**Tumblr API**

Welcome to the Tumblr API! There isn't anything we enjoy more than seeing talented designers and engineers using Tumblr to invent whole new forms of creative expression. We've put a tremendous amount of care into making this API functional and flexible enough for any projects you throw at it. Drop a line in the [GitHub Issues](https://github.com/tumblr/docs/issues) if you have ideas for what could be better, or want to discuss all the amazing things you're going to build with it. Follow our [Engineering Blog](https://engineering.tumblr.com/) for important news and announcements. Please use the API responsibly and read the [Application Developer and API License Agreement](https://www.tumblr.com/docs/api_agreement). If you run into problems with the Tumblr API, please use [our official support form](https://www.tumblr.com/support) to request help. Enjoy!

Tumblr also supports an API that delivers content according to the [oEmbed standard](http://oembed.com/). Our oEmbed API endpoint is https://www.tumblr.com/oembed, which supports post URLs in the format https://\*.tumblr.com/post/\*.

If you're looking for documentation for the old v1 API, you can find it [here](https://www.tumblr.com/docs/api/v1).

**Table of Contents**

* [What You Need](https://www.tumblr.com/docs/en/api/v2#what-you-need)
* [Console](https://www.tumblr.com/docs/en/api/v2#console)
* [API Overview](https://www.tumblr.com/docs/en/api/v2#api-overview)
  + [URI Structure](https://www.tumblr.com/docs/en/api/v2#uri-structure)
  + [Request User Agent](https://www.tumblr.com/docs/en/api/v2#request-user-agent)
  + [Request Content Types](https://www.tumblr.com/docs/en/api/v2#request-content-types)
  + [Post Identifiers](https://www.tumblr.com/docs/en/api/v2#post-identifiers)
  + [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers)
  + [Response Format](https://www.tumblr.com/docs/en/api/v2#response-format)
  + [Rate Limits](https://www.tumblr.com/docs/en/api/v2#rate-limits)
  + [Reporting Issues](https://www.tumblr.com/docs/en/api/v2#reporting-issues)
  + [About the API Documentation](https://www.tumblr.com/docs/en/api/v2#about-the-api-documentation)
* [Official Tumblr API Client Libraries](https://www.tumblr.com/docs/en/api/v2#official-tumblr-api-client-libraries)
* [Authentication](https://www.tumblr.com/docs/en/api/v2#authentication)
* [OAuth1 Authorization](https://www.tumblr.com/docs/en/api/v2#oauth1-authorization)
  + [Temporary Credentials Endpoint](https://www.tumblr.com/docs/en/api/v2#temporary-credentials-endpoint)
  + [Resource Owner Authorization Endpoint](https://www.tumblr.com/docs/en/api/v2#resource-owner-authorization-endpoint)
  + [Access Token Endpoint](https://www.tumblr.com/docs/en/api/v2#access-token-endpoint)
* [OAuth2 Authorization](https://www.tumblr.com/docs/en/api/v2#oauth2-authorization)
  + [/oauth2/authorize - Authorization Request](https://www.tumblr.com/docs/en/api/v2#oauth2authorize---authorization-request)
  + [/v2/oauth2/token - Authorization Code Grant Request](https://www.tumblr.com/docs/en/api/v2#v2oauth2token---authorization-code-grant-request)
  + [/v2/oauth2/token - Refresh Token Grant Request](https://www.tumblr.com/docs/en/api/v2#v2oauth2token---refresh-token-grant-request)
  + [/v2/oauth2/exchange - OAuth1 to OAuth2 Token Exchange](https://www.tumblr.com/docs/en/api/v2#v2oauth2exchange---oauth1-to-oauth2-token-exchange)
* [Common Response Elements](https://www.tumblr.com/docs/en/api/v2#common-response-elements)
  + [Links](https://www.tumblr.com/docs/en/api/v2#links)
  + [Tag Objects](https://www.tumblr.com/docs/en/api/v2#tag-objects)
  + [Neue Post Format objects](https://www.tumblr.com/docs/en/api/v2#neue-post-format-objects)
* [Blog Methods](https://www.tumblr.com/docs/en/api/v2#blog-methods)
  + [/info - Retrieve Blog Info](https://www.tumblr.com/docs/en/api/v2#info---retrieve-blog-info)
  + [/avatar — Retrieve a Blog Avatar](https://www.tumblr.com/docs/en/api/v2#avatar--retrieve-a-blog-avatar)
  + [/blocks – Retrieve Blog's Blocks](https://www.tumblr.com/docs/en/api/v2#blocks--retrieve-blogs-blocks)
  + [/blocks – Block a Blog](https://www.tumblr.com/docs/en/api/v2#blocks--block-a-blog)
  + [/blocks/bulk – Block a list of Blogs](https://www.tumblr.com/docs/en/api/v2#blocksbulk--block-a-list-of-blogs)
  + [/blocks – Remove a Block](https://www.tumblr.com/docs/en/api/v2#blocks--remove-a-block)
  + [/likes — Retrieve Blog's Likes](https://www.tumblr.com/docs/en/api/v2#likes--retrieve-blogs-likes)
  + [/following — Retrieve Blog's following](https://www.tumblr.com/docs/en/api/v2#following--retrieve-blogs-following)
  + [/followers — Retrieve a Blog's Followers](https://www.tumblr.com/docs/en/api/v2#followers--retrieve-a-blogs-followers)
  + [/followed\_by — Check If Followed By Blog](https://www.tumblr.com/docs/en/api/v2#followed_by--check-if-followed-by-blog)
  + [/posts – Retrieve Published Posts](https://www.tumblr.com/docs/en/api/v2#posts--retrieve-published-posts)
  + [/posts/queue — Retrieve Queued Posts](https://www.tumblr.com/docs/en/api/v2#postsqueue--retrieve-queued-posts)
  + [/posts/queue/reorder — Reorder Queued Posts](https://www.tumblr.com/docs/en/api/v2#postsqueuereorder--reorder-queued-posts)
  + [/posts/queue/shuffle — Shuffle Queued Posts](https://www.tumblr.com/docs/en/api/v2#postsqueueshuffle---shuffle-queued-posts)
  + [/posts/draft — Retrieve Draft Posts](https://www.tumblr.com/docs/en/api/v2#postsdraft--retrieve-draft-posts)
  + [/posts/submission — Retrieve Submission Posts](https://www.tumblr.com/docs/en/api/v2#postssubmission--retrieve-submission-posts)
  + [/notifications — Retrieve Blog's Activity Feed](https://www.tumblr.com/docs/en/api/v2#notifications--retrieve-blogs-activity-feed)
  + [/post — Create a New Blog Post (Legacy)](https://www.tumblr.com/docs/en/api/v2#post--create-a-new-blog-post-legacy)
  + [/post/edit – Edit a Blog Post (Legacy)](https://www.tumblr.com/docs/en/api/v2#postedit--edit-a-blog-post-legacy)
  + [/post/reblog – Reblog a Post (Legacy)](https://www.tumblr.com/docs/en/api/v2#postreblog--reblog-a-post-legacy)
  + [/posts - Create/Reblog a Post (Neue Post Format)](https://www.tumblr.com/docs/en/api/v2#posts---createreblog-a-post-neue-post-format)
  + [/posts/{post-id} - Fetching a Post (Neue Post Format)](https://www.tumblr.com/docs/en/api/v2#postspost-id---fetching-a-post-neue-post-format)
  + [/posts/{post-id} - Editing a Post (Neue Post Format)](https://www.tumblr.com/docs/en/api/v2#postspost-id---editing-a-post-neue-post-format)
  + [/post/delete – Delete a Post](https://www.tumblr.com/docs/en/api/v2#postdelete--delete-a-post)
  + [/posts/{post-id}/mute - Muting a Post's Notifications](https://www.tumblr.com/docs/en/api/v2#postspost-idmute---muting-a-posts-notifications)
  + [/notes - Get notes for a specific Post](https://www.tumblr.com/docs/en/api/v2#notes---get-notes-for-a-specific-post)
  + [/pages - Retrieve Pages](https://www.tumblr.com/docs/en/api/v2#pages---retrieve-pages)
  + [/pages/{page-name} - Retrieve a Page](https://www.tumblr.com/docs/en/api/v2#pagespage-name---retrieve-a-page)
* [User Methods](https://www.tumblr.com/docs/en/api/v2#user-methods)
  + [/user/info – Get a User's Information](https://www.tumblr.com/docs/en/api/v2#userinfo--get-a-users-information)
  + [/user/limits – Get a User's Limits](https://www.tumblr.com/docs/en/api/v2#userlimits--get-a-users-limits)
  + [/user/dashboard – Retrieve a User's Dashboard](https://www.tumblr.com/docs/en/api/v2#userdashboard--retrieve-a-users-dashboard)
  + [/user/likes — Retrieve a User's Likes](https://www.tumblr.com/docs/en/api/v2#userlikes--retrieve-a-users-likes)
  + [/user/following – Retrieve the Blogs a User Is Following](https://www.tumblr.com/docs/en/api/v2#userfollowing--retrieve-the-blogs-a-user-is-following)
  + [/user/follow – Follow a blog](https://www.tumblr.com/docs/en/api/v2#userfollow--follow-a-blog)
  + [/user/unfollow – Unfollow a blog](https://www.tumblr.com/docs/en/api/v2#userunfollow--unfollow-a-blog)
  + [/user/like – Like a Post](https://www.tumblr.com/docs/en/api/v2#userlike--like-a-post)
  + [/user/unlike – Unlike a Post](https://www.tumblr.com/docs/en/api/v2#userunlike--unlike-a-post)
  + [/user/filtered\_tags - Tag Filtering](https://www.tumblr.com/docs/en/api/v2#userfiltered_tags---tag-filtering)
  + [/user/filtered\_content - Content Filtering](https://www.tumblr.com/docs/en/api/v2#userfiltered_content---content-filtering)
* [Tagged Method](https://www.tumblr.com/docs/en/api/v2#tagged-method)
  + [/tagged – Get Posts with Tag](https://www.tumblr.com/docs/en/api/v2#tagged--get-posts-with-tag)
* [Communities Methods](https://www.tumblr.com/docs/en/api/v2#communities-methods)
  + [GET /v2/communities - List joined communities](https://www.tumblr.com/docs/en/api/v2#get-v2communities---list-joined-communities)
  + [GET /v2/communities/{community-handle} - Get info about a community](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handle---get-info-about-a-community)
  + [PUT /v2/communities/{community-handle} - Edit your community](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handle---edit-your-community)
  + [GET /v2/communities/{community-handle}/invitations - View pending invites](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handleinvitations---view-pending-invites)
  + [PUT /v2/communities/{community-handle}/invitations - Invite someone to the community](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handleinvitations---invite-someone-to-the-community)
  + [DELETE /v2/communities/{community-handle}/invitations/{blog-identifier} - Cancel an invitation](https://www.tumblr.com/docs/en/api/v2#delete-v2communitiescommunity-handleinvitationsblog-identifier---cancel-an-invitation)
  + [GET /v2/communities/{community-handle}/invitations/{blog-identifier} - Check invitation status](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handleinvitationsblog-identifier---check-invitation-status)
  + [POST /v2/communities/{community-handle}/invite\_hash - Regenerate the invite URL](https://www.tumblr.com/docs/en/api/v2#post-v2communitiescommunity-handleinvite_hash---regenerate-the-invite-url)
  + [GET /v2/communities/{community-handle}/members - Get community members](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handlemembers---get-community-members)
  + [PUT /v2/communities/{community-handle}/members - Join a community](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handlemembers---join-a-community)
  + [DELETE /v2/communities/{community-handle}/members/{blog-identifier} - Remove a member](https://www.tumblr.com/docs/en/api/v2#delete-v2communitiescommunity-handlemembersblog-identifier---remove-a-member)
  + [PUT /v2/communities/{community-handle}/members/{blog-identifier} - Change a member's role](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handlemembersblog-identifier---change-a-members-role)
  + [DELETE /v2/communities/{community-handle}/mute - Unmute a community](https://www.tumblr.com/docs/en/api/v2#delete-v2communitiescommunity-handlemute---unmute-a-community)
  + [PUT /v2/communities/{community-handle}/mute - Mute a community](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handlemute---mute-a-community)
  + [PUT /v2/communities/{community-handle}/posts/{post-id}/reactions - Add reaction to post](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handlepostspost-idreactions---add-reaction-to-post)
  + [DELETE /v2/communities/{community-handle}/posts/{post-id}/reactions/{reaction-id} - Remove reaction from post](https://www.tumblr.com/docs/en/api/v2#delete-v2communitiescommunity-handlepostspost-idreactionsreaction-id---remove-reaction-from-post)
  + [GET /v2/communities/{community-handle}/moderation/posts/{post-id} - View a moderated post](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handlemoderationpostspost-id---view-a-moderated-post)
  + [DELETE /v2/communities/{community-handle}/moderation/posts/{post-id} - Moderate a post](https://www.tumblr.com/docs/en/api/v2#delete-v2communitiescommunity-handlemoderationpostspost-id---moderate-a-post)
  + [PUT /v2/communities/{community-handle}/moderation/posts/{post-id} - Restore a post](https://www.tumblr.com/docs/en/api/v2#put-v2communitiescommunity-handlemoderationpostspost-id---restore-a-post)
  + [GET /v2/communities/{community-handle}/timeline - Get posts from a community](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handletimeline---get-posts-from-a-community)

**What You Need**

To get an OAuth key you must [register an application](https://www.tumblr.com/oauth/apps).

You'll need this to get your API key, even if you don't ever need to use a fully signed OAuth request.

For more details, see [Authentication](https://www.tumblr.com/docs/en/api/v2#authentication) below.

**Console**

Check out our [developer API console](https://api.tumblr.com/console) to test calls and see live examples of using our official clients.

**API Overview**

Before you dive in, review these essentials.

**URI Structure**

All Tumblr API requests start with https://api.tumblr.com.

The next segment of the URI path depends on the type of request you want to make. For example, getting blog data or to make a post on a blog uses the /v2/blog/{blog-identifier}/... endpoint, so the full URL would be https://api.tumblr.com/v2/blog/{blog-identifier}/...

See the different Methods sections below for complete details on the available routes.

**Request User Agent**

The User-Agent header is required and every application is expected to use a consistent value across requests. Your application may be suspended if it uses a variety of values for the User-Agent header.

**Request Content Types**

Unless otherwise noted, Tumblr API endpoints that accept body parameters expect the body parameters to be encoded using one of the following content types:

* application/json
* application/x-www-form-urlencoded
* multipart/form-data

Note that the Content-Type header must be set accordingly, and the absence of a Content-Type header may result in unexpected behavior.

**Post Identifiers**

Please note that Tumblr's Post IDs are 64-bit integers. Some languages, like Javascript, don't handle 64-bit numbers very well. With that in mind, there is also an id\_string field available so you can handle these safely.

**Blog Identifiers**

Each blog has a few different unique identifiers that can be used to reference that specific blog in the API payload. They are interchangeable and can be used in any place that {blog-identifier} appears in these API docs. The three types of blog identifiers are:

* **Blog name**: The name of the blog that appears in the URL. Examples:
  + staff for https://staff.tumblr.com/
  + changes for https://www.tumblr.com/changes or https://www.tumblr.com/@changes
* [**Hostname**](https://www.tumblr.com/docs/en/api/v2#blog-unique-hostnames): The hostname that's used to access the blog, excluding www.tumblr.com. Examples:
  + staff.tumblr.com for https://staff.tumblr.com/
  + changes.tumblr.com for https://www.tumblr.com/changes (*not* www.tumblr.com)
  + www.davidslog.com for https://www.davidslog.com/
* [**Universally unique identifier**](https://www.tumblr.com/docs/en/api/v2#blog-unique-identifiers): The UUID of the blog [retrieved from the API](https://www.tumblr.com/docs/en/api/v2#info---retrieve-blog-info). Examples:
  + t:0aY0xL2Fi1OFJg4YxpmegQ for staff, staff.tumblr.com, and even t:0aY0xL2Fi1OFJg4YxpmegQ
  + t:WX5xItRbCiJH\_GJDsIdhAQ for changes, changes.tumblr.com, and even t:WX5xItRbCiJH\_GJDsIdhAQ

**Blog Unique Hostnames**

Each blog has a unique hostname which can be used as its **identifier**. The hostname can be **standard** or **custom**.

