

Online Photo! Enhancement Platform API specification

Content

[1. General information](#)

[2. Query Parameters](#)

[3. Request format](#)

[3.1. Task queue request](#)

[3.2. Get result request](#)

[4. Available methods and their parameters](#)

[4.1. Image enhancement](#)

[4.2. Artistic effects](#)

[4.3. Funny collages](#)

[4.4. Funny cartoons](#)

[4.5. Animation effects](#)

[5. Templates system](#)

[6. Sample requests](#)

[6.1. Queued call requests using a free key with a Pho.to watermark](#)

[6.2. A series of sequential requests using the 'get_full_image' parameter and a free key with the Pho.to watermark limit](#)

[6.3. A series of sequential requests for getting unlimited result pictures \(a free key with the limitation of visiting the Pho.to site page\)](#)

[6.4. A series of sequential requests for getting result pictures using the commercial key \(pay by result\)](#)

[6.5. A series of sequential requests for getting result pictures using the commercial key \(pay by processing\)](#)

1. General information

The server for Online Photo! Enhancement Platform (image editing platform) is located at ope-api.pho.to. Interaction with the server is done through the API (Application Programming Interface) of the HTTP protocol platform. The API is asynchronous and consists of a task queue request (<http://ope-api.pho.to/queued.php>) and a get result request (<http://ope-api.pho.to/get-result.php>), and also some service requests (see [p.6](#)).

2. Query Parameters

The API supports the following query parameters in direct and queued POST and GET requests (transmitting parameters format is given in the following sections):

key=xxxxx – key issued by <http://pho.to> server;

image_limit=watermark or **image_limit=preview_size**. This parameter should be specified if a free key is used, or a commercial key with payment for each saved result. If you choose **image_limit=watermark**, a resulting image will be marked with the Pho.to watermark (in case of using a commercial key, a client's logo can also be used). If you choose **image_limit=preview_size**, a resulting image will be resized to a small size (less than or equal to 200 pixels).

image_url=url1|url2|... – the URL address of a source image. The delimiter character “|” can be used to specify several images for special methods;
rect=(left,top,right,bottom) attribute inside of **image_url** tag defines crop rectangle for the image (supported only in POST format);

image_id=xxxxx-xxxxx-xxxxx-xxxxx – previous request ID, whose result you need to use as an input image for the current request. This parameter can be used instead of **image_url** parameter in the sequence of consecutive actions with an image. For instance, in case of using **image_limit=watermark**, a previous watermark is removed from an image before the next processing;

methods_list – specifies the sequence of image processing methods (filters) and their parameters in the following format:

method1_name – method name; params1 – the string containing method parameters (if the string is absent, the default values are used). The parameters string (params1) has the following format:
param1=value1; param2=value2;...; paramn=valuen, where param1 is parameter name, value1 is parameter value;

watermark=[url=xxxxx; tl=(x.xx, x.xx);tr=(x.xx, x.xx);bl=(x.xx, x.xx);br=(x.xx, x.xx);mid=(x.xx, x.xx)] – marks the resulting image with an image located at the specified URL. The image specified by **url=xxxxx** is inscribed into a specified area.

The inscribing area may be defined by one of the points (in this case the size of the inscribed image does not change):

tl – upper left point (top left);
tr - upper right point (top right);
bl – lower left point (bottom left):



br – lower right point (bottom right);
mid – middle point;

or by two points (in this case the inscribed image is resized according to a specified rectangle):

tl – upper left point (top left);
br – lower right point (bottom right);

The coordinates (x.xx) are defined with a value from 0.00 to 1.00 – measured in the parts of corresponding resulting image side size;

result_format=xxxx – file format of the resulting image. Supported file formats are jpg, png, gif, bmp, tiff (jpg is a default, for the animated images only gif is supported);

result_size=xxxx – the size of the longest side of the resulting image in pixels (can only be less than the longest side of a source image), also this parameter could be set in WIDTHxHEIGHT format, where WIDTH and HEIGHT are numeric bounds for the corresponding image dimension, one of them could be missed; examples:
result_size=256 (on maximal dimension), result_size=320x200 (on width and height), result_size=640x (on width), result_size=x480 (on height);

result_quality=xx – resulting image quality is defined by a value from 0 to 100 (in percents). If this parameter is absent, the original quality of the source image is preserved by default.

