

Purpose

The purpose of the Pinboard API is twofold:

1. To give Pinboard users full programmatic access to their data.
2. To let people make cool stuff.

Anything that doesn't interfere with purpose #1 is likely to get my sunny approval.

Access

The API endpoint is <https://api.pinboard.in/v2/>.

Authentication

All requests require a valid username and API token. Users can find this information on their settings/password page.

There are two ways to pass the API token. Either include it as a query parameter:

```
https://api.pinboard.in/v1/method?auth_token=user:123
```

Or put it in an HTTP header:

```
X-Auth-Token: user:123
```

If both formats are present, Pinboard will ignore the HTTP header and use the query string.

App Identifiers

Third-party apps that make requests on behalf of Pinboard users are required to include an app identifier in every API request.

[link for registering App ID]

A 'third-party app' is something that users buy or install as a pre-compiled binary. It does not include scripts, libraries, or open-source projects that people compile themselves. If in doubt, ask me.

Just like the auth token, the app ID can go in the query string:

```
https://api.pinboard.in/v1/method?auth_token=user:123&app_id=pinception+1.3
```

Or in the HTTP headers:

```
X-App-ID: pinception 1.3
```

The purpose of the app ID is to help me better track and understand how the API is being used, so I can make improvements more effectively. It also gives me a point of contact if something goes wrong.

Encoding

All entities are encoded as UTF-8. When length limits are given, they refer to logical characters rather than bytes.

All arguments should be passed URL-encoded.

Where multiple arguments are allowed, they should be separated by URL-encoded whitespace ([apple+pear+orange](#))

Send an `Accept-Encoding: gzip` header if you can handle compressed output.

HTTP Verbs

If client bugs or other restrictions limit your ability to use `DELETE` or `PUT`, you can "tunnel" those methods using a `POST` request with an added HTTP header:

```
X-Http-Method: DELETE
```

Conditional Requests

Many API requests will return an `Etag` and `Last-Modified` header.

Developers are encouraged to use `If-Modified-Since` and `If-Match` headers in their requests where appropriate.

API requests that return a `304 Not Modified` response will not be counted against API rate limits.

Formats

All requests are returned as JSON, with MIME type `application/json`.

All responses have a top-level 'status' property. See the section on [error reporting](#) for how to check it.

```
{ status : "ok|not_ok",
  key1  : "...",
  key2  : "...",
}
```

HTTP Status Codes

The API uses HTTP error codes where appropriate. Clients must be ready to handle the following:

- 200 OK
- 201 Created
- 304 Not Modified
- 401 Unauthorized
- 404 Not Found
- 410 Gone
- 405 Method Not Allowed
- 429 Rate Limited ([see below](#))
- 503 Service Unavailable.

See the [method reference](#) for details. Methods that return something other than `200 OK` include HTTP headers in the example response.

Error Reporting

In many cases, the server will return a 200 OK response even when something goes wrong.

You must always check the JSON response.

A response with no errors has its 'status' field set to 'ok':

```
{ 'status' : 'ok',
  'foo' { 'bar' ....} }
```

A response with an error will have the 'status' field set to 'not_ok', and will have an 'error' field with a predefined code:

```
{ 'status' : 'not_ok',  
  'error' : 'bad_url',  
  'error_detail' : "....."  
  .... }
```

The possible values of 'error' are enumerated in the documentation for each API call.

'error_detail' is a free-form field that may contain more information.

Data Types

The Pinboard API recognizes the following data types:

tag	Up to 255 characters. May not contain commas or whitespace. Please be aware that tags beginning with a period are treated as private and trigger special private tag semantics .
url	URL as defined by RFC 3986 . Allowed schemes are http , https , javascript , mailto , ftp and file .
string	Arbitrary UTF-8 string.
Boolean	The value 0 or 1
integer	Integer in the range [0,65535]
id	Hexadecimal string. Not case sensitive.
datetime	Date and time in ISO 8601 format (2010-12-11T19:48:02Z).

User Rate Limits

The general rate limit is 400 API requests every fifteen minutes.

Requests that use conditional GET and return a 304 **do not** count against the rate limit.

API calls will return two headers:

```
X-Throttle-Remaining: 123  
X-Throttle-Reset: 423
```

The `X-Throttle-Reset` header indicates how many seconds remain until the next measuring period starts.

Rate limits can change without warning, so always check the headers.

If you hit the rate limit, you'll start to see [429 Too Many Requests](#) server errors, and should back off appropriately. If possible, keep doubling the interval between requests until you stop receiving errors.

Once a day, you are allowed to reset your rate limit by sending a `POST` request to [/site/amnesty/](#). This is intended to make life easier for developers; please don't use it in production.

The purpose of user rate limits is to keep the API as fast and useful as possible for everyone.

Sitewide Rate Limits

Requests to certain parts of the site are subject to an additional rate limit:

```
/url/*  
/site/recent/
```

```
/site/popular/  
/site/search/*  
/site/t:tag/*
```

You may make a total of 1000 requests per day to these URLs.

The limit is global (so making 400 requests to `/url/` + 500 requests to `/site/popular/` counts the same as making 900 requests to `/url/`).

This is an additional rate limit. Global requests are still subject to the time-based rate limits above.

If you need to make unusually heavy use of the API, or have a use case that makes these rate limits onerous, please [talk to me about it](#).

URL Normalization

All URLs submitted are normalized by default. This means:

- Shortened URLs get de-shortened
- Case-insensitive portions of the URL are converted to lower case
- `utc_*` and similar tracking barnacles are stripped from the query string.

If you do not want incoming URLs normalized, set the `exact_url` flag to true.

Privacy Semantics

There are two nonobvious behaviors to watch out for:

Users who have the [PRIVACY LOCK](#) setting enabled cannot save public bookmarks, or make their current bookmarks public. Attempts to do this will not raise an error, but the bookmark will be private.

Any tag beginning with a dot (.example) is treated as a private tag within Pinboard. Private tags are only visible to the user who created them. There's no way to override this behavior.

Restrictions

Your app or service can't pretend to be Pinboard, or allow users to form the impression that it's Pinboard.

You can't use the API to build up a data collection on your own service. This holds especially for sitewide data like search results, global tags, and URL metadata.

You can't use any data you get from this API for ad targeting or marketing.

You can't share user data you get from the API with outside parties. This stuff is between me, you, and our users.

Don't lie to the API server about who you are (by using a fake User-Agent string, for example).

If you want to use the API to assemble data for a research project, talk to me. Chances are I can save you some time.

If you distribute a library to interact with the API, it MUST check certificate validity with each API connection, even though that is annoying.

Support

To report bugs in the API or this documentation, please contact me at support@pinboard.in.

If you have an API feature idea, or complaint, please post it to the [pinboard-dev Google group](#) for discussion.

You can find me on Twitter as [@pinboard](#) and on IRC as [#pinboard](#) on freenode.