

Photobucket API Introduction

The Photobucket API allows you to create an interface to "talk" to the Photobucket web site. This means that users do not have to leave your site to perform certain actions on the Photobucket site, such as:

- Uploading images.
- Uploading video.
- Getting all recent media (videos and images) for:
 - A specific user.
 - All users.
 - Group albums.
- Searching media matching a specific term or terms in:
 - One user's account.
 - All user accounts.
 - Group albums.
- Getting all details associated with one piece of media, such as:
 - Link URLs.
 - Thumbnail URL.
- Updating titles, descriptions, and tags.

Using the Help

The documentation for the Photobucket API is provided as an online help. It contains the following sections:

Topic/Book	Description
Photobucket API Introduction	An overview of the Photobucket API.
What's New	Provides a list of changes since the last release.
Getting Started Book	Provides an overview of requirements that must be met before using the Photobucket API, including: • Code conventions. • An explanation of request and response formats. • A "how to" for authentication. • An overview of how to use Photobucket branding, including logo images, font colors, and font faces on your site
Examples Book	Provides examples, including code samples, for the most commonly performed tasks in Photobucket: • Logging In • Uploading Media • Searching Media • Getting a User's Media • Updating a Media Tag
Methods Book	Lists the available methods in the API.

Topic/Book	Description
References Book	Provides lists of code libraries, method throttle limits, and error codes.
Terms of Service	The Photobucket Terms of Service for using the API.

What's New

25-Jun-10

Updated

The [Get Comments](#) and [Get Ratings](#) methods have been reinstated.

Getting Started

The Photobucket API consists of a set of callable methods. The HTTP method, REST path, parameters, example requests and responses, and error codes are listed for each method, a list of which can be found in the [Methods](#) book in the Table of Contents. Methods are also listed for each Example topic. See the [Examples](#) book in the Table of Contents.

Note: The Photobucket API exposes identifiers for users, albums, media, and other uniquely identifiable objects. The format of the IDs can change over time, so relying on the current format may cause problems in the future.

Requirements

Before using the API, you must:

1. Go to the Photobucket developer web site at developer.photobucket.com, agree to the terms of service, sign up, and get the API key.
2. Understand the code sample conventions. See [Conventions](#).
3. Understand the REST request and response format. See [Request Format](#) and [Response Formats](#).
4. Implement OAuth request signing. See [Consumer Authentication](#) for details.
5. Determine which requests you want to send. Occasionally, a request requires data from a previous response.
6. Understand the error codes that may be sent in a response. See [Error Codes](#) for a list of error codes.
7. Review the [throttle limits](#) and if necessary, request a commercial key.

Important! For more information about the documentation, see [Using the Help](#).

Gotchas!

Request Throttling

Each method is given a predetermined number of uses per time period. After this number of uses is used up for the time period, access may be disallowed to that method until the time period expires. Throttling is evaluated per method, per consumer key.

Deauthorized Access Tokens

At any time, a user can turn off your API's access their user account, or a token can expire. The result is an invalid access token. Responses to requests you send with an invalid access token receive a "401: Authorization required" error. You must have the user reauthenticate before you can send requests.

General Service Outages

At times, there may be services outages and methods may not work. To check for a service outage, use the [Ping](#) method and review the error code you receive.

Redirections

Photobucket separates user data into "silos", and you may be required to access a specific host for user-based operations. This affects the way OAuth base strings are generated: all URLs will have the hostname 'api.photobucket.com'. See [Consumer Authentication](#) for details. To allow access to the host information, Consumers should follow 30x "Location:" headers (redirects) for all methods (GET, POST, PUT, and DELETE) and understand the "redirection exception" sent in the body of 30x messages.

API Changes

As needed, changes and updates may be made to the API and the documentation. Remember to check back to ensure you are using the latest version. Note that changes will be tracked in some manner on the developer site.

Options

You can use Photobucket branding and logos when you use the API, if you choose. See [Using Photobucket Branding & Logos](#) for details and images.

REST Request Format

The Photobucket API uses Representational State Transfer (REST) for requests and responses, so that domain-specific data can be transferred over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies. For an explanation of REST conventions, see <http://microformats.org/wiki/rest/urls>

The REST Request Format is a simple HTTP GET, POST, PUT, or DELETE action. Any object that is associated with a user must use the URL provided by Photobucket, via the login process or via the [Get User URL](#) method. Any object not associated with a user can use an anonymous request, sent to <http://api.photobucket.com>.

Sample Method Call Request

```
http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=9fd7bbc4a79d9c593ae3fe80b7d79b53&oauth_signature=%2Fdkp7tjDlMuZjPxbfuuBYFnjvec%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205248683&oauth_version=1.0
```

By default, REST requests send a REST XML response.

Overriding HTTP Method

In some cases, Consumers will be unable to use the required HTTP method to send a request. To override the actual HTTP method, use the `_method` parameter. For example:

```
GET /ping?_method=post
```

Note: The HTTP method part of the OAuth signature does NOT change. `_method` is simply a parameter.

See [Response Formats](#) for details about REST responses.

REST Response Formats

The Photobucket API uses Representational State Transfer (REST) for requests and responses, so that domain-specific data can be transferred over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies. For an explanation of REST conventions, see <http://microformats.org/wiki/rest/urls>.

The REST response format is a simple XML block that includes the action in a <method> tag: HTTP GET, POST, PUT, or DELETE. When using the REST request method, the response defaults to REST. The response format can be specified in the request. See [Request Format](#) for details.

The default response format is XML. All responses are wrapped in a "response envelope" that contains:

- "status" - The response status.
- "format" - The current format.
- "method" - The method that was used to request.
- "timestamp" - The timestamp of the request.
- "content" - The contents of the response.

Exception messages are formatted the same way as success messages, with the exception of a 404 (wrong method) message. A 404 is always in plain text because Photobucket may be unable to construct a response properly.

In the case of an error response (or exception), a "code" and "message" are included which detail the exception case (see [Error Codes](#)). Most likely, there will not be any content. For example:

Sample Failed Method Call Response

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>Exception</status>
  <message>Authentication failed signature check failed</message>
  <code>7</code>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1205248783</timestamp>
</response>
<!-- denlapi05 @ Tue, 11 Mar 2008 09:19:43 -0600 -->
```

Overriding Response Method

In some cases, Consumers will want to receive a response in a format other than XML. Current valid response formats are:

- xml (default)
- json
- jsonp
- phpserialize

To set the response format, use the **format** parameter in the request.

Format: XML (Default)

XML is the default format.

Notes:

- Because of user inputs, there may be XML that does not validate in all parsers.
- Elements may have many child elements and attributes.

Sample Request

```
http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=9fd7bbc4a79d9c593ae3fe80b7d79b53&oauth_signature=%2Fdkp7tjDlMuZjPxbfuBYFnjvec%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205248683&oauth_version=1.0
```

Sample Successful Response

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <pong>(read)</pong>
    <request>
      <format>xml</format>
      <oauth_consumer_key>00000000</oauth_consumer_key>
      <oauth_nonce>9fd7bbc4a79d9c593ae3fe80b7d79b53</oauth_nonce>
      <oauth_signature>/dkp7tjDlMuZjPxbfuBYFnjvec=</oauth_signature>
      <oauth_signature_method>HMAC-SHA1</oauth_signature_method>
      <oauth_timestamp>1205248683</oauth_timestamp>
      <oauth_version>1.0</oauth_version>
    </request>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1205248685</timestamp>
</response>
<!-- denlapi05 @ Tue, 11 Mar 2008 09:18:06 -0600 -->
```

Format: json

JSON is the Javascript Object Notation format. The entire response is one object, with nested properties and objects.

Note: Some values will be one layer deep in an "_attribs" value. Compare this to an elements attributes in XML.

Sample Request

```
http://api.photobucket.com/ping?format=json&oauth_consumer_key=00000000&oauth_
nonce=3b190e959c7e6de518cb1cc31f2c6abc&oauth_signature=f5Arsh02H9KKT4Lg1BOBROgmj00%3D&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1215623625&oauth_version=1.0
```

Sample Response

```
{ "status": "OK", "content": { "pong": "(read)", "request": { "jsonp_callback": "eatResponse", "oauth_consumer_
key": "00000000", "oauth_nonce": "3b190e959c7e6de518cb1cc31f2c6abc", "oauth_
signature": "f5Arsh02H9KKT4Lg1BOBROgmj00=", "oauth_signature_method": "HMAC-SHA1", "oauth_
timestamp": "1215623625", "oauth_
version": "1.0" } }, "format": "jsonp", "method": "GET", "timestamp": 1215623625 }
```

Format: jsonp

JSONP, which is json with a callback, is a protocol supported by YouTube, Google, Social Graph, Twitter and Digg. It provides a way to get to resources with Javascript outside the current domain. To use, set "format=jsonp", and specify a callback function name in the jsonp_callback parameter. The callback can be word characters and dot, supporting OO code. See Format: json (above) for other specifics.

Notes:

- Because jsonp can load scripts from a "foreign" domain, ensure that any code that generates requests is well reviewed to avoid XSS attacks.
- This method does NOT support uploads, and you must force the "_method" parameter, since <script src=""> is always a GET.

Code

```
<script language="text/javascript"> function eatResponse(obj) { alert (obj.timestamp); } </script>
<script language="text/javascript" src="http://api.photobucket.com/ping?format=jsonp&jsonp_
callback=eatResponse&oauth_consumer_key=00000000&oauth_nonce=3b190e959c7e6de518cb1cc31f2c6abc&oauth_
signature=f5Arsh02H9KKT4Lg1BOBR0gmj00%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1215623625&oauth_version=1.0"></script>
```

Sample Request

```
http://api.photobucket.com/ping?format=jsonp&jsonp_callback=eatResponse&oauth_consumer_
key=00000000&oauth_nonce=3b190e959c7e6de518cb1cc31f2c6abc&oauth_
signature=f5Arsh02H9KKT4Lg1BOBR0gmj00%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1215623625&oauth_version=1.0
```

Sample Response

```
eatResponse({"status":"OK","content":{"pong":"(read)","request":{"jsonp_
callback":"eatResponse","oauth_consumer_key":"00000000","oauth_
nonce":"3b190e959c7e6de518cb1cc31f2c6abc","oauth_signature":"f5Arsh02H9KKT4Lg1BOBR0gmj00=","oauth_
signature_method":"HMAC-SHA1","oauth_timestamp":"1215623625","oauth_
version":"1.0"}},"format":"jsonp","method":"GET","timestamp":1215623625});
```

Format: phpserialize

The response is contained in a PHP array, containing values or other arrays.

Note: Some values will be one layer deep in an '_attribs' value. Compare this to an elements attributes in XML

Sample Request

```
http://api.photobucket.com/ping?format=phpserialize&oauth_consumer_key=00000000&oauth_
nonce=3b190e959c7e6de518cb1cc31f2c6abc&oauth_signature=f5Arsh02H9KKT4Lg1BOBR0gmj00%3D&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1215623625&oauth_version=1.0
```

Sample Response

```
a:5:{s:6:"status";s:2:"OK";s:7:"content";a:2:{s:4:"pong";s:6:"(read)";s:7:"request";a:6:{s:18:"oauth_
consumer_key";s:8:"00000000";s:11:"oauth_nonce";s:32:"3b190e959c7e6de518cb1cc31f2c6abc";s:15:"oauth_
signature";s:28:"f5Arsh02H9KKT4Lg1BOBR0gmj00=";s:22:"oauth_signature_method";s:9:"HMAC-
SHA1";s:15:"oauth_timestamp";s:10:"1215623625";s:13:"oauth_
version";s:3:"1.0";}}s:6:"format";s:12:"phpserialize";s:6:"method";s:3:"GET";s:9:"timestamp";i:1215623625;}
```

Consumer Authentication

To secure and verify API Consumers, each API call must be authenticated with the following procedure. This procedure uses the OAuth Consumer/Service Provider authentication protocol.

Notes:

- See <http://oauth.net/core/1.0> and <http://oauth.net/> for information about OAuth.
- See <http://destroytoday.com/blog/2010/02/encoding-for-oauth-using-as3/> for information on using OAuth for AS3 applications.

Requirements

Before you can send requests to Photobucket, you must:

- Get an OAuth Consumer Key (<consumer_key>) and an OAuth Consumer Secret (<consumer_secret>) from level-oper.photobucket.com.
- Get the current timestamp (<ts>) from *your* server. This is the seconds since the epoch, GMT (typically time() or Date.now()). The timestamp relies on the hash generator having a clock that is close to correct.

Note: Client computers frequently have clocks that are incorrect. You may need to get the time from an independent

source.

- Get a random string per timestamp (<nonce>) to add randomness and prevent replay attacks. This requires a random number generator that is random enough for each request.
- Have the ability to generate HMAC hashes, using the SHA1 algorithm, to generate the request signature. Most languages have libraries available to do this.

Authentication

After you have satisfied the requirements, you can request a page anonymously or request a page as a user.

Notes:

- The signatures DO expire. This timeout is set by Photobucket as needed, so always generate a new signature per request.
- Signed requests are required for EVERY operation.

Requesting a Page Anonymously

Many requests to Photobucket do not require a user to be logged in, such as search or getting public albums. The following process follows the [OAuth Consumer Request](#) procedure.

- Use the Photobucket API libraries that are provided, and use the Consumer Key and Consumer Secret as necessary. This signs your request appropriately.
- Use another OAuth request library. (There are a few available from [OAuth.net's repository](#)). The request procedure is similar to the "request token" procedure from the OAuth Core specification, so OAuth libraries can use that procedure. Remember that the URL **ALWAYS** has the hostname "api.photobucket.com", regardless of the actual host being called, to avoid confusion during redirects that may happen.

Signing Process

1. Gather your parameters:
 - Your Consumer Key (<consumer_key>) - Received from Photobucket.
 - Your Consumer Secret (<consumer_secret>) - Received from Photobucket.
 - Timestamp of Request (<timestamp>) - Current seconds since epoch, GMT.
 - Random Nonce string (<nonce>) - Random string for this request.
 - OAuth Version (<version>) - Always '1.0' for this API.
 - Signature Method (<signature_method>) - Always 'HMAC-SHA1' for this API.
 - URL to call (<url>) - Whole URL, up to the '?' of the querystring that you are requesting. All calls to this API have the hostname "api.photobucket.com" regardless of the hostname you are actually calling. E.g., you might request api123.photobucket.com, based on the user you are accessing, but the string provided to the OAuth "Base String" in step 2 is always "http://api.photobucket.com".
 - Request Parameters (<params>) - List of request parameters to call, typically represented as an array of (key,value) pairs.
2. Generate the Base String:
 - Normalize the parameters:
 - Add the OAuth specific parameters for this request to the input parameters, including:
 - oauth_consumer_key = <consumer_key>
 - oauth_timestamp = <timestamp>
 - oauth_nonce = <nonce>

- `oauth_version = <version>`
- `oauth_signature_method = <signature_method>`
- Sort the parameters by name lexicographically (byte ordering, the standard sorting, not natural or case insensitive). If the parameters have the same name, then sort by the value.
- Encode the parameter values as in [RFC3986 Section 2](#) (i.e., urlencode).
- Create parameter string (`<paramstring>`). This is the same format as HTTP 'postdata' or 'querystring', that is, each parameter represented as `name=value` separated by `&`. For example, `a=1&b=2&c=hello%20there&c=something%20else`
- Construct the request URL (`<url>`). For Photobucket, this ALWAYS begins with `http://api.photobucket.com`. Then, add the path, ALWAYS without the trailing slash. For example, `http://api.photobucket.com/search/term`.
- Get the HTTP Request Method (`<method>`) - (GET, PUT, POST, DELETE).
- Create the Base String `<base>` by concatenating the elements and encoding each element, separated by `'&'` (urlencode via [RFC3986 Section 2](#)). The string is like `encode(<method>)+&+encode(<url>)+&+encode(<paramstring>)` where `encode()` is the urlencode function and `+` is concatenation. For example, `GET&http%3A%2F%2Fapi.photobucket.com%2Fsearch%2Fterm&a%3D1%26b%3D2%26c%3Dhello%2520there%26c%3Dsomething%2520else`

3. Generate the Signature Hash:

- Calculate the signature value via HMAC-SHA1 with the Base String `<base>` as the data and the Consumer Secret `<consumer_secret>` appended with `&` as the key (`<signature>`). This is done by `hmac_sha1(<base>, <consumer_secret>+&)` where `hmac_sha1` is the hash function and `+` is concatenation. For example, `hmac_sha1(GET&http%3A%2F%2Fapi.photobucket.com%2Fsearch%2Fterm&a%3D1%26b%3D2%26c%3Dhello%2520there%26c%3Dsomething%2520else&consumerkeystring&)`

4. Generate the authenticated URL.

- By appending the signature to the parameters as `oauth_signature`, urlencoded, you can then construct the URL to send as the request. `<authedurl> = <url>?<paramstring>&oauth_signature=tR3%2BTy81lMeYAr%2FFid0kMTYa%2FWM%3D`

Requesting a Page as a Logged-in User

Requests that get private content or write to a user's data (uploads, properties, etc.) must be authenticated as a user. The signing process that follows amends the process in *Requesting a Page Anonymously* and follows the [OAuth Section 6](#) procedure for getting a user-based access token. See [Web User Login](#) for specific details on how to log in as a user.

- Use the Photobucket API libraries that are provided, and use the Consumer Key and Consumer Secret as necessary. Use the login procedure to receive the access token necessary.
- Use another OAuth request library. (There are a few available from [OAuth.net's repository](#)). Remember that the URL **ALWAYS** has the hostname "api.photobucket.com", regardless of the actual host being called, to avoid confusing during redirects that may happen. Use that library to follow the login procedure to receive the access token necessary.

Signing Process

The procedure from the *Requesting a Page Anonymously* section is mostly copied here. Parts added for user authentication are noted in ***bold green italics***.

1. Gather your parameters:

- Your Consumer Key (`<consumer_key>`) - Received from Photobucket.
- Your Consumer Secret (`<consumer_secret>`) - Received from Photobucket.
- ***User Token Key (`<token>`) - Received from access request.***
- ***User Token Secret (`<token_secret>`) - Received from access request.***

- Timestamp of Request (<timestamp>) - Current seconds since epoch, GMT.
- Random Nonce string (<nonce>) - Random string for this request.
- OAuth Version (<version>) - Always '1.0' for this API.
- Signature Method (<signature_method>) - Always 'HMAC-SHA1' for this API.
- URL to call (<url>) - Whole URL, up to the '?', of the querystring that you are requesting. All calls to this API have the hostname "api.photobucket.com" regardless of the hostname you are actually calling. E.g., you might request api123.photobucket.com, based on the user you are accessing.
- Request Parameters (<params>) - List of request parameters to call, typically represented as an array of (key,value) pairs.

2. Generate the Base String:

- Normalize the parameters:
 - Add the OAuth specific parameters for this request to the input parameters, including:
 - oauth_consumer_key = <consumer_key>
 - **oauth_token = <token>**
 - oauth_timestamp = <timestamp>
 - oauth_nonce = <nonce>
 - oauth_version = <version>
 - oauth_signature_method = <signature_method>
 - Sort the parameters by name lexicographically (byte ordering, the standard sorting, not natural or case insensitive). If the parameters have the same name, then sort by the value.
 - Encode the parameter values as in [RFC3986 Section 2](#) (i.e., urlencode).
 - Create parameter string (<paramstring>). This is the same format as HTTP 'postdata' or 'querystring', that is, each parameter represented as name=value separated by &. For example, a=1&b=2&c=hello%20there&c=something%20else
- Construct the request URL (<url>). For Photobucket, this ALWAYS begins with http://api.photobucket.com. Then, add the path, ALWAYS without the trailing slash. For example, http://api.photobucket.com/search/term.
- Get the HTTP Request Method (<method>) - (GET, PUT, POST, DELETE).
- Concatenate the elements by encoding each element (urlencode via [RFC3986 Section 2](#)), separated by '&'. This is the Base String <base>. The string is like encode(<method>)+&+encode(<url>)+&+encode(<paramstring>) where encode() is the urlencode function and + is concatenation. For example, GET&http%3A%2F%2Fapi.photobucket.com%2Fsearch%2Fterm&a%3D1%26b%3D2%26c%3Dhello%2520there%26c%3Dsomething%2520else

3. Generate the Signature Hash:

- Calculate the signature value via HMAC-SHA1 with the Base String <base> as the data and the Consumer Secret <consumer_secret> appended with &, **and appended with the User Token Secret <token_secret>**, as the key (<signature>). This is done by hmac_sha1(<base>, <consumer_secret>+&+<token_secret>) where hmac_sha1 is the hash function and + is concatenation. For example, hmac_sha1(GET&http%3A%2F%2Fapi.photobucket.com%2Fsearch%2Fterm&a%3D1%26b%3D2%26c%3Dhello%2520there%26c%3Dsomething%2520else, consumersecretstring&**tokensecretstring**)

4. Generate the authenticated URL.

- By appending the signature to the parameters as oauth_signature, urlencoded, you can then construct the URL to send as the request. <authedurl> = <url>?<paramstring>&oauth_signature=tR3%2BTy81lMeYAr%2FFid0kMTYa%2FWm%3D

See the code examples in [Logging In](#), and also see [End User Authentication](#).

End-User Authentication

Consumers must get permission from both the End-User and Photobucket to access private data for a user or to write data to a user's resources. Consumers must get the same permissions to work with group albums. Like personal albums, uploads and edits can only be performed in a group album after the user is logged in, and the user can only act on media or groups to which that user has contributed or owns.

Photobucket has implemented the OAuth authentication method to allow you to get these permissions, and you can use web or direct authentication. See [Consumer Authentication](#) for the specifics of signing a request. See [Web Authentication](#) and [Direct Authentication](#) for details about the types of authentication.

Recommended Reading

- [OAuth Core Spec](#)
- [Photobucket Consumer Authentication](#)

Web Authentication

Requirements

To implement End-User authentication, you must:

- Read, understand, and implement the Photobucket [Consumer Authentication](#) requirements.
- Have the ability to show End-Users an external web page (HTML, CSS, Javascript).

As well, End-Users must be able to accept cookies.

If you cannot meet these requirements, please contact Photobucket to request alternatives. Or, if you are a partner, you can use the [Direct Authentication](#) method.

Condensed Procedure

The following web authentication procedure is the condensed version.

Note: The web authentication procedure outlined here is the greatly preferred method of authenticating users.

1. Get a Login Request token by an HTTP POST to <http://api.photobucket.com/login/request>.
 - Receive an `oauth_token` and `oauth_token_secret`: `oauth_token=<token>&oauth_token_secret=<token_secret>`
 - You may also receive extra information in this response, formatted in the same way. `next_step=<url>` contains the URL of the next step of the End-User Login page for your convenience.
2. Direct the End-User to Login page at: `http://photobucket.com/apilogin/login?oauth_token=<token>`.
 1. End-User logs in.
 2. End-User authorizes your application.
 3. End-User is directed back to your application.
3. Receive a "ready" response from Photobucket in the callback redirect: `<callback>oauth_token=<token>&-username=<user>&subdomain=<api###.photobucket.com>`
4. Get a User Access token by an HTTP POST to: `http://api.photobucket.com/login/access?oauth_token=<token>&<oauth_parameters>`
 - Receive a `oauth_token` and `oauth_token_secret`: `oauth_token=<token>&oauth_token_secret=<token_secret>`
 - You may also receive extra information in the response, formatted the same way.
`username`=username contains the user who logged in.
`subdomain`=api###.photobucket.com contains the sub-domain that is required to access the logged in user's

media.

homeurl=url contains the url of the logged in user's photo album.

See the method [Web User Login](#) for details.

Procedure

End-User web authentication happens in essentially four parts, per [OAuth Section 6](#):

1. Consumer gets a Request Token.
2. Consumer directs End-User to User Login/Authorization site on Photobucket.
3. Photobucket directs End-User back to Consumer, to notify that Login/Authorization is complete.
4. Consumer gets a User Access Token by providing the now authorized Request Token.

Notes:

- Request tokens DO expire if any of the steps take too long. If a Request token expires, the procedure can stall. The expiration date is refreshed at each step. So as long as activity occurs, the Request token should not expire.
- Remember to sign the User Access Token method as you might a user request, but with the request token and request token secret.
- Information sent to the End-User login from the Consumer is available to the End User - do not put too much information here.

See the method [Web User Login](#) for details.

1. Consumer gets a Request token.

The Request Token is an unauthorized token that can only be used to request an access token after it has been authorized. See [Requesting a Page Anonymously](#) for details about the Request Token signing procedure.

- Use the Photobucket API libraries that are provided, and issue a "login request".
- Use another OAuth request library and issue a POST to "/login/request". (There are a few available from [OAuth.net's repository](#)).

The response from either of the preceding login requests contains an `oauth_token` (<token>) and an `oauth_token_secret` (<token_secret>), in the text format "`oauth_token=<token>&oauth_token_secret=<token_secret>`", as specified in the OAuth spec.

This token and token secret are used in the login and access requests. The response also contains `next_step`, which is the URL to the Login page for the user.

2. Consumer directs End-User to user login and authorization on Photobucket.

The End-User must be directed to Photobucket to continue the login process and authorize the request token. Typically, a Consumer site redirects the End-User to the Photobucket API Login and Authorization page at: <http://photobucket.com/apilogin/login>.

Requests to this page are NOT signed via the OAuth signature methods.

The Photobucket API Login and Authorization page requires the `oauth_token` parameter, and allows the extra parameter:

- **oauth_token (required)** - The token gathered from step 1: "Consumer gets a request token".
- **extra (optional)** - Use this parameter to store a session ID, etc., in case the Consumer needs it after the login is complete. The values for extra are limited to the characters `[a-zA-Z0-9_+%-]` (word characters, numbers, and urlencoded values). The extra parameter is not otherwise modified by Photobucket and is passed back to the Consumer as-is.

The redirect therefore is minimally:

```
http://photobucket.com/apilogin/login?oauth_token=<token>
```

Login Page

The End-User is presented an HTML web page that contains inputs for their login name and password, and a checkbox to remember their login between browser sessions. On submit:

- If the login is successful, a cookie is issued to the End-User that represents their login on Photobucket, and the End-User is directed to the API authorization page.
- If the login is not successful, the End-User is presented the login page again.