thumb1_size=xxxx, thumb2_size=xxxx – the size of the longest side of the thumbnail images in pixels (can only be less than the result_size), supports WIDTHxHEIGHT format, like the **result_size** parameter;

lang=xx – preferred response language, a two-letter language identifier from ISO 639-1 (by default, en (English) is accepted);

animation_frames_count=xx – the number of frames in a resulting animation image (for animation effects only, see p. [4.5](#));

animation_speed_factor=xxx – a whole number, whose negative value denotes slowing down of the animation, and positive value denotes speeding up of the animation. The value of 10 denotes twofold speedup;

template_watermark=true/false – in case of false value removes pho.to template watermark in collage and animated_effect methods (valid only for unlimited result_url image, not for limited_image_url).

3. Request format

The address requests should be sent to is: <http://ope-api.pho.to/queued.php>
Requests of this kind are called asynchronous, and a response comes immediately. All incoming image processing tasks are placed in a queue, and are handled depending on the server workload. A resulting image is received by sending a request at <http://ope-api.pho.to/get-result.php> (see details below).

3.1 Task queue request

3.1.1. Format of a GET request for enqueueing a task

The address requests are sent to is:

http://ope-api.pho.to/queued.php?key=xxxxx&response =
[.....]&image_limit=xxxxx&image_url=xxxxx
&methods_list=method1_name:params1|method2_name:params2|...
|methodn_name:paramsn*&watermark=[url=xxxxx;tl=(x.xx, x.xx);tr=(x.xx, x.xx);*
bl=(x.xx, x.xx);br=(x.xx, x.xx);mid=(x.xx, x.xx)]&result_size=xxxx&result_quality=xx&
thumb1_size=xxxx&thumb1_quality=xx&thumb2_size=xxxx&thumb2_quality=xx&
lang=xxx&animation_frames_count =xx&animation_speed_factor=xxx

Note! The parameters shown in italics and in lilac may be skipped. To carry out a series of sequential requests, **image_id** parameter can be used instead of **image_url** parameter (see the details in [p.2](#) above).

3.1.2. Format of a POST request for enqueueing a task

The parameters in POST requests are contained in a request body as an XML-file satisfying the following schema:

http://files.pho.to/documentation/editor-platform/ope_api_post_request.xsd.

This is a sample XML request structure:

```
<image_process_call>
  <key>string</key>
  <image_limit>string</image_limit>
  <lang>xx</lang>
  <image_url order="1" rect=(0,0,320,240)>string</image_url>
  <image_url order="2">string</image_url>
  <image_url order="3">string</image_url>
  <result_size>int_val</result_size>
  <result_quality>int_val</result_quality>
  <thumb1_size>int_val</thumb1_size>
  <thumb1_quality>int_val</thumb1_quality>
  <thumb2_size>int_val</thumb2_size>
  <thumb2_quality>int_val</thumb2_quality>
  <watermark>
    url=http://... ;tl=(0.0,0.0);br=(0.1,0.1)
  </watermark>
  <methods_list>
    <method order="1">
      <name>string</name>
      <params>xxxx</params>
    </method>
    <method order="2">
      <name>string</name>
      <params>xxxx</params>
    </method>
  </methods_list>
```

```
<animation_frames_count>xx</animation_frames_count>
<animation_speed_factor>xxx</animation_speed_factor>
<template_watermark>>false</template_watermark>
</image_process_call>
```

The order="1" attribute can be skipped for <method> and <image_url> tags if only one method / image is used.

3.1.3. Response to a task queue request

This is an example of a response structure:

```
<image_process_response>
<request_id>GUID</request_id>
<status>STATUS</status>
<description/>
<err_code>CODE</err_code>
</image_process_response>
```

The possible values of the STATUS tag are:

- a. OK - means that the task was successfully queued; you can use the request for getting a result (see section 4.2) with returned request_id to check a task completion status and get the resulting image.
- b. Service Unavailable - means that the service is temporarily unavailable.

The possible values of the CODE tag are:

- 0 – no errors;
- 5 – service unavailable.

