}/{path-segment-n}*/{image}

Description

Return the requested image exactly with the requested size by (width, height). The original image is cropped to the requested size of (width, height). The given top left point is the origin of the picture detail.

Parameter

parameter	description
width	(exact) width of the image
height	(exact)height of the image
top	the y-offset of the top left corner of the image
left	the x-offset of the top left corner of the image
path-segment-n *	folder/path where image is located
image	filename of the image

Example

- get the picture detail of size 230x460, with the top left point at (280, 150)
 - the picture detail is the highlighted one
 - <http://image.redbull.com/rbcom/0010/3/230/460/150/280/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

CROP IMAGE BY RECTANGLE

/images/0010/4/{top}/{left}/{bottom}/{right}/{path-segment-n}*/{image}

Description

Return the requested image exactly with the requested size by (width=right-left+1, height=bottom-top+1). The top left and bottom right corner of the image define the picture detail corners.

Parameter

parameter	description
top	the y-offset of the top left corner of the image
left	the x-offset of the top left corner of the image
bottom	the y-offset of the bottom right corner of the image
right	the x-offset of the bottom right corner of the image
path-segment-n *	folder/path where image is located
image	filename of the image

Example

- get the picture detail of size 230x480, with the top left point at (500, 150) and the bottom right point at (729, 629)
 - width = right - left + 1 = 729 - 500 + 1 = 230
 - height = bottom - top + 1 = 629 - 150 + 1 = 480
 - the picture detail is the highlighted one
 - <http://image.redbull.com/rbcom/0010/4/150/500/629/729/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found

CROP IMAGE BY RECTANGLE (AND) SCALE IMAGE PROPORTIONATELY ON WIDTH

```
/images/0012/0/{top}/{left}/{bottom}/{right}/{width}/{path-segment-n}*/{image}
```

Description

This operation is a combined one which executes first 'CROP IMAGE BY RECTANGLE' on the image. The operation 'SCALE IMAGE PROPORTIONATELY ON WIDTH' is then applied to the result of the first step. The height of the result image is determined by the given width and the aspect ratio of the intermediate result.

Parameter

parameter	description
top	the y-offset of the top left corner of the image
left	the x-offset of the top left corner of the image
bottom	the y-offset of the bottom right corner of the image
right	the x-offset of the bottom right corner of the image
width	(exact) width of the image
path-segment-n *	folder/path where image is located
image	filename of the image

Example

- get the picture detail of size 230x480, with the top left point at (500, 150) and the bottom right point at (729, 629), scale the intermediate image to fit the width of 100px
 - $\text{width} = \text{right} - \text{left} + 1 = 729 - 500 + 1 = 230$
 - $\text{height} = \text{bottom} - \text{top} + 1 = 629 - 150 + 1 = 480$
 - the picture detail is the highlighted one
 - aspect ratio of the intermediate image is approximately 0.48 (230/480)
 - the resulting image has an geometry of 100x209 pixels, due to the aspect ratio of approximately 0.48 (100px/0.48 approx. 208.7px -> **209px**)
 - <http://image.redbull.com/rbcom/0012/0/150/500/629/729/100/010/wiki/uci-wide,0.jpg>

Return Codes

Return Code	Description
200	image found, successefully scaled and send
400	malformed request, image data could not be supplied, image codec not supported, image size is too large
404	image could not be found