

Remove The Background™ API Reference

Remove The Background™

Email: cs@removethebackground.com

www.removethebackground.com

Contents

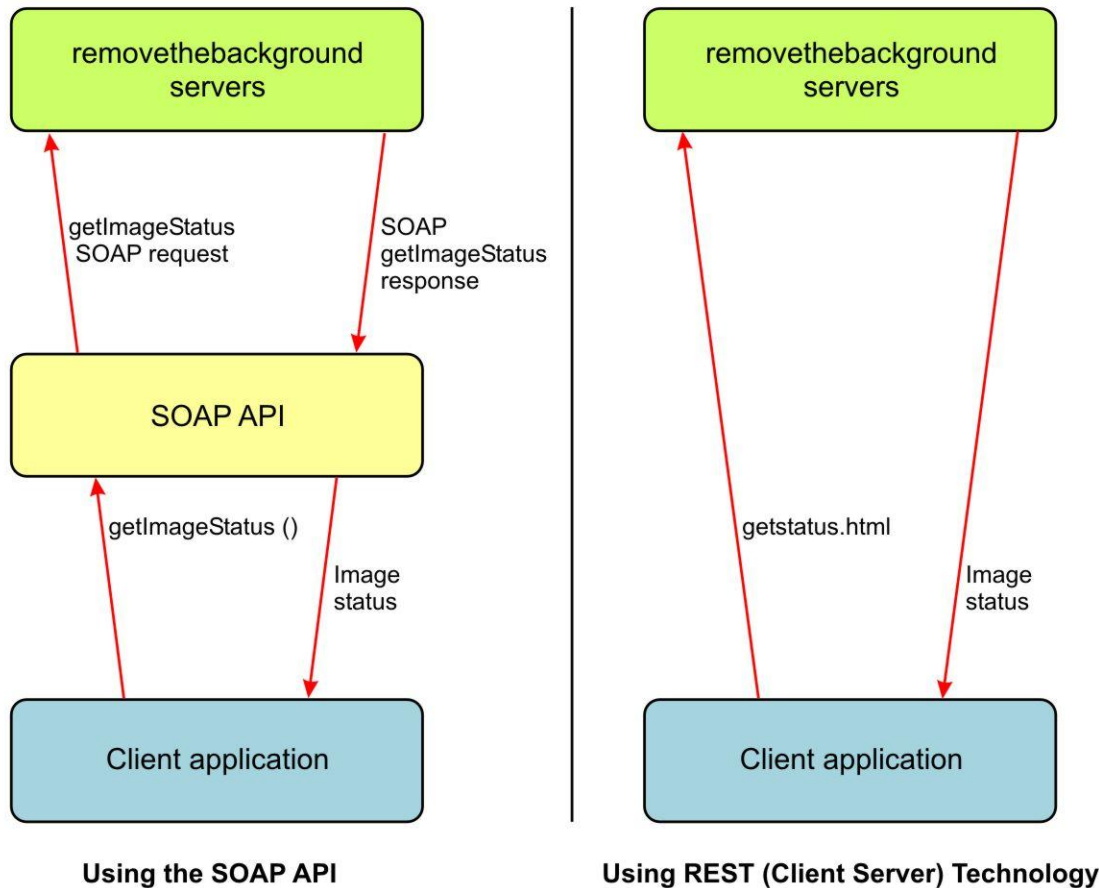
Contents	i
Introduction.....	1
SOAP	2
REST	2
SOAP API Reference	3
addImageToQueue.....	3
URL:	3
Method:.....	3
Authentication:	3
Prototype:	3
Parameters:	3
Returns:	4
Request:	4
Response:	4
Example:.....	5
addImageToQueueSimpleTypes	7
URL:	7
Method:.....	7
Authentication:	7
Prototype:	7
Parameters:	7
Returns:	8
Request:	8
Response:	8
Example:.....	9
deleteImageFromQueue	10
URL:	10
Method:.....	10
Authentication:	10
Prototype:	10
Parameters:	10
Returns:	10
Request:	10
Response:	11
Example:.....	11
getImageStatus	12
URL:	12
Method:.....	12
Authentication:	12
Prototype:	12
Parameters:	12
Returns:	12
Request:	13
Response:	13
Example:.....	14
getTemplates.....	15
URL:	15
Method:.....	15

Authentication:	15
Prototype:	15
Parameters:	15
Returns:	15
Request:	15
Response:	16
Example:.....	16
REST API Reference.....	17
addImage	17
URL:	17
Input:	17
Output:.....	17
getStatus	18
URL:	18
Input:	18
Output:.....	18
getTemplates.....	19
URL:	19
Input:	19
Output:.....	19

Introduction

Remove The Background™ provides a Developer API that can be used to integrate our service with third-party applications such as CMS and shopping cart systems. Remove The Background™ provides both a SOAP API as well as a REST API to access the service. Thus, developers can use whichever system they prefer.