3.2 Get result request

In order to get a result, requests should be sent to the following address: **http://ope-api.pho.to/get-result.php?request_id=...&remove_limit=xxxxx**

This request type is necessary to get a result in case of an asynchronous task queue request or in order to get a resulting image with no limits applied.

The server returns an HTTP response which is XML content satisfying the following schema:

http://files.pho.to/documentation/editor-platform/ope_api_result_response.xsd.

In case of successful processing, the response comes as an XML text file of the following kind:

```
<image_process_response>
<request_id>GUID</request_id>
<status>STATUS</status>
<description>DESCRIPTION</description>
<status>OK</status><err_code>ERROR_CODE</err_code>
<limited_image_url>http://.../tmpimage1.jpg</limited_image_url>
```

```
<result_url>http://.../tmpimage2.jpg</result_url>
<thumb1_url>http://.../tmpimage3.jpg</thumb1_url>
<thumb2_url>http://.../tmpimage4.jpg</thumb2_url>
</image_process_response>
```

If processing was not successful, the response comes as an XML text file of the following kind:

```
<image_process_response>
<request_id>F4D48515-FE2B-1424-15F1-37C6DDB97AC0</request_id>
<status>STATUS</status>
<description>DESCRIPTION</description>
</image_process_response>
```

The possible values of STATUS tag:

- a. OK – the image processing task is complete. The response is guaranteed to contain at least the <limited_image_url> with the URL of a resulting image with some restrictions (resized to a smaller size or watermarked).
- b. InProgress – the task is in progress.
- c. Error – error has occurred while processing the task.
- d. WrongID – there is no task with such request_id.

Getting a resulting image without any limitations.

In the following cases the response will also contain the following tags: the <result_url> tag with the unrestricted resulting image, and the <thumb1_url> and <thumb2_url> tags with undersized images (in case the size is specified in the request):

1. In case you use a commercial key with payment for each call.
2. In case you use a commercial key with payment for each saved result and the *remove_limit=approve* parameter is used (then you pay for a saved result).

If a free key is used, it is possible to get a resulting image with no limitations by visiting pho.to web site. For this, it is necessary to use the *remove_limit=visit_page* parameter. In this case the response will have the

<page_to_visit>http://pho.to/visit_url</page_to_visit> tag, containing the URL that has to be visited. A user is required to visit the http://pho.to/visit_url&redirect_url=YOURSERVERURL, after which he will be redirected to the page of the type:

YOURSERVERURL+result_url=RESULTURL&thumb1_url=THUMB1URL&thumb2_url=THUMB2URL, where the “+” character can be & or ? depending on the YOURSERVERURL syntax. In the YOURSERVERURL string you can specify all needed parameters, such as:

- user ID
- album ID
- and other parameters that you may need.

The RESULTURL is an address of a resulting image, available for download (an image is stored at this URL for 10 hours, that is why it needs to be downloaded to your server).

THUMB1URL and THUMB2URL are URLs of two different sized thumbnails of a resulting image. These URLs are available only if a commercial key is used. The YOURSERVERURL must be URL encoded. The RESULTURL, THUMB1URL, THUMB2URL addresses are also URL encoded.

4. Available methods and their parameters

All methods are characterized with a certain complexity factor, depending on which the final cost of processing is calculated for commercial usage of the Platform.

4.1. Image enhancement

4.1.1. Automatic makeup

name: **makeup**

parameters:

- use_auto_red_eye – removes red eye effect
 - range: [false, true]
 - default value: true
- use_skin_healing - removes pimples and skin defects
 - range: [false, true]
 - default value: true
- use_flash_healing – removes flash or oily skin glare
 - range: [false, true]
 - default value: true
- use_eyes_enhancement – enhances eye contour and eye color
 - range: [false, true]
 - default value: true
- use_teeth_whitening - whitens the teeth
 - range: [false, true]
 - default value: true
- use_portrait_filters - “glamorizes” an image with a soft focus effect and color tinting
 - range: [false, true]
 - default value: true
- use_skin_color_corrector – enhances color balance of a photography based on a face skin color
 - range: [false, true]
 - default value: false

Note: this methods works only with portrait photos.

