News

QUERY BUILDER SEARCH API DOCS LIVEFEED API DOCS

Table of Contents

Overview

The News Search API is an HTTP API to search in our news index.

API Endpoint

Endpoint for fetching documents from the API, all queries are passed to a single search endpoint that accepts POST requests with JSON bodies and responds with JSON.

HTTP options

- Base URL: `https://data.twingly.net/news/b/search/v1/search`
- HTTP method: `POST`
- TLS SNI client support required for HTTPS
- HTTP compression available

Required headers

- `Authorization: apikey <API_KEY>`
- `Content-Type: application/json; charset=utf-8`

- `Accept: application/json; charset=utf-8`

Search query syntax

A query is described with a JSON object, the following keywords exist:

- `all` (*array*) - All words or phrases provided have to exist in a result posting. Max 250 words or phrases in `all`, `any` and `none` combined.
  - **Note:** `all` allows nesting, see section nested boolean operators.
- `any` (*array*) - At least one word or phrase provided has to exist in a result posting. Max 250 words or phrases in `all`, `any` and `none` combined.
  - **Note:** `any` allows nesting, see section nested boolean operators.
- `none` (*array*) - None of the word or phrase provided may exist in a result posting. Max 250 words or phrases in `all`, `any` and `none` combined.
- `languages` (*array*) - Restricts results to the given languages. ISO 639-1 language code expected. By default all languages are matched.
- `locations` (*array*) - Restricts results to the given locations. ISO 3166-1 alpha-2 country code expected. By default all locations are matched.
- `authors` (*array*) - Restricts results to the given authors. Max 10 authors in `authors`.
- `sort` (*string*) - How to sort the results, possible values are: `timestamp`. Default is `timestamp`.
- `order` (*string*) - In which order to sort the results, possible values are: `desc`. Default is `desc`.
- `timestamp` (*object*)
  - `since` (*string*) - Return results dated after this date. Default is "now" minus 7 days. No more than 30 days before "now" is allowed for this parameter. Expected format is `YYYY-MM-DDThh:mm:ssZ` (UTC ISO 8601).
  - `until` (*string*) - Return results dated up to this date. Default is "now". Must be greater than `since`. Expected format is `YYYY-MM-DDThh:mm:ssZ` (UTC ISO 8601).
- `size` (*integer*) - How many documents to return in the response. Max 250. Default is 10.
- `url` (*string*) - Restrict results to sites or articles matching the given URL. See the section about URL matching below.
- `domains` (*array*) - Restrict results to sites or articles matching the given domains. Max 10 domains in `domains`. See the section about URL matching below.
- `site_id` (*integer*) - Restrict results to the site with the given `site_id`.
- `article_ids` (*array*) - Restrict results to articles with the given `article_id`s. Must be used together with `site_id`. Max 250 IDs in `article_ids`.
- `article_readership` (*boolean*) - If true, `number_of_potential_readers` and `site_average_number_of_potential_readers` will be calculated and returned for each article in the response. **Note:** This parameter requires the readership feature to

be enabled for your API key. To learn more about this feature, contact sales@twingly.com.

- `article_topics_and_entities` (*boolean*) If true, `topics` and `entities` will be included in the response field. **Note:** This parameter requires the topics and entities feature to be enabled for your API key. To learn more about this feature, contact sales@twingly.com.
- `group_identical_documents` (*boolean*) If true, articles that we consider to be identical will be grouped inside the `identical_documents` response field, instead of being delivered as separate documents.
- `update_search` (*boolean*) - If true, tells our system to only return articles which has been indexed since the last time the same query was run. **Note:** This is an experimental feature and it might not work for all cases. See the update search section for more details.

Requirement: The query needs to contain at least one of: `all`, `any`, `authors`, `site_id` and `url`.

Note: The search query syntax can be extended with additional new keywords and functionality without notice. Additions will be noted in the changelog when introduced. Keywords will never be removed, all introduced fields will exist as long the API version (`v1`) exist.

The default query values may change without notice, we recommend specifying any keywords that you rely on in production.

Keywords and phrases

- `all`, `any` and `none` accept an array of search keywords and phrases.
- To search for an exact phrase
  - `"all": ["This will match the exact phrase"]`
  - `"all": ["And this will", "match two exact phrases"]`
- To search for exact keywords
  - `"all": ["This", "will", "match", "all", "words", "but", "order", "does", "not", "matter"]`
- Quotes (`"`) are removed, these have no effect on the query, all searches are exact unless it's a wildcard search.

Nested boolean operators

The operators `all` and `any` support nesting. When an array is placed inside `all`, the `any` operator is applied to the elements of the inner array. Similarly, when an array is placed inside `any`, the `all` operator is applied to the elements of the inner array.

