

Psst! Check out our brand-new Web API Reference in beta! (/documentation/web-api/reference-beta)
And be sure to tweet us your feedback at
[@SpotifyPlatform](https://twitter.com/SpotifyPlatform) on Twitter!
(<http://twitter.com/SpotifyPlatform>)

Search for an Item

Get Spotify Catalog information about artists, albums, tracks or playlists that match a keyword string.

Endpoints

GET <https://api.spotify.com/v1/search>

Request Parameters

Header Fields

HEADER FIELD	VALUE
Authorization	<i>Required.</i> A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.

Query Parameters

QUERY PARAMETER	VALUE
q	<i>Required.</i> Search query keywords and optional field filters and operators. For example: <code>q=roadhouse%20blues</code> .

QUERY PARAMETER	VALUE
type	<p><i>Required.</i></p> <p>A comma-separated list of item types to search across.</p> <p>Valid types are: album , artist , playlist , and track .</p> <p>Search results include hits from all the specified item types.</p> <p>For example: q=name:abacab&type=album,track returns both albums and tracks with “abacab” included in their name.</p>
market	<p><i>Optional.</i></p> <p>An ISO 3166-1 alpha-2 country code (https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string from_token .</p> <p>If a country code is specified, only artists, albums, and tracks with content that is playable in that market is returned.</p> <p>Note:</p> <ul style="list-style-type: none"> - Playlist results are not affected by the market parameter. - If market is set to from_token , and a valid access token is specified in the request header, only content playable in the country associated with the user account, is returned. - Users can view the country that is associated with their account in the account settings (https://www.spotify.com/se/account/overview/). A user must grant access to the user-read-private scope prior to when the access token is issued.
limit	<p><i>Optional.</i></p> <p>Maximum number of results to return.</p> <p>Default: 20</p> <p>Minimum: 1</p> <p>Maximum: 50</p> <p>Note: The limit is applied within each type, not on the total response. For example, if the limit value is 3 and the type is artist,album , the response contains 3 artists and 3 albums.</p>
offset	<p><i>Optional.</i></p> <p>The index of the first result to return.</p> <p>Default: 0 (the first result).</p> <p>Maximum offset (including limit): 10,000.</p> <p>Use with limit to get the next page of search results.</p>
include_external	<p><i>Optional.</i></p> <p>Possible values: <i>audio</i></p> <p>If <i>include_external=audio</i> is specified the response will include any relevant audio content that is hosted externally.</p> <p>By default external content is filtered out from responses.</p>

Writing a Query - Guidelines

Encode spaces with the hex code `%20` or `+`.

Keyword matching: Matching of search keywords is *not* case-sensitive. Operators, however, should be specified in uppercase. Unless surrounded by double quotation marks, keywords are matched in any order. For example: `q=roadhouse&20blues` matches both “Blues Roadhouse” and “Roadhouse of the Blues”. `q="roadhouse&20blues"` matches “My Roadhouse Blues” but not “Roadhouse of the Blues”.

Searching for **playlists** returns results where the query keyword(s) match any part of the playlist’s name or description. Only popular public playlists are returned.

Operator: The operator NOT can be used to exclude results.

For example: `q=roadhouse%20NOT%20blues` returns items that match “roadhouse” but excludes those that also contain the keyword “blues”.

Similarly, the OR operator can be used to broaden the search: `q=roadhouse%20OR%20blues` returns all the results that include either of the terms. Only one OR operator can be used in a query.

Note: Operators must be specified in uppercase. Otherwise, they are handled as normal keywords to be matched.

Wildcards: The asterisk (*) character can, with some limitations, be used as a wildcard (maximum: 2 per query). It matches a variable number of non-white-space characters. It cannot be used:

- in a quoted phrase
 - in a field filter
 - when there is a dash (“-”) in the query
 - or as the first character of the keyword string
- Field filters:** By default, results are returned when a match is found in *any* field of the target object type. Searches can be made more specific by specifying an **album**, **artist** or **track** field filter.

For example: The query `q=album:gold%20artist:abba&type=album` returns only albums with the text “gold” in the album name and the text “abba” in the artist name.

To limit the results to a particular **year**, use the field filter year with album, artist, and track searches.

For example: `q=bob%20year:2014`

Or with a date range. For example: `q=bob%20year:1980–2020`

To retrieve only albums released in the last two weeks, use the field filter `tag:new` in album searches.

To retrieve only albums with the lowest 10% popularity, use the field filter `tag:hipster` in album searches. *Note:* This field filter only works with album searches.