Authorization Page

After successfully logging in, the End-User is presented with an HTML web page that has button option to "Accept" or "Deny" your application from accessing their account. On submit:

- If "Accept" is selected, the request token is signed and the End-User is redirected back to the Consumer via the Consumer's pre-set callback URL. Proceed to step 3 and the *Callback* section.
- If "Deny" is selected, the request token is NOT signed. The End-User is redirected back to the Consumer via the Consumer's pre-set callback URL. Proceed to step 3 and the *Callback* section.
- Regardless of whether the End-User selects "Accept" or "Deny", if a callback was not set, the End-User is directed to a "Finish" page. Proceed to step 3 and the *Login Complete Page* section.

3. Photobucket directs End-User back to Consumer.

Callback

After the End-User has completed the authorization, the End-User is redirected to the Consumer via the pre-set callback URL. The callback URL is appended with these parameters:

- **oauth_request** - Request token that started the chain of authorization.
- **status** - Status of the callback:
 - **ready** - The request token is authorized and ready.
 - **denied** - The request token is not signed because the End-User denied access to their account on the Authorization page.
 - **canceled** - The request token is not signed because the End-User canceled the login process on the Login page.
- **extra** - Optional parameter provided at the beginning of the login/authorization process. It is sent back as-is.

The Consumer is expected to use the callback URL parameters to continue the login process.

Login Complete Page

If a callback URL was not set by the Consumer, an HTML page that states the process is complete is shown.

- If the End-User allowed the Consumer to access their account, the Login Complete page contains the request token used during the login process. If the application requires it, the End-User can use those items to continue the login process.
- If the End-User canceled the process or denied access, the Login Complete page is the end of the process.

4. Consumer gets an authorized User Access token.

The User Access token is used for all other user-authorized requests. See [Requesting a Page as a Logged-in User](#) as that is the same signing procedure for getting an access token from a request token. Only the Get User Access token method takes both the Request token <token> and the Request token secret <token_secret>.

- Use provided Photobucket API libraries and issue an "access request" command, providing the <token> and <token_secret> to the method.
- Use another OAuth request library, and issue a POST to "/login/access", providing the <token> and <token_secret> to the method.

The response from either of these contains an oauth_token <token> and an oauth_token_secret <token_secret>, in the text format 'oauth_token=<token>&oauth_token_secret=<token_secret>' as specified in the OAuth spec.

This User Access token and User Access token secret are used in all subsequent requests requiring a user login.

The following addition information is supplied:

- **Username** – Username of the user who logged in.
- **Subdomain** – API sub-domain that contains the user's media.
- **Homeurl** = URL of the user's home directory (this is an HTML page).

Direct Authentication

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT WITH PHOTOBUCKET](#)

Consumers must get permission from the End User, as well as Photobucket, to access private data for a user, or to write data to a user's resources. To do this, Photobucket implements OAuth's authentication method. See [Consumer Authentication](#) for the specifics of signing a request. This direct login procedure actually bypasses the web authentication method specified in OAuth. It is only intended for partners who do not have access to a full browser.

Requirements

- Read, understand, and implement the Photobucket [Consumer Authentication](#) requirements.
- Agreement and setup with Photobucket to access the direct login method.

See the method [Non-Web Login](#) for details.

Procedure

This procedure is simple, to let mobile and device developers easily log in as a user.

1. POST the username and password to the /login/direct method, signed with an oauth anonymous request (see [Requesting a Page Anonymously](#)). For example: POST /login/direct/username?password=passwd.
2. Receive the oauth_token and oauth_token_secret in the response envelope (via JSON, receipt is in response.content.oauth_token and response.content.oauth_token_secret) . These are the full token and token secret for future requests for this user.

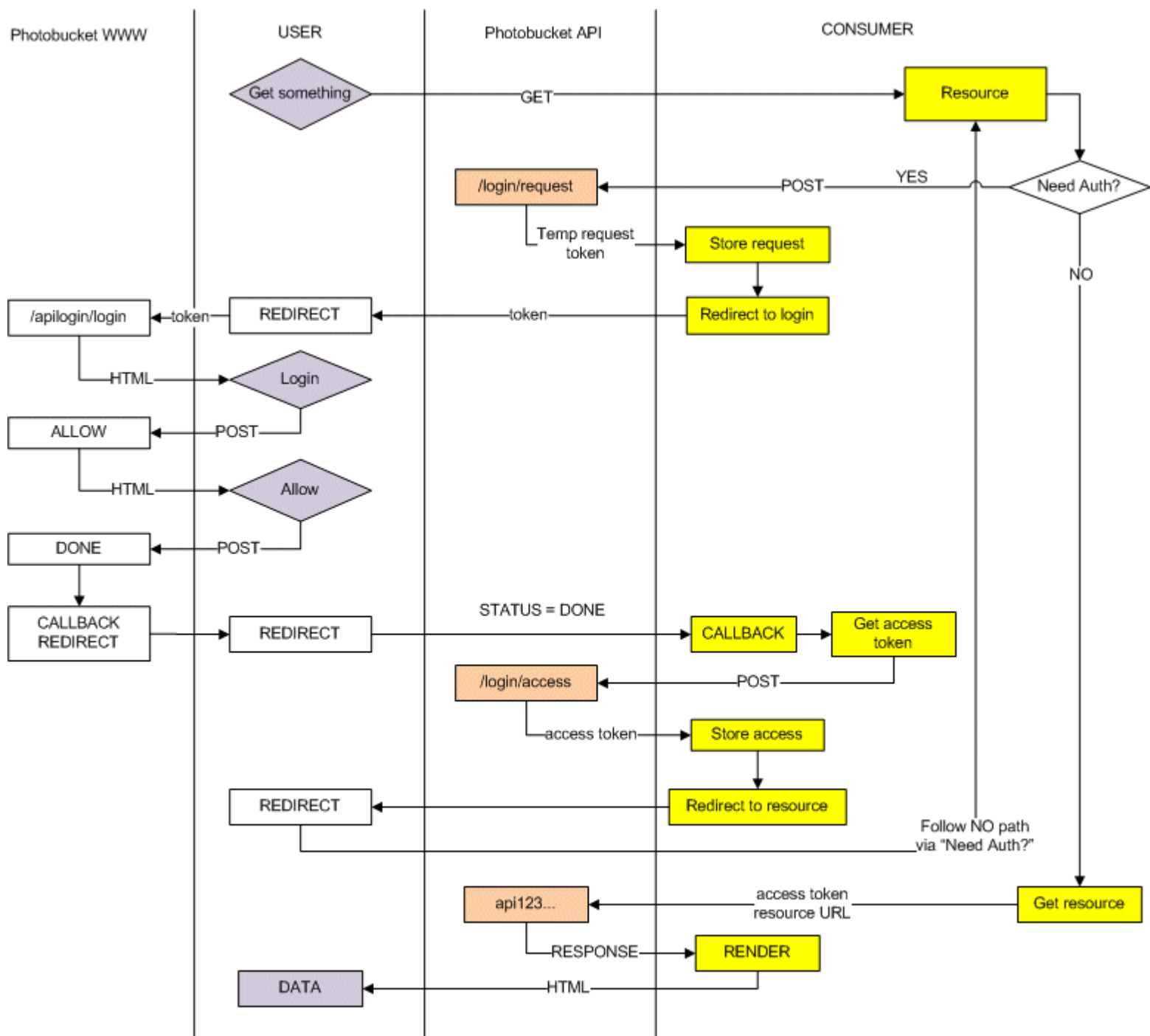
Notes:

- The oauth_token can expire or be reset by the user. Always anticipate an "authorization required", and be prepared to have the user login again.
- This information is not currently encrypted in any way. In the future, there may be a specified hashing or encryption procedure. Consumers are not permitted to store usernames or passwords in any case, and the oauth_token and token_secret should only be stored as absolutely needed.

User Authentication Process Flow

The following flow chart illustrates how a user request flows from the consumer to the Photobucket API, as explained in [Web Authentication](#). When necessary, a request may also flow through www.photobucket.com to authenticate a user before it can be processed by the Photobucket API.

- Purple indicates user interaction is required.
 - Orange indicates API requests.
 - Yellow indicates consumer application actions.
-



Conventions

The Conventions section explains the documentation conventions, the code conventions that are used in the examples and methods, as well as general Photobucket conventions.

Documentation Conventions

In the documentation, the following conventions are observed:

- All Sample Requests in each method use **bold, red text** to call out the action that the request is performing.
- All Sample Responses in each method are XML.
- POST is represented by the request URL, then the www-urlencoded-data of the post in any Sample Requests that use POST.

Example

POST **http://api.photobucket.com/login/access**

oauth_consumer_key=00000000&oauth_nonce=GOUFIE&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1207064755&oauth_token=req_1a8d7b9bb26f0f25c62691481558f_81448bac652&oauth_signature=2ggT37NAys3P6xZGoSDhp40twF0=

- PUT is represented like GET or DELETE, with all options in the query string, assuming an empty payload in any Sample Request that uses PUT.

Example

PUT **http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=997c5f008a7fb22e76df4cef4633845e&oauth_signature=jXH3fqtAmGGcbajk269e5H%2FhxXg%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443499&oauth_token=28.1&oauth_version=1.0**

Code Conventions

Code Convention	Description
Boolean	All boolean values can be entered as 1/0 or true/false.
Method	All HTTP methods actions are returned in the <method> tag in the successful response. You can override the HTTP method in the request string when necessary.

Photobucket Conventions

Photobucket provides specific URL structures that can be used by applications, supports specific media types, and requires that requests be encoded in a specific manner. The Photobucket web site also has certain structural conventions that pertain to object encoding.

Supported Media Types

Photobucket supports image and video media types. The accepted media types are:

- Image – jpg, png, gif, base 64 encoded image, or bmp (Photobucket re-encodes bmp to jpg).
- Video – 3g2, 3gp, 3gp2, 3gpp, 3p, asf, avi, divx, dv, dvx, flv, gif, moov, mov, mp4, mpeg4, mpg4, mpe, mpeg, mpg, qt, swf, wmv, xvid, or rm. (Photobucket re-encodes all videos to Flash video).

URL Structures

Photobucket provides "special" URL prefaces that you should use for various types of display and linking. The following table pertains to images and videos, lists the sub-domain preface, and provides a description of how it should be used.

Sub-domain	Description
http://s123...	"s" before the sub-domain links to an album or full view (browseurl) in a browser. "s" links display HTML for a full web browser to view only. Using the "s" link is strongly discouraged.
http://g123...	"g" before the sub-domain links to a group album or full view (browseurl) in a browser. "g" links display HTML for a full web browser to view only. Using the "g" link is strongly discouraged.

Sub-domain	Description
http:// i 123...	"i" before the sub-domain links directly to an image. You may want to replace "i" with "mob", "mobth", or "w", depending on your application.
http:// gi 123...	"gi" before the sub-domain links directly to an image. You may want to replace "g" with "mob", "mobth", or "w", depending on your application.
http:// vid 123...	"vid" before the sub-domain links directly to a video.
http:// gvid 123...	"gvid" before the sub-domain links directly to a video in a group album.
http:// w 123...	"w" before the sub-domain loads images assets, the slide show core, and individual slide shows.
http:// mob 123...	"mob" before the sub-domain resizes images to 240x240.
http:// mobth 123...	"mobth" before the sub-domain links to a thumbnail 75x75.
http:// feed 123..	"feed" before the sub-domain links directly to an RSS feed. See Photobucket RSS Feeds .
http:// gfeed 123..	"gfeed" before the sub-domain links directly to a group album RSS feed. See Photobucket RSS Feeds .

Request Encoding

The following conventions must be observed when sending requests:

- All GET requests must be encoded with URL encoded parameters.
- All POST, PUT, and DELETE requests must be encoded with URL encoded parameters or with text/www-urlencoded-data.
- The **only** exception is a file upload POST request which must be encoded with multipart/form-data.

Example Multipart Encoded POST Upload Request

```
POST http://api202.photobucket.com/album/pbapi/upload
Content-Type: multipart/form-data; boundary=xx4831d34894cec
--xx4831d34894cec
content-disposition: form-data; name="oauth_version"

1.0
--xx4831d34894cec
content-disposition: form-data; name="oauth_nonce"

6d6961436882002b1a74f0ed2c049d30
--xx4831d34894cec
content-disposition: form-data; name="oauth_timestamp"

1211224904
--xx4831d34894cec
content-disposition: form-data; name="oauth_consumer_key"
```

```

00000000
--xx4831d34894cec
content-disposition: form-data; name="type"

image
--xx4831d34894cec
content-disposition: form-data; name="format"

xml
--xx4831d34894cec
content-disposition: form-data; name="oauth_token"

31.6_1211224899
--xx4831d34894cec
content-disposition: form-data; name="oauth_signature_method"

HMAC-SHA1
--xx4831d34894cec
content-disposition: form-data; name="oauth_signature"

yYYko9YCDLzE2QcEi0fkiaPkhoU=
--xx4831d34894cec
content-disposition: form-data; name="uploadfile"; filename="blueraven.gif"
content-type: mimetype
content-transfer-encoding: binary

GIF89ad_....<rest of binary data>
--xx4831d34894cec

```

Object Identifiers

The Photobucket web site structure is:

User name > User album > Sub-albums > Sub-sub-albums, etc.

The **user name** is ALSO the **main user album name**.

For example, imagine a Photobucket user, **pbapi**. This user has three sub-albums, sub1, sub2, and sub3, and one sub-album, sub4 in album sub1. The user has media in these albums, image1, image2, and video1. SO:

User Name= pbapi

User Album = pbapi

Albums = sub1, sub2, sub3, and sub1/sub4

Media = image1 in the user album, image2 in album sub1, and video1 in album sub1/sub4

Additionally, group albums have their own "group hash value" or **Group ID** in their URLs. The Group ID is a short, all-caps string on the URL for a group. For example, the URL for the featured group album "Woof" is:

<http://gs275.photobucket.com/groups/jj285/20IF1LMOT/?featuredgroupalbum=Wooof>

where **20IF1LMOT** is the Group ID.

Get Album Request Examples

To GET the user album for user "pbapi", the request is:

```
http://api.photobucket.com/album/pbapi
```

To GET album "sub1" for user "pbapi", the request is:

```
http://api.photobucket.com/album/pbapi%2Fsub1
```

To GET album "sub4" for user "pbapi", the request is:

```
http://api.photobucket.com/album/pbapi%2Fsub1%2Fsub4
```

To GET the user album for the group "Woof" (hash value 20IF1LMOT), the request is:

```
http://api.photobucket.com/group/20IF1LMOT
```

Get Media Request Examples

To GET media from the user album (URL= <http://api.photobucket.com/albums/gg17/pbapi/image1.gif>), the request is:

```
http://api.photobucket.com/media/http%3A%2F%2Fphotobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fimage1.gif
```

Note: The "gg17" that is part of the URL is assigned by Photobucket. You can get the Photobucket-assigned value with any request that returns media or user information, for example, the Get User URL request or the Search Media request.

To GET media from the album "sub1" (URL= <http://api.photobucket.com/albums/gg17/pbapi/sub1/image2.gif>), the request is:

```
http://api.photobucket.com/media/  
http%3A%2F%2Fphotobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsub1%2Fimage2.gif
```

To GET media from the album "sub1/sub4" (URL= <http://api.photobucket.com/albums/gg17/pbapi/sub1/sub4/video1.flv>), the request is:

```
http://api.photobucket.com/media/  
http%3A%2F%2Fphotobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsub1%2Fsub4%2Fvideo1.flv
```

To GET media from the group album "Woof" (hash value 20IF1LMOT), the request is:

```
http://api.photobucket.com/media/http%3A%2F%2Fphotobucket.com%2Fgroup%2Fjj285%2F20IF1LMOT%2Fimage1.gif
```

More examples are provided in the FAQ. See the topic [What is the \[identifier\] in the Methods, and why does it look the way it does](#).

Alternative Identifier Specifications

Due to incompatibilities with some libraries, encoding slashes (/ to %2F) are not supported in the URL. In situations where slashes must be encoded, it is difficult to sign and make requests to albums and media objects. An alternative identifier specification has been devised as a work-around.

Put "!" in the URI in place of the identifier. The code assumes you are specifying "!" in the parameters (querystring or post).

If you put "!" in the first position (noted as 'identifier' in the documentation), the parameter is "id". E.g., [/user/!/url?id=pbapi](#) or [/album/!?id=pbapi%2Fsecond](#)

If you put "!" in the second position (noted as 'actionIdentifier' in the documentation), the parameter is "aid". E.g., [/user/!/tag/!?id=pbapi&aid=tagname](#) or [/group/!/contributor/!?id=FFCGKTE2&aid=pbapi](#)

Current User Identifier

When a user is logged in, some actions automatically support working on the current logged in user. To do this, specify "-" as the identifier.

Users: All actions (except register and suggest).

Albums: All actions (specifies user's root album).

Using Photobucket Branding and Logos

When you use the Photobucket API, you can use the following font colors and faces for correct branding. Note that font sizes may be adjusted for your particular application, but font colors and faces are a requirement.

Heading 1 – Arial, bold, black

Heading 2 – Arial, bold, blue (#1C64B4)

Heading 3 – Arial regular, black

Re-sizeable Logo

When you use the Photobucket logo, you must use it AS IS. The following requirements must be met:

- The logo should ONLY be used over a solid color background, NEVER over a patterned background.
- Nothing should ever be added to the logo, e.g., extra copy or images.
- Nothing should ever be removed from the logo, e.g., the camera.

Download a [zip file](#) with logos in a variety of sizes.

Examples

The *Examples* section provides end-to-end code samples for various common tasks. Each task may have code samples in multiple languages. Each sample provides a description of the tasks being performed, the prerequisites (if any), and the methods that are used. Methods can be viewed in the [Methods](#) book.

The most common tasks performed on the Photobucket site are:

- [Logging In](#)
- [Uploading Media](#)
- [Searching for Media](#)
- [Getting a User's Media](#)
- [Updating a Media Tag](#)

Values for Examples

The following values are used in all examples:

- Consumer Key <consumer_key>: 1020304
- Consumer Secret <consumer_secret>: 9eb84090956c484e32cb6c08455a667b
- OAuth Version <oauth_version>: 1.0
- OAuth Signature Method <oauth_signature_method>: HMAC-SHA1
- PHP Object init with: \$api = new PBAPI(\$consumer_key, \$consumer_secret)
- Logged In User: pbapi
- Login oauth_token: 32.5726296_264620465
- Login oauth_token_secret: c06406a063b06a06

As well, each example provides raw code and API code. For example:

Ping: GET /ping

Raw Request Example

1. <timestamp> = 1208289833
2. <nonce> = cc09413d20742699147d083d755023f7
3. <method> = GET
4. <url> = http://api.photobucket.com/ping
5. <params> (None)
6. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=cc09413d20742699147d083d755023f7&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208289833&oauth_version=1.0
7. <base> = GET&http%3A%2F%2Fapi.photobucket.com%2Fping&oauth_consumer_key%3D1020304%26oauth_nonce%3Dcc09413d20742699147d083d755023f7%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208289833%26oauth_version%3D1.0 8.
8. <signature> = 2j4y6ocB4d4tTxDoHaNulKpA46c=
9. <authedurl> = http://api.photobucket.com/ping?oauth_consumer_key=1020304&oauth_nonce=cc09413d20742699147d083d755023f7&oauth_signature=2j4y6ocB4d4tTxDoHaNulKpA46c%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208289833&oauth_version=1.0

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <pong>(read)</pong>
    <request>
      <oauth_consumer_key>1020304</oauth_consumer_key>
      <oauth_nonce> cc09413d20742699147d083d755023f7</oauth_nonce>
      <oauth_signature>2j4y6ocB4d4tTxDoHaNulKpA46c=</oauth_signature>
      <oauth_signature_method>HMAC-SHA1</oauth_signature_method>
      <oauth_timestamp>1208289833</oauth_timestamp>
      <oauth_version>1.0</oauth_version>
    </request>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1208289833</timestamp>
</response>
```

API Example (PHP)

```
$resp = $api->ping()->get()->getResponseString();
```

Logging In

Many requests require you to log into Photobucket.

Use the following API methods:

Web User Login

POST /login/request

User action

POST /login/response

Log a User into Photobucket

The examples that follow log a user into the Photobucket system.

Request a Login Token

Raw Request Example

1. <timestamp> = 1208290131
2. <nonce> = 5c4b38eb7e7f0a0603452dfb3bf149d7
3. <method> = POST
4. <url> = http://api.photobucket.com/login/request
5. <params> = (none)
6. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=5c4b38eb7e7f0a0603452dfb3bf149d7&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290131&oauth_version=1.0
7. <base> = POST&http%3A%2F%2Fapi.photobucket.com%2Flogin%2Frequest&oauth_consumer_key%3D1020304%26oauth_nonce%3D5c4b38eb7e7f0a0603452dfb3bf149d7%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208290131%26oauth_version%3D1.0
8. <signature> = jjF2U5hzxxEkTPFrX6GUkr8yl80=
9. <authedurl> = http://api.photobucket.com/login/request
 - a. Postdata: oauth_consumer_key=1020304&oauth_nonce=5c4b38eb7e7f0a0603452dfb3bf149d7&oauth_signature=jjF2U5hzxxEkTPFrX6GUkr8yl80%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290131&oauth_version=1.0

Raw Response Example

oauth_token=req_495c96a95b95e95f95_27502645&oauth_token_secret=a60ab06c06402d528ff&next_step=http%3A%2F%2Fapi.photobucket.com%2Fapi/login%2Flogin

1. Parse response to extract oauth_token and oauth_token_secret for later use.
2. Use next_step for the destination of the user in the Direct User to Login Page step that follows.

API Example (PHP)

1. \$resp = \$api->login('request')->post()->getResponseString();
2. \$api->loadTokenFromResponse();

Direct User to Login Page

Raw Request Example

(HTTP Header) Location: http://photobucket.com/apilogin/login?oauth_token= req_495c96a95b95e95f95_27502645

Raw Response Example

(Anticipate callback at specified URL)

API Example (PHP)

\$api->goRedirect('login');

Get Access Token

Raw Request Example

1. <timestamp> = 1208290327
2. <nonce> = 06ab696fc74d6fcf4a98f80ceb1c427f
3. <method> = POST
4. <url> = http://api.photobucket.com/login/access
5. <params> = oauth_token= req_495c96a95b95e95f95_27502645
6. <token_secret> = a60ab06c06402d528ff (from request)
7. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=06ab696fc74d6fcf4a98f80ceb1c427f&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290327&oauth_token= req_495c96a95b95e95f95_27502645&oauth_version=1.0
8. <base> = POST&http%3A%2F%2Fapi.photobucket.com%2Flogin%2Faccess&oauth_consumer_key%3D1020304%26oauth_nonce%3D06ab696fc74d6fcf4a98f80ceb1c427f%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208290327%26oauth_token%3Dreq_495c96a95b95e95f95_27502645%26oauth_version%3D1.0
9. <signature> = 2laqyLUGXg8Y1VxqC1nMx3X3OVI=
10. <authedurl> = http://api.photobucket.com/login/access
 - a. Postdata: oauth_consumer_key=1020304&oauth_nonce=06ab696fc74d6fcf4a98f80ceb1c427f&oauth_signature=2laqyLUGXg8Y1VxqC1nMx3X3OVI%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290327&oauth_token=req_495c96a95b95e95f95_27502645&oauth_version=1.0

Raw Response Example

```
oauth_token=32.5726296_264620465&oauth_token_secret=c06406a063b06a06&username=pbapi&subdomain=api123.photobucket.com&homeurl=http%3A%2F%2Fs123.photobucket.com
```

1. Parse response to extract oauth_token and oauth_token_secret for later use.
2. Parse username to see what user logged in (for subsequent calls) .
3. Parse subdomain to get API domain where you will make user-related calls.
4. Parse homeurl if you want to link to that user's Photobucket album (HTML) .

API Example (PHP)

1. \$api->setOAuthToken(\$oauth_token, \$oauth_token_secret); //from stored request
2. \$resp = \$api->login('access')->post()->getResponseString();
3. \$api->loadTokenFromResponse();

Uploading Media

Images and videos can be uploaded to user albums.

Use the API method [Uploading Media to an Album](#).

Follow the login procedure. See [Logging In](#).

```
POST /album/pbapi/upload
```

Upload Media to a User Album

The example that follows uploads media to the "pbapi" user's album.

Raw Request Example

1. <timestamp> = 1208290538
2. <nonce> = 25246383d508a1ec1955df1b0d439390

3. <method> = POST
4. <url> = http://api123.photobucket.com/album/pbapi/upload
5. <params> = a. oauth_token=32.5726296_264620465 b. type=image c. uploadfile=<image file binary data> d. title='my upload'
6. <token_secret> = c06406a063b06a06
7. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=25246383d508a1ec1955df1b0d439390&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290538&oauth_token=32.5726296_264620465&oauth_version=1.0&title=my%20upload&type=image
8. <base> = POST&http%3A%2F%2Fapi.photobucket.com%2Falbum%2Fpbapi%2Fupload&oauth_consumer_key%3D1020304%26oauth_nonce%3D25246383d508a1ec1955df1b0d439390%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208290538%26oauth_token%3D32.5726296_264620465%26oauth_version%3D1.0%26title%3Dmy%2520upload%26type%3Dimage
9. <signature> = lymFcf1yXtA4JdRRki66ipkCLH8=
10. <authedurl> = http://api123.photobucket.com/album/pbapi/upload

a. Multipart Post Data

- i. oauth_consumer_key=1020304
- ii. oauth_nonce=25246383d508a1ec1955df1b0d439390
- iii. oauth_signature=lymFcf1yXtA4JdRRki66ipkCLH8%3D
- iv. oauth_signature_method=HMAC-SHA1
- v. oauth_timestamp=1208290538
- vi. oauth_token=32.5726296_264620465
- vii. oauth_version=1.0
- viii. title=my%20upload
- ix. type=image

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="image.gif" public="1" type="image" uploaddate="" username="">

    <browseurl>http://s123.photobucket.com/albums/v101/pbapi/?action=view&current=image.gif</browseurl>

    <url>http://i123.photobucket.com/albums/v101/pbapi/image.gif</url>
    <thumb>http://i123.photobucket.com/albums/v101/pbapi/th_image.gif</thumb>
    <title>my upload</title>
  </media>
</content>
<format>xml</format>
<method>GET</method>
  <timestamp>1208292827</timestamp>
</response>
```

API Example (PHP)

```
$resp = $api->album('pbapi')->upload(array('type' => 'image', 'uploadfile' => '@/home/pbapi/image.gif', 'title' => 'my upload'))-
>post()->getResponseString();
```

Searching for Media

You can search user accounts for media items that match a specific term, or terms.