- The query `"all": [["A", "B"], ["C", "D"]]` would essentially translate into `(A OR B) AND (C OR D)`.
- The query `"any": [["A", "B"], ["C", "D"]]` would essentially translate into `(A AND B) OR (C AND D)`.

Note: The query builder does not support this feature, and it is not possible to use more than one level of nesting.

Wildcard searches

It is possible to use an asterisk (`*`) at the end of a word or phrase to match any character. This is available for the `all`, `any`, `none` and `authors` filters.

Examples:

- `micro*` - Matches any words starting with `micro`, e.g. `microsoft`, `microscope`, `micros` etc
- `carl bil*` - Matches any phrase starting with `carl bil` e.g. `carl bildt`, `carl biltong` etc

Limitations:

- Must be combined with words of at least three characters, otherwise the `*` will be silently ignored.
  - `th*` - Does not work, `*` will be ignored, just matches `th`.
- May only be used in end of a word, not possible to use it in the beginning or middle of a word.
  - `mic*soft` - Does not work, `*` will be ignored, just matches `micsoft`.

Matching

Searches are matched against the following document fields:

- `title`
- `text`
- `summary`
- `author`
- `image.caption`

See the response example or read the response fields documentation below for more information about what each field contains.

Case-sensitivity

Searches match case-sensitively, per keyword/phrase, if there are any uppercase letters in the given keyword/phrase, case-insensitively otherwise.

Examples:

- `microsoft` will match all of: `microsoft`, `Microsoft`, `MICROSOFT`
- `Microsoft` will match only: `Microsoft`

URL matching

By using the `url` or `domains` keywords in the query, you can find articles by their URL. The difference between the two keywords is that `url` searches in the whole article URL, including the path and query parameters, while `domains` only searches in the domain part of the URL.

Examples:

- `{ "url": "nytimes.com" }` - Return articles where the URL includes `nytimes.com`
- `{ "url": "nytimes.com/2020" }` - Return articles where the URL includes `nytimes.com/2020`
- `{ "url": "microsoft" }` - Return articles where the URL includes `microsoft`
- `{ "domains": ["nytimes.com"] }` - Return articles where the domain part of the URL includes `nytimes.com`
- `{ "domains": ["bbc.com", "cnn"] }` - Return articles where the domain part of the URL includes either `bbc.com` or `cnn`
- `{ "domains": ["bbc.com"], "url": "politics" }` - Return articles where the domain part of the URL includes `bbc.com`, and `politics` is mentioned somewhere in the URL

Limitations:

- Using `url` to search for `nytimes.com` will both match articles from `nytimes.com` and articles where `nytimes` and `com` is included anywhere else in the URL, such as `example.com/nytimes_com_news`. To only match the domain part, switch to using the `domains` keyword instead.
- When using `url`, the URL scheme is automatically removed from the URL in your query, as that part of the URL is not indexed. It is therefore not possible to find all articles that starts with `https://` for example.
- During indexing, the URL is split into words, and special characters are removed. This means that characters such as `.` and `/` are not indexed, so searching for `bbc.com/news` and `bbc com news` will yield the same results. This is true for both `url` and `domains`.

Update search

Setting the `update_search` parameter in the query to `true`, tells our system to keep track of which articles has been returned for that query, so that subsequent times you run the

same query (with the parameter set to true), you'll only get new articles indexed since last time.

When we encounter a query with `update_search` enabled for the first time, our system will start setting up that query for monitoring in our backend. During the first few minutes, the query will return hits just as normal, but after the initial warm-up phase is over, any subsequent calls to the endpoint will only return new hits, excluding the articles that you've gathered before.

As the point of this parameter is to only return new hits, any timestamp filters added to the query will be silently ignored once the initial warm-up phase is over.

Note: Due to performance reasons, this parameter only works for queries returning a few hundred hits per day or less. Queries matching thousands of articles, will just be treated as a normal query. Also, if the query isn't issued at least once a day, our system will stop tracking which articles has been returned for the query, and you'll have to start over the warm-up phase described above.

Usage

Best practices

A list of best practices can be found in the best practices section.

Pagination

Pagination is done through a moving time window of `timestamp.since` and `timestamp.until`.

With a descending time order of documents, a pagination algorithm could look like this:

- Initial query:
  - Choose a starting timestamp, `timestamp.until`.
  - Set a max max response size, `size`.
- Look at the results
  - If the response documents count is less than the max size, you got all documents and you're done.
  - If not, look at the response `timestamp` of the last/oldest document and update your query's `timestamp.until` to one second before `timestamp`
  - Repeat.