Depending on object types being searched for, other field filters, include `genre` (applicable to tracks and artists), `upc`, and `isrc`. For example:

`q=lil%20genre:%22southern%20hip%20hop%22&type=artist`. Use double quotation marks around the genre keyword string if it contains spaces.

Response Format

On success:

- In the response **header** the HTTP status code is **200** OK.
- The response **body** contains an array of artist objects (</documentation/web-api/reference/object-model/#artist-object-full>), simplified album objects (</documentation/web-api/reference/object-model/#album-object-simplified>), and/or track objects (</documentation/web-api/reference/object-model/#track-object-full>) wrapped in a paging object (</documentation/web-api/reference/object-model/#paging-object>) in JSON.

On error:

- The **header** status code is an error code (</documentation/web-api/#response-status-codes>).
- The response **body** contains an error object (</documentation/web-api/#response-schema>).

Example: Response Format

```
curl -X GET "https://api.spotify.com/v1/search?q=tania%20bowra&type=artist"  
-H "Authorization: Bearer {your access token}"
```

```

{
  "artists": {
    "href": "https://api.spotify.com/v1/search?
query=tania+bowra&offset=0&limit=20&type=artist",
    "items": [ {
      "external_urls": {
        "spotify":
"https://open.spotify.com/artist/08td7MxkoHQkXnWAYD8d6Q"
      },
      "genres": [ ],
      "href":
"https://api.spotify.com/v1/artists/08td7MxkoHQkXnWAYD8d6Q",
      "id": "08td7MxkoHQkXnWAYD8d6Q",
      "images": [ {
        "height": 640,
        "url":
"https://i.scdn.co/image/f2798ddab0c7b76dc2d270b65c4f67ddef7f67
18",
        "width": 640
      }, {
        "height": 300,
        "url":
"https://i.scdn.co/image/b414091165ea0f4172089c2fc67bb35aa37cfc
55",
        "width": 300
      }, {
        "height": 64,
        "url":
"https://i.scdn.co/image/8522fc78be4bf4e83fea8e67bb742e7d3dfe21
b4",
        "width": 64
      } ],
      "name": "Tania Bowra",
      "popularity": 0,
      "type": "artist",
      "uri": "spotify:artist:08td7MxkoHQkXnWAYD8d6Q"
    } ],
    "limit": 20,
    "next": null,
    "offset": 0,
    "previous": null,
    "total": 1
  }
}

```

TRY IT (/CONSOLE/GET-SEARCH-ITEM/?Q=TANIA+BOWRA&TYPE=ARTIST)

Example: Search for Artists

```
curl -X GET "https://api.spotify.com/v1/search?
q=tania%20bowra&type=artist" -H "Authorization: Bearer {your access
token}"
```

TRY IT (/CONSOLE/GET-SEARCH-ITEM/?Q=TANIA+BOWRA&TYPE=ARTIST)

Example: Search for Artists - Using a Wildcard

```
curl -X GET "https://api.spotify.com/v1/search?q=tania*&type=artist" -H
"Authorization: Bearer {your access token}"
```

TRY IT (/CONSOLE/GET-SEARCH-ITEM/?Q=TANIA%2A&TYPE=ARTIST)

Example: Search for Artists - Name Matching “Bob”, Offset, and Limit

```
curl -X GET "https://api.spotify.com/v1/search?
q=bob&type=artist&offset=20&limit=2" -H "Authorization: Bearer {your
access token}"
```

**TRY IT (/CONSOLE/GET-SEARCH-ITEM/?
Q=BOB&LIMIT=2&TYPE=ARTIST&OFFSET=20)**

Example: Search for Albums - Name Matching “Arrival” and Artist Matching “ABBA”

```
curl -X GET "https://api.spotify.com/v1/search?
q=album:arrival%20artist:abba&type=album" -H "Authorization: Bearer
{your access token}"
```

**TRY IT (/CONSOLE/GET-SEARCH-ITEM/?
Q=ALBUM%3AARRIVAL+ARTIST%3AABBA&TYPE=ALBUM)**

Example: Search for Album - Specific UPC Code

```
curl -X GET "https://api.spotify.com/v1/search?
q=upc:00602537817016&type=album" -H "Authorization: Bearer {your access
token}"
```

**TRY IT (/CONSOLE/GET-SEARCH-ITEM/?
Q=UPC%3A00602537817016&TYPE=ALBUM)**

Example: Search for Playlists - Name or Description matching a String

```
curl -X GET "https://api.spotify.com/v1/search?q="doom
metal"&type=playlist" -H "Authorization: Bearer {your access token}"
```