Two methods of accessing data from the removethebackground.com servers



This document explains how function calls can be made using the Remove The Background™ APIs.

Note: The reader of this document should have strong knowledge of and experience with general application development. Additionally, the reader should be familiar with SOAP or REST Web Services.

The API can be used to —

- add an image to the processing queue,
- get the status of an uploaded image, and
- get a list of available templates.

The Remove The Background™ API allows you to query your server and get information by using the methods of the Remove The Background™ SOAP (or REST) service. You can also use the API to upload images to the Remove The Background™ service queue and query the queue to find out the processing status of your images.

SOAP

Simple Object Access Protocol (SOAP) is a protocol that allows computers in a network or the Internet to exchange information using Web Services. It uses XML for its message format. SOAP requests to a server and the response from the server are encoded in XML.

The Remove The Background™ SOAP API is a SOAP 1.2 implementation. It defines the following methods:

Method	Description
addImageToQueue	Adds an image to Remove The Background™'s image processing queue.
addImageToQueueSimpleTypes	Adds an image to Remove The Background™'s image processing queue using simple parameters.
deleteImageFromQueue	Deletes an image from the processing queue.
getImageStatus	Retrieves the status of an image added to the queue.
getTemplates	Retrieves the list of templates from Remove The Background™'s server.

REST

REpresentational State Transfer (REST) is an architecture that consists of clients and servers. The client makes a request for information and the server serves the requested information to the client.

The REST API consists of the following scripts:

Script	Description
addimage.html	Adds an image to Remove The Background™'s image processing queue.
getstatus.html	Retrieves the status of an image added to the queue.
gettemplates.html	Retrieves the list of templates from Remove The Background™'s server.

SOAP API Reference

addImageToQueue

Adds up to two images to the processing queue.

URL:

<http://api.removethebackground.com>

Method:

POST

Authentication:

Required.

Prototype:

String addImageToQueue(String *ContactEmail*, String *ContactAPIkey*, String *developerAPIkey*, Integer *templateId*, String *ImageDeliveryInfo*, String *Image2DeliveryInfo*, Integer *ImageRequestedDeliveryTimeInHours*, String *ImageExternalTag*, String *ImageExternalText*, String *ImageCallbackURL*)

Parameters:

Parameter	Required	Type	Description
<i>ContactEmail</i>	No	String	The registered email address of the user.
<i>ContactAPIkey</i>	No	String	The API key given to the user.
<i>developerAPIkey</i>	No	String	If you are a reseller, use the Contact API key. If you are a developer, use your own key here and the customer's API key in <i>ContactAPIkey</i> .
<i>templateId</i>	Yes	Int	The ID Number of the template selected by the user.
<i>ImageDeliveryInfo</i>	No	String	The URL of the image to be uploaded.
<i>Image2DeliveryInfo</i>	No	String	The URL of the second image to be uploaded (when the user needs to combine two images).
<i>ImageRequestedDeliveryTimeInHours</i>	Yes	Int	The number of hours after which the image needs to be sent back. Should be 4, 24, 72 or 168.
<i>ImageExternalTag</i>	No	String	This text will be returned when the image is returned.
<i>ImageExternalText</i>	No	String	This text will be returned when the image is returned.

Parameter	Required	Type	Description
<i>ImageCallbackURL</i>	No	String	<p>This is the URL of the results page where <i>imageTicket</i> and <i>ProcessedImageURL</i> will be posted. <i>imageTicket</i> will contain the ticket id.</p> <p>The page must return a HTTP status code 200 OK indicating success. All other values will indicate a failure and the system will try again later, up to 3 times all in all.</p> <p>If this value is not specified, the system will use the Template Callback URL.</p>

Q: Both *imageDeliveryInfo* and *image2DeliveryInfo* have been given as optional (*minOccurs*="0", *maxOccurs*="1"). What values will be used if these values are not specified? What image will be added if we do not specify any URL for the image?)

Returns:

When this method is called, the system sends an *addImageToQueue* request to the service. In response, the service returns an *addImageToQueueResponse* message that contains an *imageTicket* guid on success or an error message and code on failure. The system can also optionally return the tag and caption of the image.