4.1.2. Automatic image enhancement

name: **auto_enhancement**

parameters:

- use_auto_red_eye – removes red eye effect
 - range: [false, true]
 - default value: true
- use_shadow_highlight – enhances lighting

- range: [false, true]
 - default value: true
- use_color_temperature – enhances color balance
 - range: [false, true]
 - default value: true
- use_denoising – uses denoising
 - range: [false, true]
 - default value: true
- use_deblurring – uses sharpening
 - range: [false, true]
 - default value: true
- use_saturation – uses saturation correction
 - range: [false, true]
 - default value: true

4.1.3. Automatic red eye effect removal

name: **red_eyes**

parameters: ---

4.1.4. Automatic color correction

(includes lighting correction and color temperature correction)

name: **color_correction**

parameters: ---

4.1.5. Lighting correction

name: **shadow_highlight**

parameters:

- shadow_factor (correction of shadow areas)
 - range: [1.0, 6.0]
 - default value: 3.0
- highlight_factor (correction of highlight areas)
 - range: [1.0, 5.0]
 - default value: 1.4

4.1.6. Color temperature correction

name: **color_temperature**

parameters: ---

4.1.7. Color saturation correction

name: **auto_saturation**

parameters: ---

4.1.8. Denoising

name: **denoising**

parameters:

- intensity_strength (luminance noise correction)
 - range: [-2.0, 2.0]

- default value: 0.0
- color_strength (color noise correction)
 - range: [-2.0, 2.0]
 - default value: 0.0
- crispness
 - range: [0.0, 1.0]
 - default value: 0.4

4.1.9. Deblurring (sharpening)

name: **deblurring**

parameters:

- strength
 - range: [0.0, 1.0]
 - default value: 0.25
- radius
 - range: [1, 20]
 - default value: 5

4.1.10. Potrait enhancement filter

name: **portrait_filters**

parameters:

- defocus_strength
 - range: [0.0, 1.0]
 - default value: 0.5
- diffusion_strength
 - range: [0.0, 1.0]
 - default value: 0.3
- glamour_strength
 - range: [0.0, 1.0]
 - default value: 0.2

4.1.11. Rotation

name: **rotation**

parameters:

- angle
 - range: [0.0, 360.0]
 - default value: 90.0

4.1.12. Crop

name: **crop**

parameters:

- rect – formatted as (left,top,right,bottom) (or (x1, y1, x2, y2)), all dimensions are in pixel units.

4.1.13. Image resizing

name: **resampling**

parameters:

- width
 - range: [0, 5000]
- height

- range: [0, 5000]
- **preserve_aspect** – denotes whether original proportions should be preserved
 - range: [false, true]
 - default value: false

4.1.14. Text labeling

name: **write_text**

parameters:

- **text** – base64 encoded,
- **tl, tr, bl, br, mid** – the rectangle specification is the same as for watermarking,
- **red, green, blue** – RGB text color specification.

4.1.15. Face detection

name: **face_detection**

parameters:

- **enlargement** – the size of the rectangle in percent in relation to the face size
 - default value: 140
- **w_aspect, h_aspect** – define the proportions of the resulting rectangle (if other than 0)
 - default value: 0

4.2. Artistic effects

4.2.1. Converting color images into grayscale

name: **desaturation**

parameters: ---

4.2.2. Old photo effect (sepia)

name: **sepia**

parameters: ---

4.2.3. Coloring photo with a single color tone

name: **color_dominance**

parameters:

- **hue**
 - range: [0.0, 6.283]
 - default value: 4.188790

4.2.4 Sunny colors effect

name: **sunny_colors**

parameters: ---

4.2.5. Oil painting effect

name: **oil_painting**

parameters: ---

4.2.6. “Chalk and charcoal” drawing effect

name: **chalk_charcoal**

parameters:

- **draw_sketch** – outlining contours with a pencil

- range: [0, 1]
- default value: 0
- stroke_pressure – stroke pressure (contrast)
 - range: [0.1, 5.0]
 - default value: 1.0
- chalk_area – amount of chalk
 - range: [1.1, 20.0]
 - default value: 6.0
- charcoal_area – amount of charcoal
 - range: [1.1, 20.0]
 - default value: 6.0
- draw_background – fill a one-colour background or not
 - range: [false, true]
 - default value: true

4.2.7. Pencil drawing effect

name: **pencil_painting**

parameters:

- stroke_pressure – stroke pressure (contrast)
 - range: [0.1, 5.0]
 - default value: 1.0

4.2.8. Neon glow effect

name: **neon**

parameters: ---

4.2.9. Mosaic effect

name: **mosaic**

parameters:

- tile_size_koef – tile size
 - range: [0.25, 4.0]
 - default value: 0.7

4.2.10. Blur effect

name: **blur**

parameters:

- radius – blur radius (in pixels)
 - range: [1, 100]
 - default value: 5

4.2.11. Fire flames effect

name: **fire**

parameters:

- type – type of effect,
 - range: [0, 1]
 - default value: 0

4.2.12. Matrix effect

name: **matrix**

parameters:

- symbol_type – symbol type,
 - range: [0, 2]

- default value: 1

4.2.13. Pencil sketch effect

name: **sketch**

parameters:

- strength – curve pressure (intensity),
 - range: [0.0, 1.0]
 - default value: 0.7
- details – detaling,
 - range: [0.0, 1.0]
 - default value: 0.7
- color – curve color in RGB space,
 - range: (0,0,0) – (255,255,255)
 - default value: (0,0,0)

4.2.14. Underwater effect

name: **underwater**

parameters: ---

4.2.15. Infrared filming effect

name: **infrared**

parameters: ---

4.2.16. High dynamic range effect (HDR)

name: **hdr**

parameters: ---

4.2.17. Dave Hill effect

name: **dave_hill**

parameters: ---

4.3. Funny collages

name: **collage**

parameters:

- template_name – collage template name (FET type),
- crop_portrait – if the value is true, the face area will be automatically cropped. If the value is false, the whole image will be used.
 - range: [false, true]
 - default value: false
- method_name – name of method for preliminary image processing,
- text – base64 encoded text, for labeling of the text-supporting templates.
- animation – achievement of animated result for the animation-supporting templates.
 - range: [false, true]
 - default value: false

The full templates list can be received by sending a special request (see [p. 5](#))

Some templates available to use:

1. Artist Studio
2. Exhibition Room
3. Montmartre Artist

4.4. Funny cartoons

4.4.1. Caricature (emotions simulation, face morphing)

name: **caricature**

parameters:

- type – cartoon type (whole number),
 - range: [0, 5]
 - 0 – random combination (random_seed parameter must be set),
 - 1 – smile,
 - 2 – sad,
 - 3 – oops,
 - 4 – squint eyed,
 - 5 – troll,
 - 6 – wink,
 - 7 – flirt,
 - 8 – offended,
 - 9 – alien,
 - 10 – martian,
 - 11 – bulb head,
 - 12 – tough guy,
 - 13 – grotesque,
 - 14 – fat-cheeked,
 - default value: 0
- random_seed – whole number, specifies a number of a random combination (makes sense with type=0 only),
- strength – warping amount (-1.0 value means default amount),
 - range: -1.0 , [0.0, 1.0]
 - default value: -1.0
- crop_portrait – if the value is true, the face area will be automatically cropped. If the value is false, the whole image will be used,
 - range: [false, true]
 - default value: false
- cartoon – usage of the cartoonizing filter,
 - range: [false, true]
 - default value: false

Note: this methods works only with portrait images.

4.4.2. Cartoonizing filter

name: **cartoon**

parameters:

- useFD – process portrait with special algorithm,
 - range: [false, true]
 - default value: false
- use_warping – local warpings usage,
 - range: [false, true]
 - default value: false
- use_quant – luminance quantizing usage,

- range: [false, true]
 - default value: true
- quant_levels – luminance quantizing levels quantity,
 - range: [1, 100]
 - default value: 4
- target_color – target RGB color that defines output brightness, used in case of quant_levels=1
 - default value: (128, 128, 128)

4.5. Animation effects

4.5.1. Animation effect with a specified template name

name: **animated_effect**

parameters:

- template_name – name of animation effect template (AET type),
- static – achievement of the static result (without animation),
 - range: [false, true]
 - default value: false

The full templates list can be received by sending a special request (see [p. 5](#))
Some templates available to use:

1. Cartoon Smile
2. Water Reflection
3. Large Sparkles

4.5.2. Labeling with animated text

name: **animated_text**

parameters:

- text – base64 encoded,
- custom_font – index of the animated font, zero value means standard font
 - range: [0, 1]
 - default value: 1
- font – standard font name,
 - possible values: Arial, Times New Roman, Courier New, ...
 - default value: Arial
- red, green, blue – standard font color,
 - range: [0, 255]
 - default value: 0
- animation_type – label animation type,
 - range: [0, 2]
 - 0 – without label animation,
 - 1 – whole label fading,
 - 2 – letter-by-letter fading
 - default value: 0
- position – (numpad-keys like),
 - possible values: 1,2,3,4,6,7,8,9
 - default value: 2
- vertical – vertical text alignment (valid for the position values of 1,3,7,9),
 - range: [false, true]

- default value: false
- size – height of the label in percents of the image height,
 - range: [1, 100]
 - default value: 25

4.5.3. Slideshow

name: **slideshow**

parameters:

- type – type of transition effect,
 - range: [0, 3]
 - default value: 0

Note: more than one source images are required for this method.

4.5.4. Animated filter (emerging and fading)

name: **animated_filter**

parameters:

- filter – filter method name,
- other parameters are the same as for the chosen method.

4.5.5. Animated caricature

name: **animated_caricature**

parameters: same as in cartoonist, see p. [4.4](#).

Note: this methods works only with portrait images.

5. Templates system

There is a templates system used to create funny collages (collage method, p. [4.3](#)) and animation effects (**animated_effect** method, p. [4.5.1](#)). The templates are stored on the pho.to server. The list of available templates can be received by sending the following request:

- For funny collages templates (**FET** type):
http://ope-api.pho.to/service.php?method=get_templates&type=FET or
<http://ope-api.pho.to/service/templates/fet-full.xml> (joint preview for quick loading of multiple previews)
- For animation effects templates (**AET** type):
http://ope-api.pho.to/service.php?method=get_templates&type=AET

The response format is as follows:

```
<response>
  <template>
    <name>summer_love</name>
    <title>Summer Love</title>
    <date>2010-02-05 16:28:29</date>
    <tags>
      group:real,text,mphotosstrong:2,wignore,categories:Love|Nature|
      Heart|Travel|Valentine
```

```

        </tags>
        <downloads>2.1352</downloads>
        <thumb>
        http://pho.to/templates/fet/summer_love/preview.jpg
        </thumb>
    </template>
    <template>
        <name>mirror_room</name>
        <title>Mirror Room</title>
        <date>2010-01-15 09:40:15</date>
        <tags>group:real,text,categories:Fun|Glossy</tags>
        <downloads>2.0875</downloads>
        <thumb>
        http://pho.to/templates/fet/mirror_room/preview.jpg
        </thumb>
    </template>
    .....all other templates.....
</response>

```

The values are as follows:

name – template unique name;

title – template name that should be inserted as a “template_name” parameter in effects from collage и animated_effect groups;

date – date of template creation;

downloads – number of times the template has been used in the previous 3 weeks;

thumb – URL address of the template thumbnail;

tags – some of the template parameters such as:

- group:group_identifier;
- categories:category1| category2|... – categories(keywords) of the template;
- mphotos:input_images_maximal_number;
- text – labeling support for the FET template;
- animated – animation support for the FET template;
- support_static – static result support for the AET template;
- face_detection – face detection algorithm usage (portrait image needed).

The received template names can be transmitted in the **template_name** parameter of the corresponding method.

The full list of categories for funny collages templates (**FET** type) is available via the following request:

http://ope-api.pho.to/service.php?method=get_categories&type=FET.

A response format of the categories list request:


```
<fet-categories>
<category>Art</category>
<category>Billboard</category>
<category>Black&White</category>
...
</fet-categories>
```

6. Sample requests

In examples below we'll be using three types of test keys:

3CCREZBSFGLPUYAXT7RJL6KEF6EB – a free test key (with a limit on the resulting picture size or Pho.to watermark).

RMVKY9DDDQ5VNFESPGT9FKMTSLAZ – a commercial test key (with payment for result images and an ability to process up to 500 images a day).

1FBJMVRPFW81ACRG168LIHZRPY2K – a commercial test key (with payment for photo processings and an ability to process up to 500 images a day).

