Web API Reference BETA

Welcome to the improved reference for the Spotify Web API (beta). **Note this may have some missing information**, even though we try to keep this as accurate as possible.

Have feedback? Let us know on Twitter (https://twitter.com/SpotifyPlatform)!

Reference Index

Library API

Playlists API

Albums API

Tracks API

Player API

- Get a User's Available Devices
- Toggle Shuffle For User’s Playback

Browse API

- Get All Categories
- Get a Category
- Get a Category's Playlists
- Get Recommendations
- Get Recommendation Genres
- Get All New Releases
- Get All Featured Playlists

User Profile API

- Get a User's Profile
- Get Current User's Profile

Follow API

- Get Following State for Artists/Users
- Check if Users Follow a Playlist
- Follow Artists or Users
- Follow a Playlist
- Get User's Followed Artists
- Unfollow Artists or Users
- Unfollow Playlist

Artists API

- Get Multiple Artists
- Get an Artist
- Get an Artist's Albums
- Get an Artist's Top Tracks
- Get an Artist's Related Artists

Search API

- Search for an Item

Personalization API

- Get a User's Top Artists and Tracks

Objects

- AlbumObject
- ArtistObject
- AudioFeaturesObject
- CategoryObject
- ContextObject
- CurrentlyPlayingObject
- CursorObject
- CursorPagingObject
- DeviceObject
- DevicesObject
- ExternalIdObject
- PagingObject
- PlayHistoryObject
- PlaylistObject
- PlaylistTrackObject
- PrivateUserObject
- PublicUserObject
- RecommendationSeedObject
- RecommendationsResponseObject
- SavedAlbumObject
- SavedTrackObject

- SimplifiedAlbumObject
- SimplifiedTrackObject
- TrackObject
- TuneableTrackObject

Library API

↑ Back to top

Remove Albums for Current User

Remove one or more albums from the current user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Modification of the current user's "Your Music" collection requires authorization of the `user-library-modify` scope (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` *Required if the IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	*Optional*

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=4iV5W9uYEdYUVa79Axb7Rh,1301WleyT98MSxVHPZCA6M` . Maximum: 50 IDs.	**Required**

JSON BODY PARAMETER	REQUIRED
`ids` A JSON array of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `["4iV5W9uYEdYUVa79Axb7Rh", "1301WleyT98MSxVHPZCA6M"]` A maximum of 50 items can be specified in one request. *Note: if the* `ids` *parameter is present in the query string, any IDs listed here in the body will be ignored.*	*Optional*

Response

On success, the HTTP status code in the response header is `200` Success. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to remove an album when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/DELETE-CURRENT-USER-SAVED-ALBUMS/?IDS=07BYTME3BPSLB6ZBMMFI8D%2C48JYNJH7GMIE6NJQYHMMTT%2C27CZDQRQIKT3IT00338DWS)

Save Albums for Current User

Save one or more albums to the current user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Modification of the current user's "Your Music" collection requires authorization of the `user-library-modify` scope (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**

HEADER	REQUIRED
`Content-Type` *Required if the IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	*Optional*

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=4iV5W9uYEdYUVa79Axb7Rh,1301WleyT98MSxVHPZCA6M` . Maximum: 50 IDs.	**Required**

JSON BODY PARAMETER	REQUIRED
`ids` A JSON array of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `["4iV5W9uYEdYUVa79Axb7Rh", "1301WleyT98MSxVHPZCA6M"]` A maximum of 50 items can be specified in one request. *Note: if the* `ids` *parameter is present in the query string, any IDs listed here in the body will be ignored.*	*Optional*

Response

On success, the HTTP status code in the response header is `201` Created. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to add an album when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-CURRENT-USER-SAVED-ALBUMS/?IDS=07BYTME3BPSLB6ZBMMFI8D%2C48JYNJH7GMIE6NJQYHMMTT%2C27CZDQRQIKT3IT00338DWS)

Save Tracks for User

Save one or more tracks to the current user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Modification of the current user's "Your Music" collection requires authorization of the `user-library-modify` scope (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` *Required if the IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	*Optional*

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=4iV5W9uYEdYUVa79Axb7Rh,1301WleyT98MSxVHPZCA6M` . Maximum: 50 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to add a track when you do not have the user's authorization, or when you have over 10.000 tracks in Your Music, returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-CURRENT-USER-SAVED-TRACKS/?IDS=7OUMYWPWJ422JRCDASZB7P%2C4VQPORUHP5EDPBER92T6LQ%2C2TAKCWOAAZWIXQIJPHIX7B)