Request:

Note: The placeholders shown in violet need to be replaced with actual values.

```
POST /soap.asmx HTTP/1.1
Host: api.removethebackground.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <addImageToQueue xmlns="http://removethebackground.com/">
      <ContactEmail>string</ContactEmail>
      <ContactAPIkey>string</ContactAPIkey>
      <developerAPIkey>string</developerAPIkey>
      <templateId>int</templateId>
      <ImageDeliveryInfo>string</ImageDeliveryInfo>
      <Image2DeliveryInfo>string</Image2DeliveryInfo>
      <ImageRequestedDeliveryTimeInHours>int</ImageRequestedDeliveryTimeInHours>
      <ImageExternalTag>string</ImageExternalTag>
      <ImageExternalText>string</ImageExternalText>
      <ImageCallbackURL>string</ImageCallbackURL>
    </addImageToQueue>
  </soap12:Body>
</soap12:Envelope>
```

Response:

Note: The placeholders shown in violet need to be replaced with actual values.


```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <addImageToQueueResponse xmlns="http://removethebackground.com/">
      <addImageToQueueResult>
        <imageTicket>guid</imageTicket>
      </addImageToQueueResult>
    </addImageToQueueResponse>
  </soap12:Body>
</soap12:Envelope>

```

Example:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <script type='text/javascript'
src='https://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js'></script>
  <script type='text/javascript'>

    function addImage() {
      //We are passing multiple paramers to the service in json format {"ContactEmail" :
"janus@removethebackground.com", "ContactAPIkey": "apikey"}
      var varData = '{"ContactEmail": "' + $('#ContactEmail').val()
+ '", "ContactAPIkey": "' + $('#ContactAPIkey').val()
+ '", "developerAPIkey": "' + $('#developerAPIkey').val()
+ '", "templateId": "' + $('#templateId').val()
+ '", "ImageDeliveryInfo": "' + $('#ImageDeliveryInfo').val()
+ '", "Image2DeliveryInfo": "' + $('#Image2DeliveryInfo').val()
+ '", "ImageRequestedDeliveryTimeInHours": "' + $('#ImageRequestedDeliveryTimeInHours').val()
+ '", "ImageExternalTag": "' + $('#ImageExternalTag').val()
+ '", "ImageExternalText": "' + $('#ImageExternalText').val()
+ '", "ImageCallbackURL": "' + $('#ImageCallbackURL').val()
+ '"}';
      $('#JsonData').text(varData);
      $.ajax({
        type: "POST", //GET or POST or PUT or DELETE verb
        url: "/REST.svc/Image/", //Location of the service
        data: varData, //Data sent to server
        contentType: "application/json; charset=utf-8", //content type sent to server
        dataType: "xml", //Expected data format from server
        processData: false, //True or False
        success: function (msg) { //On Successfull service call
          ServiceSucceeded(msg);
        },
        error: ServiceFailed //When Service call fails
      });
    }

    function ServiceSucceeded(result) {
      if ($(result).find("errorCode").text() != 'noError') {
        alert("Error:\n\n" + $(result).find("errorCode").text() + $(result).find("message").text());
      } else {
        alert("Success. Image added to Queue");
      }
    }

    function ServiceFailed() {
      alert("Fail");
    }

```

```
function submitImageStatus() {
    $('#getStatusForm').attr('action', '/REST.svc/Image/Status/' + $('#ImageTicket').val() + '/')
    $('#getStatusForm').submit();
}
</script>
<title>addImageToQueue Sample</title>
</head>
<body>
    Add image:<br />

    <form method="post">
        ContactEmail: <input type="text" id="ContactEmail"/><br />
        ContactAPIkey: <input type="text" id="ContactAPIkey"/><br />
        developerAPIkey: <input type="text" id="developerAPIkey"/><br />
        templateId: <input type="text" id="templateId"/><br />
        ImageDeliveryInfo: <input type="text" id="ImageDeliveryInfo"/><br />
        Image2DeliveryInfo: <input type="text" id="Image2DeliveryInfo"/><br />
        ImageRequestedDeliveryTimeInHours: <input type="text" id="ImageRequestedDeliveryTimeInHours"/><br />
        ImageExternalTag: <input type="text" id="ImageExternalTag"/><br />
        ImageExternalText: <input type="text" id="ImageExternalText"/><br />
        ImageCallbackURL: <input type="text" id="ImageCallbackURL"/> (Leave empty to use Template Callback URL)<br
    />
    <input type="button" value="Post picture" onclick="javascript:addImage();" /><br />
    JSON Data:<br />