6.1. Queued call requests using a free key with a Pho.to watermark

Request №1:

http://ope-api.pho.to/queued.php?key=3CCREZBSFGLPUYAXT7RJL6KEF6EB&image_limit=watermark&image_url=%20http://pho.to/img/soft/ope/samples/17.jpg%20&methods_list=caricature:type=13

Sample response:

```
<image_process_response>
<request_id>35A4E6DF-A4B6-8504-31E5-96AF4D92DCF0</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №2 (request_id is taken from the response to Request №1):

http://ope-api.pho.to/get-result.php?request_id=35A4E6DF-A4B6-8504-31E5-96AF4D92DCF0

Sample response:

```
<image_process_response>
<request_id>35A4E6DF-A4B6-8504-31E5-96AF4D92DCF0</request_id>
<status>InProgress</status>
<description/>
</image_process_response>
```

OR later:

```
<image_process_response>
<request_id>35A4E6DF-A4B6-8504-31E5-96AF4D92DCF0</request_id>
<status>OK</status>
<limited_image_url>
http://storageX.pho.to/images....jpg
</limited_image_url>
</image_process_response>
```

The result image will be marked with the Pho.to watermark, as Request №1 has the 'image_limit=watermark' limit.

6.2. A series of sequential requests using the 'get_full_image' parameter and a free key with the Pho.to watermark limit.

Request №1:

http://open-api.pho.to/queued.php?key=3CCREZBSFGLPUYAXT7RJL6KEF6EB&image_limit=watermark&image_url=http://pho.to/img/soft/open/samples/17.jpg&methods_list=color_dominance:hue=1.5

Response:

```
<image_process_response>
<request_id>E26881CE-0C68-4714-B160-6C2D4D0AC458</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №2 (the request_id value from the response to Request №1 is inserted into the image_id parameter):

http://open-api.pho.to/queued.php?key=3CCREZBSFGLPUYAXT7RJL6KEF6EB&image_limit=watermark&image_id=F6C05D75-0004-3BE4-A1CF-002B86E4B74C&methods_list=rotation:angle=5

Response:

```
<image_process_response>
<request_id>24B7D680-D4EA-2D74-DD0F-7F503E0E1D04</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №3 (the request_id value from the response to Request №2 is inserted into the image_id parameter):

http://ope-api.pho.to/queued.php?key=3CCREZBSFGLPUYAXT7RJL6KEF6EB&image_limit=watermark&image_id=C99F5522-7DEA-3E84-816E-A510FC47E416&methods_list=cartoonist:type=3

Response:

```
<image_process_response>
<request_id>5D5DEC1E-8927-C5B4-258B-AF59188055BE</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №4 (request_id is taken from the response to Request №3):

http://ope-api.pho.to/get-result.php?request_id=806B5F22-C229-5004-F9F4-1127AFE4B5A6

Response:

```
<image_process_response>
<request_id>5D5DEC1E-8927-C5B4-258B-AF59188055BE</request_id>
<status>OK</status>
<limited_image_url>
http://storageX.pho.to/images....jpg
</limited_image_url>
</image_process_response>
```

6.3 A series of sequential requests for getting unlimited result pictures (a free key with the limitation of visiting the Pho.to site page)

Request №1:

http://ope-api.pho.to/queued.php?key=3CCREZBSFGLPUYAXT7RJL6KEF6EB&result_size=600&methods_list=collage:template_name=glossy%20magazine&image_url=http://pho.to/img/soft/ope/samples/9.jpg&result_format=jpg&thumb1_size=190&thumb2_size=50&image_limit=watermark

Response:

```
<image_process_response>
<request_id>07D2E30E-B471-3DD4-3929-49CDAF1DC267</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №2 (request_id is taken from the response to Request №1):

http://ope-api.pho.to/get-result.php?request_id=07D2E30E-B471-3DD4-3929-49CDAF1DC267

Response:

```
<image_process_response>
<request_id>07D2E30E-B471-3DD4-3929-49CDAF1DC267</request_id>
<status>OK</status>
<limited_image_url>
http://storageX.pho.to/images/....jpg
</limited_image_url>
</image_process_response>
```

Request №3 (request_id is taken from the response to Request №2):

http://ope-api.pho.to/get-result.php?request_id=07D2E30E-B471-3DD4-3929-49CDAF1DC267&remove_limit=visit_page

Response:

```
<image_process_response>
<request_id>07D2E30E-B471-3DD4-3929-49CDAF1DC267</request_id>
<status>OK</status>
<limited_image_url>
http://storage1.pho.to/images/....jpg
</limited_image_url>
<page_to_visit>
http://ope-api.pho.to/get-html-result.php?request_id=07D2E30E-B471-3DD4-3929-49CDAF1DC267
</page_to_visit>
</image_process_response>
```

To get a result picture without any limitations, a user should be redirected to a page specified in the <page_to_visit> tag.

6.4. A series of sequential requests for getting result pictures when using the commercial key (pay by result). This image (<http://my.pho.to/natali/s1/images/mhk4fif/image.jpg>) is used as a watermark image for non-resulting pictures.

Request №1:

http://ope-api.pho.to/queued.php?key=RMVKY9DDDQ5VNFESPGT9FKMTSLAZ&result_size=600&methods_list=collage:template_name=glossy%20magazine&image_url=http://pho.to/img/soft/ope/samples/9.jpg&result_format=jpg&thumb1_size=190&thumb2_size=50&image_limit=watermark&systemwatermark=http://my.pho.to/natali/s1/images/mhk4fif/image.jpg

Response:

```
<image_process_response>
<request_id>F4D48515-FE2B-1424-15F1-37C6DDB97AC0</request_id>
<status>OK</status>
<description/>
<err_code>0</err_code>
</image_process_response>
```

Request №2 (request_id is taken from the response to Request №1):

http://ope-api.pho.to/get-result.php?request_id=F4D48515-FE2B-1424-15F1-37C6DDB97AC0

Response:

```
<image_process_response>
<request_id>F4D48515-FE2B-1424-15F1-37C6DDB97AC0</request_id>
<status>OK</status>
<limited_image_url>
http://storageX.pho.to/images/....jpg
</limited_image_url>
</image_process_response>
```

Request №3 (request_id is taken from the response to Request №2):

http://ope-api.pho.to/get-result.php?request_id=F4D48515-FE2B-1424-15F1-37C6DDB97AC0&remove_limit=approve

Response (If you use the remove_limit=approve parameter in a request, you get a picture without any limitations. The processed pictures counter for this key is increased by one point.):

```
<image_process_response>
<request_id>F4D48515-FE2B-1424-15F1-37C6DDB97AC0</request_id>
<status>OK</status>
<result_url>
http://storageX.pho.to/images/....jpg
</result_url>
<thumb1_url>
http://storageX.pho.to/images/....jpg
</thumb1_url>
<thumb2_url>
http://storageX.pho.to/images/....jpg
</thumb2_url>
<limited_image_url>
http://storageX.pho.to/images/....jpg
</limited_image_url>
</image_process_response>
```

6.5. A series of sequential requests for getting result pictures using the commercial key (pay by processing).

Request №1:

http://ope-api.pho.to/queued.php?key=1FBJMVRPFW81ACRG168LIHZRPY2K&result_size=800&methods_list=collage:template_name=coal&image_url=http://pho.to/img/soft/ope/samples/9.jpg&result_format=jpg&thumb1_size=200&thumb2_size=100

Response:

```
<image_process_response>
```

```
<request_id>E941ADBA-028F-0CE4-AD49-0DEC8644053D</request_id>  
<status>OK</status>  
<description/>  
<err_code>0</err_code>  
</image_process_response>
```

Request №2 (request_id is taken from the response to Request №1):

http://ope-api.pho.to/get-result.php?request_id=E941ADBA-028F-0CE4-AD49-0DEC8644053D

Response:

```
<image_process_response>  
<request_id>E941ADBA-028F-0CE4-AD49-0DEC8644053D</request_id>  
<status>OK</status>  
<result_url>  
http://storageX.pho.to/images/...jpg  
</result_url>  
<thumb1_url>  
http://storageX.pho.to/images/...jpg  
</thumb1_url>  
<thumb2_url>  
http://storageX.pho.to/images/...jpg  
</thumb2_url>  
</image_process_response>
```

The processed pictures counter for this key is increased by one point.