Use the API method [Search Media](#).

```
GET /search/searchterm
```

Search for a Term

The example that follows searches for the terms "foo" or "bar": `/search/foo%20or%20bar`

Raw Request Example

1. `<timestamp> = 1208290873`
2. `<nonce> = 46871d8969d98952810701b705fd44ec`
3. `<method> = GET`
4. `<url> = http://api.photobucket.com/search/foo%20or%20bar`
5. `<params> = perpage=50`
6. `<paramstring> = oauth_consumer_key=1020304&oauth_nonce=46871d8969d98952810701b705fd44ec&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290873&oauth_version=1.0&perpage=50`
7. `<base> = GET&http%3A%2F%2Fapi.photobucket.com%2Fsearch%2Ffoo%2520or%2520bar&oauth_consumer_key%3D1020304%26oauth_nonce%3D46871d8969d98952810701b705fd44ec%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208290873%26oauth_version%3D1.0%26perpage%3D50`
8. `<signature> = q7CaAMa59wO7MzIGiPmPN585OhE=`
9. `<authedurl> = http://api.photobucket.com/search/foo%20or%20bar?oauth_consumer_key=1020304&oauth_nonce=46871d8969d98952810701b705fd44ec&oauth_signature=q7CaAMa59wO7MzIGiPmPN585OhE%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290873&oauth_version=1.0&perpage=50`

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <result page="1" perpage="20" totalpages="9321" totalresults="186407">
      <primary>
        <media description_id="59090811" name="foo_kl.jpg" public="1" type="image" uploaddate=""
username="ARROGANZ_Art">
          <browseurl>http://s251.photobucket.com/albums/gg284/ARROGANZ_
Art/?action=view&amp;current=foo_kl.jpg</browseurl>
          <url>http://i251.photobucket.com/albums/gg284/ARROGANZ_Art/foo_kl.jpg</url>
          <albumurl>http://s251.photobucket.com/albums/gg284/ARROGANZ_Art/</albumurl>
          <thumb>http://th251.photobucket.com/albums/gg284/ARROGANZ_Art/th_foo_kl.jpg</thumb>
          <description />
          <title>Foo</title>
        </media>
        <media description_id="58964976" name="HPIM2193.jpg" public="1" type="image" uploaddate=""
username="RockOn976">
          <browseurl>http://s244.photobucket.com/albums/gg29/RockOn976/?action=view&amp;current=HPIM2193.jpg</browseurl>
          <url>http://i244.photobucket.com/albums/gg29/RockOn976/HPIM2193.jpg</url>
```

```

        <albumurl>http://s244.photobucket.com/albums/gg29/RockOn976/</albumurl>
        <thumb>http://th244.photobucket.com/albums/gg29/RockOn976/th_HPIM2193.jpg</thumb>
    <description />
        <title>Foo foo</title>
    </media>
</primary>
<secondary>
    <media description_id="10876642" name="mcbethactiisceneiv.flv" public="1" type="video"
uploaddate="" username="carlos11_9">
        <browseurl>http://s138.photobucket.com/albums/q258/carlos11_
9/?action=view&current=mcbethactiisceneiv.flv</browseurl>

    <url>http://i138.photobucket.com/player.swf?file=http://i138.photobucket.com/albums/q258/carlos11_
9/mcbethactiisceneiv.flv</url>
        <albumurl>http://s138.photobucket.com/albums/q258/carlos11_9/</albumurl>
        <thumb>http://th138.photobucket.com/albums/q258/carlos11_9/th_
mcbethactiisceneiv.jpg</thumb>
        <description />
        <title>foo</title>
    </media>
</secondary>
</result>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1208292436</timestamp>
</response>

```

API Example (PHP)

```
$resp = $api->search('foo or bar')->get()->getResponseString();
```

Getting a User's Media

You can get all recent media (video and images) for a specific user or all users. You can also get media related to a user's URL.

Use the following API methods:

[Get Album](#) – Get a specified album.

```
GET /album/albumname
```

[Get Media Information](#) – Get media details such as URL, description, title, and tags.

```
GET /media/mediaurl
```

Get a User's Media

The examples that follow get a user's album and information on the user's media.

Get an Album

Album: pbapi/sub

Raw Request Example

1. <timestamp> = 1208290973
2. <nonce> = 79ad7e68667a8797cd3fd741eedb3c02
3. <method> = GET
4. <url> = http://api.photobucket.com/album/pbapi%2Fsub
5. <params> (none)
6. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=79ad7e68667a8797cd3fd741eedb3c02&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290973&oauth_version=1.0
7. <base> = GET&http%3A%2F%2Fapi.photobucket.com%2Falbum%2Fpbapi%252Fsub&oauth_consumer_key%3D1020304%26oauth_nonce%3D79ad7e68667a8797cd3fd741eedb3c02%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208290973%26oauth_version%3D1.0
8. <signature> = V/KrDWUd9XV/fc738bCabRBTonU=
9. <authedurl> = http://api.photobucket.com/album/pbapi%2Fsub?oauth_consumer_key=1020304&oauth_nonce=79ad7e68667a8797cd3fd741eedb3c02&oauth_signature=V%2FKrDWUd9XV%2Ffc738bCabRBTonU%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208290973&oauth_version=1.0

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <album name="pbapi/sub" photo_count="2" subalbum_count="0" username="pbapi" video_count="0">
      <media description_id="" name="Photo2.jpg" public="1" type="image" uploaddate="1195146671"
username="pbapi">

        <browseurl>http://s123.photobucket.com/albums/v101/pbapi/sub/?action=view&current=Photo2.jpg</browseurl>
        <url>http://i123.photobucket.com/albums/v101/pbapi/sub/Photo2.jpg</url>
        <thumb>http://i123.photobucket.com/albums/v101/pbapi/sub/th_Photo2.jpg</thumb>
        <description>Four Five</description>
        <title>Three 3</title>
      </media>
      <media description_id="" name="Photo1.jpg" public="1" type="image" uploaddate="1195146668"
username="pbapi">

        <browseurl>http://s123.photobucket.com/albums/v101/pbapi/sub/?action=view&current=Photo1.jpg</browseurl>
        <url>http://i123.photobucket.com/albums/v101/pbapi/sub/Photo1.jpg</url>
        <thumb>http://i123.photobucket.com/albums/v101/pbapi/sub/th_Photo1.jpg</thumb>
        <description>Two</description>
        <title>One Two</title>
      </media>
    </album>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1208292643</timestamp>
</response>
```

API Example (PHP)

```
$resp = $api->album('pbapi/sub')->get()->getResponseString();
```

Get Information on a Media Item

Media: <http://i202.photobucket.com/albums/v101/pbapi/image.gif>

Raw Request Example

1. <timestamp> = 1208291053
2. <nonce> = 1983922d23877af45a3b47581f08be6a
3. <method> = GET
4. <url> = <http://api.photobucket.com/media/http%3A%2F%2F202.photobucket.com%2Falbums%2Fv101%2Fpbapi%2Fimage.gif>
5. <params> (none)
6. <paramstring> = oauth_consumer_key=1020304&oauth_nonce=1983922d23877af45a3b47581f08be6a&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208291053&oauth_version=1.0
7. <base> = GET&http%3A%2F%2Fapi.photobucket.com%2Fmedia/http%253A%252F%252F202.photobucket.com%252Falbums%252Fv101%252Fpbapi%252Fimage.gif&oauth_consumer_key%3D1020304&oauth_nonce%3D1983922d23877af45a3b47581f08be6a&oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1208291053%26oauth_version%3D1.0
8. <signature> = E5K+2AS5XZuVxs9D6LWHBzB0W4c=
9. <authedurl> = http://api.photobucket.com/media/http%3A%2F%2F202.photobucket.com%2Falbums%2Fv101%2Fpbapi%2Fimage.gif?oauth_consumer_key=1020304&oauth_nonce=1983922d23877af45a3b47581f08be6a&oauth_signature=E5K%2B2AS5XZuVxs9D6LWHBzB0W4c%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208291053&oauth_version=1.0

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="image.gif" public="1" type="image" uploaddate="" username="">

    <browseurl>http://s123.photobucket.com/albums/v101/pbapi/sub/?action=view&current=image.gif</browseurl>
      <url>http://i123.photobucket.com/albums/v101/pbapi/sub/image.gif</url>
      <thumb>http://i123.photobucket.com/albums/v101/pbapi/sub/th_image.gif</thumb>
      <description>Two</description>
      <title>One Two</title>
    </media>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1208292827</timestamp>
</response>
```

API Example (PHP)

```
$resp = $api->media('http://i202.photobucket.com/albums/v101/pbapi/image.gif')->get()->getResponseString();
```

Updating a Media Tag

You can update titles, descriptions, tags, and layout for existing media.

Use the following API methods to update a media tag:

[Update Media Tag](#) – Update an existing tag.

Follow the login procedure. See [Logging In](#).

PUT /media/mediaurl/tag/tagid?name=tagnameedited

Update a Tag on a Media Item

The example that follows updates the tag "29465" for the media item "http://i202.photobucket.com/albums/v101/pbapi/image.gif".

Raw Request Example

1. <timestamp> = 1208291613
2. <nonce> = 00372d684d6ee246507aedd58a8181ab
3. <method> = PUT
4. <url> = http://api-
photobucket.com/media/http%3A%2F%2Fi202.photobucket.com%2Falbums%2Fv101%2Fpbapi%2Fimage.gif/tag/29465
5. <params> =
 - a. tagurl = -1
 - b. contact = stuff@example.com
 - c. auth_token = 32.5726296_264620465
6. <paramstring> = contact=stuff%40example.com&oauth_consumer_key=1020304&oauth_nonce=00372d684d6ee246507aedd58a8181ab&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1208291613&oauth_token=32.5726296_264620465&oauth_version=1.0&tagurl=-1
7. <base> = PUT&http%3A%2F%2-
Fapi.ph-
otobucket.com%2Fmedia%2Fhttp%253A%252F%252Fi202.photobucket.com%252Falbums%252Fv101%252Fpbapi%252Fimage.gif%
consumer_key%3D1020304%26oauth_nonce%3D00372d684d6ee246507aedd58a8181ab%26oauth_signature_
method%3DHMAC-SHA1%26oauth_timestamp%3D1208291613%26oauth_token%3D32.5726296_264620465%26oauth_ver-
sion%3D1.0%26tagurl%3D-1
8. <signature> = 2Qka35GLud0sKbzDBCAXf4V0/hg=
9. <authedurl> = http://api-
ph-
otobucket.com/media/http%3A%2F%2Fi202.photobucket.com%2Falbums%2Fv101%2Fpbapi%2Fimage.gif/tag/29465?contact=stuff%40
consumer_key=1020304&oauth_nonce=00372d684d6ee246507aedd58a8181ab&oauth_
signature=2Qka35GLud0sKbzDBCAXf4V0%2Fhg%3D&oauth_signature_method=HMAC-SHA1&oauth_times-
tamp=1208291613&oauth_token=32.5726296_264620465&oauth_version=1.0&tagurl=-1

Raw Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>OK</status>
  <content>
    <tagid>29465</tagid>
    <tag>stuff</tag>
    <topLeftXCoord>0</topLeftXCoord>
```

```

    <topLeftYCoord>0</topLeftYCoord>
    <bottomRightXCoord>0.5</bottomRightXCoord>
    <bottomRightYCoord>0.5</bottomRightYCoord>
    <url />
    <contactAddressId>101045</contactAddressId>
    <contactAddress>stuff@example.com</contactAddress>
  </content>
</format>xml</format>
<method>PUT</method>
<timestamp>1208293360</timestamp>
</response>

```

API Example (PHP)

```

$resp = $api->media('http://i202.photobucket.com/albums/v101/pbapi/image.gif')->tag(29465, array('contact' => 'stuff@example.com',
'tagurl' => -1))->put()->getResponseString();

```

Methods

Important! You may not have access to all the methods in the documentation. If you see a method you want to use, and you do not have access, please contact dlapi@photobucket.com, and request access to that method.

Miscellaneous

[Ping Photobucket](#) - Ping the service.

[Get Timestamp](#) - Get the timestamp.

[Web Login](#) - Log in via the Web.

[Non-Web Login](#) - Log in on a mobile device, for example.

[Get Site Login Credentials](#) - Allow the user to log in to the Photobucket end-user site so their browser can directly visit Photobucket and be logged in.

Albums

[Upload Media to an Album](#) - Upload images or videos to an album.

[Get Album URL](#) - Get the Photobucket URL sub-domain and path information for a group album.

[Get Album](#) - Get a specific album.

[Create New Album](#) - Create a new album.

[Rename Album](#) - Rename an album.

[Delete Album](#) - Remove an album.

[Get Album Organization](#) - Get the order of media in an album.

[Set Album Organization](#) - Set the order of media in an album.

[Follow an Album](#) - Follow an album.

[Stop Following an Album](#) - Stop following an album.

[Get Album Following Status](#) - Determine if a user is following a particular album.

[Get Album Privacy Settings](#) - Get privacy settings for an album.

[Update Album Privacy Settings](#) - Change privacy settings for an album.

[Get Album Theme](#) - Get the theme for a specific album.

[Get Album Vanity URL](#) - Get the album vanity URL.

[Share an Album](#) - Share an album via Facebook, MySpace, Twitter, or all.

Group Albums

[Upload Media to a Specific Group Album](#) - Upload images or videos to a group album.

[Create New Group Album](#) - Create a new group album.

[Get Group Album URL](#) - Get the Photobucket URL sub-domain and path information for a group album.

[Get Media for a Group Album](#) - Get media for a group album.

[Get Group Album Contributors](#) - Get a list of contributors for a group album.

[Get Group Album Information](#) - Get information about a group album.

[Set Group Album Information](#) - Create or update information about a group album.

[Get Media Tags for a Group Album](#) - Get tagged media in a group album.

[Get Group Album Vanity URL](#) - Get the group album vanity URL.

[Get Group Album Theme](#) - Get the theme for a specific group album.

[Follow a Group Album](#) - Follow a group album.

[Stop Following a Group Album](#) - Stop following a group album.

[Get Group Album Following Status](#) - Determine if a user is following a particular group album.

[Get Group Album Privacy Settings](#) - Get privacy settings for a group album.

[Update Group Album Privacy Settings](#) - Change privacy settings for a group album.

[Share a Group Album](#) - Share a group album via Facebook, MySpace, Twitter, or all.

Media

[Get Media Information](#) - Get information about a media item.

[Get Media Description](#) - Get a media item description.

[Set Media Description](#) - Create or update a description for a media item.

[Delete Media Description](#) - Delete a media item description.

[Get Media Title](#) - Get a media item title.

[Set Media Title](#) - Create or update a title for a media item.

[Delete Media Title](#) - Delete a media item title.

[Get Media Tag](#) - Get the tag(s) for a media item.

[Add Media Tag](#) - Create a tag for a media item.

[Update Media Tag](#) - Update a tag for a media item.

[Delete Media Tag](#) - Delete a tag for a media item.

[Get Comments](#) - Get comments on a specific piece of media.

[Get Ratings](#) - Get ratings on a specific piece of media.

[Resize Image](#) - Change the size of an image.

[Rotate Image](#) - Change the rotation of an image.

[Get Media Meta Data](#) - Get EXIF details for a media item.

[Get Media Geo-data](#) - Get geographic data for a media item.

[Update Media Geo-data](#) - Update geo-data for a media item.

[Delete Geo-data from Media](#) - Delete geo-data from a media item.

[Get Media Links](#) - Get URL links for a media item.

[Get Related Media](#) - Get media related to a specific URL.

[Rename or Move Media](#) - Rename and/or move a media item.

[Delete Media](#) - Delete a media item.

[Share Media](#) - Send a media URL to an email address or share via Facebook, MySpace, Twitter, or all options.

Search

[Search Images](#) - Search for images.

[Search Videos](#) - Search for videos.

[Search Group Albums](#) - Search group albums for images and videos.

[Get Featured Home Page Categories](#) - Get categories featured on the Home page by Photobucket.

[Get Featured Group Albums](#) - Get a list of featured group albums.

[Get Find Stuff Category Names](#) - Get the category names for the Find Stuff page.

[Get Find Stuff Category Media](#) - Get media in a specific category or featured media for all categories.

[Follow a Search Term](#) - Search for a term and follow the results.

[Stop Following a Search Term](#) - Stop following a search term.

[Get Search Term Following Status](#) - Determine if a user is following a particular search term.

Users

[Get Recent User Media](#) - Get media recently added by a user.

[Get User Contacts](#) - Get contacts for a user.

[Get User Information](#) - Get a user's account information.

[Get Media Tags for a User](#) - Get a user's media tags.

[Get User Upload Options](#) - Get the user's current upload options.

[Update User Upload Options](#) - Update the user's current upload options.

[Get User URLs](#) - Get the URL sub-domain and path information for a user's account.

[Get a User's Group Albums](#) - Get a list of the user's group albums.

[Get Items a User Follows](#) - Get a list of everything the user follows.

[Get Media a User Follows](#) - Get a list of media the user follows.

[Get User Connections](#) - Get a user's "share" connections: Facebook, MySpace, and Twitter.

[Get User Name Availability](#) - Get user name availability and make suggestions if name is not available.

[Register User](#) - Register a new user.

Ping Photobucket

Ping Photobucket to see if the system is up or in maintenance mode.

User Login Required

No.

HTTP Methods

GET, PUT, POST, or DELETE

REST Path

/ping

Parameters

None.

Request Example

All of the following requests ping the Photobucket service to determine availability and test your request.

GET

```
GET http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=4d7e0e129dff7712a331fdd4edd61016&oauth_signature=fHOjGNhDooPApKwqMhmmV%2Fqs%2Bbc%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443498&oauth_token=28.1&oauth_version=1.0
```

PUT

```
PUT http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=997c5f008a7fb22e76df4cef4633845e&oauth_signature=jXH3fqtAmGGcbajk269e5H%2FhxXg%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443499&oauth_token=28.1&oauth_version=1.0
```

POST

```
POST http://api.photobucket.com/ping
```

DELETE

```
DELETE http://api.photobucket.com/ping?format=xml&oauth_consumer_key=00000000&oauth_nonce=ae4ee9e6104d11ec6e2480066aed405f&oauth_signature=vTng%2F%2B2J7B4fSwtBwpnhVJR7qwE%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443499&oauth_token=28.1&oauth_version=1.0
```

Response Examples

All successful responses acknowledge the request that was sent and contain similar content. Only the GET response is shown.

GET

```
<response>
  <status>OK</status>
  <content>
    <PONG> (read)</PONG>
    <REQUEST>
      <format>xml</format>
```

```
<oauth_consumer_key>00000000</oauth_consumer_key>
<oauth_nonce>1bd0b23ce0441b0b3248ae9bc5b14306</oauth_nonce>
<oauth_signature>2ggT37NAys3P6xZGoSDhp40twF0=</oauth_signature>
<oauth_signature_method>HMAC-SHA1</oauth_signature_method>
<oauth_timestamp>1203452960</oauth_timestamp>
<oauth_version>1.0</oauth_version>
</REQUEST>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1202847399</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Timestamp

Get the timestamp.

Note: This method does not require OAuth consumer authentication and can be accessed via a simple GET.

User Login Required

No.

HTTP Method

GET

REST Path

/time

Parameters

None.

Request Example

The request gets a timestamp.

```
GET http://api.photobucket.com/time
```

Response Example

The successful response is the timestamp.

```
1212160711
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Web User Login

Login Token Request

Get a login request token to use during [web authentication](#).

User Login Required

No.

HTTP Method

POST

REST Path

/login/request

Parameters

None.

Request Example

```
POST http://api.photobucket.com/login/request
oauth_consumer_key=0000000&oauth_nonce=GApXBx&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1207064434&oauth_signature=jXH3fqtAmGGcbaJk269e5H%2FhxXg=
```

Response Example

```
oauth_token=req_47811741486bc734861ef20b2cd6f1ae1fbe33a0&oauth_token_secret=0oauth_token=req_
47811001486bc034861ef20b2cd6f1ae1fbe33a0&oauth_token_
secret=001gs1590a74c21b82280000c9c886b4ad4d7e5cfa&next_
step=http%3A%2F%2Fwww.photobucket.com%2Fapi/login%2Flogin
```

Get Access Token

Get the end-user access token and token secret for the request token.

User Login Required

No, but this method gets signed with the request token and request token secret. See [OAuth Section 6](#).

REST Path

/login/access

HTTP Method

POST

Parameters

None.

Request Example

```
POST http://api.photobucket.com/login/access
oauth_consumer_key=00000000&oauth_nonce=GOUFIE&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1207064755&oauth_token=req_1a8d7b9bb26f0f25c62691481558f_81448bac652&oauth_
signature=2ggT37NAys3P6xZGoSDhp40twF0=
```

Response Example

```
oauth_token=00.00&oauth_token_
secret=6c45a02b6e25ffad9168d24fb1ef40fa0126fbb&username=pbapi&subdomain=api123.photobucket.com&homeurl=http
```

Login

Interactive user login that allows an API consumer to access the user's albums. This method is secure for the user.

Note: This method is meant for end users, not consumers. Consumer access to this method is strictly disallowed.

User Login Required

No. Requests to this method are not OAuth-signed.

REST Path

http://photobucket.com/apilogin/login

Parameters

Parameter	Optional	Description	Variable
extra	Y	Extra information to send back to callback.	String
oauth_token	N	Token string given by the login request method.	String

Request Example

```
http://photobucket.com/apilogin/login?oauth_token=req_918b3db09d0754f7999fb093b98668f67f866abe
```

Response

This method redirects to the API to the callback URL as specified in API key sign-up. The parameters given back are:

- oauth_token
- extra, as specified in the parameters

Parameter	Optional	Description	Variable
oauth_consumer_key	N	My consumer key to deauthorize.	String
extra	Y	Extra information to send back to callback.	String

Non-Web User Login

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT](#) WITH PHOTOBUCKET

Use with API consumers that do not have interactive web sessions available, such as consumer electronic devices, mobile devices, or proxy services. User information is potentially available to the API consumer with this method.

Note: This method requires a request for special access (developer.photobucket.com) and an IP address.

User Login Required

No. See [Consumer Authentication](#) for details.

HTTP Method

POST

REST Path

/login/direct/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Important! The REST path `logindirect` is deprecated.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Username, email address, or ID to log in.	String
password	N	User password.	String
hashed	Y	Signals that the password field is an md5 of the actual (recommended to be true!)	Boolean

Request Example

The request logs user "pbapi" into the Photobucket service.

```
POST http://api.photobucket.com/login/direct/pbapi
```

Response Example

The successful response provides a `userauth` that you can use in successive requests:

```
<response>
  <status>OK</status>
  <content>
    <oauth_token>28.1</oauth_token>
    <oauth_token_secret>f768b5a3f56f9ba81bfe982f242ef6733baddb0a</oauth_token_secret>
    <subdomain>api123.photobucket.com</subdomain>
  </content>
```

```
<format>xml</format>
<method>POST</method>
<timestamp>1202841085</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Site Login Credentials

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT](#) WITH PHOTOBUCKET

Interactive user login that allows the user to log in to the Photobucket end-user site so their browser can directly visit Photobucket and be logged in. This method is secure for the user.

Note: This method is meant for end users, not consumers. Consumer access to this method is strictly disallowed.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/sitelogs

Parameters

Parameter	Optional	Default	Description	Variable
redirect	N	true	If true, redirecturl is required. If false, Photobucket issues a cookie, but does not redirect.	Boolean
redirecturl	Y		The Photobucket.com URL to redirect to after the user is logged in.	String

Request Example

The request gets site login credentials.

```
GET http://api.photobucket.com/sitelogs?oauth_consumer_key=00000000&oauth_
nonce=bf9699b9ffd27b32dalea99992bbaef8&oauth_signature=alSvuSzjWsLsLG6qkh%2BAatTJJhQg%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1207772186&oauth_token=28.10_1207772186&oauth_
version=1.0&redirect=false
```

Response Example

The successful response is a redirect to the site you specify in the redirecturl parameter. Note that if you say redirect=false, the response that is output is the typical redirect response, as follows, and you must handle this response appropriately.

```
<response>
  <status>OK</status>
  <format>xml</format>
  <method>GET</method>
```

```
<timestamp>1207772186</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **009**: Insufficient permissions

For a full list of possible errors, see [Error Codes](#).