    <div id="JsonData"></div>
    </form><br />
    Description: POSTS data to /REST.svc/Image/ where the post payload has the parameters as JSON collection.<br
    />
</body>
</html>
```

addImageToQueueSimpleTypes

Adds up to two images to the processing queue using simple parameters. This method is the same as `addImageToQueue`, but uses simple parameters.

URL:

<http://api.removethebackground.com>

Method:

POST

Authentication:

Required.

Prototype:

```
String addImageToQueue(String ContactEmail, String ContactAPIkey, String developerAPIkey,
Integer templateId, String ImageDeliveryInfo, String Image2DeliveryInfo, Integer
ImageRequestedDevlieryTimeInHours, String ImageExternalTag, String ImageExternalText, String
ImageCallbackURL)
```

Parameters:

Parameter	Required	Type	Description
<i>ContactEmail</i>	No	String	The registered email address of the user.
<i>ContactAPIkey</i>	No	String	The API key given to the user.
<i>developerAPIkey</i>	No	String	If you are a reseller, use the Contact API key. If you are a developer, use your own key here and the customer's API key in <i>ContactAPIkey</i> .
<i>templateId</i>	Yes	Int	The ID Number of the template selected by the user.
<i>ImageDeliveryInfo</i>	No	String	The URL of the image to be uploaded.
<i>Image2DeliveryInfo</i>	No	String	The URL of the second image to be uploaded (when the user needs to combine two images).
<i>ImageRequestedDeliveryTimeInHours</i>	Yes	Int	The number of hours after which the image needs to be sent back. Should be 4, 24, 72 or 168.
<i>ImageExternalTag</i>	No	String	This text will be returned when the image is returned.
<i>ImageExternalText</i>	No	String	This text will be returned when the image is returned.
<i>ImageCallbackURL</i>	No	String	This is the URL of the results page where <i>imageTicket</i> and <i>ProcessedImageURL</i> will be posted.

Parameter	Required	Type	Description
			<p><i>imageTicket</i> will contain the ticket id.</p> <p>The page must return a HTTP status code 200 OK indicating success. All other values will indicate a failure and the system will try again later, up to 3 times all in all.</p> <p>If this value is not specified, the system will use the Template Callback URL.</p>

Q: Both `imageDeliveryInfo` and `image2DeliveryInfo` have been given as optional (`minOccurs="0"`, `maxOccurs="1"`). What values will be used if these values are not specified? What image will be added if we do not specify any URL for the image?)

Returns:

When this method is called, the system sends an `addImageToQueueSimpleTypes` request to the service. In response, the service returns an `addImageToQueueSimpleTypesResponse` message that contains an `imageTicket` guid on success or an error message and code on failure. The system can also optionally return the tag and caption of the image.

Request:

Note: The placeholders shown in violet need to be replaced with actual values.

```
POST /soap.asmx HTTP/1.1
Host: api.removethebackground.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <addImageToQueueSimpleTypes xmlns="http://removethebackground.com/">
      <ContactEmail>string</ContactEmail>
      <ContactAPIkey>string</ContactAPIkey>
      <developerAPIkey>string</developerAPIkey>
      <templateId>int</templateId>
      <ImageDeliveryInfo>string</ImageDeliveryInfo>
      <Image2DeliveryInfo>string</Image2DeliveryInfo>
      <ImageRequestedDeliveryTimeInHours>int</ImageRequestedDeliveryTimeInHours>
      <ImageExternalTag>string</ImageExternalTag>
      <ImageExternalText>string</ImageExternalText>
      <ImageCallbackURL>string</ImageCallbackURL>
    </addImageToQueueSimpleTypes>
  </soap12:Body>
</soap12:Envelope>
```

Response:

Note: The placeholders shown in violet need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <addImageToQueueSimpleTypesResponse xmlns="http://removethebackground.com/">
      <addImageToQueueSimpleTypesResult>
        <imageTicket>guid</imageTicket>
      </addImageToQueueSimpleTypesResult>
    </addImageToQueueSimpleTypesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Example:

???

deleteImageFromQueue

Deletes an image from the processing queue.

URL:

http://api.removethebackground.com

Method:

POST

Authentication:

Required.

Prototype:

```
String deleteImageFromQueue(String ContactEmail, String ContactAPIkey, String ImageTicket)
```

Parameters:

Parameter	Required	Type	Description
<i>ContactEmail</i>	No	String	The registered email address of the user.
<i>ContactAPIkey</i>	No	String	The API key given to the user.
<i>ImageTicket</i>	No	String	The Ticket id of the image.

Q: All these parameters have been shown as optional, but shouldn't ImageTicket be mandatory?

Returns:

When this method is called, the system sends a deleteImageFromQueue request to the service. In response, the service deletes the image and returns a deleteImageFromQueueResponse message.

Request:

Note: The placeholders shown in violet need to be replaced with actual values.