* **Standard hostname**: the blog short name + .tumblr.com. Example: greentype.tumblr.com
* **Custom hostname**: these can be anything at all, as determined by a [DNS CNAME entry](https://help.tumblr.com/hc/en-us/articles/231256548-Custom-domains). Example: www.davidslog.com

**Blog Unique Identifiers**

Each blog also has a unique identifier that you can retrieve from any API response that includes a blog, in the uuid field (example: t:0aY0xL2Fi1OFJg4YxpmegQ). The {blog-identifier} placeholder can also be replaced by this unique identifier.

The benefits of using a unique identifier instead of a blog name or hostname are that the unique identifier will not change if the blog name or custom domain changes. It can be used as a stable, persistent identifier for a blog.

However, note that under exceptional circumstances, a unique identifier can change. Your own blog's unique identifier will only be changed with your knowing.

**Response Format**

Unless otherwise noted, the API returns JSON-encoded objects (Content-Type: application/json) and every response body includes these standard properties:

| **Field** | **Notes** |
| --- | --- |
| **meta** | The meta object matches the HTTP response message |
| **response** | Endpoint-specific results |

The meta object should always contain:

| **Field** | **Notes** |
| --- | --- |
| **status** | The 3-digit HTTP Status-Code (e.g., 200) |
| **msg** | The HTTP Reason-Phrase (e.g., OK) |

Even 4xx/5xx responses such as 400 Bad Request or 500 Internal Server Error should include this information, and the response may include more information about why you received the 4xx response.

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": { ... }

}

All requests made with HTTP GET are [JSONP](https://en.wikipedia.org/wiki/JSONP) enabled. To use JSONP, append jsonp= or callback= and the name of your callback function to the request. JSONP requests will always return an HTTP status code of 200 but will reflect the real status code in the meta field of the JSON response.

**Rate Limits**

Usage of the Tumblr API is rate limited in a few ways, but we respond with a 429 Limit Exceeded response whenever a consumer hits one of these limits. There is a global rate limit for all usage of the API per consumer, as well as a few per-feature rate limits, such as how many posts you can make per day. The error message or headers you receive with the 429 Limit Exceeded response should indicate what limit you've hit.

These rate limits include:

* 300 API calls per minute, per IP address.
* 18,000 API calls per hour, per IP address.
* 432,000 API calls per day, per IP address.
* 1,000 API calls per hour, per consumer key.
* 5,000 API calls per day, per consumer key.
* 250 new published posts (including reblogs) per day, per user.
* 250 images uploaded per day, per user.
* 200 follows per day, per user.
* 1,000 likes per day, per user.
* 10 new blogs per day, per user.
* 20 videos uploaded per day, per user.
* 60 minutes of total video uploaded per day, per user.

Note that these rate limits are based on the Tumblr servers' internal clocks and that the daily limits are per calendar day.

There are also overall limits to a few specific actions, including:

* A blog can only follow 5,000 other blogs at a time.
* A blog can only have 1,000 queued posts at a time.
* You can only filter up to 1,000 tags at a time.

**Reporting Issues**

If you run into problems with the Tumblr API, please use [our official support form](https://www.tumblr.com/support) to request help.

**About the API Documentation**

These docs include some working examples. Please click them to your heart's content, but don't use the embedded API key for your own nefarious purposes. [Register your application](https://www.tumblr.com/oauth/apps) to get your own API key (your OAuth Consumer Key).

**URI Conventions**

| **Notation** | **Meaning** | **Example** |
| --- | --- | --- |
| **Curly brackets { }** | Required item | api.tumblr.com/v2/blog/{blog-identifier}/posts (the [blog identifier](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) is required) |
| **Square brackets [ ]** | Optional item | api.tumblr.com/v2/blog/{blog-identifier}/posts[/type] (specifying the type is optional) |

**Official Tumblr API Client Libraries**

Here is a list of official API client which will help you get up and running with your favorite language in no time.

* [Javascript](https://github.com/tumblr/tumblr.js)
* [Ruby](https://github.com/tumblr/tumblr_client)
* [PHP](https://github.com/tumblr/tumblr.php)
* [Java](https://github.com/tumblr/jumblr)
* [Python](https://github.com/tumblr/pytumblr)
* [Objective-C](https://github.com/tumblr/TMTumblrSDK)
* [Go](https://github.com/tumblr/tumblrclient.go)

**Authentication**

The API uses three different levels of authentication, depending on the method.

* **None:** No authentication. Anybody can query the method.
* **API key:** Requires an API key. Use your OAuth Consumer Key as your api\_key.
  + Example: api\_key=PyezS3Q4Smivb24d9SzZGYSuh--IaMfAkE
  + Get an OAuth key: [register an application](https://www.tumblr.com/oauth/apps)
* **OAuth:** Requires a signed request that meets the [OAuth 1.0a Protocol](https://oauth.net/1/).

Each method description below includes a note about the authentication level.

**OAuth1 Authorization**

The API supports the [OAuth 1.0a Protocol](https://oauth.net/1/), accepting parameters via the Authorization header, with the HMAC-SHA1 signature method only.

**Temporary Credentials Endpoint**

This route is used to create a temporary oauth1 token to be used for oauth1 authorization. See <https://tools.ietf.org/html/rfc5849#section-2.1> for further information.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| https://www.tumblr.com/oauth/request\_token | POST | OAuth (client credentials) |

**Request Parameters**

None.

**Response**

Returns 200 OK or an error code. The response body parameters are url encoded (Content-Type: application/x-www-form-urlencoded), and the standard response properties are omitted.

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **oauth\_token** | String | The access token to use during the authorization process |
| **oauth\_token\_secret** | String | The access token secret to use during the authorization process |
| **oauth\_callback\_confirmed** | String | Indicates whether the protocol is OAuth 1.0a |

**Resource Owner Authorization Endpoint**

After retrieving temporary credentials, redirect the user to this endpoint so they can authorize your app. See <https://tools.ietf.org/html/rfc5849#section-2.2> for further information.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| https://www.tumblr.com/oauth/authorize | GET | None |

**Request Parameters**

Include these parameters in the query string when you redirect the user.

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **oauth\_token** | String | The temporary access token from the temporary credentials request | N/A | Yes |
| **source** | String | An optional identifier that indicates where the process started | N/A | No |

**Response**

Upon successful authorization, the user is redirected to your callback url with the following parameters in the query string:

* **oauth\_token** - The access token retrieved from the temporary credentials request
* **oauth\_verifier** - The token required to retrieve an access token
* **source** - The optional identifier that indicates where the process started

**Access Token Endpoint**

The endpoint that is used to exchange an oauth\_verifier token for an access token and secret for the user. See <https://tools.ietf.org/html/rfc5849#section-2.3> for further information.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| https://www.tumblr.com/oauth/access\_token | GET | OAuth (temporary credentials) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **oauth\_verifier** | String | The token from the authorize redirect query string | N/A | Yes |

**Response**

Returns 200 OK or an error code. The response body parameters are url encoded (Content-Type: application/x-www-form-urlencoded), and the standard response properties are omitted.

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **oauth\_token** | String | The user's access token |
| **oauth\_token\_secret** | String | The user's access token secret |

**OAuth2 Authorization**

The API supports the [OAuth 2.0 Protocol](https://oauth.net/2/), accepting a bearer token via the Authorization header like so:

Authorization: Bearer {access\_token}

API clients have access to [Authorization Code](https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.1), [Client Credentials](https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.4), and [Refresh Token](https://datatracker.ietf.org/doc/html/rfc6749#section-1.5) grants. Available scopes include basic, write, and offline\_access. Note that you must specify a valid OAuth2 redirect URL for OAuth2 to be available to your application. You can find this field by editing [one of your applications](https://www.tumblr.com/oauth/apps).

**The OAuth2 Authorization flow**

**Step One: Redirect a user to Tumblr**

Redirect the user to [Tumblr's OAuth2 authorization endpoint](https://www.tumblr.com/docs/en/api/v2#oauth2authorize---authorization-request). This will prompt the user to allow your application to access the Tumblr API on their behalf.

**Step Two: Handle the callback request**

Once a user decides to allow or disallow your application access to their account, Tumblr will redirect them to the OAuth2 redirect URL that you set when creating your application. If there was an error processing the request, the URL will contain an error query parameter. If the request was successful, the URL will contain code and state query parameters. You must verify the state value has the same value that you included when you redirected the user to Tumblr's OAuth2 authorization endpoint, and the code value can be used to retrieve an access token for the user.

**Step Three: Retrieve an access token**

Use your applications credentials and the code from the previous step to issue a request to [Tumblr's OAuth2 access token endpoint](https://www.tumblr.com/docs/en/api/v2#v2oauth2token---authorization-code-grant-request). For example, using the curl command line tool:

curl -F grant\_type=authorization\_code -F code={code} -F client\_id={Your OAuth consumer key} -F client\_secret={Your OAuth secret key} https://api.tumblr.com/v2/oauth2/token

If you provided a redirect URI in the initial authorization request, you must provide the same redirect URI in the token request (as redirect\_uri).

A successful request will yield a response that contains the following object:

{

"access\_token": "{access\_token}",

"expires\_in": 2520,

"id\_token": false,

"refresh\_token": "{refresh\_token}",

"scope": "write offline\_access",

"token\_type": "bearer"

}

*Special tip:*[*jq*](https://stedolan.github.io/jq/)*is a handy tool for formatting JSON on the command line, you can pipe the output of curl through jq to easily format the API response, like so:*

curl -F ... https://api.tumblr.com/v2/oauth2/token | jq

You'll need to store this information somewhere so you can issue requests on behalf of the user, and so you can refresh the access token when it has expired.

**Step Four: Make your API requests**

Using the access\_token, you can now talk directly to the Tumblr API. For example, this curl request will retrieve the [user's account information](https://www.tumblr.com/docs/en/api/v2#userinfo--get-a-users-information):

curl -H 'Authorization: Bearer {access\_token}' https://api.tumblr.com/v2/user/info

**Step Five: Refresh an access token**

Your application can use the expires\_in property in the access token response to determine if the user's access token has expired. If it has, you can use [the refresh token to retrieve a new access token and refresh token](https://www.tumblr.com/docs/en/api/v2#v2oauth2token---refresh-token-grant-request). For example, using curl again:

curl -F grant\_type=refresh\_token -F refresh\_token={refresh\_token} -F client\_id={Your OAuth consumer key} -F client\_secret={Your OAuth consumer secret} https://api.tumblr.com/v2/oauth2/token

**/oauth2/authorize - Authorization Request**

This request is used to request permission from a user to gain access to their account. Once a user grants your application access, they will be returned to your OAuth2 redirect Url as defined on your [OAuth Application](https://www.tumblr.com/oauth/apps).

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| www.tumblr.com/oauth2/authorize | GET | N/A |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **client\_id** | String | Your OAuth consumer key | n/a | Yes |
| **response\_type** | String | Always code | n/a | Yes |
| **scope** | String | A space delimited list of scopes | n/a | Yes |
| **state** | String | A unique value used to maintain state | n/a | Yes |
| **redirect\_uri** | String | A redirect URI | n/a | Maybe, see note below |
| **force\_login** | Boolean | An optional param to forcibly log out the current user and redirect them to login flow | n/a | No |

A redirect URI must be supplied when multiple redirect URIs are registered.

**/v2/oauth2/token - Authorization Code Grant Request**

This request allows you to exchange an authorization code for an access token.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/oauth2/token | POST | N/A |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **grant\_type** | String | Always authorization\_code | n/a | Yes |
| **code** | String | The code recieved at your OAuth2 redirect Url | n/a | Yes |
| **client\_id** | String | Your OAuth consumer key | n/a | Yes |
| **client\_secret** | String | Your OAuth consumer secret | n/a | Yes |
| **redirect\_uri** | String | A redirect URI (if it was included in the authorization request) | n/a | No |

**Response**

* 200 Operation was successful
* 401 Unauthorized

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| access\_token | String | The OAuth2 access token |
| expires\_in | Int | The access token TTL in seconds |
| token\_type | String | The type of the access token |
| scope | String | The OAuth2 access token scopes |
| refresh\_token | String | An OAuth2 refresh token (if offline\_access scope was requested) |

**/v2/oauth2/token - Refresh Token Grant Request**

This request is used to exchange an OAuth2 refresh token for a new OAuth2 access token and refresh token.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/oauth2/token | POST | OAuth2 |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **grant\_type** | String | Always a string refresh\_token | n/a | Yes |
| **client\_id** | String | Your OAuth consumer key | n/a | Yes |
| **client\_secret** | String | Your OAuth consumer secret | n/a | Yes |
| **refresh\_token** | String | An OAuth2 refresh token | n/a | Yes |

**Example:**

{

"grant\_type": "refresh\_token",

"client\_id": "...",

"refresh\_token": "{refresh\_token goes here}"

}

**Response**

* 200 Operation was successful
* 401 Unauthorized, refresh token has expired

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| access\_token | String | The OAuth2 access token |
| expires\_in | Int | The access token TTL in seconds |
| token\_type | String | The type of the access token |
| scope | String | The OAuth2 access token scopes |
| refresh\_token | String | An OAuth2 refresh token |

**/v2/oauth2/exchange - OAuth1 to OAuth2 Token Exchange**

This route is used to exchange an OAuth1 access token for an OAuth2 access token and refresh token. Clients must expect the OAuth1 token to be invalidated if the exchange is successful.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/oauth2/exchange | POST | OAuth1 |

**Request Parameters**

None.

**Response**

* 200 Operation was successful
* 400 Incorrect authorization type (requests must use OAuth1)
* 401 Authentication failed

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| access\_token | String | The OAuth2 access token |
| expires\_in | Int | The access token TTL in seconds |
| token\_type | String | The type of the access token |
| scope | String | The OAuth2 access token scopes (always basic write) |
| refresh\_token | String | An OAuth2 refresh token |

**Common Response Elements**

Beyond the [overall response format described above](https://www.tumblr.com/docs/en/api/v2#response-format), there are some elements that are common throughout the API responses.

**Links**

Links describe an in-app or navigation behavior. A response may contain one or more of these contained under the \_links response object key.

An object in the \_links hash will always contain type and href keys.

A \_links object may be of type navigation or action.

* navigation: A reference to some external URI to which the user can go.
* action: A reference to some internal client state change in the application. The most common example would be to open a different view.

A navigation link will only contain a href to which the user should be directed. An action type link will contain details on what request method as well as what query or body parameters should be used to take the action.

**Example:**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"\_links": {

"previous": {

"type": "action",

"method": "GET",

"href": "https://api.tumblr.com/v2/endpoint",

"query\_params": {

"page": 1,

}

},

"next": {

"type": "action",

"method": "GET",

"href": "https://api.tumblr.com/v2/endpoint",

"query\_params": {

"page": 3,

}

},

"terms\_of\_service": {

"type": "navigation",

"href": "https://www.tumblr.com/policy/terms-of-service",

},

...

}

}

}

**Tag Objects**

A tag object represents a tag, and could have these fields:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **tag** | String | The tag name |
| **thumb\_url** | String | Returns an image (75x75) associated with the tag; may be null |
| **is\_tracked** | Boolean | Indicates whether the requesting user is tracking this tag |
| **featured** | Boolean | Indicates whether the tag is a featured tag |

**Neue Post Format objects**

Some routes will return Posts that have type: blocks and/or is\_blocks\_post\_format: true, which means their content is available in the Neue Post Format. [See the NPF specification docs for more info!](https://www.tumblr.com/docs/npf)

Also, when making any request to our API that returns Posts, you may supply a npf=true query parameter to specify that you'd like all of the Posts' content returned in the Neue Post Format rather than the legacy format.