If there are more documents on the same second than the max size limit, you need to limit the response by adding more filters or skip to the next second (potentially missing some documents).

Limits

- Rate limit: 5 requests per second, per API key.
- Max request body size: 16 KiB
- All request query values have defined limits, see Search query syntax

Request examples

To request 10 articles mentioning "solar eclipse" (exact phrase) and "accident", but not "trump" in the US since `2017-08-20T20:00:00Z` a query could look like this:

```
{
  "all": ["solar eclipse", "accident"],
  "none": ["trump"],
  "locations": ["us"],
  "size": 10,
  "timestamp": {
    "since": "2017-08-20T20:00:00Z"
  }
}
```

curl

Complete curl example, replace `<APIKEY>` with your issued API key.

```
curl -v -s \
  --request POST \
  --header "Authorization: apikey <APIKEY>" \
  --header "Content-Type: application/json; charset=utf-8" \
  --header "Accept: application/json; charset=utf-8" \
  --data '{"all": ["solar eclipse", "accident"], "none": ["trump"], "locatic
  "https://data.twingly.net/news/b/search/v1/search"
```

Python

Complete Python example using the Requests library. Replace `<APIKEY>` with your issued API key.

```
import requests

url = "https://data.twingly.net/news/b/search/v1/search"

headers = {"Authorization": "apikey <APIKEY>",
```

```
            "Content-Type":"application/json; charset=utf-8",
            "Accept": "application/json; charset=utf-8"}

data = {"all": ["solar eclipse", "accident"],
        "none": ["trump"],
        "locations": ["us"],
        "size": 10,
        "timestamp": {"since": "2017-08-20T20:00:00Z"}}

requests.post(url, json = data, headers = headers)
```

Response example

The example has been edited for brevity.


```
{
  "number_of_documents": 1,
  "number_of_documents_estimated_total": 543297,
  "documents": [
    {
      "article_id": 147279,
      "url": "https://www.tradewindsnews.com/cruise-and-ferry/legendary-ocea
      "title": "Legendary ocean liner Queen Mary available for ‘charter' unt
      "text": "<p> Singapore-listed Eagle Hospitality Trust is set to auctio
      "summary": "  Leaseholder wants out of what is arguably the most famou
      "author": "Jonathan Boonzaier",
      "timestamp": "2021-03-12T08:26:12Z",
      "location_code": "no",
      "language_code": "en",
      "site_id": 297,
      "site_name": "TradeWinds",
      "site_url": "http://www.tradewindsnews.com",
      "images": [
        {
          "url": "https://images-global.nhst.tech/image/dGJKbnBMSmk5R1E0SkRo
          "caption": "<p>Eagle Hospitality Trust planned to redevelop the la
        }
      ],
      "article_is_paywalled": false,
      "article_has_full_text": true,
      "readership": {
        "site_number_of_monthly_visits": null,
        "site_average_number_of_potential_readers": 22121,
```

```
        "number_of_potential_readers": 15
      },
      "topics_and_entities": {
        "topics": [
          {
            "label": "economy, business and finance>economic sector>transpo
            "mediatopic_id": "20000343",
            "score": 0.6858
          }
        ],
        "location_entities": [
          {
            "entity": "Long Beach, California",
            "wikidata_id": "Q16739"
          }
        ],
        "organization_entities": [
          {
            "entity": "FTI Consulting",
            "wikidata_id": "Q5427196"
          }
        ],
        "person_entities": [
          {
            "entity": "Alan Tantleff",
            "wikidata_id": null
          }
        ]
      },
      "site_origin": {
        "state": "CT",
        "city": "Stamford"
      },
      "identical_documents": []
    }
  ]
}
```


Response fields

Note: The response can be extended with additional new fields without notice. Additions will be noted in the changelog when introduced. Fields will never be removed, all

introduced fields will exist as long the API version (`v1`) exist.