```
POST /soap.asmx HTTP/1.1
Host: api.removethebackground.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteImageFromQueue xmlns="http://removethebackground.com/">
      <ContactEmail>string</ContactEmail>
      <ContactAPIkey>string</ContactAPIkey>
      <ImageTicket>string</ImageTicket>
    </deleteImageFromQueue>
  </soap12:Body>
</soap12:Envelope>
```

Response:

Note: The placeholders shown in violet need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteImageFromQueueResponse xmlns="http://removethebackground.com/">
      <deleteImageFromQueueResult />
    </deleteImageFromQueueResponse>
  </soap12:Body>
</soap12:Envelope>
```

Example:

???

getImageStatus

Retrieves the status of an image in the processing queue.

URL:

http://api.removethebackground.com

Method:

POST

Authentication:

Required.

Prototype:

String getImageStatus(String *ContactEmail*, String *ContactAPIkey*, String *ImageTicket*)

Parameters:

Parameter	Required	Type	Description
<i>ContactEmail</i>	No	String	The registered email address of the user.
<i>ContactAPIkey</i>	No	String	The API key given to the user.
<i>ImageTicket</i>	No	String	The Ticket id of the image.

Q: All these parameters have been shown as optional, but shouldn't ImageTicket be mandatory?

Returns:

When this method is called, the system sends a getImageStatus request to the service. In response, the service returns a getImageStatusResponse message that contains an error message and code on failure or the following items on success:

Return value	Description
<i>ImageTicket</i>	The Ticket id of the image.
<i>statusName</i>	The description of the status id.
<i>statusId</i>	The status of the image.
<i>finalImageURL</i>	The URL of the final image (if the status id is 70).

The image status (*statusId*) can be one of the following values:

Return value	Description
0	Deleted by user.
10	New image.

Return value	Description
20	New image – awaiting payment.
30	Ready for production.
40	Returned for production (by the user).
60	In production.
70	Production finished.
72	Production finished – delivery failed 1st attempt.
74	Production finished – delivery failed 2nd attempt.
76	Production finished – delivery failed 3rd attempt.
80	Delivered.
90	Rejected by Remove The Background™
95	Awaiting reply from user.
100	Closed.

Request:

Note: The placeholders shown in violet need to be replaced with actual values.

```
POST /soap.asmx HTTP/1.1
Host: api.removethebackground.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getImageStatus xmlns="http://removethebackground.com/">
      <ContactEmail>string</ContactEmail>
      <ContactAPIkey>string</ContactAPIkey>
      <ImageTicket>string</ImageTicket>
    </getImageStatus>
  </soap12:Body>
</soap12:Envelope>
```

Response:

Note: The placeholders shown in violet need to be replaced with actual values.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getImageStatusResponse xmlns="http://removethebackground.com/">
      <getImageStatusResult>
        <imageStatus>
          <imageTicket>guid</imageTicket>
          <statusName>string</statusName>
          <statusId>int</statusId>
          <finalImageURL>string</finalImageURL>
        </imageStatus>
      </getImageStatusResult>
    </getImageStatusResponse>
  </soap12:Body>
</soap12:Envelope>
```

```
</imageStatus>  
</getImageStatusResult>  
</getImageStatusResponse>  
</soap12:Body>  
</soap12:Envelope>
```

Example:

???

getTemplates

Retrieves the list of image processing templates available on the Remove The Background™ servers.

URL:

`http://api.removethebackground.com`

Method:

POST

Authentication:

Required.

Prototype:

```
String getTemplates(String ContactEmail, String ContactAPIkey)
```

Parameters:

Parameter	Required	Type	Description
<i>ContactEmail</i>	No	String	The registered email address of the user.
<i>ContactAPIkey</i>	No	String	The API key given to the user.

Q: All these parameters have been shown as optional. What values will it use if no parameter is given?

Returns:

When this method is called, the system sends a `getTemplates` request to the service. In response, the service returns a `getTemplatesResponse` message that contains an error message and code on failure or an array of templates on success. The array of templates contains the following details:

Value	Description
<i>templateId</i>	The id of the template.
<i>templateName</i>	The name of the template.
<i>price</i>	The price for the template.

Request:

Note: The placeholders shown in violet need to be replaced with actual values.