Albums

The Album methods are:

- [Upload Media to an Album](#)
- [Get Album URL](#)
- [Get Album](#)
- [Create New Album](#)
- [Rename Album](#)
- [Delete Album](#)
- [Get Album Organization](#)
- [Set Album Organization](#)
- [Get Album Vanity URL](#)
- [Get Album Theme](#)
- [Follow an Album](#)
- [Stop Following an Album](#)
- [Get Album Following Status](#)
- [Get Album Privacy Settings](#)
- [Update Album Privacy Settings](#)
- [Share an Album](#)

Upload Media to an Album

Upload media to a specific album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/album/[identifier]/upload

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Album identifier.	String
type	N	image	Media type. Options are image, video, or base64.	String
uploadfile	N, unless URL.		Media encoded with multipart/form-data. Acceptable file types are: - Image – jpg, png, gif, base 64 encoded image, or bmp. - Video –3g2, 3gp, 3gp2, 3gpp, 3p, asf, avi, divx, dv, dvx, flv, gif, moov, mov, mp4, mpeg4, mpg4, mpe, mpeg, mpg, qt, swf, wmv, xvid, rm. Notes: - If the type is base64, this is a regular www-urlencoded-data field with the base64 binary. - Base64 incurs the limits of the size of a post, so be aware that larger files cannot be uploaded this way. - Neither the uploadfile parameter nor the contents should be included in the OAuth base string, which is used to sign upload requests.	String
title	Y		Searchable title to set on the media. Maximum 250 characters.	String
description	Y		Searchable description to set on the media. Maximum 2048 characters.	String
filename	Y	blank	File name to set on the media. If left blank, the file name matches the original name. Valid characters are letters, numbers, underscore (_), hyphen (-), and space.	String
scramble	Y	false	Indicates if the filename should be scrambled. Options are true or false.	Boolean
degrees	Y		Degrees of rotation in 90 degree increments.	String
size	Y		Size to resize an image to. (Images can only be made smaller.) One of: (value => resize result dimensions) 100 => 100x75 160 => 160x120 320 => 320x240 640 => 640x480 800 => 800x600 1024 => 1024x768 1280 => 1280x960, //premium owner only 1600 => 1600x1200, //premium owner only 2048 => 2048x1536, //premium owner only 2592 => 2592x1944, //premium owner only 3648 => 3648x2736, //premium owner only 4000 =>4000x3000, //premium owner only	Integer
read_geo_exif	Y	true	Read the geo-data from exif on the image. This is overwritten by the given parameters, unless this parameter is "force", where the exif will override given parameters.	Boolean or String
latitude	Y		Decimal latitude in degrees (-90,90).	Float
longitude	Y		Decimal longitude in degrees (-180,180).	Float
altitude	Y		Decimal height above sea level (-x,x) in kilometers.	Float
compass	Y		Decimal direction of orientation at image capture (0,360) in degrees.	Float
gps_times-	Y		Timestamp of GPS capture (seconds since epoch).	Float

Parameter	Optional	Default	Description	Variable
tamp				

Request Example

The request uploads the image "image1.gif" to the "pbapi" album.

```
POST http://api123.photobucket.com/album/pbapi/upload
```

Post Form

```
POST http://api123.photobucket.com/album/pbapi/upload
Content-Length: 3047 Content-Type: multipart/form-data; boundary=xYzZY
--xYzZY
Content-Disposition: form-data; name="type" image
--xYzZY
Content-Disposition: form-data; name="title" My logo
--xYzZY
Content-Disposition: form-data; name="description" Logo of PB
--xYzZY
Content-Disposition: form-data; name="uploadfile"; filename="image1.gif" Content-Type: image/gif
<encoded image data>
--xYzZY
Content-Disposition: form-data; name="scramble" false
--xYzZY--
Content-Disposition: form-data; name="format" xml
--xYzZY--
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content description_id="" name="image1.gif" public="1" uploaddate="1202841084" username="pbapi">

    <browseurl>http://s123.photobucket.com/albums/gg17/pbapi/?action=view&current=image1.gif</browseurl>
    <url>http://i123.photobucket.com/albums/gg17/pbapi/image1.gif</url>
    <thumb>http://i123.photobucket.com/albums/gg17/pbapi/th_image1.gif</thumb>
    <description>Logo of PB</description>
    <title>My Logo</title>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1202841084</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **105:** Invalid file for upload
- **106:** Upload error

- **107:** Invalid Album
- **108:** Insufficient privileges to access private album
- **109:** Album is over 1000 file limit
- **110:** No file uploaded
- **111:** Album is over the size limit of:
- **112:** File has invalid MIME type
- **113:** A file with this name already exists
- **121:** Description too long
- **123:** Invalid Media URL
- **130:** Invalid upload type
- **144:** The album name is missing

For a full list of possible errors, see [Error Codes](#).

Get Album URL

Get the Photobucket URL sub-domain and path information for a specific album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]/url

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the album URL for the album "pbapi".

```
GET http://api.photobucket.com/album/pbapi/url?format=xml&oauth_consumer_key=000000000&oauth_nonce=b7d2041c21c050fee2e59de6ae4f55c2&oauth_signature=bRkiAvxoTWO4ryM49%2FKnwzlubl%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628744&oauth_version=1.0
```

Response Example

The successful response includes the sub-domain, album, image, API, and feed, as well as the URL path. Note the **v223** in the path below. A value similar to this displays for all albums and is automatically generated by Photobucket. See [Conventions](#) for details.

```
<response>
  <status>OK</status>
  <content>
    <username>pbapi</username>
    <subdomain>
      <album>s204/album>
      <image>i204/image>
      <api>api204/api>
```

```
<feed>feed204/feed>
</subdomain>
<path>albums/v223/pbapi</path>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1236628744</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Album

Get the specified album.

If you want the most recent 100 images from an album, consider using the [RSS feed](#) for the album.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Album identifier.	String
recurse	Y	false (0)	Gets all sub-albums recursively.	Boolean
view	Y	nested	Put sub-albums in "album" containers or in one flat list. Options are nested or flat.	String
media	Y	all	Limits the media shown. Options are images, videos, none, or all.	String
paginated	Y	false (0)	Allows you to page through results.	Boolean
page	Y	1	Page number. Used if <i>paginated</i> =true.	Integer
perpage	Y	20	Number of results per page. Used if <i>paginated</i> =true	Integer
sortorder	Y	newest	How to sort the results: • newest = Newest First • oldest = Oldest First • alpha = alphabetical order • user = user's sort order	String

Request Example

The request gets albums for user "testuser2".

```
GET http://api123.photobucket.com/album/testuser2?format=xml&oauth_consumer_key=00000000&oauth_nonce=49b71f4e2ce230d163a1dc35200825f4&oauth_signature=qo1jslc4JOue35%2FyVw0S6Oawgnk%3D&oauth_
```

```
signature_method=HMAC-SHA1&oauth_timestamp=1216773002&oauth_token=28.10_1216773002&oauth_
version=1.0&page=1&paginated=true&perpage=5
```

Response Example

The successful response gets albums for the user.

```
<response>

  <status>OK</status>
  <content>

    <album name="pbapi/album2" photo_count="0" subalbum_count="0" username="pbapi"
    video_count="0" />
    <album name="UserVideos" photo_count="0" subalbum_count="0" username="" video_
    count="42" />
    <album name="USERvideos" photo_count="0" subalbum_count="0" username="" video_
    count="0" />

    <media description_id=""
    name="Afternoon200ffshores20Hermosa20Beac.jpg" public="1"
    type="image" uploaddate="1213806606" username="testuser2">

      <brow-
      seurl>http://-
      s123.photobucket.com/albums/cc276/testuser2/?action=view&amp;current=Afternoon200ffsh

      <url>http://-
      i123-
      .photobucket.com/albums/cc276/testuser2/Afternoon200ffshores20Hermosa20Beac.jpg</url>
      <thumb>http://-
      i123.photobucket.com/albums/cc276/testuser2/th_
      Afternoon200ffshores20Hermosa20Beac.jpg</thumb>
      <description />
      <title />
      <isSponsored />

    </media>
    <media description_id="" name="13xolo-1.jpg" public="1" type="image"
    uploaddate="1213044003" username="testuser2">

      <brow-
      seurl>http://-
      s123.photobucket.com/albums/cc276/testuser2/?action=view&amp;current=13xolo-
      1.jpg</browseurl>
      <url>http://-
      i123.photobucket.com/albums/cc276/testuser2/13xolo-
      1.jpg</url>
      <thumb>http://-
      i123.photobucket.com/albums/cc276/testuser2/th_13xolo-
      1.jpg</thumb>
      <description />
      <title />
      <isSponsored />

    </media>
```

```
<media description_id="" name="IMG_006.jpg" public="1" type="image"
uploaddate="1211992459" username="testuser2">

    <browseurl>http://-
s123.photobucket.com/albums/cc276/testuser2/?action=view&current=IMG_
006.jpg</browseurl>

    <url>http://-
i123.photobucket.com/albums/cc276/testuser2/IMG_
006.jpg</url>

    <thumb>http://-
i123.photobucket.com/albums/cc276/testuser2/th_IMG_
006.jpg</thumb>

    <description />
    <title />
    <isSponsored />

</media>

</album>

</content>
<format>xml</format>
<method>GET</method>
<timestamp>1203452950</timestamp>

</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Create New Album

Create a new sub-album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/album/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
name	N	Name of result. Must be between 2 and 50 characters. Valid characters are letters, numbers, underscore (_), hyphen (-), and space.	String

Request Example

The request creates an album for user "pbapi".

```
POST http://api123.photobucket.com/album/pbapi
format=xml&name=thisnewalbum&oauth_consumer_key=000000000&oauth_
nonce=ff84a82e65a6a517e753f5d34846c940&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1236627874&oauth_token=20.000000_1236627874&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1236627874</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **115:** Album name must be between 2 and 50 alphanumeric characters
- **116:** Album already exists
- **117:** Missing album name

For a full list of possible errors, see [Error Codes](#).

Rename Album

Rename the specified album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT

REST Path

/album/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
name	N	Name of result. Must be between 2 and 50 characters. Valid characters are letters, numbers, under-score (_), hyphen (-), and space.	String
identifier	N	Album identifier.	String

Request Example

The request renames the album "album3" to "album2" for user "pbapi".

```
PUT http://api123.photobucket.com/album/pbapi%2Falbum3?format=xml&name=album2&oauth_consumer_
key=00000000&oauth_nonce=7565dbed3b732c5c48a84dfb956f0bd7&oauth_
signature=XBNQcReMNl%2Fkos0Xol7MxH5Qo4I%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443486&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>1</content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202851534</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Delete Album

Delete the specified sub-album. Albums can only be deleted if they are empty (i.e., do not contain images or videos).

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/album/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request deletes "album2" from user "pbapi".

```
DELETE http://api123.photobucket.com/album/pbapi%2Falbum2?format=xml&oauth_consumer_
key=00000000&oauth_nonce=a55232163072a0f0b0d46f8536934b81&oauth_
signature=UkBnV93sgjK3Yd51TD%2BcqZ%2FD4uk%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443488&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>1</content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1202851535</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Album Organization

Get the display order of media in an album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]/organize

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the order of the media in the "pbapi" album.

```
GET http://api123.photobucket.com/album/pbapi/organize?oauth_consumer_key=00000000&oauth_nonce=ba45300583537f99cb21ff7dbd7fbd8a&oauth_signature=5h0OUPVJf%2BZXa56RPopWH5CTZj0%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1223924722&oauth_token=0.0_1223924722&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <order>manual</order>
    <manual_sort>459841_20031119_screen005.jpg</manual_sort>
    <manual_sort>Waterlilies.jpg</manual_sort>
    <manual_sort>raven-small.gif</manual_sort>
    <manual_sort>459841_20031119_screen003.jpg</manual_sort> <manual_sort>522.png</manual_sort>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1223924723</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Set Album Organization

Set the display order of media in an album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/album/[identifier]/organize

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
order_type	N	Order type. Options are: • alpha (alphabetical, a-z) • ascending (by upload age, most current first) • descending (by upload age, most current last) • manual E.g., if a file was uploaded today, yesterday, and the day before, an ascending order_type would place the files in that order (today, yesterday, day before). A descending order_type would place the files in the order day before, yesterday, today.	String
order	Y	Order of media in the album. Must be sent in comma delimited format, with file names in the intended order.	String

Request Example

The request sets the order type for the "pbapi" album.

```
POST http://api123.photobucket.com/album/pbapi/organize
format=xml&oauth_consumer_key=000000000&oauth_nonce=29b9028452cd857ab46784f81899a194&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1236628177&oauth_token=20.000000_1236628177&oauth_
version=1.0&order=gt500.jpg%2Csig3.gif%2Criver.png&order_type=manual
```

Response Example

The successful response

```
<response>
  <status>OK</status>
  <content>1</content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1236628177</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Follow an Album

Follow an album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/album/[identifier]/follow/[actionIdentifier]

This was previously `/subscribe/` but this usage is deprecated. Please use `/follow/`

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

The following parameters have been deprecated: comment, comment_approve, upload_approve, and album_module.

Parameter	Optional	Default	Description	Variable
identifier	N		Album identifier.	String
actionidentifier	Y		What to follow: - comment - Follow comments in the album (must be album owner to pre-approve). Options are true or false. - comment_approve - Follow comment approvals. - upload_approve - Follow approved uploads. - album_module - Follow approved uploads in your album module. Must be logged in. The album_module option displays in the user's <i>People I Follow</i> panel in their Photobucket album. - stats - Follow your stats. Follow emails are sent for any of your choices except album_module.	String
email	Y/N		Email address to which notifications will be sent. This field is required if the user is not logged in to a Photobucket account.	String

Request Example

The request subscribes a user to the album "pbapi".

```
POST http://api.photobucket.com/album/pbapi/follow/upload_approve
format=xml&oauth_consumer_key=000000000&oauth_nonce=1d711b38a422ae3c881a1c423b43a8b0&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1236628506&oauth_token=31.00_1236628505&oauth_version=1.0
```

Response Example

The successful response lists everything that a user follows. The request above follows the upload_approve. The response shows the request was successful, and it also shows that the user is following comments in the same album.

```
<response>
  <status>OK</status>
  <content>
    <comment>
      <email>1</email>
    </comment>
```

```

    <upload_approve>
      <email>1</email>
    </upload_approve>
  </content>
</format>xml</format>
<method>POST</method>
<timestamp>1236628506</timestamp>
</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Stop Following an Album

Stop following an album. This method depends on data returned in [Get Items the User Follows](#).

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/album/[identifier]/follow/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionidentifier	Y	Type to stop following: - comment - Stop following comments in the album (must be album owner to pre-approve). Options are true or false. - comment_approve - Stop following comment approvals. - upload_approve - Stop following approved uploads. - album_module - Stop following approved uploads in your album module. Must be logged in. The album_module option displays in the user's <i>People I Follow</i> panel in their Photobucket album. - stats - Stop following your stats.	String
user_subscription_id	N	User's subscription ID. Get this value from the response in Get Items the User Follows .	String

Request Example

The request deletes all types except "upload_approve".

```

DELETE http://api.photobucket.com/album/pbapi/follow/album_module?format=xml&oauth_consumer_key=0000000000&oauth_nonce=1d711b38a422ae3c881a1c423b43a8b0&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628506&oauth_token=31.00_1236628505&oauth_version=1.0&user_subscription_id=2329911

```

Response Example

The successful response removes the subscription(s) from the user's account.

```

<response>
  <status>OK</status>
  <content>
    <upload_approve>
      <album_module>1</album_module>
    </upload_approve>
  </content>
</format>xml</format>
<method>DELETE</method>
<timestamp>1236628506</timestamp>
</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Album Following Status

Determine if a user is following an album, and which parameters the user is following.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]/follow/[actionidentifier]

This was previously /subscribe/ but this usage is deprecated. Please use /follow/

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionidentifier	Y	Get who's following this item (all types or a specific type). Only if the user's logged in.	String
email	Y/N	Email address to check subscription state. This field is required if the user is not logged in to a Photobucket account.	String

Request Example

The request determines if the album "pbapi" is followed.

```

GET http://api123.photobucket.com/album/pbapi/follow?format=xml&oauth_consumer_key=000000000&oauth_nonce=1ad886a4f668d2d6e4fd84dc0a4e1750&oauth_signature=GXhavBHUhofaToduFSoSWSQJNhk%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628387&oauth_token=31.00_1236628387&oauth_version=1.0

```

Response Example

The successful response lists everything a user is following, based on the parameters in the [Follow an Album](#) method. The response below shows that the user is following comments, via email, in the "pbapi" album.

```

<response>
  <status>OK</status>
  <content>

```

```
<comment>
  <email>1</email>
</comment>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1236628388</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Album Privacy Settings

Get an album's privacy status, i.e., whether it is public or private.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]/privacy

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the privacy setting for the "pbapi" user album.

```
GET http://api123.photobucket.com/album/pbapi/privacy?format=xml&oauth_consumer_key=00000000&oauth_nonce=5d71e23471383d84163f1ff0199a52fd&oauth_signature=blOPgdpig%2F8whmeynpnSZFPoNDc%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443490&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>public</content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202851939</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **143:** The album specified was not found

For a full list of possible errors, see [Error Codes](#).

Update Album Privacy Settings

Change an album's privacy setting, including password.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT or POST

REST Path

/album/[identifier]/privacy

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
privacy	N	Whether the album is public or private.	String
identifier	N	Album identifier.	String
password	Y	Password for a private album, if <i>privacy</i> =private. Passwords must be at least six characters long. They can only contain the characters a-z, A-Z, 0-9, hyphen (-), and underscore (_). Passwords cannot contain spaces or start with a zero. The guest password must also be different than the owner's Photobucket password.	String

Request Example

The request changes the "sub" album privacy setting from private to public for user "summer".

```
PUT http://api123.photobucket.com/album/summer%2Fsub/privacy?format=xml&oauth_consumer_key=00000000&oauth_nonce=aaa16015faf7e5e60d882db02e02a404&oauth_signature=MNJU%2BE65NQSBDkgEeAzD%2BjxYOsE%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443491&oauth_token=28.1&oauth_version=1.0&privacy=public
```

Response Example

The successful response includes the <content> tag which verifies the change.

```
<response>
  <status>OK</status>
  <content>public</content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202851939</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009:** Insufficient permissions
- **107:** Invalid album
- **143:** The album specified was not found

- **144:** The album name is missing
- **145:** The album depth was exceed for public / private
- **146:** Unable to change public private settings
- **147:** Password not changed
- **148:** The privacy parameter is missing
- **149:** The privacy parameter is invalid
- **159:** Album was already public
- **203:** Invalid password format

For a full list of possible errors, see [Error Codes](#).

Get Album Vanity URL

Get the specified album URL.

User Login Required

No.

HTTP Method

GET

REST Path

/album/[identifier]/vanity

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the URL for the "pbapi" album.

```
GET http://api123.photobucket.com/album/pbapi/vanity?format=xml&oauth_consumer_key=00000000&oauth_nonce=d5c0289d1d09aac4c1dc827446e63fc9&oauth_signature=tgj89FPjvtFjlubOnd%2F1fmXSnRU%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214518358&oauth_token=28.10_1214518358&oauth_version=1.0
```

Response Example

The successful response gets the vanity name and URL.

```
<response>
  <status>OK</status>
  <content>
    <name>myvanityname</name>
    <url>http://photobucket.com/myvanityname</url>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1214518359</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **162:** No album vanity defined.

For a full list of possible errors, see [Error Codes](#).

Get Album Theme

Get the theme for a specific user album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/album/[identifier]/theme

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the theme for the user album "pbapi".

```
GET http://api123.photobucket.com/album/pbapi/theme?format=xml&oauth_consumer_key=000000000&oauth_nonce=97d22e2b286203204d86cfa22a88b265&oauth_signature=SNvOFQ6T7niSMq0filDwX%2FngOXI%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628646&oauth_version=1.0
```

Response Example

The successful response includes the theme ID, name, creator, number of uses, color palette, and more.

```
<response>
  <status>OK</status>
  <content>
    <themeID>91</themeId>
    <name>Classic Corners</name>
    <creatorUsername>photobucket_themes</creatorUsername>
    <usageCount>7</usageCount>
    <palette>
      <id>144</id>
      <name>Art Gallery</name>
      <colors>
        <bgColor>#7A7A7A</bgColor>
        <txtClrDef>#FFFFFF</txtClrDef>
        <linkClr>#DCDE7D</linkClr>
        <modClrHdr>#111111</feemodClrHrd>
        <modClrHi>#353535</modClrHi>
      </colors>
    </palette>
  </content>
</response>
```

```

        <modClrBg>#3F3F3F</modClrBg>
    </colors>
    <usageCount>9</usageCount>
</palette>
<border>
    <id>2</id>
    <name>Classic</name>
    <css>classic.css</css>
    <cssPath>/include/css/borders/ninepoint/classic.css</cssPath>
    <thumbUrl>http://pic.photobucket.com/theme/border/9slice/final/classic_blk/preview_
classic_blk.png</thumbUrl>
    <usageCount>42</usageCount>
    <type>framed</type>
</border>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1236628646</timestamp>
</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Share an Album

Share a specific album via Facebook, MySpace, Twitter, or all options.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/album/[identifier]/share/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded album to share.	String
actionIdentifier	N	Comma separated list of services to send to. Options are all, twitter, facebook, or myspace. Default is "all"	String
message	Y	Body of message.	String

Request Example

The request shares album "pbapi".

```

POST http://api123.photobucket.com/album/pbapi/share/all
message=testing1&format=xml

```

Response Example

A successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <myspace>not sent</myspace>
    <facebook>http://www.new.facebook.com/profile.php?id=100001030313912</facebook>
    <twitter>error</twitter>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1202845954</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Group Albums

The Group Album methods are:

- [Upload Media to a Group Album](#)
- [Create New Group Album](#)
- [Get Group Album URL](#)
- [Get Media for a Group Album](#)
- [Get Group Album Contributors](#)
- [Get Group Album Information](#)
- [Set Group Album Information](#)
- [Get Media Tags for a Group Album](#)
- [Get Group Album Vanity URL](#)
- [Follow a Group Album](#)
- [Stop Following a Group Album](#)
- [Get Group Album Following Status](#)
- [Get Group Album Privacy Settings](#)
- [Update Group Album Privacy Settings](#)
- [Share a Group Album](#)

Upload Media to a Group Album

Upload media to a group album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/group/[identifier]/upload

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Album identifier.	String
type	N	image	Media type. Options are image, video, or base64.	String
uploadfile	N, unless URL.		Media encoded with multipart/form-data. Acceptable file types are: - Image – jpg, png, gif, base 64 encoded image, or bmp. - Video –3g2, 3gp, 3gp2, 3gpp, 3p, asf, avi, divx, dv, dvx, flv, gif, moov, mov, mp4, mpeg4, mpg4, mpe, mpeg, mpg, qt, swf, wmv, xvid, rm. Notes: - If the type is base64, this is a regular www-urlencoded-data field with the base64 binary. - Base64 incurs the limits of the size of a post, so be aware that larger files cannot be uploaded this way. - Neither the uploadfile parameter nor the contents should be included in the OAuth base string, which is used to sign upload requests.	String
title	Y		Searchable title to set on the media. Maximum 250 characters.	String
description	Y		Searchable description to set on the media. Maximum 2048 characters.	String
filename	Y	blank	File name to set on the media. If left blank, the file name matches the original name. Valid characters are letters, numbers, underscore (_), hyphen (-), and space.	String
scramble	Y	false	Indicates if the filename should be scrambled. Options are true or false.	Boolean
degrees	Y		Degrees of rotation in 90 degree increments.	String
size	Y		Size to resize an image to. (Images can only be made smaller.) One of: (value => resize result dimensions) 100 => 100x75 160 => 160x120 320 => 320x240 640 => 640x480 800 => 800x600 1024 => 1024x768 1280 => 1280x960, //premium owner only 1600 => 1600x1200, //premium owner only 2048 => 2048x1536, //premium owner only 2592 => 2592x1944, //premium owner only 3648 => 3648x2736, //premium owner only 4000 =>4000x3000, //premium owner only	Integer

Request Example

The request uploads the image "image1-17.gif" to the "FOYJ8ZK56" album.

```
POST http://api123.photobucket.com/group/FOYJ8ZK56/upload
```

Post Form

```
POST http://api123.photobucket.com/group/FOYJ8ZK56/upload
Content-Length: 3047 Content-Type: multipart/form-data; boundary=xYzZY
--xYzZY
Content-Disposition: form-data; name="type" image
--xYzZY
Content-Disposition: form-data; name="title" My logo
--xYzZY
Content-Disposition: form-data; name="description" Logo of PB
--xYzZY
Content-Disposition: form-data; name="uploadfile"; filename="image1-17.gif" Content-Type: image/gif
<encoded image data>
--xYzZY
Content-Disposition: form-data; name="scramble" false
--xYzZY
Content-Disposition: form-data; name="format" xml
--xYzZY
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content description_id="" name="image1-17.gif" public="" type="image" uploaddate="1214504395"
username="pbapi">
    <browseurl>http://gs123.photobucket.com/groups/jj225/FOYJ8ZK56/?action=view&current=image1-
17.gif</browseurl>
    <url>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/image1-17.gif</url>
    <thumb>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/th_image1-17.gif</thumb>
    <description />
    <title />
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1214504394</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Create New Group Album

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT](#) WITH PHOTOBUCKET

Create a new group album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/group

Parameters

Parameter	Optional	Default	Description	Variable
name	N		Name of group album. Must be between 2 and 50 characters. Valid characters are letters, numbers, underscore (_), hyphen (-), and space.	String
vanity	Y		Vanity URL for the group album.	String
uploads	Y	false	Subscribe to upload notifications. Options are true or false.	Boolean
comments	Y	false	Subscribe to comment notifications. Options are true or false.	Boolean
view	Y	public	Set the group album as public or private.	String
password	Y		Password for a private album to restrict users who may access the album.	String
add	Y	viewers	Set users who can add to the group. Options are viewers, moderated_viewers, or owner. <i>Viewers</i> allows any Photobucket user to add to the group, <i>moderated_viewers</i> allows anyone to add to the group, but the owner must approve all additions, and <i>owner</i> allows only the owner to add to the group.	String
descriptions	Y		Description of the group album.	String

Request Example

The request creates a group album for user "pbapi".

```
POST http://api123.photobucket.com/group
add=moderated_
viewer&comments=1&description=description%20goes%20here&format=xml&name=apiexample&oauth_consumer_
key=00000000&oauth_nonce=5cb619d7c58ef4d202e6938eac8aba03&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1247505892&oauth_token=0.0_0000000000&oauth_version=1.0&uploads=1&vanity=api_example_
group&view=public
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <groupHash>AD XK9ZMUOD</groupHash>
    <subdomain>
      <album>gs122</album>
      <image>gi122</image>
      <api>api122</api>
      <feed>gfeed122</feed>
    </subdomain>
    <path>groups/ii129/AD XK9ZMUOD/</path>
  </content>
</format>xml</format>
```

```
<method>POST</method>
<timestamp>1247505892</timestamp>

</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album URL

Get the Photobucket URL sub-domain and path information for a specific group album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/url

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the group album URL for the album "ADP8DAIIFH".

```
GET http://api.photobucket.com/group/ADP8DAIIFH/url?format=xml&oauth_consumer_key=000000000&oauth_nonce=aa38d287dcb9472df2b0f81ee625e233&oauth_signature=moJXFB%2FcAYS%2FS5N96i%2F1Iag%2B81s%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236629245&oauth_version=1.0
```

Response Example

The successful response includes the group hash, sub-domain, album, image, API, and feed, as well as the URL path. Note the **hh251** in the path below. A value similar to this displays for all albums and is automatically generated by Photobucket. See [Conventions](#) for details.

```
<response>
  <status>OK</status>
  <content>
    <groupHash>ADP8DAIIFH</groupHash>
    <subdomain>
      <album>gs393/album>
      <image>gi393/image>
      <api>api393/api>
      <feed>gfeed393/feed>
    </subdomain>
    <path>groups/hh251/ADP8DAIIFH</path>
  </content>
</format>xml</format>
<method>GET</method>
<timestamp>1236629245</timestamp>
```

</response>

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Media for a Group Album

Get media for the specified group album.

If you want the most recent 100 images from an album, consider using the [RSS feed](#) for the album.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Group identifier.	String
media	Y	all	Limits the media shown. Options are images, videos, none, or all.	String
paginated	Y	false	Allows you to page through results.	Boolean
page	Y	1	Page number. Used if <i>paginated</i> =true.	Integer
perpage	Y	20	Number of results per page. Used if <i>paginated</i> =true	Integer
sortorder	Y	newest	How to sort the results: • newest = Newest First • oldest = Oldest First • alpha = alphabetical order • user = user's sort order	String

Request Example

The request gets the album "FOMP2B8AI".