- `number_of_documents` (*integer*) - The number of documents in the response.
- `number_of_documents_estimated_total` (*integer*) - Estimate of the total number of documents that matched the query.
- `documents` (*array*) - Array holding documents (news articles).
  - `article_id` (*integer*) - Together, the attributes `site_id` and `article_id` uniquely identifies the article.
  - `url` (*string*) - Source URL of the article.
  - `title` (*string*) - The article header.
  - `summary` (*string*) - The article lead.
  - `text` (*string*) - The text of the article. **Note**: `summary` + `text` should be considered the complete article text.
  - `author` (*string*) - The author of the article, if found.
  - `timestamp` (*string*) - ISO 8601 UTC timestamp, generally, indicating when the article was indexed. If the difference between the indexing date and the detected publishing date is greater than a certain threshold (usually two days), then the publishing date will be used instead. This allows older articles to be ignored, if needed.
  - `location_code` (*string/null*) - the ISO 3166-1 alpha-2 country code for this article (based on the original source of the article).
  - `language_code` (*string*) - the ISO 639-1 language code representing the language that the article was written in.
  - `site_id` (*integer*) - Together, the attributes `site_id` and `article_id` uniquely identifies the article.
  - `site_name` (*string*) - The name of the site where this article was published.
  - `site_url` (*string*) - The URL to the site where this article was published.
  - `images` (*array*) - Array holding images found in the article.
    - `url` (*string/null*) - URL to the image file.
    - `caption` (*string/null*) - The image caption, if one was found.
  - `article_is_paywalled` (*boolean*) - Boolean indicating whether the article was extracted from behind a paywall or not. See the paywalls page for more information.
  - `article_has_full_text` (*boolean*) - Boolean indicating whether the full text of the article was delivered or not. See the paywalls page for more information.
  - `readership` (*object*) - **Note:** The readership fields below will be empty unless your API key has the readership feature enabled. To include the numbers for potential readers, set `article_readership` to true in the search query. To learn more about this feature, contact sales@twingly.com.
    - `site_number_of_monthly_visits` ~~(*integer*) - The number of visits the site has per month.~~ **Note:** This value is no longer supported and will always be `null` .

        - `site_average_number_of_potential_readers` (*integer*) - The average number of potential readers for articles on the site.
        - `number_of_potential_readers` (*integer*) - The number of potential readers for the article. **Note:** This value can be highly inaccurate for recently published articles, as it is calculated based on where on a site the article was featured and for how long, amongst other things.
    - `topics_and_entities` (*object*) - **Note:** The topics_and_entities fields below will be empty unless your API key has the topics_and_entities feature enabled. To learn more about this feature, contact sales@twingly.com.
        - `topics` (*array*) - Topics found in the article.
            - `label` (*string*) - IPTC Media topic name. Topic names are hierarchically structured, each level separated by a >, and with the levels ordered from the most general to the most specific. Each label falls under a mediatopic_id.
            - `mediatopic_id` (*string*) - IPTC Media Code. The mediatopic_id is linked to a group of topics.
            - `score` (*float*) - Value representing the strength in confidence of the topic. Ranging between 0 and 1.
        - `location_entities` (*array*) - Location entities found in the article.
            - `entity` (*string*) - Name of the location.
            - `wikidata_id` (*string/null*) - Id in the Wikidata knowledge database.
        - `organization_entities` (*array*) - Organization entities found in the article.
            - `entity` (*string*) - Name of the organization.
            - `wikidata_id` (*string/null*) - Id in the Wikidata knowledge database.
        - `person_entities` (*array*) - Person entities found in the article.
            - `entity` (*string*) - Name of the person.
            - `wikidata_id` (*string/null*) - Id in the Wikidata knowledge database.
    - `site_origin` (*object*) - **Note:** The site_origin fields below will be empty unless your API key has the site_origin feature enabled. To learn more about this feature, contact sales@twingly.com.
        - `state` (*string/null*) - The state of origin of the site where this article was published.
        - `city` (*string/null*) - The city of origin of the site where this article was published.
    - `identical_documents` (*array*) - Array holding documents (news articles) identical to this one.

Note about HTML

The following response fields might contain HTML tags or HTML entities. Depending on your use case you may need to decode them. Otherwise, for example, `&` will look like

`&amp;`.

- `text`
- `summary`
- `title`
- `author`
- `caption`

Known issues

- If you are getting duplicate articles from two different sites, and the `site_name` for the two sites are "<Name of publisher>" and "<Name of publisher> VIP" respectively, it is safe to ignore all articles from one of them. The "VIP" site is used to deliver content that requires special licensing.

Error Codes

HTTP status code	Description
400	Invalid query. This happens if the query is missing or invalid.
401	Invalid API key. This happens if the API key is missing or invalid.
429	Rate Limit Exceeded. This indicates too many requests per rate limit window, the client should back off and reduce the requests per window.
500	Internal Server Error. Unexpected conditions were encountered, indicating a server-side bug.
503	Service Unavailable. Indicates temporary problem, please try again.

Available languages

See the list of available languages.

Available locations

See the list of available locations.

FAQ

Is it possible to add sources?

If you have noticed that we are not currently covering a particular news site, please let us know by filling out our Request Sources form. We will do our best to ensure that the site is indexed in the future. Note that adding new sources requires some manual work and comes with a small fee. For more information, please send an email to sales@twingly.com.