```
POST /soap.asmx HTTP/1.1
Host: api.removethebackground.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```

<soap12:Body>
  <getTemplates xmlns="http://removethebackground.com/">
    <ContactEmail>string</ContactEmail>
    <ContactAPIkey>string</ContactAPIkey>
  </getTemplates>
</soap12:Body>
</soap12:Envelope>

```

Response:

Note: The placeholders shown in violet need to be replaced with actual values.

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getTemplatesResponse xmlns="http://removethebackground.com/">
      <getTemplatesResult>
        <templates>
          <template>
            <templateId>int</templateId>
            <templateName>string</templateName>
            <price xsi:nil="true" />
          </template>
          <template>
            <templateId>int</templateId>
            <templateName>string</templateName>
            <price xsi:nil="true" />
          </template>
        </templates>
      </getTemplatesResult>
    </getTemplatesResponse>
  </soap12:Body>
</soap12:Envelope>

```

Example:

???

REST API Reference

Developers or resellers access the REST APIs with the following API calls, described below.

addImage

Allows the user to add up to two images to the processing queue.

URL:

<http://api.removethebackground.com/REST.svc/Image/>

Input:

Parameters are as follows:

ContactEmail – Registered email address of the user

ContactAPIkey - API key given to the user

developerAPIkey – Developer uses his/her own key here as well as customer's API key in *ContactAPIkey*. A reseller should use the Contact API key

templateId - ID Number of the template

ImageDeliveryInfo - URL of the image to be uploaded

Image2DeliveryInfo - URL of the second image to be uploaded (an optional parameter)

ImageRequestedDeliveryTimeInHours - Number of hours after which the image needs to be sent back; typical values are 4, 24, 72 or 168

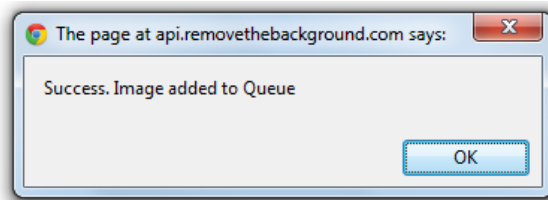
ImageExternalTag - Tag to be returned with the image (an optional parameter)

ImageExternalText - Text to be returned with the image (an optional parameter)

ImageCallbackURL- URL of the results page (an optional parameter). Leave empty to use Template Callback URL

Output:

On successfully adding an image to the queue, the following dialog box will be displayed:



On clicking on the OK button, the system will display the ImageCallback page (or the default Template Callback page, if no ImageCallback URL has been provided) displaying the Image Ticket Id, and the URL of the final processed image.

getStatus

Returns the status of an image in the processing queue.

URL:

<http://api.removethebackground.com/REST.svc/Image/Status/> [*image ticket*] ? ContactEmail=[*ContactEmail*]& ContactAPIkey=[*ContactAPIkey*]

Input:

Parameters are as follows:

image ticket – Image ticket the user gets once he/she adds an image to the Remove The Background

ContactEmail - Registered email address

ContactAPIkey – API key provided by Remove The Background

Output:

The image status returned (*statusId*) can be one of the following values:

Return value	Description
0	Deleted by user.
10	New image.
20	New image – awaiting payment.
30	Ready for production.
40	Returned for production (by the user).
60	In production.
70	Production finished.
72	Production finished – delivery failed 1st attempt.
74	Production finished – delivery failed 2nd attempt.
76	Production finished – delivery failed 3rd attempt.
80	Delivered.
90	Rejected by Remove The Background™
95	Awaiting reply from user.
100	Closed.

getTemplates

Retrieves the list of image processing templates available on the Remove The Background™ servers.

URL:

[http://api.removethebackground.com/REST.svc/Templates/? ContactEmail=\[ContactEmail\]& ContactAPIkey=\[ContactAPIkey\]](http://api.removethebackground.com/REST.svc/Templates/? ContactEmail=[ContactEmail]& ContactAPIkey=[ContactAPIkey])

Input:

Parameters are as follows:

ContactEmail - Registered email address

ContactAPIkey – API key provided by Remove The Background

Output:

The system returns a list of templates in XML. The list contains the following details:

Value	Description
<i>templateId</i>	The id of the template.
<i>templateName</i>	The name of the template.
<i>price</i>	The price for the template.