```
GET http://api123.photobucket.com/group/FOMP2B8AI?format=xml&oauth_consumer_key=00000000&oauth_nonce=985ac3bdd3ce0de82f56f35c2d29d903&oauth_signature=IdlpD7E2XAYjgvqi5INIA1f3t%2BI%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214842422&oauth_token=&oauth_version=1.0
```

Response Example

The successful response gets album "FOMP2B8AI".

Note: The media that displays is specific to the user who is logged in. The group album owner sees all media, including unapproved media. A contributor sees only approved media or media that they uploaded, but that may not be approved.

```
<response>
  <status>OK</status>
  <content>
```

```
<group group_hash_value="" name="FOMP2B8AI" photo_count="2" subalbum_count="0"
video_count="0">
```

```
  <media description_id="" name="Wallpaper_Catch_the_Waves_50th.jpg"
public="1" type="image" uploaddate="1214842223"
username="FOMP2B8AI">
```

```
    <browseurl>http://g-
s123.photobucket.com/groups/jj225/FOYJ8ZK56/?action=view&current=image1-
16.gif</browseurl>
```

```
    <url>http://g-
i123.photobucket.com/groups/jj225/FOYJ8ZK56/image1-
16.gif</url>
```

```
    <thumb>http://g-
i123.photobucket.com/groups/jj225/FOYJ8ZK56/th_image1-
16.gif</thumb>
```

```
    <description />
```

```
    <title />
```

```
    <contributor>
```

```
      <username>pbapi</username>
```

```
      <silourl>http://s234.photobucket.com/albums/cc276/pbapi/</silourl>
```

```
    </contributor>
```

```
  </media>
```

```
  <media description_id=""
name="Afternoon20Offshores20Hermosa20Beac.jpg" public="1"
type="image" uploaddate="1214842199" username="FOMP2B8AI">
```

```
    <browseurl>http://gs-
123.photobucket.com/groups/jj266/FOMP2B8AI/?action=view&current=Afternoon20Offsho
i123-
.photobucket.com/groups/jj266/FOMP2B8AI/Afternoon20Offshores20Hermosa20Beac.jpg</url>
```

```
    <thumb>http://gi123.photobucket.com/groups/jj266/FOMP2B8AI/th_
Afternoon20Offshores20Hermosa20Beac.jpg</thumb>
```

```
    <description />
```

```
    <title />
```

```
    <contributor>
```

```
      <username>pbapi</username>
```

```
      <silourl>http://-
s234-
.photobucket.com/albums/cc276/pbapi/</silourl>
```

```
    </contributor>
```

```
  </media>
```

```
</group>

</content>
<format>xml</format>
<method>GET</method>
<timestamp>1214842422</timestamp>

</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album Contributors

Get the contributors to the group album.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/contributor/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionIdentifier	Y	Username to filter on. The default (blank) gets the list of contributors.	String

Request Example

The request filters on user "pbapi" and gets the contributors for the "FOYJ8ZK56" album.

```
GET http://api123.photobucket.com/group/FOYJ8ZK56/contributor/pbapi?format=xml&oauth_consumer_
key=000000000&oauth_nonce=7167a88bbf27d54ccf49c602f4a7cddb&oauth_
signature=A7IL5YYTvztd3oEXNXHoTnQfT0s%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1214504388&oauth_token=28.10_1214504385&oauth_version=1.0
```

Response Example

The successful response looks similar to the following:

```
<response>

  <status>OK</status>
  <content>

    <media description_id="" name="image1-16.gif" public="0" type="image"
    uploaddate="1214503787" username="pbapi">

      <brow-
      seurl>http://g-
      s123.photobucket.com/groups/jj225/FOYJ8ZK56/?action=view&current=image1-
```

```

16.gif</browseurl>
<url>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/image1-
16.gif</url>
<thumb>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/th_
image1-16.gif</thumb>
<description />
<title />

</media>
<media description_id="" name="image1-15.gif" public="0" type="image"
uploaddate="1214503613" username="pbapi">

  <brow-
seurl>http://g-
s123.photobucket.com/groups/jj225/FOYJ8ZK56/?action=view&current=image1-
15.gif</browseurl>
  <url>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/image1-
15.gif</url>
  <thumb>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/th_
image1-15.gif</thumb>
  <description />
  <title />

</media>

</content>
<format>xml</format>
<method>GET</method>
<timestamp>1214504388</timestamp>

</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album Information

Get information about the group album specified.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/info

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets information for the group album “FOYJ8ZK56”.

```
GET http://api123.photobucket.com/group/FOYJ8ZK56/info?format=xml&oauth_consumer_key=00000000&oauth_nonce=7d26fdde585f1dc6b92a6eb50_26401b8&oauth_signature=Q12FWIfiyr2EQScQ2jqGE1iLcH0%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504390&oauth_token=28.10_1214504385&oauth_version=1.0
```

Response Example

The successful response gets details for the group album, thumbnail, description, and title.

```
<response>
  <status>OK</status>
  <content>
    <title>GROUP-API-TITLE</title>
    <description>group api desc.</description>
    <thumburl>http://gi123.photobucket.com/groups/jj225/FOYJ8ZK56/th_6000_black.jpg</thumburl>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1214504390</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Set Group Album Information

Set information about a group album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT or POST

REST Path

/group/[identifier]/info

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
title	Y	Text to set as the title of the group album.	String
description	Y	Text to set as the description for the group album.	String
url	Y	Urlencoded thumbnail URL.	String

Request Example

The request sets information for the group album “FOYJ8ZK56”.

PUT

```
http://api123.photobucket.com/group/FOYJ8ZK56/info?description=group%20api%20desc.&format=xml&oauth_consumer_key=00000000&oauth_nonce=6b161ed403070d3a76898858adbcd0&oauth_signature=jb15c58PShT0AuseSAkZW5YDS1I%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504391&oauth_token=28.10_1214504385&oauth_version=1.0&title=GROUP-API-TITLE&url=http%3A%2F%2Fgi123.photobucket.com%2Fgroups%2Fjj271%2FFNZ00YKL6%2Fth_6000_black-1.jpg
```

Response Example

The successful response is similar to the following:

```
<response>
  <status>OK</status>
  <content>1</content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1214504391</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Media Tags for a Group Album

Get the tag(s) for a media item.

User Login Required

No.

HTTP Method

GET

REST Path

/group/[identifier]/tag/[tagname]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Group name.	String
tagname	Y		Tag name to get media for. If not entered, tag name list is returned.	String
separate	Y	false (0)	If tagname=", separate tag list by contact and URL.	Boolean
page	Y	1	Page number to view, 1 is indexed.	Integer
perpage	Y	20	Number of media per page. Maximum of 200.	Integer

Request Example

The request gets a tag list for the "FOMP2B8AI" group album.

```
GET http://api123.photobucket.com/group/FOMP2B8AI/tag/?format=xml&oauth_consumer_key=00000000&oauth_nonce=49eab98a2b611bbccd6f7fe3b5ea8bf8&oauth_signature=oeEQ8DBQVgK9wZ91mBubGXV7CVU%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214855017&oauth_token=28.10_1214855017&oauth_version=1.0
```

Response Example

The successful response includes the tag.

Note: The media that displays is specific to the user who is logged in. The group album owner sees all media, including unapproved media. A contributor sees only approved media or media that they uploaded, but that may not be approved.

```
<response>
  <status>OK</status>
  <content>
    <name>wave</name>
    <count>1</count>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1214855018</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Follow a Group Album

Follow a group album .

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/group/[identifier]/follow/[actionIdentifier]

This was previously `/subscribe/` but this usage is deprecated. Please use `/follow/`

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

The following parameters have been deprecated: comment, comment_approve, upload, upload_approve, and album_module.

Parameter	Optional	Default	Description	Variable
identifier	N		Album identifier.	String
actionIdentifier	Y		What to follow: - comment - Follow comments in the album (this option is only available for the album owner). - comment_approve - Follow comment approvals. - upload - Follow uploads (this option is only available for the album owner). Options are true or false. - upload_approve - Follow approved uploads. - album_module - Follow approved uploads in your album module. Must be logged in. The album_module option displays in the user's <i>Groups I Follow</i> panel in their Photobucket album. Follow emails are sent for any of your choices except album_module.	
email	Y/N		Email address to which notifications will be sent. This field is required if the user is not	String

Parameter	Optional	Default	Description	Variable
			logged into a Photobucket account.	

Request Example

The request follows the group album "ADP8DAIIFH".

```
POST http://api123.photobucket.com/group/ADP8DAIIFH/follow/upload
format=xml&oauth_consumer_key=149826443&oauth_nonce=6eb50ccec70089bd59d158a3fab87658&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1236629037&oauth_token=31.14_1236629036&oauth_version=1.0&
```

Response Example

The successful response lists ALL the items that a user is following. The request above adds the upload subscription. The response shows the request was successful, and it also shows that the user is following album_module for the same album.

```
<response>
  <status>OK</status>
  <content>
    <upload_approve>
      <album_module>1</album_module>
    </upload_approve>
    <upload>
      <email>1</email>
    </upload>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1236629038</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Stop Following a Group Album

Stop following a group album. This method depends on data returned in [Get Items the User Follows](#).

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/group/[identifier]/follow/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionidentifier	Y	Type to stop following: - comment - Follow comments in the album (this option is only available for the album owner). - comment_approve - Follow comment approvals. - upload - Follow uploads (this option is only available for the album owner). Options are true or false. - upload_approve - Follow approved uploads. - album_module - Follow approved uploads in your album module. Must be logged in. The album_module option displays in the user's <i>Groups / Follow</i> panel in their Photobucket album.	String
user_subscription_id	N	User's subscription ID. Get this value from the response in Get Items the User Follows .	String

Request Example

The request deletes all types except “upload_approve”.

```
DELETE http://api.photobucket.com/group/BOY00FJX67/follow/album_module?format=xml&oauth_consumer_key=000000000&oauth_nonce=1d711b38a422ae3c881a1c423b43a8b0&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628506&oauth_token=31.00_1236628505&oauth_version=1.0&user_subscription_id=2329911
```

Response Example

The successful response removes the subscription(s) from the user's account.

```
<response>
  <status>OK</status>
  <content>
    <upload_approve>
      <album_module>1</album_module>
    </upload_approve>
  </content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1236628506</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album Following Status

Determine if a user is following a group album, and which parameters the user is following.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/follow/[actionIdentifier]

This was previously */subscribe/* but this usage is deprecated. Please use */follow/*

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionIdentifier	Y	Get who's following this item (all types or a specific type). Only if the user's logged in.	
email	Y/N	Email address to check subscription state. This field is required if the user is not logged into a Photobucket account.	String

Request Example

The request determines if the user is following the group album "ADP8DAIIFH" .

```
GET http://api123.photobucket.com/group/ADP8DAIIFH/follow?format=xml&oauth_consumer_key=000000000&oauth_nonce=bca33fba386335d034e23bd10c5bf7fa&oauth_signature=qyIAcqXNk32r3XAd%2B6XdpB87lrk%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628934&oauth_token=31.00_1236628934&oauth_version=1.0
```

Response Example

The successful response lists everything the user follows, based on the parameters in the [Follow a Group Album](#) method. The response shows that the user follows approved uploaded media, via the *Following* panel in their album in the UI.

```
<response>
  <status>OK</status>
  <content>
    <upload_approve>
      <album_module>1</album_module>
    </upload_approve>
  </content>
</format>xml</format>
<method>GET</method>
<timestamp>1236628934</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album Privacy Settings

Get a group album's privacy status, i.e., whether it is public or private.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/privacy

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the privacy setting for the "FOYJ8ZK56" group album.

```
GET http://api123.photobucket.com/group/FOYJ8ZK56/privacy?format=xml&oauth_consumer_key=00000000&oauth_nonce=e3f850b5549d83738cc4f8e370eb0e68&oauth_signature=%2BH%2FxJ2qvWsyPG9%2FI%2BMLj0dDJ6uk%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504387&oauth_token=28.10_1214504385&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <privacy>private</privacy>
    <password>password93</password>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1214504388</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Update Group Album Privacy Settings

Change a group album's privacy setting, including password.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT or POST

REST Path

/group/[identifier]/privacy

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
privacy	N	Whether the album is public or private.	String
identifier	N	Album identifier.	String
password	N	Password for a private album, if <i>privacy</i> =private. Passwords must be at least six characters long. They can only contain the characters a-z, A-Z, 0-9, hyphen (-), and underscore (_). Passwords cannot contain spaces or start with a zero. The guest password must also be different than the owner's Photobucket password.	String

Request Example

The request changes the "FOYJ8ZK56" group album privacy setting from private to public.

```
PUT http://api123.photobucket.com/group/FOYJ8ZK56/privacy?format=xml&oauth_consumer_key=00000000&oauth_nonce=6182ebf9ae9facf5491329a1d403852d&oauth_signature=vI85ZulTP0KftsmYjgHSSFwqD%2BQ%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504385&oauth_token=28.10_1214504385&oauth_version=1.0&privacy=public
```

Response Example

The successful response includes the <content> tag which verifies the change.

```
<response>
  <status>OK</status>
  <content>public</content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1214504386</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Group Album Vanity URL

Get the specified album URL.

User Login Required

No.

HTTP Method

GET

REST Path

/group/[identifier]/vanity

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String

Request Example

The request gets the URL for the "FOYJ8ZK56" group album.

```
GET http://api123.photobucket.com/group/FOYJ8ZK56/vanity?format=xml&oauth_consumer_key=00000000&oauth_nonce=8a2b725b2518ec8930d86d86f657ea63&oauth_signature=Nv2C0o%2BMP7HW2Bmqx3KExiw%2Bb4s%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504395&oauth_token=28.10_1214504385&oauth_version=1.0
```

Response Example

The successful response gets the vanity name and URL.

```
<response>
  <status>OK</status>
  <content>
    <name>Surfing Party</name>
    <url>http://photobucket.com/surfingparty</url>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1214504396</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **162**: No album vanity defined.

For a full list of possible errors, see [Error Codes](#).

Get Group Album Theme

Get the theme for a specific group album.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/group/[identifier]/theme

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Group identifier.	String

Request Example

The request gets the theme for the group album "ADP8DAIIFH".

```
GET http://api123.photobucket.com/group/ADP8DAIIFH/theme?format=xml&oauth_consumer_key=000000000&oauth_nonce=0f9a8c93aae0a52b18976d007b1782f1&oauth_
```

```
signature=KNXY%2FRlRd4YuqmJtkvcglQPEclK%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236629180&oauth_version=1.0
```

Response Example

The successful response includes the theme ID, name, creator, number of uses, color palette, and more.

```
<response>
  <status>OK</status>
  <content>
    <themeID>13</themeID>
    <name>Classic</name>
    <creatorUsername>photobucket_themes</creatorUsername>
    <usageCount>23</usageCount>
    <palette>
      <id>24</id>
      <name>Classic</name>
      <colors>
        <bgColor>#FFFFFF</bgColor>
        <txtClrDef>#000000</txtClrDef>
        <linkClr>#0202CC</linkClr>
        <modClrHdr>#BDBDBD</feemodClrHrd>
        <modClrHi>#E3E4E6</modClrHi>
        <modClrBg>#F7F7F7</modClrBg>
      </colors>
      <usageCount>105</usageCount>
    </palette>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1236629180</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Share a Group Album

Share a specific group album via Facebook, MySpace, Twitter, or all options.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/group/[identifier]/share/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded group album to share.	String
actionIdentifier	N	Comma separated list of services to send to. Options are all, twitter, facebook, or myspace. Default is "all"	String
message	Y	Body of message.	String

Request Example

The request shares the group album "KK1K352".

```
POST http://api123.photobucket.com/group/KK1K352/share/all
message=testing1&format=xml
```

Response Example

A successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <myspace>not sent</myspace>
    <facebook>http://www.new.facebook.com/profile.php?id=100001030313912</facebook>
    <twitter>http://twitter.com/test</twitter>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1273007364</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Media

All Media methods are available for both group albums and personal user albums, provided the [identifier] in the REST path and the ### in api###.photobucket.com point to the same (and desired) silo. See [Why am I redirected to api###.photobucket.com when I access an album](#) for more information.

Example

The direct link to an image in the "Macro" group album in Photobucket is:

```
http://gil112.photobucket.com/groups/n189/D8KDJRG4VO/PA287084frog.jpg
```

If you want to add a media tag, the api### would be:

```
POST
http://
api112
.photobucket.com/media/http%3A%2F%2Fgil112.photobucket.com%2Fgroups%2Fn189%2fD8KDJRG4VO%2fPA287084frog.jpg/ta
```

Because this is a POST example, the url-encoded data must also be sent.

The Media methods are:

- [Get Media Information](#)
- [Get Media Description](#)
- [Set Media Description](#)
- [Delete Media Description](#)
- [Get Media Title](#)
- [Set Media Title](#)
- [Delete Media Title](#)
- [Get Media Tag](#)
- [Add Media Tag](#)
- [Update Media Tag](#)
- [Delete Media Tag](#)
- [Get Comments](#)
- [Get Ratings](#)
- [Resize Image](#)
- [Rotate Image](#)
- [Get Media Meta Data](#)
- [Get Media Geo-data](#)
- [Update Media Geo-data](#)
- [Delete Geo-data from Media](#)
- [Get Media Links](#)
- [Get Related Media](#)
- [Rename or Move Media](#)
- [Delete Media](#)
- [Share Media](#)

Get Media Information

Get information about the media specified.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the image "pb.gif" in the "summer" album for user "pbapi".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsummer%2Fpb.gif?format=xml&oauth_consumer_
key=00000000&oauth_nonce=7c3ad83012aeb564ad6b32645a7cflab&oauth_
signature=FxOsj%2FXuSSeIO%2Bc9X%2BMzb424az0%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443491&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response gets details for the image, such as URL, thumbnail, description, title, and tags.

```
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="pb.gif" public="1" uploaddate="" username="">

    <browseurl>http://s123.photobucket.com/albums/gg17/pbapi/summer/?action=view&current=pb.gif</browseurl>

    <url>http://i123.photobucket.com/albums/gg17/pbapi/summer/pb.gif</url>
    <thumb>http://i123.photobucket.com/albums/gg17/pbapi/summer/th_pb.gif</thumb>
    <description>API pic description</description>
    <title>ApiPic</title>
    <tag bottom_right_x="0" bottom_right_y="0" tag="APITAG" top_left_x="0" top_left_y="0" url=""
  />
    <tag bottom_right_x="0" bottom_right_y="0" tag="APITAG" top_left_x="0" top_left_y="0" url=""
  />
  </media>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1202876757</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **123:** Invalid Media URL

For a full list of possible errors, see [Error Codes](#).

Get Media Description

Get the description for a media item.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/description

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the description for the image "pb.gif", in the "pbapi" user album, for user "pbapi".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fpb.gif/description?format=xml&oauth_
consumer_key=00000000&oauth_nonce=4819a7174840a64f6273e1df4d75036c&oauth_
signature=dE6rR9wxsxX2Mf7ujEkL0kbd0E8%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443494&oauth_token=28.1&oauth_version=1.0
```

Response Example

A successful response includes a description.

```
<response>
  <status>OK</status>
  <content>
    <description>DESCBEN</description>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202845958</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **123:** Invalid Media URL

For a full list of possible errors, see [Error Codes](#).

Set Media Description

Set the description (up to 2048 characters) for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST or PUT

REST Path

/media/[identifier]/description

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
description	N	Text to set as the description.	String

Request Example

The request updates the description to "updateDESC" for the "ferrari.jpg" image.

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2F
ferrari.jpg/description?description=updateDESC&format=xml&oauth_consumer_key=00000000&oauth_
nonce=d0366b583e759caf814941951453f1d8&oauth_signature=isJlPBkpoueyfZzGApFDRd5iZp0%3D&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1205278884&oauth_token=28.1&oauth_version=1.0
```

Response Example

A successful response includes the media description.

```
<response>
  <status>OK</status>
  <content>
    <media>1</media>
  </content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202845957</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009**: Insufficient permissions
- **121**: Description too long
- **141**: Media specified was not found

For a full list of possible errors, see [Error Codes](#).

Delete Media Description

Delete the description for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/media/[identifier]/description

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request deletes the description for the image "pb.gif"

```
DELETE
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fpb.gif/description?format=xml&oauth_
consumer_key=00000000&oauth_nonce=f1432b76044061590c92875c66f031d5&oauth_
signature=mbECwHb3MeN7iGJDdnIOfXoqiy8%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443492&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <media>1</media>
  </content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1202845956</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009:** Insufficient permissions
- **141:** Media specified was not found

For a full list of possible errors, see [Error Codes](#).

Get Media Title

Get the title of a media item.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/title

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the title for the "ferrari.jpg" image.

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fferrari.jpg/title?format=xml&oauth_
consumer_key=00000000&oauth_nonce=d38c5437649a4e557cf110312188b5c7&oauth_
```

```
signature=gCNfjoA%2B3eEmYgPlFGaEL%2FoxWQk%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443497&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes the title.

```
<response>
  <status>OK</status>
  <content>
    <title>ferrariTitle</title>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202845955</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **123:** Invalid Media URL

For a full list of possible errors, see [Error Codes](#).

Set Media Title

Set the title (up to 128 characters) for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST or PUT

REST Path

/media/[identifier]/title

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
title	N	Text to set as the title.	String

Request Example

The request updates the title for the image "ferrari.jpg".

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fferrari.jpg/title?format=xml&oauth_consumer_key=000000000&oauth_nonce=cc25d363e5f72fd31c2dcf2498746a35&oauth_signature=X%2FXdrpmz73gheBXzE%2F%2Bm8Aeay50%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205278888&oauth_token=28.1&oauth_version=1.0&title=ferrariTitle
```

Response Example

A successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <media>1</media>
  </content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202845954</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009:** Insufficient permissions
- **126:** Title too long
- **141:** Media specified was not found
- **152:** The title parameter is required

For a full list of possible errors, see [Error Codes](#).

Delete Media Title

Delete the title of a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/media/[identifier]/title

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request deletes the title for the image "pb.gif".

```
DELETE
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2F%2Fpbapi%2Fpb.gif/title?format=xml&oauth_consumer_
key=00000000&oauth_nonce=c44cfd734ad7e1557c544b2842614c80&oauth_
signature=E8s3wuolcsRhZUPzgfO2MlCFxk8%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443495&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <media>1</media>
  </content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1202845953</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009:** Insufficient permissions
- **141:** Media specified was not found

For a full list of possible errors, see [Error Codes](#).

Get Media Tag

Get the tag(s) for a media item.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/tag/[tagid]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
tagid	Y	Specific tag. If omitted, all tags for the media are returned.	Integer

Request Example

The request gets tag "221" for the "ferrari.jpg" image.

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fferrari.jpg/tag/221?format=xml&oauth_
consumer_key=00000000&oauth_nonce=02290b9e21d1cd9c705f62e8c57d3d7b&oauth_
signature=UXFqFdGMyP2PBfU1%2Fc569KB6Ozc%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1205278892&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes the tag.

```

<response>
  <status>OK</status>
  <content>
    <tagid>221</tagid>
    <tag>TAGNAME</tag>
    <topLeftXCoord>0.1</topLeftXCoord>
    <topLeftYCoord>0.1</topLeftYCoord>
    <bottomRightXCoord>0.1</bottomRightXCoord>
    <bottomRightYCoord>0.1</bottomRightYCoord>
    <url />
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202846938</timestamp>
</response>

```

Error Codes

The following errors may be returned:

- **123:** Invalid Media URL
- **139:** The tag was not found

For a full list of possible errors, see [Error Codes](#).

Add Media Tag

Add tag(s) to a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/media/[identifier]/tag

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
tag	N	String to add as the tag for the media. Maximum of 50 characters. Maximum of 20 tags per image.	String
toleftx	Y	Top left X coordinate, between 0 and 1 as fraction of image dimension.	Float
tolefty	Y	Top left Y coordinate, between 0 and 1 as fraction of image dimension.	Float
bottomrightx	Y	Bottom right X coordinate, between 0 and 1 as fraction of image dimension.	Float
bottomrighty	Y	Bottom right Y coordinate, between 0 and 1 as fraction of image dimension.	Float
contact	Y	Contact ID to add in the tag, or specific email address.	Integer or String
tagurl	Y	URL to link to.	String

Request Example

The request creates a tag for the "gg_avatar_50.jpg" image.

```
POST http://api123.photobucket.com/media/http%3A%2F%2Fphotobucket.com%2Falbums%2Fff180%2Fpbapi%2Fgg_avatar_50.jpg/tag
format=xml&oauth_consumer_key=000000000&oauth_nonce=331d93605b0a3bd2bccd160e025c194a&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1236630577&oauth_token=31.00_1236630577&oauth_version=1.0&tag=tagged
```

Response Example

The successful response includes the tag(s).

```
<response>
  <status>OK</status>
  <content>
    <tagid>182</tagid>
    <tag>tagged</tag>
    <topLeftXCoord>0</topLeftXCoord>
    <topLeftYCoord>0</topLeftYCoord>
    <bottomRightXCoord>0</bottomRightXCoord>
    <bottomRightYCoord>0</bottomRightYCoord>
    <url />
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1236630577</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **123:** Invalid Media URL
- **131:** The tag exceeds the maximum length
- **132:** Tag link is invalid
- **133:** Contact is invalid
- **134:** Tag dimensions are invalid
- **135:** Exceeded maximum number of tags
- **136:** Tag is required

For a full list of possible errors, see [Error Codes](#).

Update Media Tag

Update the tag(s) for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT

REST Path

/media/[identifier]/tag/[tagid]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
tagid	N	Tag to update.	String
tag	Y	String to add as the tag for the media. Maximum of 50 characters. Maximum of 20 tags per image.	String
topleftx	Y	Top left X coordinate, between 0 and 1 as fraction of image dimension.	Float
toplefty	Y	Top left Y coordinate, between 0 and 1 as fraction of image dimension.	Float
bottomrightx	Y	Bottom right X coordinate, between 0 and 1 as fraction of image dimension.	Float
bottomrighty	Y	Bottom right Y coordinate, between 0 and 1 as fraction of image dimension.	Float
contact	Y -1 to remove	Contact ID to add in the tag, or specific email address.	Integer or String
tagurl	Y -1 to remove	URL to add to the tag.	String

Request Example

The request updates tag "221" for the "ferrari.jpg" image.

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fferrari.jpg
/tag/221?bottomrightx=0.2&bottomrighty=0.2&format=xml&oauth_consumer_key=00000000&oauth_
nonce=b9e53fc35c6ad298e159c8cdb796edd7&oauth_signature=B%2BA3pAQBI8rVqgepGyv730K%2BQ04%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1205278893&oauth_token=28.1&oauth_
version=1.0&tag=TAGNAME&topleftx=0.2&toplefty=0.2
```

Response Example

The successful response includes the updated tag(s).

```
<response>
  <status>OK</status>
  <content>
    <tagid>221</tagid>
    <tag>TAGNAME</tag>
    <topLeftXCoord>0.2</topLeftXCoord>
    <topLeftYCoord>0.2</topLeftYCoord>
    <bottomRightXCoord>0.2</bottomRightXCoord>
    <bottomRightYCoord>0.2</bottomRightYCoord>
    <url />
  </content>
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202846937</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **123**: Invalid Media URL
- **131**: The tag exceeds the maximum length
- **132**: Tag link is invalid
- **133**: Contact is invalid
- **134**: Tag dimensions are invalid
- **136**: Tag is required
- **137**: Tag ID is invalid
- **138**: Tag is not owned by the authenticated user
- **139**: The tag was not found
- **140**: Either the tag or the coordinates are required for update

For a full list of possible errors, see [Error Codes](#).

Delete Media Tag

Delete the tag(s) for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/media/[identifier]/tag/[tagid]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL to delete tag from.	String
tagid	N	Specific tag ID to remove.	Integer

Request Example

The request deletes the tag "221" for the "ferrari.jpg" image.