**Blog Methods**

**/info - Retrieve Blog Info**

This method returns general information about the blog, such as the title, number of posts, and other high-level data.

**Examples**

* https://api.tumblr.com/v2/blog/scipsy.tumblr.com/info
* https://api.tumblr.com/v2/blog/good.tumblr.com/info

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/info?api\_key={key} | GET | [API Key](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **api\_key** | String | Your OAuth Consumer Key. See [Authentication](https://www.tumblr.com/docs/en/api/v2#authentication) for more details. | N/A | Yes |
| **fields** | Object | See next section: "Partial Responses". | N/A | No |

**Partial Responses**

Note that blog objects via the Tumblr API support "partial responses", which allows the request to specify which fields are returned any time a blog object is present by using the fields query parameter.

For example, if you would like to always only see the name and updated fields in blog objects:

https://api.tumblr.com/v2/blog/david/info?fields[blogs]=name,updated

That will return only the name and updated fields in the blog object. You can use this in *any* API request that includes blog objects, not just this blog info endpoint. Some fields are also "optional", and are preceded in the list by a ? (encoded to %3F).

Those optional fields are *only available upon request*, such as:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **is\_following\_you** | Boolean | Whether or not the blog is following your primary blog |
| **duration\_blog\_following\_you** | Integer | How long (in seconds) that blog has been following your primary blog (-1 if they aren't) |
| **duration\_following\_blog** | Integer | How long (in seconds) you've been following that blog (-1 if you aren't) |
| **timezone** | String | The blog's configured timezone, such as "US/Eastern". Only viewable by blog member. |
| **timezone\_offset** | String | The blog's configured timezone as a GMT offset such as "GMT+0800". Only viewable by blog member. |

To access these, specify them in the fields[blogs] query parameter list, like so:

https://api.tumblr.com/v2/blog/david/info?fields[blogs]=name,updated,%3Fis\_following\_you,%3Fduration\_following\_blog

Which will return a response like:

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": {

"name": "david",

"updated": 1591035760,

"is\_following\_you": false,

"duration\_following\_blog": 123456

}

}

}

Omitting the fields query parameter will return the default set of fields below.

**Response**

*Note: Field ordering below may differ from actual response.* Also, any fields present that aren't included here are not officially supported for third-party use.

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **title** | String | The display title of the blog |
| **posts** | Number | The total number of posts to this blog |
| **name** | String | The short blog name that appears before tumblr.com in a standard blog hostname |
| **updated** | Number | The time of the most recent post, in seconds since the epoch |
| **description** | String | You guessed it! The blog's description |
| **ask** | Boolean | Indicates whether the blog allows questions |
| **ask\_anon** | Boolean | Indicates whether the blog allows anonymous questions; returned only if ask is true |
| **followed** | Boolean | Whether you're following the blog, returned only if this request has an authenticated user |
| **likes** | Number | Number of likes for this user, returned only if this is the user's primary blog and sharing of likes is enabled |
| **is\_blocked\_from\_primary** | Boolean | Indicates whether this blog has been blocked by the calling user's primary blog; returned only if there is an authenticated user making this call |
| **avatar** | Array | An array of avatar objects, each a different size, which should each have a width, height, and URL. |
| **url** | String | The blog's url |
| **theme** | Object | The blog's general theme options, which may not be useful if the blog uses a custom theme. See next table. |

Specific fields inside of the theme object and what they mean:

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| avatar\_shape | String | "circle" or "square", this is the shape of the mask over the user's avatar. |
| background\_color | String | The intended hex color used for the blog's background color. |
| body\_font | String | The font that the blog has selected as their "body" font. |
| header\_bounds | Mixed | If the blog's header should be cropped, this is a comma-separated list of top/right/bottom/left coordinates to use. |
| header\_image | String | The URL of the blog's original, full header image. Note that this may be a default Tumblr header image. |
| header\_image\_npf | Object | [NPF image block format](https://www.tumblr.com/docs/npf#content-block-type-image) of the blog's header image. It will include one or more image sizes to choose from, in order from largest to smallest. |
| header\_image\_focused | String | If the blog cropped/repositioned their header image, this will be that version, which should be preferred over the original. |
| header\_image\_poster | String | The URL of a single-frame "poster" version of the blog's header image, if it's an animated image. Note that this may be an empty string if no poster could be made or is not needed. |
| header\_image\_scaled | String | If the blog *only* scaled their header image, this will be that scaled version. Note that this may be a default Tumblr header image in the case that they scaled *and* repositioned it, in which case, use the \_focused version. |
| header\_stretch | Boolean | Whether or not the blog's header is meant to be stretched to aspect-fill any given space where it's used. |
| link\_color | String | The intended hex color of any links in the blog's description. |
| show\_avatar | Boolean | Whether or not the blog's avatar should be displayed, even if it's given in the API payload. |
| show\_description | Boolean | Whether or not the blog's description should be displayed, even if it's given in the API payload. |
| show\_header\_image | Boolean | Whether or not the blog's header image should be displayed, even if it's given in the API payload. |
| show\_title | Boolean | Whether or not the blog's title should be displayed, even if it's given in the API payload. |
| title\_color | String | The intended hex color of the blog's title. |
| title\_font | String | The intended font to use when displaying the blog's title. |
| title\_font\_weight | String | The intended font weight to use when displaying the blog's title. |

**Example**

[https://api.tumblr.com/v2/blog/david.tumblr.com/info?api\_key={key}](https://api.tumblr.com/v2/blog/david.tumblr.com/info?api_key=fuiKNFp9vQFvjLNvx4sUwti4Yb5yGutBN4Xh10LXZhhRKjWlV4)

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": {

"title": "David's Log",

"posts": 3456,

"name": "david",

"url": "https:\/\/david.tumblr.com\/",

"updated": 1308953007,

"description": "<p><strong>Mr. Karp<\/strong> is tall and skinny, with

unflinching blue eyes a mop of brown hair.\r\n

He speaks incredibly fast and in complete paragraphs.</p>",

"ask": true,

"ask\_anon": false,

"likes": 12345

}

}

}

**/avatar — Retrieve a Blog Avatar**

You can get a blog's avatar in 9 different sizes. The default size is 64x64.

**Examples**

* https://api.tumblr.com/v2/blog/david.tumblr.com/avatar/512
* https://api.tumblr.com/v2/blog/david.tumblr.com/avatar

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/avatar[/size] | GET | None |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |
| **size** | Number | The size of the avatar (square, one value for both length and width). Must be one of the values: 16, 24, 30, 40, 48, 64, 96, 128, 512 | 64 | No |

**Response**

Requests that are *not* signed using OAuth1 will receive the requested avatar in PNG format, while requests that are signed will receive a response of the following form:

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **avatar\_url** | String | The URL of the avatar image. This is also returned in the [Location HTTP header field](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.30) (see note). | An [HTTP Location header](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.30) field is returned — the response points to the avatar image. That means you can embed this method in an img tag in HTML. |

**/blocks – Retrieve Blog's Blocks**

Get the blogs that the requested blog is currently blocking. The requesting user must be an admin of the blog to retrieve this list.

Note that this endpoint is rate limited to 60 requests per minute.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/blocks | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **offset** | Number | Block number to start at | 0 | No |
| **limit** | Number | The number of blocks to retrieve, 1-20, inclusive | 20 | No |

**Response**

Returns 200 OK or an error code.

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **blocked\_tumblelogs** | Array | Blog objects that are blocked. |  |
| **\_links** | Object | A standard \_links object with pagination objects. |  |

**Example response:**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blocked\_tumblelogs": [

{

"name": "joe",

"url": "http:\/\/www.spammyjoe.com",

"updated": 1308797076,

"title": "Spammy Joe",

"description": "Posting things you don't really care for"

},

{

"name": "harry",

"url": "harryblog19.tumblr.com",

"updated": 1808841333,

"title": "The 19th Harry Blog",

"description": ""

},

...

],

"\_links": {

"next": {

"href": "/v2/blog/andysblog.tumblr.com/blocks?offset=20",

"method": "GET",

"query\_params": {

"offset": 20

}

}

}

}

}

**/blocks – Block a Blog**

Note that this endpoint is rate limited to 60 requests per minute.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/blocks | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blocked\_tumblelog** | String | The tumblelog to block, specified by any blog identifier | N/A | Only if post\_id isn't passed |
| **post\_id** | String | The anonymous post ID (asks, submissions) to block | N/A | Only if blocked\_tumblelog isn't passed |

**Response**

Returns 201 Created or an error code.

**/blocks/bulk – Block a list of Blogs**

Note that this endpoint is rate limited to 60 requests per minute.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/blocks/bulk | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blocked\_tumblelogs** | String | Comma-separated list of tumblelogs to block, specified by any blog identifier | N/A | Yes |
| **force** | Bool | Whether to force the block to go through even if it requires canceling a Post+ Subscription | false | No |

**Response**

Returns 200 OK or an error code.

**/blocks – Remove a Block**

Note that this endpoint is rate limited to 60 requests per minute.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/blocks | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blocked\_tumblelog** | String | The tumblelog whose block to remove, specified by any blog identifier | N/A | Only if anonymous\_only isn't passed |
| **anonymous\_only** | Boolean | When passed without the blocked\_tumblelog parameter, this will clear all anonymous IP blocks | N/A | Only if blocked\_tumblelog isn't passed |

* If you pass both params, anonymous\_only is going to be ignored.
* Passing just anonymous\_only=false is the equivalent of no params. A 404 response will be returned.

**Response**

Returns 200 OK if the block on the passed blocked\_tumblelog parameter was successfully removed.

Returns 200 OK if all anonymous blocks were successfully removed when anonymous\_only=true was passed without a blocked\_tumblelog parameter in the path.

**/likes — Retrieve Blog's Likes**

This method can be used to retrieve the publicly exposed likes from a blog.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/likes?api\_key={key} | GET | [API Key](https://www.tumblr.com/docs/en/api/v2#auth) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Liked post number to start at | 0 (first post) | No |
| **before** | Integer | Retrieve posts liked before the specified timestamp | None | No |
| **after** | Integer | Retrieve posts liked after the specified timestamp | None | No |

**Notes**

* You can only provide either before, after, or offset. If you provide more than one of these options together you will get an error.
* You can still use limit with any of those three options to limit your result set.
* When using the offset parameter the maximum limit on the offset is 1000. If you would like to get more results than that use either before or after.

**Response**

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **liked\_posts** | Array | An array of post objects (posts liked by the user) |  |
| **liked\_count** | Number | Total number of liked posts |  |

**/following — Retrieve Blog's following**

This method can be used to retrieve the publicly exposed list of blogs that a blog follows, in order from most recently-followed to first.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/following | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Followed blog index to start at | 0 (most-recently followed) | No |

**Response**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **blogs** | Array | An array of short blog infos that this blog follows, in order from most recently-followed to first. |
| **total\_blogs** | Number | Total number of followed blogs |

**/followers — Retrieve a Blog's Followers**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/followers | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Result to start at | 0 (first follower) | No |

**Response**

These fields are wrapped in a users object:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **total\_users** | Number | The number of users currently following the blog |
| **users** | Array | Each item is a follower, containing these fields: |
| **name** | String | The user's name on tumblr |
| **following** | Boolean | Whether the caller is following the user |
| **url** | String | The URL of the user's primary blog |
| **updated** | Number | The time of the user's most recent post, in seconds since the epoch |

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"total\_users": 2684,

"users": [

{

"name": "david",

"following": true,

"url": "https:\/\/www.davidslog.com",

"updated": 1308781073

},

{

"name": "ben",

"following": true,

"url": "http:\/\/bengold.tv",

"updated": 1308841333

},

...

]

}

}

**/followed\_by — Check If Followed By Blog**

This method can be used to check if one of your blogs is followed by another blog.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/followed\_by | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Required?** |
| --- | --- | --- | --- |
| **query** | string | The name of the blog that may be following your blog | Yes |

Example usage:

GET https://api.tumblr.com/v2/blog/YOUR-BLOG.tumblr.com/followed\_by?query=staff

**Response**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **followed\_by** | Boolean | True when the queried blog follows your blog, false otherwise. |

**/posts – Retrieve Published Posts**

**Examples**

* https://api.tumblr.com/v2/blog/peacecorps.tumblr.com/posts/text?notes\_info=true
* https://api.tumblr.com/v2/blog/pitchersandpoets.tumblr.com/posts/photo?tag=new+york+yankees
* https://api.tumblr.com/v2/blog/staff.tumblr.com/posts?before=1496289599

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts[/type]?api\_key={key}&[optional-params=] | GET | [API Key](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **api\_key** | String | Your OAuth Consumer Key See Authentication for more details. | N/A | Yes |
| **type** | String | The type of post to return. Specify one of the following: text, quote, link, answer, video, audio, photo, chat | None – return all types | No |
| **id** | Number | A specific post ID. Returns the single post specified or (if not found) a 404 error. | None | No |
| **tag** | String or Array | Limits the response to posts with the specified tag(s), see note below | None | No |
| **limit** | Number | The number of posts to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Post number to start at | 0 (first post) | No |
| **reblog\_info** | Boolean | Indicates whether to return reblog information (specify true or false). Returns the various reblogged\_ fields. | False | No |
| **notes\_info** | Boolean | Indicates whether to return notes information (specify true or false). Returns note count and note metadata. | False | No |
| **filter** | String | Specifies the post format to return, other than HTML: text – Plain text, no HTML; raw – As entered by the user (no post-processing); if the user writes in Markdown, the Markdown will be returned rather than HTML | None (HTML) | No |
| **before** | Number | Returns posts published before a specified Unix timestamp, in seconds. | False | No |
| **after** | Number | Returns posts published after a specified Unix timestamp, in seconds. | False | No |
| **sort** | String | A specific sort order, "desc" for descending, "asc" for ascending. | "desc" | No |
| **npf** | Boolean | Returns posts' content in [NPF format](https://www.tumblr.com/docs/npf) instead of the legacy format. | False | No |

By default, the list of posts on a blog is in reverse-chronological order, using the user-set publish time which allows backdating. Note that you cannot use before and offset in the same request; before is best used for paginating through posts. Also, using after is best used with sort set to asc.

Note that tag can either be a string or an array of strings:

Fetch posts using this one tag:

/v2/blog/{blog-identifier}/posts?tag=something

or fetch posts using all of these tags:

/v2/blog/{blog-identifier}/posts?tag[0]=something&tag[1]=example

When given an array of tags, posts will be returned that have *all* of the provided tags (so in that example, posts with both "something" and "example" used on them). Also, a maximum of four tags can be provided for this use.

**Response**

Each response includes a blog object that is the equivalent of an /info [response](https://www.tumblr.com/docs/en/api/v2#blog-info). Posts are returned as an array attached to the posts field.

**Important note:** Post content can be in two formats: legacy or Neue Post Format (NPF). By default, posts returned from this endpoint (and any other endpoint that returns posts) will be in the legacy post-type-based content formats described below. NPF-created posts from the official Tumblr mobile apps will be returned as text/regular posts to maintain backwards compatibility. Note that this "backwards compatibility" from NPF to HTML is *best effort*. To help transition to an NPF-only world, you can pass along the npf=true query parameter to force *all posts* returned here to be in Neue Post Format (also described below). Eventually, all posts will be NPF, so we strongly encourage leveraging NPF JSON via npf=true for post content when consuming posts via the API.