Is it possible to backfill old articles when a new site gets added?

No, unfortunately that is not possible. We only index new articles published after we have added the site.

Is it possible to re-index articles?

It is possible, though this isn't something that we normally do. We can reindex certain articles, or articles for a certain period back in time for a site, with some volume restrictions.

Do you update articles?

Not generally, but if the same article is found on another URL than originally, it may be updated.

Why isn't the full article available in some cases?

In some cases, if an article happen to be hidden behind a paywall, but there is some free content available, the free (partial) content will be indexed.

How are country codes assigned to a site?

This is done manually when we add a new site. It is determined by the domain, whois information or any contact details found on the site itself.

Is it possible to access articles behind paywalls?

See more information about paywalls.

Changelog

- **2023-10** Extended the API response with the `identical_documents` respone field and added the `group_identical_documents` query keyword.
- **2023-09** Extended the API query with the `domains` keyword.
- **2023-08** Extended the API query with the `update_search` keyword.
- **2023-08** Extended the API query with the `authors` keyword.
- **2023-06** Added support for nested boolean operators.
- **2023-03** Added support for searching on the following characters: `=`, `/`, `:`, `.` and `_`.
- **2021-12** Extended the API response with `article_is_paywalled` and `article_has_full_text` response fields.
- **2021-11** Extended the API response with `number_of_documents_estimated_total` response field.
- **2021-09** Clarified that the `readership.site_number_of_monthly_visits` response field has been discontinued, and will from now on always contain `null`.
- **2021-08** Extended the API response with `site_origin` response fields.
- **2021-03** Added location codes for Netherlands Antilles (`an`) and Kosovo (`xk`).
- **2021-03** Extended the API response with `topics_and_entities` response fields.
- **2020-09** Extended the API query with the `url` keyword.
- **2020-08** Extended the API query with the `site_id` and `article_ids` keywords.
- **2020-07** Increased the maximum allowed value for `timestamp.since` to "now-30 days" instead of 7 days.
- **2020-06** Extended the API response with the `readership` response fields and added the `article_readership` query keyword.
- **2020-03** Added information to the FAQ about adding sources.
- **2020-01** Added information to the FAQ about articles behind paywalls.
- **2019-10** Added information to the FAQ about partial articles.
- **2019-10** Added information to the FAQ about articles being updated in certain circumstances.
- **2019-03** Added case-sensitivity information to the matching section.
- **2019-02** Added support for searching on the following characters: `&`, `®` `©` and `°`.
- **2018-12** Added data types for all values in the search query.
- **2018-08** Corrected the documentation of the `timestamp` response field to reflect that it is primarily based on indexing time, not publishing time.
- **2018-04** Added a FAQ section.
- **2018-03** Added a known issues section, with a note about duplicate articles.
- **2017-11** Clarified that the API uses ISO 639-1 two-letter language codes.
- **2017-11** The default number of documents returned by the API was changed from 250 to 10.
- **2017-10** Documented which fields are searched when querying the API.
- **2017-10** Extended the API response with `images`.

- **2017-09** Add documentation section about keywords and phrases.
- **2017-09** The query now require at least one item in `all` or `any`.
- **2017-09** Fixed wildcard documentation, not possible to use wildcard in beginning of a word.
- **2017-09** Extended the API response with `author`.
- **2017-09** Extended the API response with `site_name` and `site_url`.
- **2017-08** Introduced the API.

Languages

Languages accepted by the News Search API, and returned by News Search API and News LiveFeed API.

Note:

- Presence in the list below doesn't necessarily mean that there exists documents in that language.
- Values with regional designators (e.g. `en-US`, `fr_CA` etc.) are accepted but only the language part is considered.

Name	ISO 639-1 two-letter code
Afar	aa
Abkhazian	ab
Afrikaans	af
Akan	ak
Albanian	sq
Amharic	am
Arabic	ar
Aragonese	an
Armenian	hy

Name	ISO 639-1 two-letter code
Assamese	as
Avaric	av
Avestan	ae
Aymara	ay
Azerbaijani	az
Bashkir	ba
Bambara	bm
Basque	eu
Belarusian	be
Bengali	bn
Bihari languages	bh
Bislama	bi
Bosnian	bs
Breton	br
Bulgarian	bg
Burmese	my
Catalan	ca
Chamorro	ch

Name	ISO 639-1 two-letter code
Chechen	ce
Chinese	zh
Church Slavic	cu
Chuvash	cv
Cornish	kw
Corsican	co
Cree	cr
Czech	cs
Danish	da
Divehi	dv
Dutch	nl
Dzongkha	dz
English	en
Esperanto	eo
Estonian	et
Ewe	ee
Faroese	fo
Fijian	fj