```
DELETE
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fegg17%2Fpbapi%2Fferrari.jpg/tag/221?format=xml&oauth_
consumer_key=00000000&oauth_nonce=77c0b6ec1bb6a69ec52afecd8f2b93f1&oauth_
signature=i1S34rnnvS6XrMryoUpE38yaZJbI%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203452966&oauth_token=28.1&oauth_version=1.0&tagid=64
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
```

```
<format>xml</format>
<method>DELETE</method>
<timestamp>1202846841</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **123:** Invalid Media URL
- **137:** Tag ID is invalid
- **138:** Tag is not owned by the authenticated user
- **139:** The tag was not found
- **141:** Media specified was not found

For a full list of possible errors, see [Error Codes](#).

Get Comments

Get comments on a specific piece of media.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/comment

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Urlencoded full media URL.	String
page	Y	0	Page to start at.	Integer
offset	Y	0	Beginning offset of results. This value overrides the page parameter.	Integer
perpage	Y	10	Number of comments to return per page. Maximum of 50.	Integer

Request Example

The request searches for comments on "00_pb_tattoo-1.jpg" in the "pbuser" album.

```
GET
http://api123.photobucket.com/media/http%3A%2F%2Fi234.photobucket.com%2Falbums%2Ffv234%2Fpbuser%2F00_
pb_tattoo-1.jpg/comment?format=xml&oauth_consumer_key=00000000&oauth_
nonce=23cd652bc8ce973dce6b4f06dec9fe6a&oauth_signature=u%2B%2B9ldZ8f7rHoAATxxkkyFeRSGM%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1227204003&oauth_token=31.0_1227204003&oauth_version=1.0
```

Response Example

The successful response lists comments for the media.

```
<response>
  <status>OK</status>
  <content>
```

```

    <total_count>1</total_count>
    <comment>
      <comment_content>funky tattoo</comment_content>
      <display_date>1277332027</display_date>
      <facebook_name>Facebook User</facebook_name>
    </comment>
  </content>
</format>xml</format>
<method>GET</method>
<timestamp>1277332236</timestamp>
</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Ratings

Get ratings on a specific piece of media.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/media/[identifier]/rating

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request searches for ratings on "mass1.jpg" in the "pbapi" album.

```

GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2Ffbapi%2Fmass1.jpg/rating?format=xml&oauth_
consumer_key=000000000&oauth_nonce=93f2f443860592533fb31d55abd02948&oauth_
signature=3CiMOA17OtpqLVGW2kl0QHHYSE%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1236629580&oauth_token=31.00_1236629580&oauth_version=1.0

```

Response Example

The successful response lists ratings for the media.

```

<response>
  <status>OK</status>
  <content>
    <number>32</number>
    <type>facebook</type>

```

```
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1277332116</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Resize Image

Resize the specified image, and overwrite the existing copy.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT

REST Path

/media/[identifier]/resize

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
size	Y	Size to resize an image to. (Images can only be made smaller.) One of: (value => resize result dimensions) • 100 => 100x75 • 160 => 160x120 • 320 => 320x240 • 640 => 640x480 • 800 => 800x600 • 1024 => 1024x768 • 1280 => 1280x960, //premium owner only • 1600 => 1600x1200, //premium owner only • 2048 => 1048x1536, //premium owner only • 2240 => 2240x1680, //premium owner only	Integer

Request Example

The request resizes the image "pb.gif" to 320.

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2Fgg17%2Fpbapi%2Fsummer%2Fpb.gif/resize?format=xml&oauth_
consumer_key=00000000&oauth_nonce=0952369f3fb541d17a9649cb92e8bf28&oauth_
signature=o73mLLkdQX2%2BKJ%2B1s2E5Ymp7wz4%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443500&oauth_token=28.1&oauth_version=1.0&size=320
```

Response Example

The successful response looks like:

```
<response>
  <status>OK</status>
  <content />
  <format>xml</format>
  <method>PUT</method>
  <timestamp>1202854502</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **153:** The size parameter is required

For a full list of possible errors, see [Error Codes](#).

Rotate Image

Rotate the specified image, and overwrite the existing copy.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT

REST Path

/media/[identifier]/rotate

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
degrees	Y	Enter the degree of rotation. Images rotate clock-wise, and degrees must be a multiple of 90 (e.g., 90, 180, 270, etc.). Degrees that are not a multiple of 90 cause an error.	Integer

Request Example

The request rotates the image "pb.gif" 90 degrees.

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsummer%2F
pb.gif/rotate?degrees=90&format=xml&oauth_consumer_key=00000000&oauth_
nonce=2ebb0f3ba83f91044310701db0abfdb4&oauth_signature=QeiJvJH%2FzGnaYb%2FZyxtdeEIDmg%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1203443499&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks like:

```
<response>
  <status>OK</status>
```

```
<content />
<format>xml</format>
<method>PUT</method>
<timestamp>1202854502</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **141**: Media specified was not found
- **142**: Rotation degrees must be in 90 degree increments
- **154**: The degrees parameter is required

For a full list of possible errors, see [Error Codes](#).

Get Media Meta Data

Get EXIF details for a piece of media.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/meta

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the details for the image "13xolo-1.jpg" in the album "pbapi".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2F13xolo-1.jpg/meta?format=xml&oauth_
consumer_key=00000000&oauth_nonce=62dala304e98c77bd1d4474bb81b7254&oauth_
signature=kylNSw40n5iwBCezuSC19R8hv5k%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1206468801&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes the resolution of the image.

```
<response>
  <status>OK</status>
  <content>
    <exif>
      <Resolution>512 x 382</Resolution>
      <Comment>Copyright (C) 2002-2007 SQUARE ENIX CO., LTD. All Rights Reserved.</Comment>
    </exif>
```

```
<width>512</width>
<height>382</height>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1206468801</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Media Geo-data

Get geo-data for the media specified.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/geo

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the geo-data for the image "pb.gif" in the "summer" album for user "pbapi".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsummer%2Fpb.gif/geo?format=xml&oauth_
consumer_key=00000000&oauth_nonce=7c3ad83012aeb564ad6b32645a7cf1ab&oauth_
signature=FxOsj%2FXuSseIO%2Bc9X%2BMzb424aZ0%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443491&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response shows the geo-data.

```
<response>
<status>OK</status>
<content>
  <latitude>100.000000</latitude>
  <longitude>100.000000</longitude>
  <altitude>9.0000</altitude>
  <compass>180.0000</compass>
  <gps_timestamp>1265320576.0000</gps_timestamp>
</content>
```

```
<format>xml</format>
<method>GET</method>
<timestamp>1202876757</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Update Media Geo-data

Update the geographic data for a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT

REST Path

/media/[identifier]/geo

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Urlencoded full media URL.	String
read_geo_exif	Y	true	Read the geo-data from exif on the image. This is overwritten by the given parameters, unless this parameter is "force", where the exif will override given parameters.	Boolean or String
latitude	Y		Decimal latitude in degrees (-90,90).	Float
longitude	Y		Decimal longitude in degrees (-180,180).	Float
altitude	Y		Decimal height above sea level (-x,x) in kilometers.	Float
compass	Y		Decimal direction of orientation at image capture (0,360) in degrees.	Float
gps_timestamp	Y		Timestamp of GPS capture (seconds since epoch).	Float

Request Example

The request updates the altitude, compass, and gps_timestamp for the the "ferrari.jpg" image. It keeps the geo-data from the image for other fields.

```
PUT
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2Fgg17%2Fpbapi%2Fferrari.jpg
/geo?altitude=100&compass=180&format=xml&gps_timestamp=1265320518&oauth_consumer_key=00000000&oauth_nonce=b9e53fc35c6ad298e159c8cdb796edd7&oauth_signature=B%2BA3pAQBI8rVqgepGyv730K%2BQ04%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205278893&oauth_token=28.1&oauth_version=1.0&read_geo_exif=1
```

Response Example

The successful response includes all the geo-data.

```
<response>
<status>OK</status>
```

```
<content>
  <latitude>52.416668</latitude>
  <longitude>0.583333</longitude>
  <altitude>100.0000</altitude>
  <compass>180.0000</compass>
  <gps_timestamp>1265320576.0000</gps_timestamp>
</content>
<format>xml</format>
<method>PUT</method>
<timestamp>1202846937</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Delete Geo-data from Media

Delete the geo-data from a media item.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/media/[identifier]/geo

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request deletes the geo-data for the image "pb.gif".

```
DELETE
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2Fgg17%2Fpbapi%2Fpb.gif/geo?format=xml&oauth_consumer_
key=00000000&oauth_nonce=c44cfd734ad7e1557c544b2842614c80&oauth_
signature=E8s3wuolcsRhZUPzgF02MlCFxk8%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443495&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>1</content>
  <format>xml</format>
  <method>DELETE</method>
```

```
<timestamp>1202845953</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Media Links

Get the media's links including URL, <a href>, image source, and image code for forums and message boards.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/link

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request gets the link for the image "13xolo-1.jpg".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2F13xolo-1.jpg/link?format=xml&oauth_
consumer_key=00000000&oauth_nonce=8af50aa98191642011f318b108f5fec7&oauth_
signature=EM5bQPRRgE%2B%2By78FtbiXRlNzI8c%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1206468800&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes the URL, the <a href>, image source, and the image code.

```
<response>
  <status>OK</status>
  <content>
    <url>http://i123.photobucket.com/albums/gg17/pbapi/13xolo-1.jpg</url>
    <thumb>http://i123.photobucket.com/albums/gg17/pbapi/th_13xolo-1.jpg</thumb>
    <fullsize>http://s123.photobucket.com/albums/gg17/pbapi/?action=view&current=13xolo-
1.jpg</fullsize>
    <sharetag>&lt;a
href=&quot;http://s123.photobucket.com/albums/gg17/pbapi/?action=view&current=13xolo-1.jpg"
target=&quot;_blank&quot;&gt;&lt;img src=&quot;http://i123.photobucket.com/albums/gg17/pbapi/13xolo-
1.jpg" border=&quot;0&quot; alt=&quot;Photobucket&quot;&gt;&lt;/a&gt;</sharetag>
    <bbcodetag>{IMG]http://i123.photobucket.com/albums/gg17/pbapi/13xolo-1.jpg[/IMG]</bbcodetag>
    <album>http://s123.photobucket.com/albums/gg17/pbapi/</album>
    <tagembedtag />
  </content>
```

```
<format>xml</format>
<method>GET</method>
<timestamp>1206468800</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Related Media

Get media that is related to other media, based on title, description, size, etc.

User Login Required

No.

HTTP Method

GET

REST Path

/media/[identifier]/related

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		Urlencoded full media URL.	String
num	Y	5	Number of results to return. Maximum 20.	Integer
type	Y	images	Type of media to return. Options are images or videos.	String

Request Example

The request gets media that is related to the image "ferrari.jpg".

```
GET
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falb%2Fegg17%2Fpbapi%2Fferrari.jpg/related?format=xml&oauth_
consumer_key=000000000&oauth_nonce=8E863ad52bf0a2898c193916b31865b6c&oauth_
signature=KyHDKAgId%2Bf8MxnIlzywRmezgV4%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1205278891&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes the two images that most closely match the details associated with "ferrari.jpg".

```
<response>
  <status>OK</status>
  <content>
    <result page="1" perpage="" totalpages="0" totalresults="2">
      <media description_id="53716270" name="ferrari.gif" public="1" uploaddate=""
username="mooncor">

      <browseurl>http://s234.photobucket.com/albums/jj191/mooncor/?action=view&current=ferrari.gif</browseurl>

      <url>http://i234.photobucket.com/albums/jj191/mooncor/ferrari.gif</url>
      <albumurl>http://s234.photobucket.com/albums/jj191/mooncor/</albumurl>
```

```

        <thumb>http://th234.photobucket.com/albums/jj191/mooncor/th_ferrari.gif</thumb>
    </media>
    <media description_id="4242764" name="ferrari.gif" public="1" uploaddate="" username="jef_
jeret_29">
        <browseurl>http://s223.photobucket.com/albums/dd75/jef_jeret_
29/?action=view&current=ferrari.gif</browseurl>
        <url>http://i223.photobucket.com/albums/dd75/jef_jeret_29/ferrari.gif</url>
        <albumurl>http://s223.photobucket.com/albums/dd75/jef_jeret_29/</albumurl>
        <thumb>http://th223.photobucket.com/albums/dd75/jef_jeret_29/th_ferrari.gif</thumb>
    </media>
</result>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1202854987</timestamp>
</response>

```

Error Codes

The following error may be returned:

- **141:** Media specified was not found

For a full list of possible errors, see [Error Codes](#).

Rename or Move Media

Update the media file name and/or move the file.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

PUT or POST

REST Path

/media/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String
name	N	The new name for the media, without an extension. E.g., filename, not filename.gif	String
album	Y	The album ID (username/location) to which the media will be moved. Both albums must be owned by the same user (the usernames match in the albumid).	String

Request Example

The request updates the media file name to "river".

PUT

```

http://api342.photobucket.com/media/http%3A%2F%2Fi342.photobucket.com%2Falb%2Fv123%2Fpbuser%2Fnewalbum%2F
river&oauth_consumer_key=00000000&oauth_nonce=96527b6cbc4f600000e50f25604b71da&oauth_

```

```
signature=y3XR%2FunzsTU1zCGx%2BpYaJgcJDx8%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1228169822&oauth_token=20.000000_1228169822&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="river.png" public="1" type="image"
      uploaddate="1228169823" username="pbuser"></order>
      <brow-
        seurl>http://-
          s342.photobucket.com/albums/v123/pbuser/?action=view&amp;current=river.png</browseurl>
        <url>http://i342.photobucket.com/albums/v123/pbuser/river.png</url>
        <thumb>http://i342.photobucket.com/albums/v123/pbuser/th_
          river.png</thumb>
        <description />
        <title />
      </media>
    </content>
    <format>xml</format>
    <method>PUT</method>
    <timestamp>1228169823</timestamp>
  </response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Delete Media

Delete the specified image(s) or video(s).

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/media/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded full media URL.	String

Request Example

The request deletes image “pb.gif” from the “summer” album of user "pbapi".

```
DELETE
http://api123.photobucket.com/media/
http%3A%2F%2Fi123.photobucket.com%2Falbums%2Fgg17%2Fpbapi%2Fsummer%2Fpb.gif?format=xml&oauth_consumer_
key=00000000&oauth_nonce=747c5c107bc2a820bc7951a2dd94f41c&oauth_
signature=EXzSLBYvUnKbkIrdbfr1X1MMAFQ%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1203443492&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <deleted>1</deleted>
  </content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1202841098</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Share Media

Share the URL of specific media via email, Facebook, MySpace, and Twitter.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/media/[identifier]/share/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Note the addition of [actionIdentifier] to the path

Parameters

Parameter	Optional	Description	Variable
identifier	N	Urlencoded media to share.	String
actionIdentifier	N	Comma separated list of services to send to. Options are all, twitter, facebook, myspace, or email. Default is "all", but all does NOT include email.	String
message	Y	Body of message.	String
email	Y/N	Email to which the URL for the media will be sent. Required if actionIdentifier is "email".	String

Request Example

The request shares the image "blueraven.gif".

```
POST
http://api123.photobucket.com/media/
http%3A%2F%2Fi342.photobucket.com%2Falbums%2Fo43i%2Fpbapi%2Fblueraven.gif/share/all
message=testing1&format=xml
```

Response Example

A successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <myspace>not sent</myspace>
    <facebook>http://www.new.facebook.com/profile.php?id=100001030313912</facebook>
    <twitter>http://twitter.com/test</twitter>
  </content>
</format>xml</format>
<method>POST</method>
<timestamp>1273007222</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **009:** Insufficient permissions
- **118:** Missing email address
- **141:** Media specified was not found
- **204:** Invalid email address

For a full list of possible errors, see [Error Codes](#).

Search

The Search methods are:

- [Search Images](#)
- [Search Videos](#)
- [Search Group Albums](#)
- [Get Featured Home Page Categories](#)
- [Get Featured Group Albums](#)
- [Get Find Stuff Category Names](#)
- [Get Find Stuff Category Media](#)
- [Follow a Search Term](#)
- [Stop Following a Search Term](#)
- [Get Search Term Following Status](#)

Search Images

Search for images that match a specific search term or terms. If a search term is not specified, the response contains the most recent images.

If you want the top 100 items for a term or the top 100 recent images, consider using the [Photobucket RSS Feeds](#).

User Login Required

No.

HTTP Method

GET

REST Path

/search/[identifier]/image

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Important! The parameter "type" is deprecated. Note the new REST path for Search Images and [Search Videos](#).

Parameter	Optional	Default	Description	Variable
identifier	Y		Search term. If a search term is not entered, or is "-", recent images are returned.	String
num	Y	20	Number of results to return (recent). Maximum of 100.	Integer
perpage	Y	20	Number of results to return for main search type. Maximum of 100.	Integer
page	Y	1	Page number to display (1 indexed).	Integer
offset	Y	(page-1)*- perpage	Beginning offset of results.	Integer
secondaryperpage	Y	5	Number of images to show, per page, for secondary search type.	Integer
recentfirst	Y	false	Show images from the last seven days.	Boolean

Request Example

The request searches for recent images with the search term "me".

```
GET http://api.photobucket.com/search/me/image?format=xml&oauth_consumer_key=00000000&oauth_nonce=6e47e8127a2d7705a2d203e44cc4e983&oauth_signature=YwNedomUJ6JYW%2BGGDeM%2BYxfMnjs%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205810735&oauth_token=28.1&oauth_version=1.0&perpage=2
```

Response Example

The successful response lists media that matches the search term, including URL, album URL, description, and title. The following response does not display all 20 results.

```
<response>
  <status>OK</status>
  <content>
    <result page="1" perpage="2" totalpages="442392" totalresults="8847824">
      <primary>
        <media description_id="57388640" name="not.jpg" public="1" uploaddate=""
username="GrouchoMarx69">

        <browseurl>http://s266.photobucket.com/albums/ii260/GrouchoMarx69/?action=view&current=not.jpg</browseurl>
          <url>http://i266.photobucket.com/albums/ii260/GrouchoMarx69/not.jpg</url>
          <albumurl>http://s266.photobucket.com/albums/ii260/GrouchoMarx69/</albumurl>
          <thumb>http://th266.photobucket.com/albums/ii260/GrouchoMarx69/th_not.jpg</thumb>
          <description />
          <title>me</title>
        </media>
```

```

        <media description_id="10721932" name="as.jpg" public="1" uploaddate=""
username="kaybee81">

    <browseurl>http://s161.photobucket.com/albums/t220/kaybee81/?action=view&current=as.jpg</browseurl>
        <url>http://i161.photobucket.com/albums/t220/kaybee81/as.jpg</url>
        <albumurl>http://s161.photobucket.com/albums/t220/kaybee81/</albumurl>
        <thumb>http://th161.photobucket.com/albums/t220/kaybee81/th_as.jpg</thumb>
        <description />
        <title>me</title>
    </media>
</primary>
<secondary />
</result>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1202848217</timestamp>
</response>

```

Error Codes

The following errors may be returned:

- **103:** Invalid search query.
- **104:** Invalid characters in query.

For a full list of possible errors, see [Error Codes](#).

Search Videos

Search for videos that match a specific search term or terms. If a search term is not specified, the response contains the most recent updated videos.

If you want the top 100 items for a term or the top 100 recent images, consider using the [Photobucket RSS Feeds](#).

User Login Required

No.

HTTP Method

GET

REST Path

/search/[identifier]/video

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Important! The parameter "type" is deprecated. Note the new REST path for [Search Images](#) and [Search Videos](#).

Parameter	Optional	Default	Description	Variable
identifier	Y		Search term. If a search term is not entered, or is "-", recent videos are returned.	String
num	Y	20	Number of results to return (recent). Maximum of 100.	Integer

Parameter	Optional	Default	Description	Variable
perpage	Y	20	Number of results to return for main search type. Maximum of 100.	Integer
page	Y	1	Page number to display (1 indexed).	Integer
offset	Y	(page-1)*- perpage	Beginning offset of results.	Integer
secondaryperpage	Y	5	Number of videos to show, per page, for secondary search type.	Integer
recentfirst	Y	false	Show videos from the last seven days.	Boolean

Request Example

The request searches for recent videos with the search term "me".

```
GET http://api.photobucket.com/search/me/video?format=xml&oauth_consumer_key=00000000&oauth_nonce=216e05783e1773ab229442cbd6d3404a&oauth_signature=DD1zUrM8gxWcQF8rMHCJtEqhrI8%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214857629&oauth_token=28.10_1214857629&oauth_version=1.0
```

Response Example

The successful response lists media that matches the search term, including URL, album URL, description, and title. The following response does not display all 20 results.

```
<response>
  <status>OK</status>
  <content>
    <results page="1" perpage="20" totalpages="929" totalresults="18562">
      <primary>
        <media description_id="" name="me.flv" public="1" type="video" uploaddate=""
username="ORLANDOMAESTAS">

        <browseurl>http://s251.photobucket.com/albums/gg307/ORLANDOMAESTAS/?action=view&current=me.flv</browseurl>

        <url>http://i251.photobucket.com/player.swf?file=http://api.photobucket.com/albums/gg307/ORLANDOMAESTAS/me.flv</url>
          <albumurl>http://s251.photobucket.com/albums/gg307/ORLANDOMAESTAS/</albumurl>
          <thumb>http://th251.photobucket.com/albums/gg307/ORLANDOMAESTAS/th_me.jpg</thumb>
          <description />
          <title>me</title>
        </media>
      </primary>
      <secondary>
        <media description_id="" name="ME.jpg" public="1" type="image" uploaddate="1214357938"
username="Darsh_bucket">
          <browseurl>http://s249.photobucket.com/albums/gg232/Darsh_bucket/?action=view&current=ME.jpg</browseurl>
          <url>http://i249.photobucket.com/albums/gg232/Darsh_bucket/ME.jpg</url>
          <albumurl>http://s249.photobucket.com/albums/gg232/Darsh_bucket/</albumurl>
          <thumb>http://th249.photobucket.com/albums/gg232/Darsh_bucket/th_ME.jpg</thumb>
          <description>It's me</description>
          <title>me</title>
        </media>
      </secondary />
    </result>
```

```
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1214931982</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **103:** Invalid search query.
- **104:** Invalid characters in query.

For a full list of possible errors, see [Error Codes](#).

Search Group Albums

Search group albums for images or videos that match a specific search term or terms. If a search term is not specified, the response contains the most recent updated media.

If you want the top 100 items for a term or the top 100 recent images, consider using the [Photobucket RSS Feeds](#).

User Login Required

No.

HTTP Method

GET

REST Path

/search/[identifier]/group

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	Y		Search term. If a search term is not entered, or is "-", recent groups are returned.	String
num	Y	20	Number of results to return (recent). Maximum of 100.	Integer
perpage	Y	20	Number of results to return for main search type. Maximum of 100.	Integer
page	Y	1	Page number to display (1 indexed).	Integer
offset	Y	(page-1)*perpage	Beginning offset of results.	Integer
recentfirst	Y	false	Show images from the last seven days.	Boolean

Request Example

The request searches for recent groups with the search term "puppies".

```
GET http://api.photobucket.com/search/puppies/group?format=xml&oauth_consumer_key=00000000&oauth_nonce=3057638b7d3f9501f38b8d3ce_4974732&oauth_signature=D1LzaBa16QVQ1PMdPiGF5e0Fv%2Fw%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1214504396&oauth_token=28.10_1214504385&oauth_version=1.0
```

Response Example

The successful response lists media that matches the search term, including URL, album URL, description, and title. The following response does not display all 20 results.

```
<response>
  <status>OK</status>
```

```

<content>
  <result page="1" perpage="20" totalpages="1" totalresults="16">
    <groups>
      <id>1</id>
      <name>Cute Puppies</name>
      <description />
      <url>http://gs255.photobucket.com/groups/hh153/4JPM8I77Z</url>
      <thumbUrl>http://th255.photobucket.com/groups/hh153/4JPM8I77Z/th_
DSCN0902.jpg<thumbUrl/>
      <numUploads>601</numUploads>
      <numUsers>199</numUsers>
    </groups>
    <groups>
      <id>1</id>
      <name>Cute Puppies</name>
      <description />
      <url>http://gs255.photobucket.com/groups/hh153/4JPM8I77Z</url>
      <thumbUrl>http://th255.photobucket.com/groups/hh153/4JPM8I77Z/th_
DSCN0902.jpg<thumbUrl/>
      <numUploads>601</numUploads>
      <numUsers>199</numUsers>
    </groups>
  </result>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1202848217</timestamp>
</response>

```

Error Codes

The following errors may be returned:

- **103:** Invalid search query.
- **104:** Invalid characters in query.

For a full list of possible errors, see [Error Codes](#).

Get Featured Home Page Categories

Get categories featured on the Home page by Photobucket.

User Login Required

No.

HTTP Method

GET

REST Path

/featured/homepage

Parameters

None.