**Fields available for all Post types:**

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **blog\_name** | String | The short name used to uniquely identify a blog |  |
| **id** | Number | The post's unique ID |  |
| **id\_string** | String | The post's unique ID as a String | For clients that don't support 64-bit integers |
| **genesis\_post\_id** | String | The post's unique "genesis" ID† as a String | Only available to the post owner in certain circumstances |
| **post\_url** | String | The location of the post |  |
| **parent\_post\_url** | String | The URL of the parent post, if this is a reblog | Only available if the post is a reblog |
| **type** | String | The type of post | See the type request parameter |
| **timestamp** | Number | The time of the post, in seconds since the epoch |  |
| **date** | String | The GMT date and time of the post, as a string |  |
| **format** | String | The post format: html or markdown |  |
| **reblog\_key** | String | The key used to reblog this post | See the /post/reblog method |
| **tags** | Array of strings | Tags applied to the post |  |
| **bookmarklet** | Boolean | Indicates whether the post was created via the Tumblr bookmarklet | Exists only if true |
| **mobile** | Boolean | Indicates whether the post was created via mobile/email publishing | Exists only if true |
| **source\_url** | String | The URL for the source of the content (for quotes, reblogs, etc.) | Exists only if there's a content source |
| **source\_title** | String | The title of the source site | Exists only if there's a content source |
| **liked** | Boolean | Indicates if a user has already liked a post or not | Exists only if the request is fully authenticated with OAuth. |
| **state** | String | Indicates the current state of the post | States are published, queued, draft and private |
| **is\_blocks\_post\_format** | Boolean | Indicates whether the post is stored in the [Neue Post Format](https://www.tumblr.com/docs/en/api/v2#neue-post-format-objects) |  |
| **muted** | Boolean | Indicates whether push notifications and activity items are muted for this post by its author. | Only available to the post owner in certain circumstances |
| **mute\_end\_timestamp** | Number | See note below. | Only available to the post owner in certain circumstances |
| **total\_posts** | Number | The total number of post available for this request, useful for paginating through results |  |

† The "genesis" ID for a post is only available and different than its current ID if that post had been drafted, queued, or scheduled, and is now published. In which case, the "genesis" ID will be the original post ID generated when drafting, queuing, or scheduling that post. You cannot use this ID to look up the post after it has been published, but it can be useful for tracking a post from its pre- to post-published state.

If muted: true and mute\_end\_timestamp: 0, then the post is muted forever. Otherwise, mute\_end\_timestamp is a unix timestamp of when the mute will end. If muted: false, then mute\_end\_timestamp will be 0, but doesn't matter since the post is not muted.

**Fields available for posts from a community:**

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **community** | Object | Info about the community this post is from | See the [community info endpoint](https://www.tumblr.com/docs/en/api/v2) |
| **can\_react** | Boolean | Whether the current user can react to this post |  |
| **can\_moderate** | Boolean | Whether the current user can moderate this post | Only visible for moderators, otherwise omitted |
| **reactions** | Array | The reactions currently on this post | See the reaction object below |
| **is\_admin** | Boolean | Whether the post author is an admin of the community |  |
| **is\_moderator** | Boolean | Whether the post author is a moderator of the community |  |
| **post\_author** | String | The blog name who authored the post inside the community |  |
| **author\_blog** | Object | The blog info object for the post's author |  |

Reactions have a specific object for each reaction in the array:

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **emoji** | Object | A description of the emoji reaction, with fields for our unique id, the emoji's grapheme, variants, and base\_slug used to group them |
| **count** | Number | How many of this reaction have been used |
| **preview** | Array | A preview of the blog names of who have used this reaction |
| **reacted** | Boolean | Whether the current user has reacted to this post |

**Neue Post Format (NPF) Posts**

Posts that have type: blocks and/or is\_blocks\_post\_format: true will have three fields for the post's content:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **content** | Array | The content of the post. |
| **layout** | Array | The layout of the post content. |
| **trail** | Array | The reblog trail items, if any. |

The specification for what objects you can find in these fields is [documented here](https://www.tumblr.com/docs/npf).

**Legacy Text Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **title** | String | The optional title of the post |
| **body** | String | The full post body |

**Example**

https://api.tumblr.com/v2/blog/citriccomics.tumblr.com/posts/text?api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "citriccomics",

"id": 3507845453,

"id\_string": "3507845453",

"post\_url": "https:\/\/citriccomics.tumblr.com\/post\/3507845453",

"type": "text",

"date": "2011-02-25 20:27:00 GMT",

"timestamp": 1298665620,

"state": "published",

"format": "html",

"reblog\_key": "b0baQtsl",

"tags": [

"tumblrize",

"milky dog",

"mini comic"

],

"note\_count": 14,

"title": "Milky Dog",

"body": "<p><img src=\"https:\/\/media.tumblr.com\

/tumblr\_lh6x8d7LBB1qa6gy3.jpg\"\/><a href=\"http:\/\

/citriccomics.com\/blog\/?p=487\" target=\"\_blank\">TO READ

THE REST CLICK HERE<\/a><br\/>\n\nMilky Dog was inspired by

something <a href=\"http:\/\/gunadie.com\/naomi\"

target=\"\_blank\">Naomi Gee<\/a> wrote on twitter, I really

liked the hash tag <a href=\"https:\/\/twitter.com\/

search?q=%23MILKYDOG\" target=\"\_blank\">#milkydog<\/a>

and quickly came up with a little comic about it. You can

(and should) follow Naomi on twitter <a href=\"https:\/\

/twitter.com\/ngun\" target=\"\_blank\">@ngun<\/a> I'm on

twitter as well <a href=\"https:\/\/twitter.com\

/weflewairplanes\"target=\"\_blank\">@weflewairplanes<\/a>

<\/p>\n\nAlso, if youâ€™re a Reddit user (or even if

you're not) I submitted this there, if you could up vote

it I'd be super grateful just <a href=\"http:\/\

/tinyurl.com\/5wj3tqz\" target=\"\_blank\">CLICK HERE<\/a>"

},

...

],

"total\_posts": 3

}

}

**Legacy Photo Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **caption** | String | The user-supplied caption |
| **width** | Number | The width of the photo or photoset |
| **height** | Number | The height of the photo or photoset |
| **photos** | Array | Photo objects with properties: |
| **caption** | String | user supplied caption for the individual photo (Photosets only) |
| **alt\_sizes** | Array | alternate photo sizes, each with: |
| **width** | Number | width of the photo, in pixels |
| **height** | Number | height of the photo, in pixels |
| **url** | String | Location of the photo file (either a JPG, GIF, or PNG) |

Multi-photo Photo posts, called **Photosets**, will send return multiple photo objects in the photos array.

**Example**

https://api.tumblr.com/v2/blog/derekg.org/posts/photo?id=7431599279&api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "derekg",

"id": 7431599279,

"id\_string": "7431599279",

"post\_url": "https:\/\/derekg.org\/post\/7431599279",

"type": "photo",

"date": "2011-07-09 22:09:47 GMT",

"timestamp": 1310249387,

"format": "html",

"reblog\_key": "749amggU",

"tags": [],

"note\_count": 18,

"caption": "<p>my arm is getting tired.<\/p>",

"photos": [

{

"caption": "",

"alt\_sizes": [

{

"width": 1280,

"height": 722,

"url": "https:\/\/derekg.org\/photo\/1280\/7431599279\/1\/

tumblr\_lo36wbWqqq1qanqww"

},

{

"width": 500,

"height": 282,

"url": "https:\/\/30.media.tumblr.com\/

tumblr\_lo36wbWqqq1qanqwwo1\_500.jpg"

},

{

"width": 400,

"height": 225,

"url": "https:\/\/29.media.tumblr.com\/

tumblr\_lo36wbWqqq1qanqwwo1\_400.jpg"

},

{

"width": 250,

"height": 141,

"url": "https:\/\/26.media.tumblr.com\/

tumblr\_lo36wbWqqq1qanqwwo1\_250.jpg"

},

{

"width": 100,

"height": 56,

"url": "https:\/\/24.media.tumblr.com\/

tumblr\_lo36wbWqqq1qanqwwo1\_100.jpg"

},

{

"width": 75,

"height": 75,

"url": "https:\/\/30.media.tumblr.com\/

tumblr\_lo36wbWqqq1qanqwwo1\_75sq.jpg"

}

]

}

]

}

],

"total\_posts": 1

}

}

**Legacy Quote Posts**

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **text** | String | The text of the quote (can be modified by the user when posting) |  |
| **source** | String | Full HTML for the source of the quote Example: <a href="...">Steve Jobs</a> | See also the table of common response fields |

**Example**

https://api.tumblr.com/v2/blog/museumsandstuff.tumblr.com/posts/quote?api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "museumsandstuff",

"id": 4742980381,

"id\_string": "4742980381",

"post\_url": "https:\/\/museumsandstuff.tumblr.com\/post\/4742980381",

"type": "quote",

"date": "2011-04-19 08:52:34 GMT",

"timestamp": 1303203154,

"format": "html",

"reblog\_key": "KLA85e6c",

"tags": [],

"note\_count": 23,

"source\_url": "http:\/\/museumtwo.blogspot.com\/2011\/04\/

guest-post-convivial-museum-photo-essay.html",

"source\_title": "museumtwo.blogspot.com",

"text": "Why do visitors still report discomfort, confusion,

elitism, exclusion?",

"source": "<a href=\"http:\/\/museumtwo.blogspot.com\/2011\/04\/

guest-post-convivial-museum-photo-essay.html\"

target=\"\_blank\">Museum 2.0: Guest Post: The Convivial

Museum Photo Essay<\/a> (via <a href=\"http:\/\/

www.joshrobinson.org\/\"target=\"\_blank\">joshrobinsonblog

<\/a>)"

},

...

],

"total\_posts": 9

}

}

**Legacy Link Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **title** | String | The title of the page the link points to |
| **description** | String | A user-supplied description |
| **url** | String | The link! |
| **link\_author** | String | The author of the article the link points to |
| **excerpt** | String | An excerpt from the article the link points to |
| **publisher** | String | The publisher of the article the link points to |
| **photos** | Array | Photo objects with properties: |
| **caption** | String | The thumbnail caption |
| **original\_size** | Object | The photo at its original size |
| **width** | Number | Width of the image, in pixels |
| **height** | Number | Height of the image, in pixels |
| **url** | String | Location of the image file (either a JPG, GIF, or PNG) |
| **alt\_sizes** | Array | Alternate photo sizes, each with the same properties as above. |

**Example**

https://api.tumblr.com/v2/blog/travellingcameraclub.com/posts/link?api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "travellingcameraclub",

"id": 688472164,

"id\_string": "688472164",

"post\_url": "https:\/\/travellingcameraclub.com\/post\/688472164",

"type": "link",

"date": "2010-06-11 23:17:08 GMT",

"timestamp": 1276298228,

"format": "html",

"reblog\_key": "RWhJJWia",

"tags": [],

"note\_count": 9,

"title": "Esther Aarts Illustration | News and Blog: Faq: How do

you make those marker doodles?",

"url": "https:\/\/blog.estadiezijn.nl\/post\/

459075181\/faq-how-do-you-make-those-marker-doodles",

"link\_author": "Ester Aarts",

"excerpt": "How I make marker doodles",

"publisher": "blog.estadiezijn.nl",

"photos": [

{

"caption": "",

"original\_size":

{

"width": 500,

"height": 500,

"url": "https:\/\/40.media.tumblr.com

\/1393850e5c331da2e3c9fb478a30310d

\/tumblr\_inline\_nm3lwntw8k1rplry2\_500.jpg"

},

"alt\_sizes": []

}

],

"description": "<blockquote>\n<p>On my Ipad, of course!<\/p>

<p>Nothing better than the latest technology to get the job done.

Look at all my apps!<\/p> <p><img height=\"555\" width=\"500\"

src=\"http:\/\/farm5.static.flickr.com\/4006\/

4445161463\_31da0327c2\_o.jpg\" alt=\"my iPad\"\/><\/p>

<p>My favourite markers are an Edding 400, a Sharpie and a

Copic Ciao. The white gouache is from Dr Martins and does a

decent job covering up whatever needs to be covered up, and

flows.<\/p><\/blockquote>"

}

...

],

"total\_posts": 7

}

}

**Legacy Chat Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **title** | String | The optional title of the post |
| **body** | String | The full chat body |
| **dialogue** | Array | Array of objects with the following properties: |
| **name** | String | Name of the speaker |
| **label** | String | Label of the speaker |
| **phrase** | String | Text. |

**Example**

https://api.tumblr.com/v2/blog/www.davidslog.com/posts/chat?api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "david",

"id": 5845345725,

"id\_string": "5845345725",

"post\_url": "https:\/\/www.davidslog.com\/5845345725\/if-youre-okay-with-it-destroying-tumblr",

"type": "chat",

"date": "2011-05-25 22:32:00 GMT",

"timestamp": 1306362720,

"format": "html",

"reblog\_key": "wlxAsElf",

"tags": [],

"note\_count": 114,

"title": null,

"body": "David: https://staff.tumblr.com/post/5845153911

[draft] Topherchris: are you freaking serious\r\n

David: If you're okay with it - I'd love to :)\r\n

Topherchris: i'm okay with it, if you're okay with

it destroying tumblr",

"dialogue": [

{

"label": "David:",

"name": "David",

"phrase": "https://staff.tumblr.com/post/5845153911 [draft]\r",

},

{

"label": "Topherchris:",

"name": "Topherchris",

"phrase": "are you freaking serious\r"

},

{

"label": "David:",

"name": "David",

"phrase": "If you're okay with it - I'd love to :)"

},

{

"label": "Topherchris:",

"name": "Topherchris",

"phrase": "i'm okay with it, if you're okay with it

destroying tumblr",

}

]

}

...

],

"total\_posts": 7

}

}

**Legacy Audio Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **caption** | String | The user-supplied caption |
| **player** | String | HTML for embedding the audio player |
| **plays** | Number | Number of times the audio post has been played |
| **album\_art** | String | Location of the audio file's ID3 album art image |
| **artist** | String | The audio file's ID3 artist value |
| **album** | String | The audio file's ID3 album value |
| **track\_name** | String | The audio file's ID3 title value |
| **track\_number** | Number | The audio file's ID3 track value |
| **year** | Number | The audio file's ID3 year value |

**Example**

https://api.tumblr.com/v2/blog/derekg.org/posts?id=5578378101&api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "derekg",

"id": 5578378101,

"id\_string": "5578378101",

"post\_url": "http:\/\/derekg.org\/post\/5578378101/otis-redding",

"type": "audio",

"date": "2011-05-17 16:21:05 GMT",

"timestamp": 1305649265,

"format": "html",

"reblog\_key": "Wa65rR5l",

"tags": [],

"note\_count": 3,

"source\_url": "http:\/\/soundcloud.com\/mariam-cabal\/otis-redding",

"source\_title": "SoundCloud \/ Mariam Caballero",

"id3\_title": "Otis Redding - Cigarettes And Coffee",

"caption": "<p>Otis Redding never fails me.\u00a0<\/p>",

"embed": "<iframe class=\"tumblr\_audio\_player tumblr\_audio\_player\_5578378101\" src=\"http:\/\/derekg.org\/post\/5578378101/otis-redding/audio\_player\_iframe" frameborder=\"0\" allowtransparency=\"true\" scrolling=\"no\" width=\"540\" height=\"85\"></iframe>",

"plays": 1576

}

],

"total\_posts": 1

}

}

**Legacy Video Posts**

| **Response Field** | **Type** | **Description** | **Notes** |
| --- | --- | --- | --- |
| **caption** | String | The user-supplied caption |  |
| **player** | Array of embed objects | Object fields within the array: | Values vary by video source |
| **width** | Number | Width of video player, in pixels |  |
| **embed\_code** | String | HTML for embedding the video player |  |

**Example**

https://api.tumblr.com/v2/blog/john.io/posts/video?api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "john",

"id": 6522991678,

"id\_string": "6522991678",

"post\_url": "http:\/\/john.io\/post\/6522991678",

"type": "video",

"date": "2011-06-14 15:51:21 GMT",

"timestamp": 1308066681,

"format": "html",

"reblog\_key": "sWRdVJrI",

"tags": [],

"note\_count": 17,

"source\_url": "http:\/\/www.WatchMojo.com",

"source\_title": "WatchMojo.com",

"caption": "<p><a href=\"https:\/\/foreverneilyoung.tumblr.com\/

post\/6522738445\" target=\"\_blank\">foreverneilyoung<\/a>:

<\/p>\n<blockquote>\n<p><a href=\"https:\/\/watchmojo.tumblr.com\/

post\/6521201320\" target=\"\_blank\">watchmojo<\/a>:<\/p>\n

<blockquote>\n<p>Neil Young\u2019s live album \u201cA Treasure\

u201d is available today. To celebrate, we take a look at the

life and career of the Canadian singer-songwriter.