Remove User's Saved Tracks

Remove one or more tracks from the current user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Modification of the current user's "Your Music" collection requires authorization of the `user-library-modify` scope (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` *Required if the IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	*Optional*

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=4iV5W9uYEdYUVa79Axb7Rh,1301WleyT98MSxVHPZCA6M` . Maximum: 50 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200` Success. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to remove an album when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/DELETE-CURRENT-USER-SAVED-TRACKS/?IDS=7OUMYWPWJ422JRCDASZB7P%2C4VQPORUHP5EDPBER92T6LQ%2C2TAKCWOAAZWIXQIJPHIX7B)

Check User's Saved Albums

Check if one or more albums is already saved in the current Spotify user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The `user-library-read` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) must have been authorized by the user.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the albums. Maximum: 50 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a JSON array of `true` or `false` values, in the same order in which the `ids` were specified. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER-CONTAINS-SAVED-ALBUMS/?IDS=0PJJGBZJ26QNE1NSQUXAB0%2C5ZAKZV4ZIA5GT7Z29OYHV0)

Get User's Saved Tracks

Get a list of the songs saved in the current Spotify user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The `user-library-read` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) must have been authorized by the user.	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of objects to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first object to return. Default: 0 (i.e., the first object). Use with `limit` to get the next set of objects.	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token`. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of saved track objects (/documentation/web-api/reference/object-model/#saved-track-object) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER-SAVED-TRACKS/)

Check User's Saved Tracks

Check if one or more tracks is already saved in the current Spotify user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The `user-library-read` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) must have been authorized by the user.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the tracks. Maximum: 50 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a JSON array of `true` or `false` values, in the same order in which the `ids` were specified. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER-CONTAINS-SAVED-ALBUMS/?IDS=0PJJGBZJ26QNE1NSQUXAB0%2C5ZAKZV4ZIA5GT7Z29OYHV0)

Get User's Saved Albums

Get a list of the albums saved in the current Spotify user's 'Your Music' library.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The `user-library-read` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) must have been authorized by the user.	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of objects to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*

QUERY PARAMETER	REQUIRED
`offset` The index of the first object to return. Default: 0 (i.e., the first object). Use with `limit` to get the next set of objects.	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of album objects (/documentation/web-api/reference/object-model/#album-object-full) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. Each album object is accompanied by a timestamp (/documentation/web-api/#timestamps) (`added_at`) to show when it was added. There is also an **etag** in the header that can be used in future conditional requests (/documentation/web-api/#conditional-requests).

On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER-CONTAINS-SAVED-ALBUMS/?IDS=0PJJGBZJ26QNE1NSQUXAB0%2C5ZAKZV4ZIA5GT7Z29OYHV0)

Playlists API

↑ Back to top

Replace a Playlist's Tracks

Replace all the tracks in a playlist, overwriting its existing tracks. This powerful request can be useful for replacing tracks, re-ordering existing tracks, or clearing the playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Setting tracks in the current user's public playlists requires authorization of the `playlist-modify-public scope` ; setting tracks in the current user's private playlist (including collaborative playlists) requires the `playlist-modify-private scope` . See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` if URIs are passed in the request body, otherwise ignored._ The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

QUERY PARAMETER	REQUIRED

`uris`
A comma-separated list of Spotify track URIs (/documentation/web-api/#spotify-uris-and-ids) to set. For example:
`uris=spotify:track:4iV5W9uYEdYUVa79Axb7Rh,spotify:track:1301WleyT98MSxVHPZCA6M`
A maximum of 100 tracks can be set in one request.
	Optional

JSON BODY PARAMETER	REQUIRED
`uris` A JSON array of the Spotify track URIs (/documentation/web-api/#spotify-uris-and-ids) to set. For example: `{"uris": ["spotify:track:4iV5W9uYEdYUVa79Axb7Rh", "spotify:track:1301WleyT98MSxVHPZCA6M"]}` Currently, a maximum of 100 tracks can be set. Note: if the uris parameter is present in the query string, any URIs listed here in the body will be ignored.	*Optional*

Response

On success, the HTTP status code in the response header is `201` Created. 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), and the existing playlist is unmodified. Trying to set a track when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PLAYLIST-TRACKS/)

Get a List of a User's Playlists

Get a list of the playlists owned or followed by a Spotify user.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Private playlists are only retrievable *for the current user* and requires the `playlist-read-private` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) to have been authorized by the user. Note that this scope alone will not return collaborative playlists, even though they are always private. Collaborative playlists are only retrievable *for the current user* and requires the `playlist-read-collaborative` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) to have been authorized by the user.	**Required**