Name	ISO 639-1 two-letter code
Finnish	fi
French	fr
Western Frisian	fy
Fulah	ff
Georgian	ka
German	de
Gaelic	gd
Irish	ga
Galician	gl
Manx	gv
Greek, Modern (1453-)	el
Guarani	gn
Gujarati	gu
Haitian	ht
Hausa	ha
Hebrew	he
Herero	hz
Hindi	hi

Name	ISO 639-1 two-letter code
Hiri Motu	ho
Croatian	hr
Hungarian	hu
Igbo	ig
Icelandic	is
Ido	io
Sichuan Yi	ii
Inuktitut	iu
Interlingue	ie
Interlingua (International Auxiliary Language Association)	ia
Indonesian	id
Inupiaq	ik
Italian	it
Javanese	jv
Japanese	ja
Kalaallisut	kl
Kannada	kn
Kashmiri	ks

Name	ISO 639-1 two-letter code
Kanuri	kr
Kazakh	kk
Central Khmer	km
Kikuyu	ki
Kinyarwanda	rw
Kirghiz	ky
Komi	kv
Kongo	kg
Korean	ko
Kuanyama	kj
Kurdish	ku
Lao	lo
Latin	la
Latvian	lv
Limburgan	li
Lingala	ln
Lithuanian	lt
Luxembourgish	lb

Name	ISO 639-1 two-letter code
Luba-Katanga	lu
Ganda	lg
Macedonian	mk
Marshallese	mh
Malayalam	ml
Maori	mi
Marathi	mr
Malay	ms
Malagasy	mg
Maltese	mt
Mongolian	mn
Nauru	na
Navajo	nv
Ndebele, South	nr
Ndebele, North	nd
Ndonga	ng
Nepali	ne
Norwegian Nynorsk	nn

Name	ISO 639-1 two-letter code
Bokmål, Norwegian	nb
Norwegian	no
Chichewa	ny
Occitan (post 1500)	oc
Ojibwa	oj
Oriya	or
Oromo	om
Ossetian	os
Panjabi	pa
Persian	fa
Pali	pi
Polish	pl
Portuguese	pt
Pushto	ps
Quechua	qu
Romansh	rm
Romanian	ro
Rundi	rn

Name	ISO 639-1 two-letter code
Russian	ru
Sango	sg
Sanskrit	sa
Sinhala	si
Slovak	sk
Slovenian	sl
Northern Sami	se
Samoan	sm
Shona	sn
Sindhi	sd
Somali	so
Sotho, Southern	st
Spanish	es
Sardinian	sc
Serbian	sr
Swati	ss
Sundanese	su
Swahili	sw

Name	ISO 639-1 two-letter code
Swedish	sv
Tahitian	ty
Tamil	ta
Tatar	tt
Telugu	te
Tajik	tg
Tagalog	tl
Thai	th
Tibetan	bo
Tigrinya	ti
Tonga (Tonga Islands)	to
Tswana	tn
Tsonga	ts
Turkmen	tk
Turkish	tr
Twi	tw
Uighur	ug
Ukrainian	uk

Name	ISO 639-1 two-letter code
Urdu	ur
Uzbek	uz
Venda	ve
Vietnamese	vi
Volapük	vo
Welsh	cy
Walloon	wa
Wolof	wo
Xhosa	xh
Yiddish	yi
Yoruba	yo
Zhuang	za
Zulu	zu

Locations

Locations available in News Search API and News LiveFeed API.

Country	ISO 3166-1 alpha-2 code
Andorra	ad

Country	ISO 3166-1 alpha-2 code
United Arab Emirates	ae
Afghanistan	af
Antigua and Barbuda	ag
Anguilla	ai
Albania	al
Armenia	am
Netherlands Antilles	an
Angola	ao
Argentina	ar
American Samoa	as
Austria	at
Australia	au
Aruba	aw
Azerbaidjan	az
Bosnia-Herzegovina	ba
Barbados	bb
Bangladesh	bd
Belgium	be

Country	ISO 3166-1 alpha-2 code
Burkina Faso	bf
Bulgaria	bg
Bahrain	bh
Burundi	bi
Benin	bj
Bermuda	bm
Brunei Darussalam	bn
Bolivia	bo
Brazil	br
Bahamas	bs
Bhutan	bt
Botswana	bw
Belarus	by
Belize	bz
Canada	ca
Cocos (Keeling) Islands	cc
The Democratic Republic of the Congo	cd
Central African Republic	cf