<\/p>\n<\/blockquote>\n<p>Neil 101 for you new fans out

there.<\/p>\n<\/blockquote>\n<p><strong>If you don't

know\/appreciate Neil Young's impressive body of work,

this will help<\/strong><\/p>",

"player": [

{

"width": 250,

"embed\_code": "<object width=\"248\" height=\"169\"><param

name=\"movie\" value=\"http:\/\/www.youtube.com\/

v\/4Q1aI7xPo0Y&rel=0&egm=0&

showinfo=0&fs=1\"><\/param><param name=\"wmode\"

value=\"transparent\"><\/param><param name=\"

allowFullScreen\" value=\"true\"><\/param><embed

src=\"http:\/\/www.youtube.com\/v\/

4Q1aI7xPo0Y&rel=0&egm=0&showinfo=

0&fs=1\" type=\"application\/x-shockwave-flash\"

width=\"248\" height=\"169\" allowFullScreen=\"true\"

wmode=\"transparent\"><\/embed><\/object>"

},

{

"width": 400,

"embed\_code": "<object width=\"400\" height=\"251\">

<param name=\"movie\" value=\"http:\/\/www.youtube.com\/

v\/4Q1aI7xPo0Y&rel=0&egm=0&showinfo=0&fs=1\"><\/

param><param name=\"wmode\" value=\"transparent

\"><\/param><param name=\"allowFullScreen\" value=

\"true\"><\/param><embed src=\"http:\/\/www.youtube.com

\/v\/4Q1aI7xPo0Y&rel=0&egm=0&

showinfo=0&fs=1\" type=\"application\/

x-shockwave-flash\" width=\"400\" height=\"251\"

allowFullScreen=\"true\" wmode=\"transparent\">

<\/embed><\/object>"

},

{

"width": 500,

"embed\_code": "<object width=\"500\" height=\"305\"><param

name=\"movie\" value=\"http:\/\/www.youtube.com\/

v\/4Q1aI7xPo0Y&rel=0&egm=0&

showinfo=0&fs=1\"><\/param><param name=\"wmode\"

value=\"transparent\"><\/param><param name=\"

allowFullScreen\" value=\"true\"><\/param><embed

src=\"http:\/\/www.youtube.com\/v\/4Q1aI7xPo0Y&

rel=0&egm=0&showinfo=0&fs=1\"

type=\"application\/x-shockwave-flash\" width=\"500\"

height=\"305\" allowFullScreen=\"true\"

wmode=\"transparent\"><\/embed><\/object>"

}

]

},

...

],

"total\_posts": 48

}

}

**Legacy Answer Posts**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **asking\_name** | String | The blog that sent this ask, or answered it if it was privately answered |
| **asking\_url** | String | The blog URL that sent this ask, or answered it if it was privately answered |
| **question** | String | The question being asked |
| **answer** | String | The answer given |

**Example**

https://api.tumblr.com/v2/blog/david.tumblr.com/posts?id=7504154594&api\_key={key}

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"blog": { ... },

"posts": [

{

"blog\_name": "david",

"id": 7504154594,

"id\_string": "7504154594",

"post\_url": "https://www.davidslog.com/7504154594",

"type": "answer",

"date": "2011-07-11 20:24:14 GMT",

"timestamp": 1310415854,

"format": "html",

"reblog\_key": "HNvqLd5G",

"tags": [],

"asking\_name": "aperfectfacade",

"asking\_url": "https://aperfectfacade.tumblr.com/",

"question": "I thought Tumblr started in 2007, yet you

have posts from 2006?",

"answer": "<p>Good catch! Tumblr <strong>launched</strong> in

February 2007. We were testing it for a few months before

then.</p>\n<p><strong>Tumblr Trivia:</strong> Before Tumblr,

my blog (davidslog.com) was a manually edited, single page,

HTML tumblelog.</p>"

}

],

"total\_posts": 1

}

}

**/posts/queue — Retrieve Queued Posts**

Gives you a list of the currently queued posts for the specified blog.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/queue | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **offset** | String | Post number to start at | 0 (first post) | No |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **filter** | String | Specifies the post format to return, other than HTML: text – Plain text, no HTML; raw – As entered by the user (no post-processing); if the user writes in Markdown, the Markdown will be returned rather than HTML | None (HTML) | No |

**/posts/queue/reorder — Reorder Queued Posts**

This allows you to reorder a post within the queue, moving it after an existing queued post, or to the top.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/queue/reorder | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **post\_id** | String/Integer | Post ID to move | N/A | Yes |
| **insert\_after** | String/Integer | Which post ID to move it after, or 0 to make it the first post | 0 | No |

**/posts/queue/shuffle - Shuffle Queued Posts**

This randomly shuffles the queue for the specified blog.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/queue/shuffle | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**/posts/draft — Retrieve Draft Posts**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/draft | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **before\_id** | Number | Return posts that have appeared before this ID; Use this parameter to page through the results: first get a set of posts, and then get posts since the last ID of the previous set. | 0 | No |
| **filter** | String | Specifies the post format to return, other than HTML: text – Plain text, no HTML; raw – As entered by the user (no post-processing); if the user writes in Markdown, the Markdown will be returned rather than HTML | None (HTML) | No |

**/posts/submission — Retrieve Submission Posts**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/submission | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **offset** | String | Post number to start at | 0 (first post) | No |
| **filter** | String | Specifies the post format to return, other than HTML: text – Plain text, no HTML; raw – As entered by the user (no post-processing); if the user writes in Markdown, the Markdown will be returned rather than HTML | None (HTML) | No |

**Response**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **id** | Number | The ID of the submitted post |
| **id\_string** | String | The ID of the submitted post, in String format for clients that do not support 64-bit integers |
| **post\_url** | String | The location of the post |
| **slug** | String | Short text summary to the end of the post URL |
| **type** | String | The type of post. One of the following: text, photo, quote, link, video |
| **date** | String | The GMT date and time of the post |
| **timestamp** | Number | The time of the post, in seconds since the epoch |
| **state** | String | Indicates the current state of the post (submission) |
| **format** | String | Format type of post. |
| **reblog\_key** | String | The reblog key for the post |
| **tags** | Array | Tags applied to the post |
| **short\_url** | String | Short url for the post |
| **post\_author** | String | Author of post, only available when submission is not anonymous |
| **is\_submission** | Boolean | Indicates post is a submission (true) |
| **anonymous\_name** | String | Name on an anonymous submission |
| **anonymous\_email** | String | Email on an anonymous submission |

**/notifications — Retrieve Blog's Activity Feed**

Retrieve the activity items for a specific blog, in reverse chronological order.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/notifications | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **before** | Integer | Unix epoch timestamp that begins the page, defaults to request time. | Request time | No |
| **types** | String[] | An array of one or more types to filter by, or none if you want all | None | No |
| **rollups** | Boolean | Whether to "roll up" similar activity items into single items | true | No |
| **omit\_post\_ids** | String[] | An array of one or more of your own Post IDs to filter out | None | No |

Available "types" include:

| **Type** | **Description** |
| --- | --- |
| **like** | A like on your post |
| **reply** | A reply on your post |
| **follow** | A new follower |
| **mention\_in\_reply** | A mention of your blog in a reply |
| **mention\_in\_post** | A mention of your blog in a post |
| **reblog\_naked** | A reblog of your post, without commentary |
| **reblog\_with\_content** | A reblog of your post, with commentary |
| **ask** | A new ask received |
| **answered\_ask** | An answered ask that you had sent |
| **new\_group\_blog\_member** | A new member of your group blog |
| **post\_attribution** | Someone using your post content in their post |
| **post\_flagged** | A post of yours being flagged |
| **post\_appeal\_accepted** | An appeal accepted |
| **post\_appeal\_rejected** | An appeal rejected |
| **what\_you\_missed** | A post we think you missed |
| **conversational\_note** | A conversational note (reply, reblog with comment) on a post you're watching |

**Response**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **notifications** | Array | An array of activity item objects, see below. |
| **\_links** | Object | A pagination links object with a next key to use, to load more activity items. |

Note that different activity item objects vary field schema by activity item type. Some common fields include:

* id -- The ID of the activity item.
* type -- The type of activity item, from the list above.
* timestamp -- A unix epoch timestamp of when the event happened.
* unread -- A boolean indicating whether this is new/unread as of the last activity read time.
* target\_post\_id -- If the activity has to do with one of your blog's posts, this will be its ID.
* from\_tumblelog\_name -- If the activity is coming from another blog, like a Like or Reblog, this will be its name.
* post\_id -- For activity like Reblogs and Replies, this will be the relevant post's ID.
* post\_tags -- An array of used in the reblog, if any.
* added\_text -- For reblogs with comment, this will be a summary of the added content.
* reply\_text -- For replies, this will be the text of the reply.

**/post — Create a New Blog Post (Legacy)**

These legacy posting flows are still available, but we encourage you to use the [Neue Post Format creation route](https://www.tumblr.com/docs/en/api/v2#posts---createreblog-a-post-neue-post-format).

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/post | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

These parameters are used for /post, /post/edit, and /post/reblog methods.

**Fields available for all Post types:**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **type** | String | The type of post to create. Specify one of the following: text, photo, quote, link, chat, audio, video | text | Yes |
| **state** | String | The state of the post. Specify one of the following: published, draft, queue, private | published | No |
| **tags** | String | Comma-separated tags for this post | None | No |
| **tweet** | String | Manages the autotweet (if enabled) for this post: set to off for no tweet, or enter text to override the default tweet | None | No |
| **date** | String | The GMT date and time of the post, as a string | The date and time of the POST request | No |
| **format** | String | Sets the format type of post. Supported formats are: html & markdown | html | No |
| **slug** | String | Add a short text summary to the end of the post URL | Dynamically generated if enabled on the blog | No |
| **native\_inline\_images** | Boolean | Convert any external image URLs to Tumblr image URLs | false | No |

**Text Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **title** | String | The optional title of the post, HTML entities must be escaped | None | No |
| **body** | String | The full post body, HTML allowed | None | Yes |

**Photo Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **caption** | String | The user-supplied caption, HTML allowed | None | No |
| **link** | String | The "click-through URL" for the photo | None | No |
| **source** | String | The photo source URL | None | Yes, either source or data or data64 |
| **data** | Array (URL-encoded binary contents) | One or more image files (submit multiple times to create a slide show), limit 10MB | None | Yes, either source or data or data64 |
| **data64** | String | The contents of an image file encoded using base64, limit 10MB | None | Yes, either source or data or data64 |

**Quote Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **quote** | String | The full text of the quote, HTML entities must be escaped | None | Yes |
| **source** | String | Cited source, HTML allowed | None | No |

**Link Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **title** | String | The title of the page the link points to, HTML entities should be escaped | None | No |
| **url** | String | The link | None | Yes |
| **description** | String | A user-supplied description, HTML allowed | None | No |
| **thumbnail** | String | The url of an image to use as a thumbnail for the post | None | No |
| **excerpt** | String | An excerpt from the page the link points to, HTML entities should be escaped | None | No |
| **author** | String | The name of the author from the page the link points to, HTML entities should be escaped | None | No |

**Chat Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **title** | String | The title of the chat | None | No |
| **conversation** | String | The text of the conversation/chat, with dialogue labels (no HTML) | None | Yes |

**Audio Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **caption** | String | The user-supplied caption | None | No |
| **external\_url** | String | The URL of the site that hosts the audio file (not tumblr) | None | Yes, either external\_url or data |
| **data** | String (URL-encoded binary contents) | An audio file, limit 10MB | None | Yes, either external\_url or data |

**Video Posts**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **caption** | String | The user-supplied caption | None | No |
| **embed** | String | HTML embed code for the video or a URI to the video. If you provide an unsupported service's URI you may receive a 400 response. | None | Yes, either a URI, embed, or data. |
| **data** | String (URL-encoded binary contents) | A video file, limited to 500MB and 10 minutes | None | Yes, either embed or data |

**Response**

Returns 201: Created or an error code.

**/post/edit – Edit a Blog Post (Legacy)**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/post/edit | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

These parameters are in addition to the common parameters listed under [/post](https://www.tumblr.com/docs/en/api/v2#post--create-a-new-blog-post-legacy).

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the post to edit | None | Yes |

**Response**

Returns 200: OK (successfully edited) or an error code.

**/post/reblog – Reblog a Post (Legacy)**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/post/reblog | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

These parameters are in addition to the common parameters listed under [/post](https://www.tumblr.com/docs/en/api/v2#post--create-a-new-blog-post-legacy).

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the reblogged post on tumblelog | None | Yes |
| **reblog\_key** | Number | The reblog key for the reblogged post – get the reblog key with a /posts request | None | Yes |
| **comment** | String | A comment added to the reblogged post | None | No |
| **native\_inline\_images** | Boolean | Convert external image URLs in the comment to Tumblr image URLs | false | No |

**Response**

Returns 201: Created or an error code.

**/posts - Create/Reblog a Post (Neue Post Format)**

This route allows you to create posts (and reblogs) using the Neue Post Format. The specification for NPF is [here](https://www.tumblr.com/docs/npf).

**Note about Post States**

Posts can be in the following "states" as indicated in requests to the post creation/editing endpoints:

* "published" means the post should be publicly published immediately.
* "queue" means the post should be added to the end of the blog's post queue.
* "draft" means the post should be saved as a draft.
* "private" means the post should be privately published immediately.
* "unapproved" means the post is a new submission.

If omitted, the state parameter on a new post defaults to "published".

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **content** | Array | An array of [NPF content blocks](https://www.tumblr.com/docs/npf) to be used to make the post; in a reblog, this is any content you want to add. | [] | Yes, for new posts |
| **layout** | Array | An array of [NPF layout objects](https://www.tumblr.com/docs/npf) to be used to lay out the post content. | [] | No |
| **state** | String | The initial [state of the new post](https://www.tumblr.com/docs/en/api/v2#note-about-post-states), such as "published" or "queued". | "published" | No |
| **publish\_on** | String | The exact *future* date and time (ISO 8601 format) to publish the post, if desired. This parameter will be ignored unless the state parameter is "queue". | Now | No |
| **date** | String | The exact date and time (ISO 8601 format) in the past to *backdate* the post, if desired. This backdating does not apply to when the post shows up in the Dashboard. | Now | No |
| **tags** | String | A comma-separated list of tags to associate with the post. | None | No |
| **source\_url** | String | A source attribution for the post content. | None | No |
| **send\_to\_twitter** | Boolean | Whether or not to share this via any connected Twitter account on post publish. Defaults to the blog's global setting. | false | No |
| **is\_private** | Boolean | Whether this should be a private answer, if this is an answer. | false | No |
| **slug** | String | A custom URL slug to use in the post's permalink URL | Automatically generated based on the post's content | No |
| **interactability\_reblog** | String | Who can interact with this when reblogging; 'everyone' or 'noone' | 'everyone' | No |

**If the post being created is a reblog, all of the above parameters are expected, along with:**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **parent\_tumblelog\_uuid** | String | The unique public identifier of the Tumblelog that's being reblogged from. | N/A | Yes |
| **parent\_post\_id** | Integer | The unique public post ID being reblogged. | N/A | Yes |
| **reblog\_key** | String | The unique per-post hash validating that this is a genuine reblog action. | N/A | Yes |
| **hide\_trail** | Boolean | Whether or not to hide the reblog trail with this new post. Defaults to false. | false | No |
| **exclude\_trail\_items** | Array | Instead of **hide\_trail**, use this to specify an array of specific reblog trail item indexes to *exclude* from your reblog. | [] | No |

Note that when making a reblog, the trail from the post you're reblogging is not required, and will be ignored if given. Any content you provide when making a reblog will be added at the end of the reblog trail, in your post.