**TRY IT (/CONSOLE/GET-SEARCH-ITEM/?
Q=%22DOOM%20METAL%22&TYPE=PLAYLIST)**

Example: Search only for Tracks Available Only in a specific market

```
curl -X GET "https://api.spotify.com/v1/search?
q=abba&type=track&market=US" -H "Authorization: Bearer {your access
token}"
```

TRY IT (/CONSOLE/GET-SEARCH-ITEM/?Q=ABBA&TYPE=TRACK&MARKET=US)

Fields Reference

Album Object - Simplified

KEY	VALUE TYPE	VALUE DESCRIPTION
album_type	string	“album”, “single”, or “compilation”.
artists	array of simplified artist objects (/documentation/web-api/reference/object-model/#artist-object-simplified)	The artists of the album. Each artist object includes a link in <code>href</code> to further information about the artist.

KEY	VALUE TYPE	VALUE DESCRIPTION
available_markets	array of strings	The markets in which the album is available: ISO 3166-1 alpha-2 country codes (https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Note that an album is considered available in a market when at least 1 of its tracks is available in that market.
external_urls	an external URL object (/documentation/web-api/reference/object-model/#external-url-object)	Known external URLs for this album.
href	string	A link to the Web API endpoint providing full details of the album.
id	string	The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the album.
images	array of image objects	The cover art for the album in various sizes, widest first.
name	string	The name of the album. In case of an album takedown, the value may be an empty string.
type	string	The object type: "album"
uri	string	The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the album.

Artist Object - Full

KEY	VALUE TYPE	VALUE DESCRIPTION
external_urls	an external URL object (/documentation/web-api/reference/object-model/#external-url-object)	Known external URLs for this artist.

KEY	VALUE TYPE	VALUE DESCRIPTION
followers	A followers object (/documentation/web-api/reference/object-model/#followers-object)	Information about the followers of the artist.
genres	array of strings	A list of the genres the artist is associated with. For example: "Prog Rock" , "Post-Grunge" . (If not yet classified, the array is empty.)
href	string	A link to the Web API endpoint providing full details of the artist.
id	string	The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the artist.
images	array of image objects (/documentation/web-api/reference/object-model/#image-object)	Images of the artist in various sizes, widest first.
name	string	The name of the artist
popularity	int	The popularity of the artist. The value will be between 0 and 100, with 100 being the most popular. The artist's popularity is calculated from the popularity of all the artist's tracks.
type	string	The object type: "artist"
uri	string	The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the artist.

Track Object - Full

KEY	VALUE TYPE	VALUE DESCRIPTION
album	a simplified album object (/documentation/web-api/reference/object-model/#album-object-simplified)	The album on which the track appears. The album object includes a link in href to full information about the album.

KEY	VALUE TYPE	VALUE DESCRIPTION
artists	an array of simplified artist objects (/documentation/web-api/reference/object-model/#artist-object-simplified)	The artists who performed the track. Each artist object includes a link in href to more detailed information about the artist.
available_markets	array of strings	A list of the countries in which the track can be played, identified by their ISO 3166-1 alpha-2 (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) code.
disc_number	integer	The disc number (usually 1 unless the album consists of more than one disc).
duration_ms	integer	The track length in milliseconds.
explicit	Boolean	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown).
external_ids	an external ID object (/documentation/web-api/reference/object-model/#external-id-object)	Known external IDs for the track.
external_urls	an external URL object (/documentation/web-api/reference/object-model/#external-url-object)	Known external URLs for this track.
href	string	A link to the Web API endpoint providing full details of the track.
id	string	The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.
is_playable	boolean	Part of the response when Track Relinking (/documentation/general/guides/track-relinking-guide/) is applied. If true , the track is playable in the given market. Otherwise false .

KEY	VALUE TYPE	VALUE DESCRIPTION
linked_from	a linked track object (/documentation/web-api/reference/object-model/#track-link)	Part of the response when Track Relinking (/documentation/general/guides/track-relinking-guide/) is applied, and the requested track has been replaced with different track. The track in the linked_from object contains information about the originally requested track.
name	string	The name of the track.
popularity	integer	<p>The popularity of the track. The value will be between 0, for least popular, and 100 for most popular.</p> <p>The popularity of a track is a value between 0 and 100, with 100 being the most popular. Popularity is based mainly on the total number of playbacks. Duplicate tracks, such as both in a single and in an album, are popularity rated differently.</p> <p>Note: This value is not updated in real-time and may therefore lag behind in actual popularity.</p>
preview_url	string	A link to a 30 second preview (MP3 format) of the track. Can be null
track_number	integer	The number of the track. If an album has several discs, the track number is the number on the specified disc.
type	string	The object type: "track".
uri	string	The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the track.