Country	ISO 3166-1 alpha-2 code
Congo	cg
Switzerland	ch
Côte d'Ivoire (Ivory Coast)	ci
Cook Islands	ck
Chile	cl
Cameroon	cm
China	cn
Colombia	co
Costa Rica	cr
Cuba	cu
Cape Verde	cv
Christmas Island	cx
Cyprus	cy
Czech Republic	cz
Germany	de
Djibouti	dj
Denmark	dk
Dominica	dm

Country	ISO 3166-1 alpha-2 code
Dominican Republic	do
Algeria	dz
Ecuador	ec
Estonia	ee
Egypt	eg
Western Sahara	eh
Eritrea	er
Spain	es
Ethiopia	et
Finland	fi
Fiji	fj
Falkland Islands	fk
Micronesia	fm
Faroe Islands	fo
France	fr
Gabon	ga
United Kingdom	gb
Grenada	gd

Country	ISO 3166-1 alpha-2 code
Georgia	ge
French Guyana	gf
Ghana	gh
Gibraltar	gi
Greenland	gl
Gambia	gm
Guinea	gn
Guadeloupe (French)	gp
Equatorial Guinea	gq
Greece	gr
Guatemala	gt
Guam (USA)	gu
Guinea Bissau	gw
Guyana	gy
Hong Kong	hk
Honduras	hn
Croatia	hr
Haiti	ht

Country	ISO 3166-1 alpha-2 code
Hungary	hu
Indonesia	id
Ireland	ie
Israel	il
Isle of Man	im
India	in
British Indian Ocean Territory	io
Iraq	iq
Iran	ir
Iceland	is
Italy	it
Jersey	je
Jamaica	jm
Jordan	jo
Japan	jp
Kenya	ke
Kyrgyz Republic (Kyrgyzstan)	kg
Cambodia	kh

Country	ISO 3166-1 alpha-2 code
Kiribati	ki
Comoros	km
Saint Kitts & Nevis Anguilla	kn
North Korea	kp
South Korea	kr
Kuwait	kw
Cayman Islands	ky
Kazakhstan	kz
Laos	la
Lebanon	lb
Saint Lucia	lc
Liechtenstein	li
Sri Lanka	lk
Liberia	lr
Lesotho	ls
Lithuania	lt
Luxembourg	lu
Latvia	lv

Country	ISO 3166-1 alpha-2 code
Libya	ly
Morocco	ma
Monaco	mc
Moldavia (Moldova)	md
Montenegro	me
Madagascar	mg
Marshall Islands	mh
Macedonia	mk
Mali	ml
Myanmar	mm
Mongolia	mn
Macau	mo
Northern Mariana Islands	mp
Martinique (French)	mq
Mauritania	mr
Montserrat	ms
Malta	mt
Mauritius	mu

Country	ISO 3166-1 alpha-2 code
Maldives	mv
Malawi	mw
Mexico	mx
Malaysia	my
Mozambique	mz
Namibia	na
New Caledonia (French)	nc
Niger	ne
Norfolk Island	nf
Nigeria	ng
Nicaragua	ni
Netherlands	nl
Norway	no
Nepal	np
Nauru	nr
Niue	nu
New Zealand	nz
Oman	om

Country	ISO 3166-1 alpha-2 code
Panama	pa
Peru	pe
Polynesia (French)	pf
Papua New Guinea	pg
Philippines	ph
Pakistan	pk
Poland	pl
Saint Pierre and Miquelon	pm
Pitcairn Island	pn
Puerto Rico	pr
Palestine	ps
Portugal	pt
Palau	pw
Paraguay	py
Qatar	qa
Reunion (French)	re
Romania	ro
Serbia	rs

Country	ISO 3166-1 alpha-2 code
Russian Federation	ru
Rwanda	rw
Saudi Arabia	sa
Solomon Islands	sb
Seychelles	sc
Sudan	sd
Sweden	se
Singapore	sg
Saint Helena	sh
Slovenia	si
Svalbard and Jan Mayen Islands	sj
Slovak Republic	sk
Sierra Leone	sl
San Marino	sm
Senegal	sn
Somalia	so
Suriname	sr
Saint Tome (Sao Tome) and Principe	st

Country	ISO 3166-1 alpha-2 code
El Salvador	sv
Syria	sy
Swaziland	sz
Turks and Caicos Islands	tc
Chad	td
Togo	tg
Thailand	th
Tadjikistan	tj
Tokelau	tk
East Timor	tl
Turkmenistan	tm
Tunisia	tn
Tonga	to
Turkey	tr
Trinidad and Tobago	tt
Tuvalu	tv
Taiwan	tw
Tanzania	tz