**User Uploaded Media**

In order to support user uploaded media (photos, videos, etc.), the creation and edit routes also support a multi-part form request where the first part contains the JSON body and subsequent parts contain media data. The Content-Type: multipart/form-data header must be used in this case.

To specify which media data pertains to which block, we use a unique identifier in the JSON body and this identifier is used as the key in the form data.

{

"content": [

{

"type": "image",

"media": [

{

"type": "image/jpeg",

"identifier": "some-identifier",

"width": 250,

"height": 200

}

]

}

]

}

--TumblrBoundary

Content-Disposition: form-data; name="json"

Content-Type: application/json

{JSON encoded parameters}

--TumblrBoundary

Content-Disposition: form-data; name="some-identifier"; filename="filename.jpg"

Content-Type: image/jpeg

{image data}

--TumblrBoundary--

Uploading a video for a video content block follows the same identifier pattern as above:

{

"content": [

{

"type": "video",

"media": {

"type": "video/mp4",

"identifier": "some-identifier",

"width": 250,

"height": 200

}

}

]

}

Normally we would require a provider field inside the video block, but on post creation this is assumed to be tumblr if none is provided.

**Response**

This returns a 201 Created on successful post creation, or an error code. The response JSON object will contain the id of the created post, which is intentionally a string instead of an integer for 32bit device compatibility.

**Example response:**

{

"meta": {

"msg": "Created",

"status": 201

},

"response": {

"id": "1234567891234567"

}

}

**Errors and Error Subcodes**

The error codes here are the HTTP statuses that'll be returned in error states, the error subcodes are the specific subcodes also returned in some situations. The response should have an errors array at the top level alongside meta, containing what error(s) occurred.

* 400 Bad Request when the request has not provided the required data.
  + 400.8001 when an NPF JSON parameter is invalid or a bad format.
  + 400.8002 when the reblog parent post and/or blog is invalid or cannot be found or cannot be interacted with.
  + 400.8005 when the uploaded media is an invalid format we cannot accept.
  + 400.8016 when there is something invalid about the format of the answer content or layout.
* 401 Unauthorized when the requester is an unauthorized client.
* 403 Forbidden when the requester is not allowed to reblog the parent post they specified.
  + 403.8004 when the requesting user cannot upload more media today.
  + 403.8008 when trying to upload a video in reblog content, which is not allowed.
  + 403.8010 when there is a video upload still transcoding, so you can't upload another video yet.
  + 403.8011 when the requesting user cannot upload more videos today.
  + 403.8022 when the blog's queue limit has been reached.
  + 403.8023 when the blog's daily posting limit has been reached.
* 404 Not Found when the request is not HTTPS.
* 404 Not Found when the Tumblelog identifier in the path does not resolve to an existing blog.
* 404 Not Found when the given post ID for editing/fetching doesn't exist.
* 500 Internal Server Error when something has gone very wrong.
  + 500.8006 when there was an unknown upload error (not video related).
  + 500.8009 when there was an unknown upload error (video related).
* 503 Service Unavailable when all posting, or posting via the API, is disabled.

For example:

{

"meta": {

"msg": "Bad Request",

"status": 400

},

"errors": [

{

"title": "'content' must be an array'",

"code": 8001

}

]

}

**/posts/{post-id} - Fetching a Post (Neue Post Format)**

This route can fetch a post, either legacy or NPF, and return it in either in the legacy or NPF format. The specification for NPF is [here](https://www.tumblr.com/docs/npf). Note that NPF to legacy conversion is not valid yet, so the available conversions are: NPF -> NPF, legacy -> legacy, and legacy -> NPF.

The intention of this route is to fetch a post for editing in either the NPF or legacy format.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/{post-id} | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |
| **post-id** | String | Post ID. See the [Post Identifiers](https://www.tumblr.com/docs/en/api/v2#post-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **post\_format** | String | The format to serve the post as, either npf or legacy. | npf | No |

**Response**

This returns a 200 OK on successful post fetching, or an error code.

The response JSON object will contain:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **object\_type** | String | The timeline object type, always post. |
| **type** | String | The post type. If formatting as NPF, the type will be blocks; if formatting as legacy, the type will be one of the original legacy types (text, photo, quote, chat, link, video, audio). |
| **id** | String | The post ID, intentionally a string instead of an integer, for 32bit device compatibility. |
| **tumblelog\_uuid** | String | The posting blog's unique identifier. |
| **parent\_post\_id** | String | The parent post ID, if the post being fetched is a reblog. |
| **parent\_tumblelog\_uuid** | String | The parent posting blog's UUID, if the post being fetched is a reblog. |
| **reblog\_key** | String | The reblog key needed as an additional authentication token. |
| **trail** | Array | Array of trail items, if the post being fetched is a reblog. |
| **content** | Array | Array of the content blocks of the post itself. |
| **layout** | Array | Array of the post's layout objects. |
| **queued\_state** | String | If the post's state is queued, this field will indicate whether it's actually a scheduled post. |
| **scheduled\_publish\_time** | Integer | If the post's state is queued, this will provide the scheduled publishing timestamp. |
| **publish\_on** | String | If the queued\_state is "scheduled", this is the timestamp of when the post is scheduled to publish, in ISO 8601 format. |
| **interactability\_reblog** | String | Who can interact with this when reblogging; 'everyone' or 'noone' |

Note about queued/scheduled posts: the queued\_state field will only exist for a state: queued post, and will indicate either queued or scheduled. The scheduled\_publish\_time field will tell you when the post is currently scheduled for publishing, as a Unix epoch timestamp. These fields are read-only; to change post's state or scheduled publish time, please use the state and publish\_on fields when editing the post.

**Errors and Error Subcodes**

See the notes from the NPF Post creation route for info about this.

**/posts/{post-id} - Editing a Post (Neue Post Format)**

This route allows you to edit posts using the Neue Post Format. Note that you can only edit posts in NPF if they were originally created in NPF, or are legacy text posts. The specification for NPF is [here](https://www.tumblr.com/docs/npf).

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/{post-id} | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |
| **post-id** | String | Post ID. See the [Post Identifiers](https://www.tumblr.com/docs/en/api/v2#post-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

For editing a post, all of the request parameters from the NPF Post Creation route are expected (depending on if it's an original post or reblog), along with the Post's ID in the query path. If you are editing a scheduled post, make sure to include its publish\_on value.

Note that the exclude\_trail\_items parameter will edit the reblog's current trail at the time of editing, not the trail that existed when the reblog was created. So if you had created a reblog and excluded trail items at creation, then are editing it to exclude trail items again, you are now editing that reblog's trail, not the parent post's trail. Basically: you can't "bring back" trail items without reblogging the parent post all over again.

**Response**

This returns a 200 OK on successful post editing, or an error code. The response JSON object will contain the post\_id of the edited post, which is intentionally a string instead of an integer for 32bit device compatibility.

**Example response:**

{

"meta": {

"msg": "OK",

"status": 200

},

"response": {

"id": "1234567891234567"

}

}

**Errors and Error Subcodes**

See the notes from the NPF Post creation route for info about this.

**/post/delete – Delete a Post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/post/delete | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the post to delete | N/A | Yes |

**Response**

Returns 200: OK (successfully deleted) or an error code.

**/posts/{post-id}/mute – Muting a Post's Notifications**

Muting a post's notifications means the author of the post will no longer receive activity items or push notifications about the post, until the post is unmuted, or until the temporary mute expires.

**Methods**

The POST request mutes the given post, and the DELETE request can unmute a post.

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/{post-id}/mute | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |
| api.tumblr.com/v2/blog/{blog-identifier}/posts/{post-id}/mute | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |
| **post-id** | String | The post ID to mute. | N/A | Yes |

**Body Parameters**

To make the mute temporary, you can supply how long you want the mute to last.

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **mute\_length\_seconds** | Integer | How many seconds the mute should last | 0 (forever) | No |

Giving 0 or omitting mute\_length\_seconds entirely means the mute will last until the post is unmuted by the author.

**Response**

Returns 200 OK on successful muting or unmuting, or an error code.

Returns 400 Bad Request if the given mute\_length\_seconds is less than 0.

**/notes - Get notes for a specific Post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/notes | GET | [API Key](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the post to fetch notes for | N/A | Yes |
| **before\_timestamp** | Number | Fetch notes created before this timestamp, for pagination. This is a unix timestamp in seconds precision, but microsecond precision for conversation mode. | N/A | No |
| **mode** | String | The response formatting mode, see list below. | "all" | No |

The mode field can be one of a few things, most of which act as filters for the types of notes returned. All return notes in reverse chronological order.

* "all" loads all notes for the post.
* "likes" loads only likes for the post.
* "conversation" loads only replies and reblogs with added text commentary, with the rest of the notes (likes, reblogs without commentary) in a rollup\_notes field.
* "rollup" loads only like and reblog notes for the post in the notes array.
* "reblogs\_with\_tags" loads only the reblog notes for the post, and each note object includes a tags array field (which may be empty).

And different modes can cause the response to contain different things.

Example request:

GET https://api.tumblr.com/v2/blog/YOUR-BLOG.tumblr.com/notes?id=1234567890000&mode=all&before\_timestamp=1234567890

**Response**

Returns 200: OK with an array of notes, along with more info, and a \_links object to load more notes.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **notes** | Array | An array of note objects, which may be formatted differently based on the mode and note type |
| **rollup\_notes** | Array | In "conversation" mode, this contains notes not listed in the notes array. |
| **total\_notes** | Number | The total notes, which can change depending on the mode. |
| **total\_likes** | Number | The total likes, when mode is conversation. |
| **total\_reblogs** | Number | The total reblogs, when mode is conversation. |
| **\_links** | Object | Contains a next link object for pagination. |

Example response:

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"notes": [

{

"type": "reblog",

"timestamp": 1595527574,

"blog\_name": "test-blog",

"blog\_uuid": "t:abcd1234",

"blog\_url": "https://test-blog.tumblr.com/",

"followed": false,

"avatar\_shape": "square",

"post\_id": "1123456688899000",

"reblog\_parent\_blog\_name": "other-test-blog"

},

{

"type": "like",

"timestamp": 1595527569,

"blog\_name": "test-blog",

"blog\_uuid": "t:abcd1234",

"blog\_url": "https://test-blog.tumblr.com/",

"followed": false,

"avatar\_shape": "square"

}

],

"total\_notes": 2

}

}

**/pages - Retrieve Pages**

This route allows you to retrieve a list of pages for a blog. Pages are returned based on their 'position' setting.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/pages | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Post number to start at | 0 (first post) | No |

**Response**

This returns a 200 OK on successful page fetching, or an error code.

**Example response:**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"pages": [

{

"path": "/test-page",

"title": "test test",

"content": "<p>:)</p>\n<p>hello world...</p>",

"position": 1,

"updated\_at": 1733165107,

"custom\_template": false,

"show\_link": false

},

{

"path": "/another-page",

"title": "TEST!",

"content": "hello world",

"position": 2,

"updated\_at": 1733165107,

"custom\_template": false,

"show\_link": true

},

{

"path": "/custom-layout",

"title": "",

"content": "<strong>test test test</strong>",

"position": 3,

"updated\_at": 1733165107,

"custom\_template": true,

"show\_link": false

}

],

"total\_pages": 12,

"\_links": {

"next": {

"href": "/v2/blog/test-blog/pages?limit=3&offset=3",

"method": "GET",

"query\_params": {

"limit": 3,

"offset": 3

}

}

}

}

}

**Errors and Error Subcodes**

The error codes here are the HTTP statuses that'll be returned in error states, The response should have an errors array at the top level alongside meta, containing more information about what error(s) occurred.

* 401 Unauthorized when the requester is not logged in.
* 400 Bad Request when an invalid tumblelog identifier is given.
* 403 Forbidden when the requester is not allowed to access the given blog.
* 404 Not Found when the given blog is not found.

**/pages/{page-name} - Retrieve a Page**

This route allows you to retrieve a single page for a blog.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/blog/{blog-identifier}/pages/{page-name} | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **blog-identifier** | String | Any blog identifier. See the [Blog Identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) section for more details. | N/A | Yes |
| **page-name** | String | The name of the page to retrieve. The page name should be url-encoded if requesting a subpath (e.g. %2F for /). | N/A | Yes |

**Response**

This returns a 200 OK on successful page fetching, or an error code.

**Example response:**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"path": "/test-page",

"title": "Test test",

"content": "<p>Hello World!</p>\n<p>This is a test page.</p>",

"position": 1,

"updated\_at": 1733165107,

"custom\_template": false,

"show\_link": false

}

}

**Errors and Error Subcodes**

The error codes here are the HTTP statuses that'll be returned in error states, The response should have an errors array at the top level alongside meta, containing more information about what error(s) occurred.

* 401 Unauthorized when the requester is not logged in.
* 400 Bad Request when an invalid tumblelog identifier is given.
* 403 Forbidden when the requester is not allowed to access the given blog.
* 404 Not Found when the given blog is not found or the given page is not found.

**User Methods**

**/user/info – Get a User's Information**

Use this method to retrieve the user's account information that matches the OAuth credentials submitted with the request.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/info | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Response**

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **following** | Number | The number of blogs the user is following |
| **default\_post\_format** | String | The default posting format - html, markdown, or raw |
| **name** | String | The user's tumblr short name |
| **likes** | Number | The total count of the user's likes |
| **blogs** | Array | Each item is a blog the user has permissions to post to, containing these fields: |
| **blogs.name** | String | the short name of the blog |
| **blogs.url** | String | the URL of the blog |
| **blogs.title** | String | the title of the blog |
| **blogs.primary** | Boolean | indicates if this is the user's primary blog |
| **blogs.followers** | Number | total count of followers for this blog |
| **blogs.tweet** | String | indicate if posts are tweeted auto, Y, N |
| **blogs.type** | String | indicates whether a blog is public or private |

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"user": {

"following": 263,

"default\_post\_format": "html",

"name": "derekg",

"likes": 606,

"blogs": [

{

"name": "derekg",

"title": "Derek Gottfrid",

"url": "https://derekg.org/",

"tweet": "auto",

"primary": true,

"followers": 33004929,

},

{

"name": "ihatehipstrz",

"title": "I Hate Hipstrz",

...

}

]

}

}

}

**/user/limits – Get a User's Limits**

Use this method to retrieve information about the various limits for the current user.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/limits | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Response**

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"user": {

"blogs": {

"description": "the number of secondary blogs you can create per day",

"limit": 10,

"remaining": 4,

"reset\_at": 1670043600

},

"follows": {

"description": "the number of blogs you can follow per day",

"limit": 200,

"remaining": 199,

"reset\_at": 1670043600

},

...

}

}

}

**/user/dashboard – Retrieve a User's Dashboard**

Use this method to retrieve the dashboard that matches the OAuth credentials submitted with the request.

⚠️ Note: Please don't re-implement the Dashboard, and don't recreate complete Tumblr functions or clients on a platform where Tumblr already has an official client. See our API policies [here](https://www.tumblr.com/docs/api_agreement).

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/dashboard | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Post number to start at | 0 (first post) | No |
| **type** | String | The type of post to return. Specify one of the following: text, photo, quote, link, chat, audio, video, answer | None – return all types | No |
| **since\_id** | Number | Return posts that have appeared after this ID; Use this parameter to page through the results: first get a set of posts, and then get posts since the last ID of the previous set. | 0 | No |
| **reblog\_info** | Boolean | Indicates whether to return reblog information (specify true or false). Returns the various reblogged\_ fields. | False | No |
| **notes\_info** | Boolean | Indicates whether to return notes information (specify true or false). Returns note count and note metadata. | False | No |
| **npf** | Boolean | Returns posts' content in [NPF format](https://www.tumblr.com/docs/npf) instead of the legacy format. | False | No |

**Response**

Dashboard responses include the fields returned in /posts [responses](https://www.tumblr.com/docs/en/api/v2#posts--retrieve-published-posts) (with all the various type-specific fields), but without the blog object. Instead, a blog\_name field identifies the blog for each post returned.