Request Example

The request searches for categories featured on the Photobucket Home page.

```
GET http://api.photobucket.com/featured/homepage?format=xml&oauth_consumer_key=00000000&oauth_nonce=4ca0908f1c841c184770bde4664c4073&oauth_signature=iuLVofVY9AOdA2W8AoCSUvDqYv8%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1207585112&oauth_token=28.10&oauth_version=1.0
```

Response Example

The successful response lists featured categories. The following response does not display all results.

```
<response>
  <status>OK</status>
  <content>
    <result page="1" perpage="9" totalpages="1" totalresults="9">
      <isSponsored>
        <searchSponsor>
          <primary>
            <media description_id="" name="" public="1" type="image" uploaddate="" username="">
              <browseurl>http://api.photobucket.com/findstuff/Photography/</browseurl>
              <thumb>http://i0006.photobucket.com/albums/0006/findstuff22/Best%20Images/Photography/photographyswing1.jpg</thumb>
              <description>Check out amazing photos and some cool photography techniques!</description>
              <title>Photography</title>
              <isSponsored/>
            </media>
            <media description_id="" name="" public="1" type="image" uploaddate="" username="">
              <browseurl>http://api.photobucket.com/images/color splash/</browseurl>
              <thumb>http://i0006.photobucket.com/albums/0006/findstuff22/Best%20Images/Photography/colorsplashthumb22edit.jpg</thumb>
              <description>It's a cool effect of a black and white photo with a splash of color.</description>
              <title>Color Splash</title>
              <isSponsored/>
            </media>
            <media description_id="" name="" public="1" type="image" uploaddate="" username="">
              <browseurl>http://api.photobucket.com/findstuff/Just for Fun/</browseurl>
              <thumb>http://i0006.photobucket.com/albums/0006/findstuff22/Best%20Images/Just%20For%20Fun/peace1.jpg</thumb>
              <description>From tattoos, PostSecret and glitter, this is your destination for all things fun.</description>
              <title>Just for Fun</title>
              <isSponsored/>
            </media>
          </primary>
        </result>
      </content>
```

```
<format>xml</format>
<method>GET</method>
<timestamp>1207609439</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Featured Group Albums

Get public group albums featured by Photobucket.

User Login Required

No.

HTTP Method

GET

REST Path

/featured/group

Parameters

None.

Request Example

The request searches for Photobucket's featured group albums.

```
GET http://api.photobucket.com/featured/group?format=xml&oauth_consumer_key=00000000&oauth_
nonce=f9a1c972bd394d6c685f1738482826 c&oauth_signature=3RV3oVCYono4tQwT83Q5SGsxGb4%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1214504397&oauth_token=28. 10_1214504385&oauth_version=1.0
```

Response Example

The successful response lists featured media. The following response does not show all results.

```
<response>
  <status>OK</status>
  <content>
    <group>
      <name>mccuen</name>
      <url>http://gs123.photobucket.com/groups/11104/FAZED9KFU/?featuredgroupalbum=mccuen</url>
      <thumb>http://gs123.photobucket.com/groups/11104/FAZED9KFU/th_cavityHeadlight5-0-
39.jpg</thumb>
      <mediacount>1</mediacount>
      <contributors>1</contributors>
    </group>
    <group>
      <name>gregz</name>
      <url>http://gs234.photobucket.com/groups/11228/FAXFC5IGQ/?featuredgroupalbum=gregz</url>
      <thumb>http://gs234.photobucket.com/groups/11228/FAXFC5IGQ/th_coolbike.jpg</thumb>
      <mediacount>1</mediacount>
      <contributors>1</contributors>
    </group>
```

```
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1214504398</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Find Stuff Category Names

Get categories for Find Stuff featured media.

User Login Required

No.

HTTP Method

GET

REST Path

/findstuff

Parameters

None.

Request Example

The request gets the categories for the Photobucket Find Stuff option.

```
GET http://api.photobucket.com/findstuff?format=xml&oauth_consumer_key=00000000&oauth_
nonce=b33c0e5e14ec3b44ef4f49ae63de0b6f&oauth_signature=oX8%2B3Dh1ikz6XZSUsJ1WY7%2BgWJw%3D&oauth_
signature_method=HMAC-SHA1&oauth_timestamp=1247506316&oauth_version=1.0
```

Response Example

The successful response lists the Find Stuff categories. The following response does not display all results.

```
<response>
  <status>OK</status>
  <content>
    <category_name>Funny</category_name>
    <category_name>Motivational Posters</category_name>
    <category_name>Random Funny</category_name>
    ... (275+ lines)
  </content>
</format>xml</format>
<method>GET</method>
<timestamp>1247506316</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Find Stuff Category Media

Get media for a specific Find Stuff category, or get categories for the Find Stuff landing page.

User Login Required

No.

HTTP Method

GET

REST Path

/findstuff/[identifier]/category

Parameters

Parameter	Optional	Description	Variable
identifier	Y	Find stuff category name. If identifier is empty or is "-", Find Stuff landing page categories are returned.	String

Request Example

The request gets media in the "Gaming" Find Stuff category.

```
GET http://api.photobucket.com/findstuff/Gaming/category?format=xml&oauth_consumer_key=00000000&oauth_nonce=65fea894c7f1e6fa0ce0e7599f067f86&oauth_signature=zU%2FV00K6omAneIK81LeoyM0V8xo%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1247506517&oauth_version=1.0
```

Response Example

The successful response gets media in the specified Find Stuff category. The following response does not display all results.

```
<response>
  <status>OK</status>
  <content>
    <category>
      <category_name>Gaming</category_name>
      <description>Give your callused thumbs a break and view the hottest video
games.</description>
      <thumb_
url>http://i410.photobucket.com/albums/pp190/FindStuff2/Best%20Images/Gaming/wowthumb.png</thumb_
url>
      <url>http://major.local.dev.photobucket.com/findstuff/Gaming/</url>
      <search_term>video games</search_term>
      <sub_categories>
        <category_name>World of Warcraft</category_name>
        <description>Check out unique characters created and witness the world of
Azeroth. </description>
        <thumb_
url>http://i410.photobucket.com/albums/pp190/FindStuff2/Best%20Images/Gaming/wowthumb.png</thumb_
url>
        <url>http://major.local.dev.photobucket.com/images/world of warcraft/</url>
        <search_term>world of warcraft</search_term>
      </sub_categories>
    </category>
  </content>
</response>
```

```

        <sub_categories>
            <category_name>Final Fantasy</category_name>
            <description>As one of the longest games in production, Final Fantasy is a
timeless classic. </description>
            <thumb_
url>http://i410.photobucket.com/albums/pp190/FindStuff2/Best%20Images/Gaming/wowthumb.png</thumb_
url>

            <url>http://major.local.dev.photobucket.com/images/final_fantasy/</url>
            <search_term>final_fantasy</search_term>
        </sub_categories>
        ... (many more)
    </category>

</content>
<format>xml</format>
<method>GET</method>
<timestamp>1247506517</timestamp>
</response>

```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Follow a Search Term

Use a search term to follow images or videos. The album_module parameter displays in the user's *Following* panel in their Photobucket album.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

POST

REST Path

/search/[identifier]/follow/[actionIdentifier]

This was previously /subscribe/ but this usage is deprecated. Please use /follow/

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

The following parameters have been deprecated: search_term and album_module.

Parameter	Optional	Description	Variable
identifier	N	Search term.	String
actionIdentifier	Y	What to follow: - search_term - Follow a search term by email. - album_module- Follow search terms in your album module. Must be logged in. The album_module displays in the user's <i>Searches / Follow</i> panel in their Photobucket account.	String
email	Y/N	Email address to which notifications will be sent. This field is required if the user is not logged into a Photobucket account.	String

Request Example

The request follows the search term "lolcat".

```
POST http://api.photobucket.com/search/lolcat/follow/search_term
format=xml&oauth_consumer_key=000000000&oauth_nonce=9bdfdc260bd2bb206395278905c7b4d&oauth_signature_
method=HMAC-SHA1&oauth_timestamp=1236630795&oauth_token=31.00_1236630795&oauth_version=1.0&
```

Response Example

The successful response lists everything the user is following. The request above adds the search_term subscription. The response shows the request was successful.

```
<response>
  <status>OK</status>
  <content>
    <search_term>
      <email>l</email>
    </search_term>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1236630796</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Stop Following a Search Term

Stop following a search term. This method depends on data returned in [Get Items the User Follows](#).

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

DELETE

REST Path

/search/[identifier]/follow/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Album identifier.	String
actionidentifier	Y	Type to stop following: - search_term - Follow a search term by email. - album_module- Follow search terms in your album module. Must be logged in.	String
user_subscription_id	N	User's subscription ID. Get this value from the response in Get Items the User Follows .	String

Request Example

The request stops a user from following the search term "lolcats".

```
DELETE http://api.photobucket.com/saarch/jessica alba//search_term?format=xml&oauth_consumer_key=000000000&oauth_nonce=1d711b38a422ae3c881a1c423b43a8b0&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628506&oauth_token=31.00_1236628505&oauth_version=1.0&user_subscription_id=2327562
```

Response Example

The successful responder removes the subscription from the user's account.

```
<response>
  <status>OK</status>
  <content>
    <search_term>
      <album_module>1</album_module>
    </search_term>
  </content>
  <format>xml</format>
  <method>DELETE</method>
  <timestamp>1236628506</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get Search Term Following Status

Determine if a user is following a search term, and which parameters the user is following.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/search/[identifier]/follow

This was previously `/subscribe/` but this usage is deprecated. Please use `/follow/`

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Search term.	String
email	Y/N	Email address to check following state. This field is required if the user is not logged into a Photobucket account.	String

Request Example

The request determines if the user is following the search term "something".

```
GET http://api.photobucket.com/search/something/follow?format=xml&oauth_consumer_key=000000000&oauth_nonce=af8ea6f4ce806be34187e18bfd0fceb1&oauth_signature=110ZZf8rDJ456vaLnQhhkNwE6es%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236630699&oauth_token=31.00_1236630699&oauth_version=1.0
```

Response Example

The successful response lists the search terms that a user is following, based on the parameters in the [Follow a Search Term](#) method. The following response shows that the user follows the search term "something".

```
<response>
  <status>OK</status>
  <content>
    <search_term>
      <email>1</email>
    </search_term>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1236630700</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Users

The User methods are:

- [Get Recent User Media](#)
- [Get User Contacts](#)
- [Get User Information](#)
- [Get Media Tags for a User](#)
- [Get User Upload Options](#)
- [Update User Upload Options](#)
- [Get User URLs](#)
- [Get a User's Group Albums](#)
- [Get Media that a User Follows](#)
- [Get User Connections](#)
- [Get User Name Availability](#)
- [Register User](#)

Get Recent User Media

Get media recently added by a specific user.

If you want the most recent 100 items for a user, consider using the [RSS feed](#).

User Login Required

Optional (returns private). If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/search/[actionidentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		User name.	String
actionidentifier	Y	recent uploads	Search term.	String
page	Y	1	Page number to view; 1 indexed.	Integer
perpage	Y	20	Number of media per page. Maximum of 200.	Integer
type	Y	all	Filter media by type. Options are image, video, all.	String

Request Example

The request gets recent images added by user "pbapi".

```
GET http://api123.photobucket.com/user/pbapi/search?format=xml&oauth_consumer_key=00000000&oauth_nonce=420eed7a5a7c33a5911878569dc52c67&oauth_signature=2QvZVrFYG32XJr0vMKRFmkMDo0w%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1205810738&oauth_token=28.1&oauth_version=1.0&perpage=2&type=image
```

Response Example

The successful response includes a list of media that matches the search request.

```
<response>
  <status>OK</status>
  <content page="1" perpage="2" totalpages="35" username="pbapi">
    <media description_id="" name="IMG_0026.jpg" public="1" uploaddate="1205793318"
username="pbapi">
      <browseurl>http://s123.photobucket.com/albums/gg17/pbapi/?action=view&current=IMG_
0026.jpg</browseurl>
      <url>http://i123.photobucket.com/albums/gg17/pbapi/IMG_0026.jpg</url>
      <thumb>http://i123.photobucket.com/albums/gg17/pbapi/th_IMG_0026.jpg</thumb>
      <description />
      <title />
    </media>
    <media description_id="" name="IMG_0011.jpg" public="1" uploaddate="1205793311"
username="pbapi">
      <browseurl>http://s123.photobucket.com/albums/gg17/pbapi/?action=view&current=IMG_
0011.jpg</browseurl>
      <url>http://i123.photobucket.com/albums/gg17/pbapi/IMG_0011.jpg</url>
      <thumb>http://i123.photobucket.com/albums/gg17/pbapi/th_IMG_0011.jpg</thumb>
      <description />
      <title />
    </media>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202849793</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **102:** User not found

For a full list of possible errors, see [Error Codes](#).

Get User Contacts

Get the user's contacts.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/contact

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String

Request Example

The request gets the contacts for user "pbapi".

```
GET http://api123.photobucket.com/user/pbapi/contact?format=xml&oauth_consumer_key=00000000&oauth_nonce=e74a83eff96f9d275e31277399529792&oauth_signature=h9e4zIMZcFu8wkX2EBPGFjI%2BPV0%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443505&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes a list of contact names and email addresses for the selected user.

```
<response>
  <status>OK</status>
  <content contacts_count="7">
    <contact>
      <first>george</first>
      <last />
      <primary_email>george@email.com</primary_email>
    </contact>
    <contact>
      <first>bill</first>
      <last />
      <primary_email>bill@email.com</primary_email>
    </contact>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202848776</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **009**: Insufficient permissions

For a full list of possible errors, see [Error Codes](#).

Get User Information

Get user information such as user name, album URL, megabytes allowed and used, etc.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	Y	User name. If not entered, user information for the user who is currently logged-in is retrieved.	String

Request Example

The request gets information about user "pbapi".

```
GET http://api123.photobucket.com/user/pbapi?format=xml&oauth_consumer_key=00000000&oauth_nonce=23db2b4c97846dfa4b5c6ede85149be0&oauth_signature=7sX4GmfjkVtWCJXIfvWibiyBIL4%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443505&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes user name, album URL, megabytes allowed and used, whether the user is a premium member, total number of pictures (includes images and video), and preferred picture size.

```
<response>
  <status>OK</status>
  <content>
    <username>pbapi</username>
    <album_url>http://s123.photobucket.com/albums/gg17/pbapi</album_url>
    <megabytes_used>9</megabytes_used>
    <megabytes_allowed>500</megabytes_allowed>
    <premium>0</premium>
    <public>1</public>
    <total_pictures>20</total_pictures>
    <preferred_picture_size>1024</preferred_picture_size>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202849065</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **007**: Authentication failed
- **102**: User not found

For a full list of possible errors, see [Error Codes](#).

Get Media Tags for a User

Get media tags for a user.

User Login Required

Optional (returns private). If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/tag/[tagname]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		User name.	String
tagname	Y		Tag name to get media for. If not entered, tag name list is returned.	String
separate	Y	false (0)	If tagname="", separate tag list by contact and URL.	Boolean
page	Y	1	Page number to view, 1 is indexed.	Integer
perpage	Y	20	Number of media per page. Maximum of 200.	Integer

Request Example 1

Request Example 1 looks for tags for user "pbapi", but does not include a tag name.

```
GET http://api123.photobucket.com/user/pbapi/tag/?format=xml&oauth_consumer_key=12344321&oauth_nonce=44ca06813a1a872932156f0c1dfbb3a1&oauth_signature=0H2muE3NNZZBA%2B1UklS2JWL3UXU%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1206376180&oauth_token=28.1&oauth_version=1.0
```

Response Example 1

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <name>allie</name>
    <count>1</count>
  </content>
  <content>
    <name>Apple Store</name>
    <count>1</count>
  </content>
```

```
<format>xml</format>
<method>GET</method>
<timestamp>1206376181</timestamp>
</response>
```

Request Example 2

Request Example 2 looks for tags for user "pbapi" and includes the tag name "Corvette".

```
GET http://api123.photobucket.com/user/pbapi/tag/Corvette?format=xml&oauth_consumer_
key=12344321&oauth_nonce=50ea7f6738d0a6a23ff80a170d1f5c15&oauth_
signature=lpUqOCZyl0yOtfhSILN1V39iWlc%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1206376431&oauth_token=28.1&oauth_version=1.0
```

Response Example 2

The successful response returns media with the tag "Corvette".

```
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="corvette.jpg" public="0" uploaddate="" username="pbapi">

    <browseurl>http://s123.photobucket.com/albums/gg17/pbapi/TagStuff/?action=view&current=corvette.jpg</browse
      <url>http://i123.photobucket.com/albums/gg17/pbapi/TagStuff/corvette.jpg</url>
      <thumb>http://i123.photobucket.com/albums/gg17/pbapi/TagStuff/th_corvette.jpg</thumb>
      <description />
      <title />
    </media>
  </content>
</format>xml</format>
<method>GET</method>
<timestamp>1206376431</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **102:** User not found

For a full list of possible errors, see [Error Codes](#).

Get User Upload Options

Get the user's current upload options.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/uploadoption

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String

Request Example

The request gets the upload options for user “pbapi”.

```
http://api123.photobucket.com/user/pbapi/uploadoption?format=xml&oauth_consumer_key=12344321&oauth_nonce=3065f89aff4eae903b066d15b8fa484e&oauth_signature=q%2FAp7d5sByCIRHqbpvOKMNWzxR0%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1206377327&oauth_token=28.1&oauth_version=1.0
```

The request gets the upload options for user “pbapi”.

```
http://api.photobucket.com/user/pbapi/uploadoption?format=xml&oauth_consumer_key=12344321&oauth_nonce=3065f89aff4eae903b066d15b8fa484e&oauth_signature=q%2FAp7d5sByCIRHqbpvOKMNWzxR0%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1206377327&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response returns the default image size set for the user and whether auto-tagging is enabled.

```
<response>
  <status>OK</status>
  <content>
    <defaultimagesize>800</defaultimagesize>
    <autotagging>>false</autotagging>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1206377327</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **102:** User not found
- **119:** Missing url

For a full list of possible errors, see [Error Codes](#).

Update User Upload Options

Update the user's current upload options.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

POST or PUT

REST Path

/user/[identifier]/uploadoption

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		User name.	String
defaultimagesize	Y	nochange	Default image upload size number. One of: (value => resize result dimensions) • 100 => 100x75 • 160 => 160x120 • 320 => 320x240 • 640 => 640x480 • 800 => 800x600 • 1024 => 1024x768 • 1280 => 1280x960, //premium owner only • 1600 => 1600x1200, //premium owner only • 2048 => 1048x1536, //premium owner only • 2240 => 2240x1680, //premium owner only	String
autotagging	Y	nochange	Automatically tag after upload.	Boolean

Request Example

The request changes the default image size for user "pbapi".

```
PUT http://api123.photobucket.com/user/pbapi/uploadoption?defaultimagesize=800&format=xml&oauth_consumer_key=00000000&oauth_nonce=8971751e238547ddbfa781bd07de684&oauth_signature=XQ5YKkp%2FquZxKIrjw%2FSKYXZqi5c%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443507&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response looks similar to:

```
<response>  
  <status>OK</status>  
  <format>xml</format>  
  <method>PUT</method>  
  <timestamp>1202851025</timestamp>  
</response>
```

Error Codes

The following errors may be returned:

- **009**: Insufficient permissions
- **124**: Invalid parameter:
- **125**: Invalid image size

For a full list of possible errors, see [Error Codes](#).

Get User URLs

Get the Photobucket URL sub-domain and path information for a specific user account.

User Login Required

No.

HTTP Method

GET

REST Path

/user/[identifier]/url

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String

Request Example

The request gets the user album URL for user "pbapi".

```
GET http://api123.photobucket.com/user/pbapi/url?format=xml&oauth_consumer_key=00000000&oauth_nonce=3f738df2387d5e8b018f171211c32e44&oauth_signature=GaK%2FfKXr7pjjV1h41F03948s41s%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1216404672&oauth_token=&oauth_version=1.0
```

Response Example

The successful response includes the sub-domain, album, image, API, and feed, as well as the URL path. Note the **gg17** in the path below. A value similar to this displays for all albums and is automatically generated by Photobucket. See [Conventions](#) for details.

```
<response>
  <status>OK</status>
  <content>
    <username>pbapi</username>
    <subdomain>
      <album>s236/album>
      <image>i236/image>
      <api>api236/api>
      <feed>feed236/feed>
    </subdomain>
    <path>albums/gg17/pbapi</path>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1216404672</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **102:** User not found

For a full list of possible errors, see [Error Codes](#).

Get User Group Albums

Get a list of group albums owned by the user and contributed to by the user.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/group

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String

Request Example

The request gets allgroup albums from the user account "pbapi".

```
GET http://api.photobucket.com/user/pbapi/group?format=xml&oauth_consumer_key=000000000&oauth_nonce=764d6ad91184162bb29e0842fe6a84e2&oauth_signature=1BlSNk6bKXBp2hQP6oFsyZzvMs8%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236630917&oauth_token=31.00_1236630917&oauth_version=1.0
```

Response Example

The successful response includes the group albums the user owns and the group albums to which the user has contributed.

```
<response>
  <status>OK</status>
  <content>
    <owned>
      <name>zipzipzip</name>
      <id>ADP8DAIIFH</id>
      <url>http://gs403.photobucket.com/groups/hh251/ADP8DAIIFH</url>
    </owned>
    <contributed>
      <id>FCCQJ65B1</id>
      <url>http://gs204.photobucket.com/groups/hh202/FCCQJ65B1</favurl>
    </contributed>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1236630918</timestamp>
</response>
```

Error Codes

There are no specific error codes for this method. See [Error Codes](#) for a full list.

Get Items the User Follows

Get a list of all items that a logged-in user follows and who follows that user. Or, get a list of all items another user follows and who follows that user, from that user's public profile.

User Login Required

Optional. If you are not logged in, you only see public data. If you are logged in, you see all data. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/following/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String
actionIdentifier	Y	One of people, searches, groups, or followers. Default retrieves all types.	String

Request Example 1

The request gets a list of everything the logged in user "pbapi" is following and the users who are following pbapi.

```
GET http://api123.photobucket.com/user/pbapi/following?format=xml&oauth_consumer_key=000000000&oauth_nonce=88b3b1f6b981fd7691e2ca81357b7859&oauth_signature=O3m9GYgT4xU%2Bo%2FaNy%2B6mI1xaPMg%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236631030&oauth_token=31.00_1236631030&oauth_version=1.0
```

Response Example 1

The successful response includes everything the user "pbapi" is following. The following response is not a complete list.

```
<response>
  <status>OK</status>
  <content>
    <followers>
      <name>test_user</name>
      <fullname />
      <owner_type>group</owner_type>
      <url>http://api204.photobucket.com/albums/ww71/test_user/</url>
    </followers>
    <people>
      <subalbums>
        <name>Funny</name>
        <ownerType>user</ownerType>
        <count>1</count>
        <url>/users/@89750225/Best
Images/Funny?sort=ascending&amp;albumpath=findstuff22%2FBest+Images%2FFunny&amp;subscription=2335103</url>
        <albumPath>findstuff22/Best Images/Funny</albumPath>
        <userSubscriptionId>2335103</userSubscriptionId>
      </subalbums>
      <subalbums>
        <name>Quotes and Sayings</name>
        <ownerType>user</ownerType>
        <count>1</count>
        <url>/users/@89750225/Best Images/Quotes and
Sayings?sort=ascending&amp;albumpath=findstuff22%2FBest+Images%2FQuotes+and+Sayings&amp;subscription=2335103
        <albumPath>findstuff22/Best Images/Quotes and Sayings</albumPath>
        <userSubscriptionId>2335103</userSubscriptionId>
```

```

        </subalbums>
        <name>Find Stuff User</name>
        <ownerType>user</ownerType>
        <count>0</count>
        <url>/users/@89750225?sort=ascending</url>
        <albumPath />
        <userSubscriptionId>2335103</userSubscriptionId>
    </people>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1236631030</timestamp>
</response>

```

Request Example 2

The request gets a list of everything ANOTHER user ("test_user") is following and who follows that user, from that user's public profile.

```

GET http://api123.photobucket.com/user/test_user/following?format=xml&oauth_consumer_
key=000000000&oauth_nonce=88b3b1f6b981fd7691e2ca81357b7859&oauth_
signature=03m9GYgT4xU%2Bo%2FaNy%2B6mIlxaPMg%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1236631030&oauth_token=31.00_1236631030&oauth_version=1.0

```

Response Example 2

The successful response includes everything the user "test_user" is following. The following response is not a complete list.

```

<response>
  <status>OK</status>
  <content>
    <people>
      <name>pbapi</name>
      <url>http://photobucket.com/albums/ww71/pbapi</url>
    </people>
    <people>
      <name>findstuff22</name>
      <url>http://photobucket.com/home/findstuff22</url>
    </people>
    <searches>
      <name>color splash</name>
      <url>http://photobucket.com/images/photography?filter=newest</url>
    </searches>
    <groups>
      <name>Amateur Photography Group</count>
      <url>http://photobucket.com/grouphome/1XNAK16IES</url>
    </groups>
    <followers>
      <name>tester2</name>
      <url>http://photobucket.com/albums/ww71/tester2</url>
    </followers>
  </content>
</format>xml</format>

```

```
<method>GET</method>
<timestamp>1236631030</timestamp>
</response>
```

Error Codes

There are no specific error codes for this method. See [Error Codes](#) for a full list.

Get Media a User Follows

Get a list of media for the latest activity that a user follows.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/followingmedia/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Default	Description	Variable
identifier	N		User name.	String
actionIdentifier	Y		One of user, search_term, or group. Default retrieves all types.	String
perpage	Y	20	Number of results to return for main search type. Maximum of 100.	Integer
page	Y	1	Page number to display (1 indexed).	Integer

Request Example

The request gets a list of everything the user "pbapi" is following.

```
GET http://api123.photobucket.com/user/pbapi/followingmedia?format=xml&oauth_consumer_
key=000000000&oauth_nonce=88b3b1f6b981fd7691e2ca81357b7859&oauth_
signature=03m9GYgT4xU%2Bo%2FaNy%2B6mI1xaPMg%3D&oauth_signature_method=HMAC-SHA1&oauth_
timestamp=1236631030&oauth_token=31.00_1236631030&oauth_version=1.0
```

Response Example

The successful response includes the media in Latest Activity that the user follows.

