



Wiki

Pages & Files

VIEW

Twitter Search API Method: search

last edited by [Taylor Singletary](#) 12 months ago[Page history](#)[<< Back to Twitter API Documentation](#)

search

Returns tweets that match a specified query.

method status | [report a bug](#)

Notice: As of April 1st 2010, the Search API provides an option to retrieve "popular tweets" in addition to real-time search results. In an upcoming release, this will become the default and clients that don't want to receive popular tweets in their search results will have to explicitly opt-out. See the *result_type* parameter below for more information.

URL:

<http://search.twitter.com/search>.format

Formats:

json, atom

HTTP Method:

GET

Requires Authentication ([about authentication](#)):

false

API rate limited ([about rate limiting](#)):

1 call per request

Parameters:

- callback: Optional. **Only available for JSON format.** If supplied, the response will use the JSONP format with a callback of the given name.
 - Example: <http://search.twitter.com/search.json?callback=foo&q=twitter>
- lang: Optional. Restricts tweets to the given language, given by an [ISO 639-1 code](#).
 - Example: <http://search.twitter.com/search.atom?lang=en&q=devo>
- locale: Optional. Specify the language of the query you are sending (only ja is currently effective). This is intended for language-specific clients and the default should work in the majority of cases.
 - Example: <http://search.twitter.com/search.atom?q=東京大地震&locale=ja>
- max_id: Optional. Returns tweets with status ids less than the given id.
 - Example: http://search.twitter.com/search.atom?q=twitter&max_id=1520639490
- q: Optional. The text to search for. See the example queries section for examples of the syntax supported in this parameter
 - Example: <http://search.twitter.com/search.json?q=twitter>
- rpp: Optional. The number of tweets to return per page, up to a max of 100.
 - Example: <http://search.twitter.com/search.atom?q=devo&rpp=15>
- page: Optional. The page number (starting at 1) to return, up to a max of roughly 1500 results (based on rpp * page. Note: [there are pagination limits](#).
 - Example: <http://search.twitter.com/search.atom?q=devo&rpp=15&page=2>
- since: Optional. Returns tweets with since the given date. Date should be formatted as YYYY-MM-DD

Search this workspace

Navigator

[API Methods](#)[Pages](#) [No Files](#)[options](#)

SideBar

[FAQ](#)[Twitter API Documentation](#)[Getting Started](#)[API Changelog](#)[Libraries](#)[Tutorials](#)[Open source](#)

Recent Activity

[sabthan](#)removed by [Matt Harris](#)[Getting Started](#)edited by [Matt Harris](#)[API Overview](#)edited by [Matt Harris](#)[Things Every Developer Should Know](#)edited by [Matt Harris](#)[Twitter API Documentation](#)edited by [Matt Harris](#)[Twitter REST API Method: help test](#)edited by [Matt Harris](#)[Twitter REST API Method: help test](#)edited by [Matt Harris](#)[More activity...](#)

- Example: <http://search.twitter.com/search.atom?q=twitter&since=2010-02-28>
- `since_id`: Optional. Returns tweets with status ids greater than the given id.
 - Example: http://search.twitter.com/search.atom?q=twitter&since_id=1520639490
- `geocode`: Optional. Returns tweets by users located within a given radius of the given latitude/longitude. The location is preferentially taking from the Geotagging API, but will fall back to their Twitter profile. The parameter value is specified by "latitude,longitude,radius", where radius units must be specified as either "mi" (miles) or "km" (kilometers). Note that you cannot use the **near** operator via the API to geocode arbitrary locations; however you can use this `geocode` parameter to search near geocodes directly.
 - Example: <http://search.twitter.com/search.atom?geocode=40.757929%2C-73.985506%2C25km>
- `show_user`: Optional. When true, prepends "<user>:" to the beginning of the tweet. This is useful for readers that do not display Atom's author field. The default is false.
 - Example: http://search.twitter.com/search.atom?q=twitterapi&show_user=true
- `until`: Optional. Returns tweets with generated before the given date. Date should be formatted as YYYY-MM-DD
 - Example: <http://search.twitter.com/search.atom?q=twitter&until=2010-03-28>
- `result_type`: Optional. Specifies what type of search results you would prefer to receive.
 - Valid values include:
 - `mixed`: In a future release this will become the default value. Include both popular and real time results in the response.
 - `recent`: The current default value. Return only the most recent results in the response.
 - `popular`: Return only the most popular results in the response.
 - Example: http://search.twitter.com/search.atom?q=Twitter&result_type=mixed
 - Example: http://search.twitter.com/search.json?q=twitterapi&result_type=popular
 - Example: http://search.twitter.com/search.atom?q=justin+bieber&result_type=recent

Usage Notes:

- Query strings should be [URL encoded](#).
- Queries are limited 140 URL encoded characters.
- [Some users](#) may be absent from search results.
- The `since_id` parameter will be removed from the `next_page` element as it is not supported for pagination. If `since_id` is removed a warning will be added to alert you.
- This method will return an HTTP 404 error if `since_id` is used and is too old to be in the search index.
- If you are having trouble constructing your query, use the [advanced search form](#) to construct your search, then add the format. For example <http://search.twitter.com/search?q=twitter> would become <http://search.twitter.com/search.json?q=twitter>
- Applications must have a meaningful and unique User Agent when using this method. A HTTP Referrer is expected but not required. Search traffic that does not include a User Agent will be [rate limited](#) to fewer API calls per hour than applications including a User Agent string.
- After April 1st, 2010 we have a new feature for returning popular tweets in beta. After the beta period the value of `result_type=mixed` will become the default.