**Important note:** Post content can be in two formats: legacy or Neue Post Format (NPF). By default, posts returned from this endpoint (and any other endpoint that returns posts) will be in the legacy post-type-based content formats [described here](https://www.tumblr.com/docs/en/api/v2#posts--retrieve-published-posts). NPF-created posts from the official Tumblr mobile apps will be returned as text/regular posts to maintain backwards compatibility. To help transition to an NPF-only world, you can pass along the npf=true query parameter to force *all posts* returned here to be in Neue Post Format (also [described here](https://www.tumblr.com/docs/en/api/v2#posts--retrieve-published-posts)).

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"posts": [

{

"blog\_name": "laughingsquid",

"id": 6828225215,

"id\_string": "6828225215",

"post\_url": "https:\/\/links.laughingsquid.com\/post\/6828225215",

...

}

]

}

}

**/user/likes — Retrieve a User's Likes**

Use this method to retrieve the liked posts that match the OAuth credentials submitted with the request.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/likes | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Liked post number to start at | 0 (first post) | No |
| **before** | Integer | Retrieve posts liked before the specified timestamp | None | No |
| **after** | Integer | Retrieve posts liked after the specified timestamp | None | No |

* You can only provide either before, after, or offset. If you provide more than one of these options together you will get an error.
* You can still use limit with any of those three options to limit your result set.
* When using the offset parameter the maximum limit on the offset is 1000. If you would like to get more results than that use either before or after.

**Response**

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **liked\_posts** | Array | An array of post objects (posts liked by the user) |
| **liked\_count** | Number | Total number of liked posts |

**/user/following – Retrieve the Blogs a User Is Following**

Use this method to retrieve the blogs followed by the user whose OAuth credentials are submitted with the request.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/following | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **offset** | Number | Result number to start at | 0 (first followee) | No |

**Response**

These fields are wrapped in a blog object:

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **total\_blogs** | Number | The number of blogs the user is following |
| **blogs** | Array | Each item is a blog that's being followed, containing these fields: |
| **name** | String | the user name attached to the blog that's being followed |
| **url** | String | the URL of the blog that's being followed |
| **updated** | Number | the time of the most recent post, in seconds since the epoch |
| **title** | String | the title of the blog |
| **description** | String | the description of the blog |

**Example**

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"total\_blogs": 259,

"blogs": [

{

"name": "david",

"url": "https:\/\/www.davidslog.com",

"updated": 1308781073,

"title": "David’s Log",

"description": "“Mr. Karp is tall and skinny, with unflinching blue eyes and a mop of brown hair. He speaks incredibly fast and in complete paragraphs.” – NY Observer"

},

{

"name": "matthew",

"url": "https:\/\/matthew.tumblr.com",

"updated": 1306436424,

"title": "Matt Hackett",

"description": " I obsess over the engineering and dissemination of technology for creative people.

\nI'm happily torn between work on the visible (marketing) and the invisible (architecture, management) components of media-product-making machinery.\n

\nMost recently, I saw as Tumblr's team grew from 10 to 110 and monthly audience from 25 million to 150 million from my vantage point as VP Engineering and later Head of Brand Strategy.

\nMore about Matt Hackett"

},

...

]

}

}

**/user/follow – Follow a blog**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/follow | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| url | String | The URL of the blog to follow | None | Must supply url or email |
| email | String | The email of the blog to follow. A blog is only followable by email if it has the Let people find your blogs through this address. setting enabled on [tumblr.com/settings/account](https://www.tumblr.com/settings/account). | None | Must supply url or email |

**Response**

Returns 200: OK (blog successfully followed) or a 404 (blog was not found)

| **Response Field** | **Type** | **Description** |
| --- | --- | --- |
| **blog** | Object | The followed blog info |

**/user/unfollow – Unfollow a blog**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/unfollow | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **url** | String | The URL of the blog to unfollow | None | Yes |

**Response**

Returns 200: OK (blog successfully unfollowed) or a 404 (if the blog was not found)

**/user/like – Like a Post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/like | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the post to like | None | Yes |
| **reblog\_key** | String | The reblog key for the post id | None | Yes |

**Response**

Returns 200: OK (post successfully liked ) or a 404 (post ID or reblog\_key was not found)

**/user/unlike – Unlike a Post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/user/unlike | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **id** | Number | The ID of the post to unlike | None | Yes |
| **reblog\_key** | String | The reblog key for the post id | None | Yes |

**Response**

Returns 200: OK (post successfully unliked ) or a 404 (post ID or reblog\_key was not found)

**/user/filtered\_tags - Tag Filtering**

This endpoint lets you manage the tagged content you'd like covered on your dashboard. See more info about this feature [here](https://tumblr.zendesk.com/hc/articles/115015814708-Tag-filtering). Note that this filtering relies on the content being tagged, but also filters tag recommendations and other tag-based content. For filtering the text content of posts, see [Content Filtering](https://www.tumblr.com/docs/en/api/v2#userfiltered_content---content-filtering).

**Methods**

All of the endpoints require [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) authentication.

| **URI** | **HTTP Method** | **Description** |
| --- | --- | --- |
| api.tumblr.com/v2/user/filtered\_tags | GET | Retrieve a list of currently-filtered tag strings. |
| api.tumblr.com/v2/user/filtered\_tags | POST | Add one or more tags to filter. |
| api.tumblr.com/v2/user/filtered\_tags/{tag} | DELETE | Remove a tag filter. |

**Request Parameters**

For adding new tag filters, you can add one at a time or many at once via the POST body:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **filtered\_tags** | Array of Strings | One or more than one tag string to add to your list of filters |

Example:

POST https://api.tumblr.com/v2/user/filtered\_tags

Content-Type: application/json

{ "filtered\_tags": [ "something" ] }

POST https://api.tumblr.com/v2/user/filtered\_tags

Content-Type: application/json

{ "filtered\_tags": [ "something", "technology" ] }

For deleting a tag filter, pass along the tag string in the path:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **tag** | String | Tag to stop filtering; this is expected to be ["raw" URL encoded](http://www.faqs.org/rfcs/rfc3986.html). |

Example:

DELETE https://api.tumblr.com/v2/user/filtered\_tags/tag%20here

That would delete the tag "tag here".

**Response**

For GET requests, the endpoint will return a 200 OK on success, along with the list of tags:

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"filtered\_tags": [

"technology",

"something",

"and something else"

]

}

}

For POST requests, the endpoint will return a 201 Created on success, with an empty response object.

For DELETE requests, the endpoint will return a 200 OK on success, with an empty response object.

**Errors**

* 400 Bad Request if given an invalid/empty tag to filter or delete.

**/user/filtered\_content - Content Filtering**

This endpoint lets you manage the plain text content you'd like covered on your dashboard, including blog names in the reblog trail. See more info about this feature [here](https://tumblr.zendesk.com/hc/articles/360046752174). For filtering based on tags, see [Tag Filtering](https://www.tumblr.com/docs/en/api/v2#userfiltered_tags---tag-filtering).

**Limits**

* Each user can have a maximum of 200 filtered strings.
* Each filtered string cannot be more than 250 characters in length.
* Filtered strings can contain any characters that are valid UTF-8.

**Methods**

All of the endpoints require [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) authentication.

| **URI** | **HTTP Method** | **Description** |
| --- | --- | --- |
| api.tumblr.com/v2/user/filtered\_content | GET | Retrieve a list of currently-filtered content strings. |
| api.tumblr.com/v2/user/filtered\_content | POST | Add one or more content strings to filter. |
| api.tumblr.com/v2/user/filtered\_content | DELETE | Remove a content filter string. |

**Request Parameters**

For adding new content filters, you can add one at a time or many at once via the POST body:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **filtered\_content** | String or Array of Strings | One or more than one string to add to your list of filters |

Example:

POST https://api.tumblr.com/v2/user/filtered\_content

Content-Type: application/x-www-form-urlencoded

filtered\_content=something

POST https://api.tumblr.com/v2/user/filtered\_content

Content-Type: application/json

{ "filtered\_content": [ "something", "technology" ] }

For deleting a content filter, pass along the string in the query params:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **filtered\_content** | String | Content filter string to remove. |

Example:

DELETE https://api.tumblr.com/v2/user/filtered\_content?filtered\_content=something

**Response**

For GET requests, the endpoint will return a 200 OK on success, along with the list of strings:

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"filtered\_content": [

"technology",

"something",

"and something else"

]

}

}

For POST requests, the endpoint will return a 201 Created on success, with an empty response object.

For DELETE requests, the endpoint will return a 200 OK on success, with an empty response object.

**Errors**

* 400 Bad Request if given an invalid/empty string to filter or delete, or if given an already-filtered string.
* 403 Forbidden if at the maximum number of allowed filters.

**Tagged Method**

**/tagged – Get Posts with Tag**

**Examples**

* https://api.tumblr.com/v2/tagged?tag=gif
* https://api.tumblr.com/v2/tagged?tag=lol

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/tagged | GET | [API Key or OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **tag** | String | The tag on the posts you'd like to retrieve | None | Yes |
| **before** | Integer | The timestamp of when you'd like to see posts before. If the Tag is a "featured" tag, use the "featured\_timestamp" on the post object for pagination. | Current timestamp | No |
| **limit** | Number | The number of results to return: 1–20, inclusive | 20 | No |
| **filter** | String | Specifies the post format to return, other than HTML: text – Plain text, no HTML; raw – As entered by the user (no post-processing); if the user writes in Markdown, the Markdown will be returned rather than HTML; | None (HTML) | No |

**Communities Methods**

⚠️ **NOTE:** Communities is a brand new beta feature that's under active development. These endpoints may change in backwards-incompatible ways until the feature is fully released.

Communities endpoints often require a "community handle" path parameter, which can be given in one of a few different forms:

* The name of the community, without the double-@ prefix, i.e. communities-feedback.
* The community's unique Tumblr UUID, like [blog unique identifiers](https://www.tumblr.com/docs/en/api/v2#blog-unique-identifiers). Community objects will include this.

**GET /v2/communities - List joined communities**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Response**

The response field is an array of your joined communities. See [the next endpoint](https://www.tumblr.com/docs/en/api/v2#get-v2communitiescommunity-handle---get-info-about-a-community) for the community object definition.

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": [

{ ... }, // community object

{ ... }, // community object

]

}

**GET /v2/communities/{community-handle} - Get info about a community**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle} | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **context** | String | The context of what the info will be used for, see note below | None | No |

The context request parameter can be set to edit (i.e. ?context=edit) to get additional editable parts of the community, like its settings.

**Response**

Community objects contain these fields in the "full" version of the object. Some of them are only visible to certain roles or specific situations.

The "short" abbreviated version of the community object contains only name, uuid, title, member\_count, mention\_tag, can\_view, and avatar\_image.

| **Field** | **Type** | **Description** | **Visibility** |
| --- | --- | --- | --- |
| **name** | String | The name/handle of the community, without the double-@ | Visible to all |
| **uuid** | String | The [unique identifier](https://www.tumblr.com/docs/en/api/v2#blog-unique-identifiers) of the community | Visible to all |
| **title** | String | The title of the community | Visible to all |
| **description** | String | The description of the community | Visible to all |
| **about** | String | The About information of the community | Visible to all |
| **guidelines** | Array | An array of strings, each a community guideline | Visible to all |
| **visibility** | String | The community's visibility, public or private | Visible to all |
| **member\_count** | Number | How many members are in this community | Visible to all |
| **members\_online\_count** | Number | How many members are "online" now | Visible to all |
| **mention\_tag** | String | The handle of the community, with the double-@ | Visible to all |
| **is\_member** | Boolean | Whether the current user is a member of this community | Visible to all |
| **can\_view** | Boolean | Whether the community can be viewed by the current user | Visible to all |
| **can\_post** | Boolean | Whether the current user can post to this community | Visible to all |
| **can\_edit** | Boolean | Whether the current user can edit this community | Visible to all |
| **can\_interact** | Boolean | Whether the current user can interact with the posts in this community | Visible to all |
| **can\_invite** | Boolean | Whether the current user can invite others to this community | Visible to all |
| **can\_view\_comments** | Boolean | Whether the current user can view comments in posts in this community | Visible to all |
| **post\_count** | Number | The number of posts published in this community | Visible to all |
| **unread\_post\_count** | Number | The number of "unread" posts for the current user | Visible to all |
| **tags** | Array | The tags used to describe this community | Visible to all |
| **created\_ts** | Number | The unix epoch timestamp of when this community was created | Visible to all |
| **invitation** | Object | Information about the current user's invitation | Visible if the user has an invitation |
| **handle\_can\_change** | Boolean | Whether the community handle can be changed | Visible to admins in "edit" context |
| **handle\_next\_change\_ts** | Boolean | If the handle cannot be changed, this is the next time it can be changed | Visible to admins in "edit" context |
| **can\_edit\_content\_label** | Boolean | Whether the content label can be changed | Visible to admins in "edit" context |
| **settings** | Object | The community's current settings values | Visible to admins in "edit" context |
| **avatar\_image** | Array | The community's avatar, in various sizes | Visible to all |
| **header\_image** | Array | The community's header image, in various sizes | Visible to all |
| **population\_cap** | Number | The population cap of the community, can be null if there is none | Visible to all |
| **join\_type** | String | Whether the community is invite-only or free-to-join | Visible to all |
| **has\_content\_label** | Boolean | Whether this community has a "mature" content label | Visible to all |
| **content\_label\_categories** | Array | The specific content labels this community has applied | Visible to all |
| **content\_label\_last\_reporter** | String | The origin of this community's mature classification, can be null if the community is not mature | Visible to all |
| **invite\_link** | String | The invite link to join the community, can be null if disabled | Visible to admins |
| **pending\_invitations\_count** | Number | How many pending invitations there are | Visible to admins |

For those who can edit the community, the settings object contains:

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **invite\_links\_enabled** | Boolean | Whether an invite link is available for this community |
| **join\_type** | String | What join options non-members have, can be "free" or "invite". Note that the top-level join\_type field takes precedence. |
| **mods\_can\_invite** | Boolean | Whether moderators in the community can invite new members, or only admins |
| **automod\_links** | Boolean | Whether to automatically block new posts with suspicious links from publishing |

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"name": "community-name",

"uuid": "t:abcd123",

"title": "Community Title",

"description": "A description of the community",

"about": "Nothing here yet",

"guidelines": [],

"visibility": "public",

"member\_count": 12,

"members\_online\_count": 10,

"mention\_tag": "@@community-name",

"is\_member": true,

"can\_view": true,

"can\_post": true,

"can\_edit": true,

"can\_interact": true,

"can\_invite": true,

"can\_view\_comments": true,

"post\_count": 8,

"unread\_post\_count": 1,

"tags": ['tumblr communities'],

"created\_ts": 1709221499,

"invitation": {

"type": "direct",

"sender": { ... } // blog info object

},

"handle\_can\_change": false,

"handle\_next\_change\_ts": 1234567890,

"settings": {}, // settings object

"avatar\_image": [

{

"width": 512,

"height": 512,

"url": "https://64.media.tumblr.com/bac5b311618185a605d83e7244bfdec3/b26b33c94f644486-87/s512x512u\_c1/39187fb2bbf11f92e91f40ca9a15f92041149ee6.pnj"

},

{

"width": 128,

"height": 128,

"url": "https://64.media.tumblr.com/bac5b311618185a605d83e7244bfdec3/b26b33c94f644486-87/s128x128u\_c1/424063b0e2e23d9060dedb40428e05402bfc0078.pnj"

},

{

"width": 96,

"height": 96,

"url": "https://64.media.tumblr.com/bac5b311618185a605d83e7244bfdec3/b26b33c94f644486-87/s96x96u\_c1/25bfccd20b0777a82aa8cb201b6e3d48714a2c41.pnj"

},

{

"width": 64,

"height": 64,

"url": "https://64.media.tumblr.com/bac5b311618185a605d83e7244bfdec3/b26b33c94f644486-87/s64x64u\_c1/a2ed9c2bec1c1c9e39d53fc5df564467dd01f7be.pnj"

}

],

"population\_cap": 9001,

"join\_type": "invite",

"invite\_link": "https://www.tumblr.com/join/abcd123",

"header\_image": [

{

"type": "image/jpeg",

"width": 2048,

"height": 1542,

"url": "https://64.media.tumblr.com/2ad713f0fa15a3f0517e7c5b217b4c54/32cbed0b10ff7b6c-23/s2048x3072/dad0b45b4740f15510fcb0373b52ecf6a2f7ddee.jpg"

},

{

"type": "image/jpeg",

"width": 1280,

"height": 963,

"url": "https://64.media.tumblr.com/2ad713f0fa15a3f0517e7c5b217b4c54/32cbed0b10ff7b6c-23/s1280x1920/7eb927d8f7e3a563c05e1ba39f3a24c3b1913791.jpg"

},

{

"type": "image/jpeg",

"width": 640,

"height": 482,

"url": "https://64.media.tumblr.com/2ad713f0fa15a3f0517e7c5b217b4c54/32cbed0b10ff7b6c-23/s640x960/4a7d164a3d0e2bffc1225489217e4d172c5f4822.jpg"

}

],

"pending\_invitations\_count": 2

}

}

**Errors**

* 403 Forbidden if you are not allowed to view this:
  + Error code 23020 signifies that you have been banned from the community.
  + Error code 23001 signifies that the community is private and you are not a member.
  + Error code 23002 signifies that you are not a member and you need to be to see the content.
* 404 Not Found if the community was not found.