Country	ISO 3166-1 alpha-2 code
Ukraine	ua
Uganda	ug
United States	us
Uruguay	uy
Uzbekistan	uz
Holy See (Vatican City State)	va
Saint Vincent & Grenadines	vc
Venezuela	ve
Virgin Islands (British)	vg
Virgin Islands (USA)	vi
Vietnam	vn
Vanuatu	vu
Wallis and Futuna Islands	wf
Samoa	ws
Kosovo	xk
Yemen	ye
Mayotte	yt
South Africa	za

Country	ISO 3166-1 alpha-2 code
Zambia	zm
Zimbabwe	zw

Paywalls

To access news articles behind paywalls, you need to supply Twingly with the following:

- Credentials (e.g. username and password) for the newspaper site in question
- Proof of approval from the newspaper to collect the articles behind the paywall

If you are interested in getting articles behind paywalls delivered, and can fulfill the above requirements, please contact us for more information.

Receive titles of articles behind paywalls

Articles which have been extracted from behind a paywall on behalf of a customer, will normally only be made available through the APIs for that customer.

However, it's possible for you to get access to the title of articles found behind paywalls, even if you don't have access to the paywalled article. If you enable this feature, it will be possible for you to search in the full text of articles found behind a paywall, and get the title of the article back in the API response. To learn more about this feature, contact sales@twingly.com.

Determine if an article was found behind a paywall or not

If you provide us with credentials for a site, or if you enable the above mentioned feature, you'll get paywalled articles delivered back in the API results. To be able to determine if the article was found behind a paywall or not, you can use the `article_is_paywalled` and `article_has_full_text` response fields. These are the possible combination of fields you can get back in the response:

Combination of response fields	Explanation
`article_is_paywalled: true` `article_has_full_text: false`	This is an article extracted behind a paywall, using another customer's credentials

Combination of response fields	Explanation
`article_is_paywalled: true` `article_has_full_text: true`	This is an article extracted behind a paywall on your behalf
`article_is_paywalled: false` `article_has_full_text: true`	This is a regular article, which can be accessed outside of a paywall. **Note**: This also includes paywalled articles for sites which we don't have any credentials for, and thus can only access the part of the article that is outside of the paywall.

Sites owned by Bonnier News

Note that some news sites owned by Bonnier News in Sweden, has a special licence tied to them. For this reason, we can only deliver a snippet of the article text, even for articles which are not behind a paywall.

Due to the special license, the `article_has_full_text` response field will contain `true` for these sites, even though only a snippet of the text is returned in the results.

The following news sites owned by Bonnier are currently covered by this license:

Site ID	Site Name	Site URL
80	Dagens Nyheter	https://www.dn.se
113	Sydsvenskan	https://www.sydsvenskan.se
169	Expressen	https://www.expressen.se
167421	Di Weekend	https://weekend.di.se
93	Helsingborgs Dagblad	https://www.hd.se
413	Privata Affärer	https://www.privataaffarer.se
2423	Dagens Medicin.se	https://www.dagensmedicin.se
7274	Teknikens Värld	https://teknikensvarld.se

Site ID	Site Name	Site URL
10720	VeckoRevyn	https://veckorevyn.com
10727	Damernas Värld	https://www.damernasvarld.se
705	Resumé	https://www.resume.se
2477	Aktuell Hållbarhet	https://www.aktuellhallbarhet.se
3565	Allt om Mat	https://alltommat.se
10687	Family Living	https://www.familyliving.se
18346	M-magasin	https://www.m-magasin.se
28310	Feber	http://feber.se
41030	Styleby	https://styleby.nu
49340	DI Trader	https://trader.di.se
53547	Allt om Trädgård	https://www.alltomtradgard.se
97879	Tjock	http://tjock.se
135863	Expressen Omtalat	http://www.expressen.se/omtalat
155641	Di Digital - Di.se	https://digital.di.se
157501	Mitt Kök - Expressen	https://mittkok.expressen.se
166331	Dagens Industri	https://www.di.se
189009	Di Play	https://www.di.se/ditv

Best practices

- Subscribe to updates from our statuspage to get notified about scheduled maintenance or any other disruption in our services
- Use the latest API version available
- Ensure your server code handles transient network errors
    - Ensure you retry your request in case there are network errors, server errors, or you are rate limited
    - Use a back off strategy (exponential backoff for example) for the retry logic
- Ensure your server code can handle additional response fields as new ones can be added without notice (these will be noted in the changelog found at the bottom of each API's documentation)
- Log the HTTP response header `X-Request-Id` for debugging (e.g. in error handling code), please attach the Request ID if you need to contact Twingly Support.
- Use HTTP compression to receive data faster
- Provide a user agent header that identifies you