Playlist Object - Simplified

KEY	VALUE TYPE	VALUE DESCRIPTION
collaborative	Boolean	This will always return false to prevent malicious use of modifying collaborative playlists.

KEY	VALUE TYPE	VALUE DESCRIPTION
external_urls	an external URL object (/documentation/web-api/reference/object-model/#external-url-object)	Known external URLs for this playlist.
href	string	A link to the Web API endpoint providing full details of the playlist.
id	string	The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.
images	an array of image objects (/documentation/web-api/reference/object-model/#image-object)	Images for the playlist. The array may be empty or contain up to three images. The images are returned by size in descending order. See Working with Playlists (/documentation/general/guides/working-with-playlists/). <i>Note: If returned, the source URL for the image (<code>url</code>) is temporary and will expire in less than a day.</i>
name	string	The name of the playlist.
owner	a user object (/documentation/web-api/reference/object-model/#user-object-public)	The user who owns the playlist
public	Boolean or <code>null</code>	The playlist's public/private status: <code>true</code> the playlist is public, <code>false</code> the playlist is private, <code>null</code> the playlist status is not relevant. For more about public/private status, see Working with Playlists (/documentation/general/guides/working-with-playlists/).
snapshot_id	string	The version identifier for the current playlist. Can be supplied in other requests to target a specific playlist version

KEY	VALUE TYPE	VALUE DESCRIPTION
tracks	a tracks object	A collection containing a link (<code>href</code>) to the Web API endpoint where full details of the playlist's tracks can be retrieved, along with the total number of tracks in the playlist.
type	string	The object type: "playlist"
uri	string	The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the playlist.

Paging Object

KEY	VALUE TYPE	VALUE DESCRIPTION
href	string	A link to the Web API endpoint returning the full result of the request.
items	an array of objects	The requested data.
limit	integer	The maximum number of items in the response (as set in the query or by default).
next	string	URL to the next page of items. (<code>null</code> if none)
offset	integer	The offset of the items returned (as set in the query or by default).
previous	string	URL to the previous page of items. (<code>null</code> if none)
total	integer	The total number of items available to return.

Note:

- Currently, you cannot fetch sorted results.
- You cannot search for playlists that contain a certain track.
- You can search only one genre at a time.
- You cannot search for playlists within a user's library.
- In an effort to keep the response small, but include as much information as possible, Spotify has expanded the response and intends to continue and improve the Search endpoint.

- To query based on a release date query at a year level using the year scope. For example:

The query

`https://api.spotify.com/v1/search?q=bob%20year:2014&type=album`

Returns albums released in 2014 with their names or artist names containing “bob”. You can also use the tag:new field filter to get just these albums, as well as compilations and singles, released in the last 2 weeks.

DOCS (/DOCUMENTATION/)

General (/documentation/general/guides/)

Web API (/documentation/web-api/)

Web Playback SDK (/documentation/web-playback-sdk/)

iOS (/documentation/ios/)

Android (/documentation/android/)

Widgets (/documentation/widgets/)

COMMUNITY (/COMMUNITY/NEWS/)

News (/community/news/)

Showcase (/community/showcase/)

USE CASES (/USE-CASES)

Mobile Apps (/use-cases/mobile-apps/)

Hardware (/use-cases/hardware/)

SUPPORT
([HTTPS://DEVELOPER.SPOTIFY.COM/SUPPORT/](https://developer.spotify.com/support))

DISCOVER (/DISCOVER/)

CONSOLE (/CONSOLE/)

DASHBOARD (/DASHBOARD/)

USE CASES (/USE-CASES)

DESIGN & BRANDING GUIDELINES (/BRANDING-GUIDELINES)

LEGAL (/TERMS/)

[Terms of Service \(/terms/\)](/terms/)

[Third Party Licenses \(/legal/third-party-licenses/\)](/legal/third-party-licenses/)

[Legal \(https://www.spotify.com/legal/\)](https://www.spotify.com/legal/)

© 2020 Spotify AB

[Cookies \(https://www.spotify.com/legal/privacy-policy/#cookies\)](https://www.spotify.com/legal/privacy-policy/#cookies)