**PUT /v2/communities/{community-handle} - Edit your community**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle} | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Body Parameters**

The different community fields that are editable:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **title** | String | The title of the community |
| **description** | String | The description/tagline of the community |
| **handle** | String | The handle/URL of the community, see note below |
| **privacy** | Number | 0 for public, 1 for private |
| **tags** | Array | An array of strings, each a tag to describe your community |
| **about** | String | More information about your community |
| **guidelines** | Array | An array of strings, each a guideline that members need to follow |
| **invite\_links\_enabled** | Boolean | Whether invite links are enabled for your community |
| **join\_type** | String | The type of options for joining, 'invite' or 'free' |
| **mods\_can\_invite** | Boolean | Whether moderators can invite new members for your community |
| **has\_content\_label** | Boolean | Whether this community has a "mature" content label |
| **content\_label\_categories** | Array | The specific content labels, see note below |
| **automod\_links** | Boolean | Whether to automatically block/moderate new posts with links from publishing |

Note that the handle can only be edited once every 7 days. When reading a community object, the handle\_can\_change and handle\_next\_change\_ts fields will tell you if you can edit this.

Also note that updating the handle will invalidate the previous community URL and free up that handle to be taken by another community.

Also note that content\_label\_categories can include sexual\_themes, violence, and/or drug\_use specific labels.

**Response**

200 OK on successful save, along with the JSON object for your community, including the additional fields as if context=edit had been passed.

**Errors**

* 400 Bad Request when you've provided an invalid parameter. Error code 23025 means you cannot edit the content label.
* 403 Forbidden when you do not have permission to edit this community.
* 503 Service Unavailable when we've encountered a problem saving your community.

**GET /v2/communities/{community-handle}/invitations - View pending invites**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/invitations | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Response Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **before** | Number | The timestamp to use for the next page boundary | None | No |

**Response**

200 OK on success.

The response will have a invitations array of invitations, sorted in reverse chronological order, each with fields:

* sender -- a blog object representing who sent the invite.
* recipient -- a blog object representing who was invited.
* sent\_on -- the unix timestamp of when the invite was sent.

If there are more than 20 invitations, a \_links object alongside the invitations array will provide the next page info.

**PUT /v2/communities/{community-handle}/invitations - Invite someone to the community**

Admins can send invites at a maximum rate of 1 per second and 250 per day across all their communities.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/invitations | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **recipient\_blog** | String | The blog to invite, accepts any [blog identifier](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) | None | Yes |
| **message** | String | A message to send with the invite, max length 100 characters | None | No |

**Response**

200 OK on success.

**Errors**

* 400 Bad Request if you're missing a receipient blog in the request.
  + with error code 5041 if the message is too long.
* 403 Forbidden for anyone trying to do this who isn't an admin of the community. Various error codes:
  + 23003 if the community is at capacity already.
  + 23008 if the recipient blog, or its owner, has been banned from the community, or otherwise cannot join.
  + 23009 if the recipient blog is actually already a member.
  + 23011 if the recipient blog already has an invitation.
* 429 Limit Exceeded if sending invites too fast. See note above.

**Limits**

Admins can send 250 invites per day, at a maximum rate of 1 per second, across all of their communities.

**DELETE /v2/communities/{community-handle}/invitations/{blog-identifier} - Cancel an invitation**

This allows an admin to cancel a pending invitation to a specific blog, and it allows a receipient of an invitation to decline the invite, and disallow future invitations.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/invitations/{blog-identiifer} | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **blog-identifier** | String | The blog to cancel, see [blog identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) | None | Yes |

**Response Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **do\_not\_invite** | Boolean | Whether to disallow any future re-invitations | None | No |

**Response**

200 OK on success.

**GET /v2/communities/{community-handle}/invitations/{blog-identifier} - Check invitation status**

This allows you to check the invitation status for a specific blog recipient belonging to the current user.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/invitations/{blog-identiifer} | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **blog-identifier** | String | The blog to check, see [blog identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) | None | Yes |

**Response**

200 OK on success, with a sender object with a name field indicating who sent the invite.

**Errors**

* 404 Not Found if the given blog does not have an invitation to the community.

**POST /v2/communities/{community-handle}/invite\_hash - Regenerate the invite URL**

Allows admins to regenerate the invite URL with a new hash. Note that this invalidates the previous invite link.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/invite\_hash | POST | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Response**

200 OK on success, with a invite\_link field containing the new invite link.

**Errors**

* 403 Forbidden with error code 23017 if invite links have been disabled for the community.
  + ... or error code 23012 if the user trying to regenerate the link is not allowed to do so.

**GET /v2/communities/{community-handle}/members - Get community members**

The list of all members can only be fetched by members, those with a pending invite, an invite code, or if the community is free-to-join and public.

The lists of admins and moderators is always accessible for public communities.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/members | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **page** | Number | The page of results to fetch | 1 | No |
| **filter** | String | Filter to a specific set of members | 'all' | No |

The filter field, if provided, can have one of these values:

* 'all' for all members.
* 'admin' to list only admins.
* 'moderator' to list only moderators.
* 'following' to list any blogs the requesting user is following.
* 'mutual' to list any blogs the requesting user has a mutual following relationship with.

**Response**

200 OK on success. The response will have a members array, which contains standard blog info objects for each member.

Each member blog object will also contain additional fields:

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| **joined\_ts** | Number | The unix epoch timestamp of when this member joined. |
| **is\_admin** | Boolean | Whether this member has the "admin" role. |
| **is\_moderator** | Boolean | Whether this member has the "moderator" role. |
| **is\_following\_you** | Boolean | Whether this member is following the requesting user's primary blog. |
| **followed** | Boolean | Whether this member is being followed by the requesting user. |
| **is\_mutual** | Boolean | Whether this member is a mutual of the requesting user. |

**Errors**

* 403 Forbidden with all the error codes from the "get info" endpoint.

**PUT /v2/communities/{community-handle}/members - Join a community**

Users can join a maximum of 25 communities per hour and 100 per day via free-to-join or invite link.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/members | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Body Parameters**

To join, the request must have one of the following:

* The correct invite\_hash,
* or the community is free-to-join,
* or the requesting user must have a pending invitation.

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **invite\_hash** | String | The invite code/hash | None | No |

**Response**

200 OK on successful join.

**Errors**

* 403 Forbidden with all the error codes from the "get info" endpoint, but also:
  + Error code 23003 if the community is already at capacity.
  + Error code 23015 if the pending invitation no longer exists or was already declined.
* 400 Bad Request with error code 23014 if the given invite hash is invalid.
* 429 Limit Exceeded when too many joins have happened too quickly. See note above.

**DELETE /v2/communities/{community-handle}/members/{blog-identifier} - Remove a member**

Remove another member from the community, if you have permission, or your own blog

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/members/{blog-identifier} | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **blog-identifier** | String | The blog to remove, see [blog identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) | None | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **reason** | String | The reason for removing the member, see note below | None | Yes |
| **note** | String | Why the member is being removed, max length 1,000 characters, see note below | None | Depends |
| **ban\_user** | Boolean | Whether to also ban the user permenantly from rejoining | false | No |

The reason value can be:

* community\_guidelines for community guidelines violations.
* off\_topic if their content is off topic.
* spam if their content is considered spam.
* inappropriate\_behavior for inappropriate behavior in the community.
* tumblr\_guidelines for violating Tumblr's user guidelines.
* other for another reason -- **this requires the "note" parameter to be used as well**

**Response**

200 OK on success.

**Errors**

* 403 Forbidden for anyone trying to do this who isn't an admin of the community, or not trying to remove their own blog
* 400 Bad Request if an invalid reason is given, or if a note wasn't given when reason = 'other'

**PUT /v2/communities/{community-handle}/members/{blog-identifier} - Change a member's role**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/members/{blog-identifier} | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **blog-identifier** | String | The blog to change, see [blog identifiers](https://www.tumblr.com/docs/en/api/v2#blog-identifiers) | None | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **role** | String | Their new role, we currently support member, moderator, and admin | None | Yes |

**Response**

200 OK on success.

**Errors**

* 400 Bad Request if you provided an invalid role, or are trying to demote yourself as the last admin.
* 403 Forbidden for anyone trying to do this who isn't an admin of the community.

**DELETE /v2/communities/{community-handle}/mute - Unmute a community**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/mute | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Response**

200 OK on successful save.

**Errors**

* 403 Forbidden when you're trying to unmute a community you're not a member of.

**PUT /v2/communities/{community-handle}/mute - Mute a community**

Mute the community. Note that this only hides the unread count right now.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/mute | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Response**

200 OK on successful save.

**Errors**

* 403 Forbidden when you're trying to mute a community you're not a member of.

**PUT /v2/communities/{community-handle}/posts/{post-id}/reactions - Add reaction to post**

React to a specific post in a community.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/posts/{post-id}/reactions | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **post-id** | String/Number | The post ID to react to | None | Yes |

**Request Body Parameters**

The client must supply **one of these two** body parameters, and slug will take precedence if both are provided:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| **slug** | String | A specific slug to use, i.e. :thumbs\_up: |
| **grapheme** | String | A unicode emoji to use, i.e. 👍 |

When using **grapheme**, we'll make a best effort to use that specific unicode character, if we support it.

**Response**

200 OK on successful save. The API may return a timeline response with a celebration banner on the first reaction used, or nothing.

**Errors**

* 400 Bad Request with subcode 22001 - when the post has reached its overall reaction limit.
* 400 Bad Request with subcode 22002 - when the current user has reached their reaction limit on this post.
* 403 Forbidden with subcode 23002 - when you're trying to react to a post in a community you're not a member of.

**DELETE /v2/communities/{community-handle}/posts/{post-id}/reactions/{reaction-id} - Remove reaction from post**

Remove your reaction from a specific post in a community.

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/posts/{post-id}/reactions/{reaction-id} | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **post-id** | String/Number | The post ID to remove a reaction from | None | Yes |
| **reaction-id** | String | The unique identifier of the reaction to remove, or its grapheme | None | Yes |

**Response**

200 OK on successful removal.

**GET /v2/communities/{community-handle}/moderation/posts/{post-id} - View a moderated post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/moderation/posts/{post-id} | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **post-id** | String/Number | The moderated post ID to view | N/A | Yes |

**Response**

Information about a moderated post can only be viewed by admins/ mods of the community, and the author of the post. Information about which moderator performed the action will not be returned for the post author.

| **Field** | **Type** | **Description** | **Visibility** |
| --- | --- | --- | --- |
| **action** | String | The most recent moderation action taken on this post, eg. moderate\_post, restore\_post | Visible to mods and author |
| **reason** | String | The reason chosen when moderating a post, if the most recent action was a post removal | Visible to mods and author |
| **note** | String | An optional note from the moderator when they took the moderation action. Max length 1,000 characters | Visible to mods and author |
| **ts** | Number | The unix epoch timestamp of when the most recent moderation action occurred | Visible to mods and author |
| **post** | Object | The post that was moderated | Visible to mods and author |
| **moderator** | Object | The community moderator who performed the most recent moderation action | Visible to mods |

{

"meta": {

"status": 200,

"msg": "OK"

},

"response": {

"action": "moderate\_post",

"reason": "off\_topic",

"note": "This is a cat post and we are a dog community!",

"ts": 1298665620,

"post": { ... }, // blog object

"moderator": { ... }, // blog object

}

}

**Errors**

* 403 Forbidden if you are not allowed to view this
* 404 Not Found if the post was not found.

**DELETE /v2/communities/{community-handle}/moderation/posts/{post-id} - Moderate a post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/moderation/posts/{post-id} | DELETE | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **post-id** | String/Number | The post ID to moderate | N/A | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **reason** | String | The reason for removing the post, see note below | None | Yes |
| **note** | String | Why the post is being removed, max length 1,000 characters, see note below | None | Depends |

The reason value can be:

* community\_guidelines for community guidelines violations.
* off\_topic if their content is off topic.
* spam if their content is considered spam.
* inappropriate\_behavior for inappropriate behavior in the community.
* tumblr\_guidelines for violating Tumblr's user guidelines.
* other for another reason -- **this requires the "note" parameter to be used as well**

**Response**

204 No Content on success.

**Errors**

* 400 Bad Request if an invalid reason is given, or if a note wasn't given when reason = 'other'
* 403 Forbidden when the requester is not allowed to remove the post.
* 404 Not found
  + 404.24005 when the post was not found or has been deleted.
* 409 Conflict
  + 409.24006 if the post was already removed by a moderator

**PUT /v2/communities/{community-handle}/moderation/posts/{post-id} - Restore a post**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/moderation/posts/{post-id} | PUT | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |
| **post-id** | String/Number | The post ID to restore | N/A | Yes |

**Request Body Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **note** | String | Why the post is being restored, max length 1,000 characters | None | Yes |

**Response**

204 No Content on success.

**Errors**

* 403 Forbidden when the requester is not allowed to restore the post.
* 404 Not found
  + 404.24005 when the post was not found or has been deleted.
* 409 Conflict
  + 409.24007 if the post was already restored by a moderator
* 503 Service Unavailable when we've encountered a problem saving the restore action.

**GET /v2/communities/{community-handle}/timeline - Get posts from a community**

**Method**

| **URI** | **HTTP Method** | **Authentication** |
| --- | --- | --- |
| api.tumblr.com/v2/communities/{community-handle}/timeline | GET | [OAuth](https://www.tumblr.com/docs/en/api/v2#authentication) |

**Request Path Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **community-handle** | String | The community handle, see [Communities methods](https://www.tumblr.com/docs/en/api/v2#communities-methods) | None | Yes |

**Request Query Parameters**

| **Parameter** | **Type** | **Description** | **Default** | **Required?** |
| --- | --- | --- | --- | --- |
| **type** | String | The type of timeline to visit, see note below | recent | No |

We currently support timelines of type recent, polls, and discover.

**Response**

200 OK with a standard timeline response

**Errors**

* 403 Forbidden when you do not have permission to view this community, see the info endpoint for specific error codes.

Note that these docs are generated from [our public documentation on Github](https://github.com/tumblr/docs).