Example queries:

- Containing a word: <http://search.twitter.com/search.atom?q=twitter>
- From a user: <http://search.twitter.com/search.atom?q=from%3Aal3x>
- Replying to a user (tweet starts with @mzsanford): <http://search.twitter.com/search.atom?q=to%3Amzsanford>
- Mentioning a user (tweet contains @biz): <http://search.twitter.com/search.atom?q=%40biz>

- Containing a hashtag (up to 16 characters): <http://search.twitter.com/search.atom?q=%23haiku>
- Combine any of the operators together: <http://search.twitter.com/search.atom?q=happy+hour&until=2009-03-24>
- Originating from an application: <http://search.twitter.com/search.atom?q=landing+source:tweetie>

Search operators:

Most [search operators](#) can be used with API queries.

Boolean operators:

- OR to combine queries:
 - Mentioning @twitterapi OR @twitter: <http://search.twitter.com/search.atom?q=%40twitterapi+OR+%40twitter>
- Negation: place - in front of the operator.
 - Referencing a user but not from that user: <http://search.twitter.com/search.atom?q=dougw+-from%3Adougw>

Operator Limits:

- since: and until operators:
 - since: will return a HTTP 404 if it is too old
 - until: cannot be in the future
 - since: and until: do not accept the negation (-) operation
- filter:links operator:
 - results are limited to 7 days
- source: operator:
 - results are limited to 7 days
 - queries must contain at least one keyword parameter with the source: operator
- lang= operator:
 - results are limited to 7 days
- location operator:
 - results are limited to 7 days
- near:
 - cannot be used in API search
- geocode:
 - does not support disjunction (OR)

Response ([about return values](#)):

Warning: The user ids in the Search API are different from those in the REST API ([about the two APIs](#)). This defect is being tracked by [Issue 214](#). This means that the to_user_id and from_user_id field vary from the actual user id on Twitter.com. Applications will have to perform a screen name-based lookup with the [users/show](#) method to get the correct user id if necessary.

The metadata node will sometimes contain a result_type field with a value of either "recent" or "popular" -- though other values may be possible in the future. Popular results are derived by an algorithm that Twitter computes, and up to 3 will appear in "mixed mode" at the top of the result set. Popular results will also include another node to metadata called "recent_retweets" and will indicate how many retweets the tweet was bestowed recently. The metadata node will contain more fields as time goes on.

JSON example (truncated):

```
{ "results": [
  {
    "text": "@twitterapi http://tinyurl.com/ctrefg",
    "to_user_id": 396524,
    "to_user": "TwitterAPI",
    "from_user": "jkoum",
    "metadata": {
      {
        "result_type": "popular",
        "recent_retweets": 109
      }
    }
  },

```

```

      "id":1478555574,
      "from_user_id":1833773,
      "iso_language_code":"nl",
      "source":"<a href='\"http://twitter.com/\">twitter</a>",
      "profile_image_url":"http://
\\s3.amazonaws.com\\twitter_production\\profile_images\\118412707
\\2522215727_a5f07da155_b_normal.jpg",
      "created_at":"Wed, 08 Apr 2009 19:22:10 +0000",
      ... truncated ...,
      "since_id":0,
      "max_id":1480307926,
      "refresh_url":"?since_id=1480307926&q=%40twitterapi",
      "results_per_page":15,
      "next_page":"?page=2&max_id=1480307926&q=%40twitterapi",
      "completed_in":0.031704,
      "page":1,
      "query":"%40twitterapi"
    }
  }
}

```

Refreshing Search Results

For those using client side search widgets, by default the first request might include popular results. If you want to display these you can use the `result_type` attribute to visually differentiate them. If you don't want to display these you can always just pass the `"result_type"` parameter with a value of `"recent"` along with your request and they'll never be included.

In the case that you are displaying them, you may not want to not redisplay them every time your widget refreshes. Each result payload will include the `since_id` that you should use in subsequent requests to get the latest results. If you are using this `since_id` (which all compliant search widgets should) then the popular results will be excluded in subsequent requests. The specific `since_id` value to pass up to the query string is included in the payload. Here are examples in various formats:

JSON

```
"refresh_url":"?since_id=9290798834&q=i"
```

atom

```

<link type="application/atom+xml"
href="/http://search.twitter.com/search.atom?
q=i&since_id=9290738270" rel="refresh"/>

```

rss

```

<twitter:refresh_url>http://search.twitter.com/search.rss?
q=i&since_id=9290775688</twitter:refresh_url>

```

Usage examples:

cURL ([about cURL](#)):

```
curl http://search.twitter.com/search.json?q=%40twitterapi
```

Do you have an example to share in the language of your choice? [Please share!](#)

[<< Back to Twitter API Documentation](#)

Comments (0)

You don't have permission to comment on this page.

[PDF version](#)