```
<response>
  <status>OK</status>
  <content>
    <media description_id="" name="musicquotes1.jpg" public="1" type="image" uploaddate=""
username="findstuff22">

    <browseurl>http://api.photobucket.com/albums/0006/findstuff22/Best%20Images/Quotes%20and%20Sayings/?action=

    <url>http://api.photobucket.com/albums/0006/findstuff22/Best%20Images/Quotes%20and%20Sayings/musicquotes1.jpg
```

```

<thumb>http://api.photobucket.com/albums/0006/findstuff22/Best%20Images/Quotes%20and%20Sayings/th_
musicquotes1.jpg</thumb>
  <description />
  <title />
  <isSponsored />
  <type>user</type>
</media>
  <media description_id="" name="15.jpg" public="1" type="image" uploaddate=""
username="special_buddy_zip">
  <browseurl>http://api.photobucket.com//albums/j94/special_buddy_
zip/Avatars/Jessica%20Alba/?action=view&current=15.jpg</browseurl>
  <url>http://api.photobucket.com/albums/j94/special_buddy_
zip/Avatars/Jessica%20Alba/15.jpg</url>
  <thumb>http://api.photobucket.com/albums/j94/special_buddy_zip/Avatars/Jessica%20Alba/th_
15.jpg</thumb>
  <description />
  <title>Jessica Alba</title>
  <isSponsored />
  <search_term>jessica alba</search_term>
  <type>search_term</type>
</media>
</content>
<format>xml</format>
<method>GET</method>
<timestamp>1236631030</timestamp>
</response>

```

Error Codes

There are no specific error codes for this method. See [Error Codes](#) for a full list.

Get User Connections

Get a user's "share" connections.

User Login Required

Yes. See [End-User Authentication](#) for details.

HTTP Method

GET

REST Path

/user/[identifier]/share/[actionIdentifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name.	String
actionIdentifier	N	Service name. Options are all, twitter, facebook, or myspace. Default is "all"	String

Request Example

The request gets the connections for the user "pbapi".

```
GET http://api123.photobucket.com/user/pbapi/share
```

Response Example

A successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <facebook>
      <connected>true</connected>
      <name>Test</name>
      <img>http://static.ak.connect.facebook.com/pics/q_silhouette.gif</img>
      <link>http://www.new.facebook.com/profile.php?id=100001030313912</link>
    </facebook>
    <myspace>
      <connected>>false</connected>
    </myspace>
    <twitter>
      <connected>true</connected>
      <name>test</name>
      <img>http://s.twimg.com/a/1272578449/images/default_profile_6_normal.png</img>
      <link>http://twitter.com/test</link>
    </twitter>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1273007041</timestamp>
</response>
```

Error Codes

For a full list of possible errors, see [Error Codes](#).

Get User Name Availability

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT](#) WITH PHOTOBUCKET

Get availability of a user name, and make user name suggestions if the desired name is not available.

User Login Required

No.

HTTP Method

GET

REST Path

/user/[identifier]/suggest

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	User name to check and make suggestions against.	String

Request Example

The request asks for alternate username suggestions for "billy".

```
GET http://api.photobucket.com/user/billy/suggest?format=xml&oauth_consumer_key=00000000&oauth_nonce=9110ce8c0caea3fa89a0ef58700531fb&oauth_signature=CJvOT2mDefx15A2CSwCdAeQTB%2BQ%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1203443506&oauth_token=28.1&oauth_version=1.0
```

Response Example

The successful response includes suggestions for available user names.

```
<response>
  <status>OK</status>
  <content>
    <available>0</available>
    <suggestion>
      <suggestion>billy_2008</suggestion>
    </suggestion>
    <suggestion>
      <suggestion>billy_027</suggestion>
    </suggestion>
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1202850227</timestamp>
</response>
```

Error Codes

The following error may be returned:

- **155:** Username missing

For a full list of possible errors, see [Error Codes](#).

Register User

THIS METHOD CAN BE ENABLED PER YOUR [COMMERCIAL AGREEMENT](#) WITH PHOTOBUCKET

Create a Photobucket account for a user.

User Login Required

No.

HTTP Method

POST

REST Path

/user/[identifier]

See [Object Identifiers](#) for an [identifier] explanation.

Parameters

Parameter	Optional	Description	Variable
identifier	N	Username to register. Usernames must be 3 to 50 characters, can only contain the characters a-z, A-Z, 0-9, -, _, and must begin with either a letter, a number, or an underscore. Spaces are not allowed.	String
password	N	Password (plain text). Passwords must be at least 6 characters long. They can only contain the characters a-z, A-Z, 0-9, -, _, cannot contain spaces, and cannot start with a zero.	String
email	N	User's email address.	String
birthmonth	N	User's month of birth. E.g., 04 or 4.	Integer
birthyear	N	User's year of birth. E.g., 1980.	Integer
birthday	N	User's day of birth. E.g., 05 or 5.	Integer
gender	N	M = Male, F = Female	String
country	N	Two letter country code. E.g., CA. See the ISO list of country codes at ISO Country Codes .	String
tosaccepted	N	Terms of Service accepted.	Boolean
cobrand	Y	Name of the co-branding sponsor.	String
firstname	Y	User's first name.	String
lastname	Y	User's last name.	String
zipcode	Y	User's postal or zip code (Canada and US only).	String
cellphone	Y	User's cell phone number, in 10-digit format (e.g., 3035551212).	Integer

Request Example

The following request registers a user.

```
POST http://api.photobucket.com/user/newusername
```

Response Example

The successful response looks similar to:

```
<response>
  <status>OK</status>
  <content>
    <username>newusername</username>
    <album_url>http://api.photobucket.com/albums/nn182/newusername/</album_url>
    <user_id>56021898</user_id>
    <megabytes_used>0</megabytes_used>
    <megabytes_allowed>500<megabytes_allowed/>
    <premium>0</premium>
    <public>1</public>
    <preferred_picture_size>1024</preferred_picture_size>
  </content>
  <format>xml</format>
  <method>POST</method>
  <timestamp>1206398738</timestamp>
</response>
```

Error Codes

The following errors may be returned:

- **200**: Required field missing:
- **201**: Invalid username format
- **202**: Username unavailable
- **203**: Invalid password format
- **204**: Invalid email address:
- **205**: Account already exists for this email address:
- **206**: Invalid birthmonth
- **207**: Invalid birthday
- **208**: Invalid birthyear
- **209**: User must be at least 13 years old to register
- **210**: Invalid gender
- **211**: Invalid country

For a full list of possible errors, see [Error Codes](#).

Frequently Asked Questions

The Frequently Asked Questions is divided into sections:

API Overview

[What is the API?](#)

[Who should use the API?](#)

[What other options do I have?](#)

[Is the API free to use?](#)

[How do I use the API?](#)

[How do I know if I need a commercial key?](#)

[What kind of support does Photobucket provide for the API?](#)

[How often is the API updated, and how do I know there are updates?](#)

[Do I have to submit my application to Photobucket before I use it?](#)

[Can I use the API as a white-box media host, without referencing or linking back to Photobucket](#)

[How do I request API functionality?](#)

Developer and User Accounts

[I have a Photobucket account. Do I automatically have a Developer account?](#)

[How do I create a Developer account?](#)

[Can I log into my Photobucket account and my Developer account at the same time?](#)

Application Gallery

[How do I get my application in the Application Gallery?](#)

[What is the application of the day? How do I get my application featured?](#)

Technical and OAuth

[Does Photobucket provide client libraries I can use?](#)

[Why am I redirected to api###.photobucket.com when I access an album?](#)

[How does the OAuth base-string encoding work?](#)

[What is the \[identifier\] in the Methods, and why does it look the way it does?](#)

[What do I do if I'm using .Net and my client is fighting me?](#)

[How do I use Web Authentication with my desktop application?](#)

[Why am I getting a "timestamp invalid-\[time\]" message?](#)

FAQ: API Overview

[What is the API?](#)

[Who should use the API?](#)

[What other options do I have?](#)

[Is the API free to use?](#)

[How do I use the API?](#)

[How do I know if I need a commercial key?](#)

[What kind of support does Photobucket provide for the API?](#)

[How often is the API updated, and how do I know there are updates?](#)

[Do I have to submit my application to Photobucket before I use it?](#)

[Can I use the API as a white-box media host, without referencing or linking back to Photobucket](#)

[How do I request API functionality?](#)

What is the API?

The Photobucket API is an application programming interface that enables software developers to give Photobucket users access to their online images and videos from environments outside of Photobucket, including other Web sites, Web applications, desktop applications, browser plug-ins, and devices that go beyond the desktop, such as mobile phones, home-entertainment systems, digital photo frames, and digital cameras.

The API allows you to programmatically access much of the functionality of the Photobucket site. This includes Search, Uploading, User Albums, and Media Tagging and Titling. See [Methods in the API Documentation](#).

Who should use the API?

You should consider using the API for your application if:

- You are a developer who knows what REST is, what an API is, and you know XML and/or JavaScript.
- You are creating a product, and you want to support Photobucket users and their content.
- You have an application that requires Photobucket, but you do not want to display the Photobucket interface.

What other options do I have?

In addition to the API, Photobucket provides two additional options:

- **Media Plug-in** – The Photobucket Media Plug-in is a free Plug-in that lives in an iFrame on your web site and allows your users to upload and publish content from their Photobucket account without leaving your site. See the [Photobucket Media Plug-in](#) documentation for implementation information.

- **RSS feeds** – Photobucket provides RSS (Really Simple Syndication) feeds as a quick way to monitor album activity, recent media activity, specific search terms, and categories. Photobucket RSS feeds only contain public media, they are ordered by upload date, and they are only published every hour. See the [Photobucket RSS feeds](#) documentation for more information.

Is the API free to use?

The API is free to use with a non-commercial key. If you expect to exceed the [Method Throttle Limits](#), expect your application to generate more than \$5000 USD/month, or you are writing a mobile or consumer electronic application, you may require a commercial key.

How do I use the API?

Before you can use the API, you must:

1. Sign up for a Photobucket User account (if you do not already have one).
2. Go to the Photobucket developer web site at developer.photobucket.com, agree to the terms of service, and sign up as a Developer. A non-commercial API key will be emailed to you.
3. Understand the code sample conventions and REST request and response formats.
4. Implement OAuth request signing for your application.
5. Determine which requests you want to send. Occasionally, a request requires data from a previous response.
6. Understand the error codes that may be sent in a response.
7. Review the [throttle limits](#) and if necessary, request a commercial key.

Information about the requirements listed in steps 3 to 7 is provided in the [API Documentation](#).

The API is accessed via HTTP commands to the API, as outlined in the documentation, and produces responses in your specified format (e.g., XML or JSON). Access is controlled via OAuth authentication.

How do I know if I need a commercial key?

You need a commercial key if you:

- Expect to exceed the [Method Throttle Limits](#).
- Expect your application to generate more than \$5000 USD/month.
- Are writing a mobile or consumer electronic application.

What kind of support does Photobucket provide for the API?

When you sign up as a Developer to use the API, you can access the Developer forum where many common questions are answered. Additionally, Photobucket provides client libraries and help documentation. Although Photobucket makes the best effort to help whenever possible, our resources are limited. As much as Photobucket would like to write more code samples and correct code that you write, support at that level is not guaranteed.

How often is API updated, and how do I know there are updates?

The API is updated as needed to support new functionality or fix defects. When the API and the API documentation are updated, an entry is published on the [Developer Blog](#) on the Developer forum. Subscribe to the RSS feed for the Developer Blog to receive prompt notification of new entries.

Do I have to submit my application to Photobucket before I use it?

You do not have to submit your application to Photobucket unless you want it considered for inclusion in the Application Gallery.

Can I use the API as a white-box media host, without referencing or linking back to Photobucket?

No. The API is meant to augment the features on the Photobucket service, not replace the site.

How do I request API functionality?

If there is functionality you want to see in the API, post your request on the [Photobucket API Support](#) forum. If another developer has already requested the functionality, add your request for functionality to the existing thread.

FAQ: Developer and User Accounts

[I have a Photobucket account. Do I automatically have a Developer account?](#)

[How do I create a Developer account?](#)

[Can I log into my Photobucket account and my Developer account at the same time?](#)

I have a Photobucket account. Do I automatically have a Developer account?

You must agree to the Developer Terms of Use and sign up for an API key before your Photobucket User account becomes a Photobucket Developer account, too.

How do I create a Developer account?

1. Log into your Photobucket account. If you do not have a Photobucket account, [join now](#).
2. Click **developers** in the menu at the bottom of the page.
3. Click **Register a key** in the menu on the left.
4. Enter the required information. Provide as much of the optional information as you can.
5. Read the Developer Addendum and the Photobucket Agreement, then select the checkbox that indicates you have read these documents.
6. Click **Register**. A click-through email will be sent to the email address you provided.
7. Follow the instructions in the email.

Can I log into my Photobucket account and my Developer account at the same time?

After you create a Developer account, when you log into your Photobucket account (with your albums and media), you are also logged into your Developer account.

FAQ: Application Gallery

[How do I get my application in the Application Gallery?](#)

[What is the Application of the Day? How do I get my application featured?](#)

How do I get my application into the Application Gallery?

You can submit your application to Photobucket to be considered for the Application Gallery.

1. Log into the [developer site](#).
2. Click **Edit** beside the application you want to submit, then fill in the Developer information and Gallery requirements.
3. Click **Add to gallery** to submit your application.

Photobucket may also ask certain applications if they wish to be included in the gallery.

What is the Application of the Day? How do I get my application featured?

The Application of the Day showcases specific applications that use the Photobucket API. The Application of the Day is selected by Photobucket, from the applications that are submitted to the Application Gallery.

FAQ: Technical and OAuth

[Does Photobucket provide client libraries I can use?](#)

[Why am I redirected to api###.photobucket.com when I access an album?](#)

[How does the OAuth base-string encoding work?](#)

[What is the \[identifier\] in the Methods, and why does it look the way it does?](#)

[What do I do if I'm using .Net and my client is fighting me?](#)

[How do I use Web Authentication with my desktop application?](#)

[Why am I getting a "timestamp invalid -\[time\]" message?](#)

Does Photobucket provide client libraries I can use?

Photobucket provides [client libraries](#) for you to use and a [forum](#) dedicated to client library support.

Why am I redirected to api###.photobucket.com when I access an album?

Photobucket separates users into 'silos'; each one accessible by a specific number. If you check the User URLs response for a given user, you can see which sub-domains that user can be accessed by. Photobucket also provides this information in the token access responses. To access a user's content, you must interact with that specific silo, and Photobucket provides the redirects as a convenience.

Each user and group may have a different subdomain. The subdomain is based on the album (user or group) you are accessing. This holds true for Album, Group, Media and User methods. You can get these subdomains by issuing a GET request to [/album/\[identifier\]/url](#), [/user/\[identifier\]/url](#), or [/group/\[identifier\]/url](#), or by following and caching the redirects.

Some HTTP clients cannot handle these redirects, and Photobucket ultimately prefers that you use the information given, rather than relying on the redirects. The silo information for a user can safely be cached for a time, per user, to save calls. The silo hostnames can present a problem with OAuth signing, which requires the entire hostname and URL in the signature. For this reason, we require ALL requests, regardless of the silo being accessed, to use "http://api.photobucket.com" as the beginning of the URL being hashed in the OAuth signature process to help work around the issue of redirects and hostnames.

How does the OAuth base-string encoding work?

The OAuth basestring will look like certain parts are double encoded (mainly, the parameters to the request). This is normal, and you should not see characters other than ALPHA / DIGIT / "-" / "." / "_" / "~", and obviously the %XX characters in the encoded fields, separated by unencoded "&"

Parts:

- methodString = urlencode(method)
 - Actual HTTP method being used.
For example, "GET"
- urlString = urlencode(url)
 - URL (not including query string) being accessed, no trailing slash.
For example, http://api.photobucket.com/user/username/url
- for each param (name and value)
 - Push on paramArray = name+'='+urlencode(value).
For example, "foo=bar%20baz"
- baseString = methodString+'&'+urlString+'&'+urlencode(join('&', paramArray))
 - For example:
GET&http%3A%2F%2Fapi.photobucket.com%2Fuser%2Fusername%2Furl&format%3Dxml%26oauth_consumer_key%3D000010100%26oauth_nonce%3Dd7ehf0hdx0vhf0w%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1205433090%26oauth_version%3D1.0

Using a pre-built library from OAuth.net can do this for you.

What is the [identifier] in the Methods, and why does it look the way it does?

The [identifier] is based on the Photobucket web site URL structure:

User name > User album > Sub-album > Sub-album, etc.

The user name is ALSO the *main user album name*.

For example, imagine a Photobucket user, *pbapi*. This user has one sub-album (sub1) in the main album, and one sub-album (sub4) in album "sub1". The user has media (image1, image2, and video1) in these albums. SO:

User Name= pbapi

User Album = pbapi

Albums = sub1 and sub1/sub4

Media = image1 in pbapi, image2 in album sub1, and video1 in album sub1/sub4

To GET album "sub4" for user "pbapi", the REST path is:

/album/[identifier]

And the request is:

GET http://api123.photobucket.com/album/pbapi%2Fsub1%2Fsub4

- "123" is the silo number in which the user's media is stored (see [Why am I redirected to api###.photobucket.com when I access an album](#))
- "pbapi" is the user name/user album
- %2F is the encoding of the slash "/"
- "pbapi%2Fsub1%2Fsub4" is the path to the user's sub4 album

Group albums have their own "group hash value" or **Group ID** in their URLs. The Group ID is a short, all-caps string on the URL for a group. For example, the group ID for the featured group album "Woof" is "20IF1LMOT", as displayed in the group URL:

http://gs275.photobucket.com/groups/jj285/**20IF1LMOT**?featuredgroupalbum=WooF

To get media for the group album "Woof", the REST path is:

/group/[identifier]

And the request is:

GET http://api275.photobucket.com/group/**20IF1LMOT**

- "275" is the silo number in which the group album is stored (see [Why am I redirected to api###.photobucket.com when I access an album](#))
- **20IF1LMOT** is the group identifier

These conventions hold true for all methods (GET, PUT, POST, DELETE), including uploads. See [Conventions](#) for more information. For an upload, the POST URL is still /album/[identifier]/upload, and a multipart request may look like http://pic.photobucket.com/dev_help/pbapi_upload.txt

What do I do if I'm using .Net and my client is fighting me?

If you have trouble accessing sub-albums, there is a known incompatibility with using the '/' character in URLs when using .NET. To help get around this, Photobucket provides Alternate Identifier Specifications to assist developers who are using .Net. See the [Alternative Identifier Specifications](#) section in the Conventions topic.

How do I use Web Authentication with my desktop application?

Open the [photobucket.com] page with the system's default http protocol handler.

- Do not enter a callback URL. Photobucket displays a page that says to close this window. Or, redirect to your own landing page that says the same thing. Show a "continue" button in your application, and exchange your tokens.
- Capture the 'end' page through a control in a manner similar to the preceding point. Do not use the default handler. Instead, use something like the Windows built-in for rendering a web page, and handle the end page.
- Register (temporarily?) a custom protocol handler, and use that as the callback (like, x-yourappname://oauth-done), which can let your program continue. This is the Photobucket preferred method.

If for any reason you cannot implement one of the preceding three suggestions (for example, you have a consumer electronics device, like a set-top box), contact Photobucket for further assistance.

Why am I getting a "timestamp invalid -[time]" message?

Important! Make sure you use UTC time for the source.

When you see a timestamp difference (the number in the message) that is greater than a few thousand, check your server's timestamp and compare it to the message. Typically, the time stamps are close or the same. A difference greater than a few thousand means the API server did not receive a proper timestamp field in your request. The difference can occur for a number of reasons:

1. Your OAuth request was not signed.
2. Your OAuth request was signed, but on a redirect (response code 301-307), your library did not resend all appropriate data. This is a common problem with POSTs. The [HTTP RFC \(for 30x\)](#) says that the method should be maintained for at least 301 and 307, but does not specifically state that the request payload (in the case of POST here) should also be resent. This is an ambiguity. Because the Photobucket API needs that POST payload to operate on, you **MUST** re-send all data on 301 and 307 response codes. For many libraries (including cURL) extra code around the built-in libraries is required for proper handling.
3. Your server's (or client's) clock is off by more than Photobucket allows. For small values of the time-delta in the exception message, this is the most likely case. You may need to use a more reliable time source to initially seed your application's time stamps. Photobucket recommends getting the time from a reliable source (such as the /time method on the Photobucket API) at the invocation of the client, calculating the delta between the reliable timestamp and the host machine's timestamp, and using that as an offset on later requests. For example, if Photobucket replies with a time of 1220566660, but your local install reports 1220561160, the difference is +5500. You can apply that difference to later operations by taking the local install's time (for example, 1220592112) and adding the offset (resulting in 1220597612) to reduce the delta between your request's timestamp and the server timestamp.

Make sure that you send all the appropriate headers (content-type, etc) so the Photobucket server knows you are sending a multipart/form-data type message. Also be sure that you have a valid mime payload. If the whole message is not formatted properly, the server may discard the payload.

References

The References section contains:

- [Code libraries](#) that you can use in your development efforts.
- [Error Codes](#) that you may receive when you use the API methods
- [Method Throttle Limits](#) that are applied by Photobucket to the API Methods

Code Libraries

Photobucket provides a number of unsupported, as-is code libraries that you can use.

C#

[PhotobucketNet](#) by user schallm2

[Silverlight 2](#) targeted library by Michael S. Scherotter (mscherotter) ([thread](#)) - has unit tests

PHP

[PHP5 Fluent library](#) by Justin Hart (jhart, Photobucket developer) ([thread](#))

ActionScript 3

[Actionscript 3 Photobucket API Library](#) by Jared Lewark (jlewark, Photobucket developer) ([thread](#))

Java

[Photobucket Java API](#) by Todd Benge (t_benge, Photobucket developer)

Error Codes

You may receive any of the following error codes when using the Photobucket API.

Error Code	Description
General Errors	
001	Service currently unavailable
002	Missing scid
003	Missing sig
005	Missing service_token
006	Login failed: Invalid Session Key
007	Authentication failed
008	Authentication failed: User did not authorize API call.
009	Insufficient permissions
010	Missing version
011	Invalid version
012	Service Token Expired
013	User Authentication failed
100	Invalid method
101	Missing parameter
102	User not found
103	Invalid search query
104	Invalid characters in query
105	Invalid file for upload
106	Upload error
107	Invalid Album
108	Insufficient privileges to access private album
109	Album is over 1000 file limit
110	No file uploaded
111	Album is over the size limit of:
112	File has invalid MIME type
113	A file with this name already exists
114	Video server error:
115	Album name must be between 2 and 50 alphanumeric characters
116	Album already exists:
117	Missing album name
118	Missing email address
119	Missing url

Error Code	Description
120	Missing email body
121	Description too long
122	Filename must contain only letters, numbers, underscores and hyphens
123	Invalid Media URL
124	Invalid parameter:
125	Invalid image size:
126	Title too long
127	SWF file uploading is no longer available on free accounts...
128	Album not empty
129	Unable to delete album
130	Invalid upload type
131	The tag exceeds the maximum length
132	Tag link is invalid
133	Contact is invalid
134	Tag dimensions are invalid
135	Exceeded maximum number of tags
136	Tag is required
137	Tag ID is invalid
138	Tag is not owned by the authenticated user
139	The tag was not found
140	Either the tag or the coordinates are required for update
141	Media specified was not found
142	Rotation degrees must be in 90 degree increments
143	The album specified was not found
144	The album name is missing
145	The album depth was exceed for public / private
146	Unable to change public private settings
147	Failed changing the password
148	The privacy parameter is missing
149	The privacy parameter is invalid
150	Unable to save the media
151	The description parameter is required
152	The title parameter is required
153	The size parameter is required
154	The degrees parameter is required
155	Username missing
156	Password missing
159	Album was already public

Error Code	Description
162	No album vanity defined
169	Rating should be between 1 and 5
170	User has already rated this media
Registration Errors	
200	Required field missing
201	Invalid username format
202	Username unavailable
203	Invalid password format
204	Invalid email address
205	Account already exists for this email address
206	Invalid birthmonth
207	Invalid birthday
208	Invalid birthyear
209	User must be at least 13 years old to register
210	Invalid gender
211	Invalid country
Image Manipulation Errors	
300	Source file does not exist
301	Internal system command error
302	Source file is over the file size limit of
303	Malformed source file path
304	We are sorry, this particular image cannot be processed by Photobucket.
305	Cannot increase image size
306	I refuse to resize files with this extension
307	This animation has too many frames to resize
310	Resulting file does not exist
311	Resulting image has no:
312	Resize error:
314	Resulting image is too large
315	Unable to create symlink
Unknown Error	
999	An unknown error has occurred

Method Throttle Limits

The following table lists the throttle limits for generally available methods. Numbers shown are requests per day.

Method Name	Throttle Limit
Ping Photobucket	10000

Web User Login : Get Access Token	10000
Web User Login: Login Token Request	10000
Upload Media to an Album	10000
Get Album	2000
Create New Album	200
Rename Album	100
Delete Album	50
Get Album Privacy Settings	500
Update Album Privacy Settings	100
Get Media Info	2000
Delete Media	50
Get Media Description	2000
Set Media Description	200
Delete Media Description	200
Get Media Links	2000
Get Media Meta Data	500
Get Related Media	1000
Resize Image	200
Rotate Image	200
Share Media via Email	50
Add Media Tag	500
Get Media Tag	2000
Update Media Tag	200
Delete Media Tag	100
Get Media Title	2000
Set Media Title	200
Get Media Tags for a Group Album	5000
Get User Contacts	50
Get User Information	5000
Get Recent User Media	5000
Get User Upload Options	200
Delete Media Title	200
Upload Media to a Group Album	5000
Update User Upload Options	200
Get User URLs	20000
Get Media Tags for a User	5000
Update Album Privacy Settings	100
Get Album Vanity URL	500
Get Featured Group Albums	10000

Get Featured Home Page Media	10000
Get Group Album Contributors	1000
Get Media for a Group Album	5000
Get Group Album Information	1000
Set Group Album Information	200
Get Group Album Privacy Settings	1000
Update Group Album Privacy Settings	100
Get Group Album Vanity URL	500
Search Group Albums	10000
Search Images	10000
Search Videos	10000

Terms of Service

Your use of these materials is at your own risk and implies that you have a signed Non-Disclosure Agreement with Photobucket (if applicable), that you accepted the Developer Addendum at <http://photobucket.com/devaddendum> when you signed up for a key, and that you also accepted the Photobucket Terms of Service at <http://photobucket.com/terms>.