PATH PARAMETER	REQUIRED
`{user_id}` The user's Spotify user ID (/documentation/web-api/#spotify-uris-and-ids).	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of playlists to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first playlist to return. Default: 0 (the first object). Maximum offset: 100.000. Use with `limit` to get the next set of playlists.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of simplified playlist objects (/documentation/web-api/reference/object-model/#playlist-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-PLAYLISTS/?USER_ID=WIZZLER)

Change a Playlist's Details

Change a playlist's name and public/private state. (The user must, of course, own the playlist.)

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Changing a public playlist for a user requires authorization of the `playlist-modify-public` scope; changing a private playlist requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

JSON BODY PARAMETER	REQUIRED
`name` The new name for the playlist, for example `"My New Playlist Title"`	*Optional*
`public` If `true` the playlist will be public, if `false` it will be private.	*Optional*
`collaborative` If `true` , the playlist will become collaborative and other users will be able to modify the playlist in their Spotify client. *Note: You can only set `collaborative` to `true` on non-public playlists.*	*Optional*
`description` Value for playlist description as displayed in Spotify Clients and in the Web API.	*Optional*

Response

On success the HTTP status code in the response header is `200` OK.

On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to change a playlist when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PLAYLIST/)

Upload a Custom Playlist Cover Image

Replace the image used to represent a specific playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. This access token must be tied to the user who owns the playlist, and must have the scope `ugc-image-upload granted` . In addition, the token must also contain `playlist-modify-public` and/or `playlist-modify-private` , depending the public status of the playlist you want to update . See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` The content type of the request body: `image/jpeg`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

Response

If you get status code `429` , it means that you have sent too many requests. If this happens, have a look in the `Retry-After` header, where you will see a number displayed. This is the amount of seconds that you need to wait, before you can retry sending your requests.

Notes

The request should contain a Base64 encoded JPEG image data, maximum payload size is 256 KB.

Rate Limiting: If you get status code `429` , it means that you have sent too many requests. If this happens, have a look in the `Retry-After` header, where you will see a number displayed. This is the amount of seconds that you need to wait, before you can retry sending your requests.

Reorder a Playlist's Tracks

Reorder a track or a group of tracks in a playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Reordering tracks in the current user's public playlists requires authorization of the `playlist-modify-public` scope; reordering tracks in the current user's private playlist (including collaborative playlists) requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` if URIs are passed in the request body, otherwise ignored._ The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

JSON BODY PARAMETER	REQUIRED
`range_start` The position of the first track to be reordered.	**Required**
`insert_before` The position where the tracks should be inserted. To reorder the tracks to the end of the playlist, simply set *insert_before* to the position after the last track. Examples: To reorder the first track to the last position in a playlist with 10 tracks, set *range_start* to 0, and *insert_before* to 10. To reorder the last track in a playlist with 10 tracks to the start of the playlist, set *range_start* to 9, and *insert_before* to 0.	**Required**
`range_length` The amount of tracks to be reordered. Defaults to 1 if not set. The range of tracks to be reordered begins from the *range_start* position, and includes the *range_length* subsequent tracks. Example: To move the tracks at index 9-10 to the start of the playlist, *range_start* is set to 9, and *range_length* is set to 2.	*Optional*
`snapshot_id` The playlist's snapshot ID against which you want to make the changes.	*Optional*

Response

On success, the response body contains a `snapshot_id` in JSON format and the HTTP status code in the response header is `200` OK. The `snapshot_id` can be used to identify your playlist version in future requests.

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), and the existing playlist is unmodified.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PLAYLIST-TRACKS/)

Add Tracks to a Playlist

Add one or more tracks to a user's playlist.

Request

HEADER	REQUIRED

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Adding tracks to the current user's public playlists requires authorization of the `playlist-modify-public` scope; adding tracks to the current user's private playlist (including collaborative playlists) requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` *Required if URIs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

QUERY PARAMETER	REQUIRED

`uri`
A comma-separated list of Spotify track URIs (/documentation/web-api/#spotify-uris-and-ids) to add. For example:
`uris=spotify:track:4iV5W9uYEdYUVa79Axb7Rh, spotify:track:1301WleyT98MSxVHPZCA6M`
A maximum of 100 tracks can be added in one request. *Note: it is likely that passing a large number of track URIs as a query parameter will exceed the maximum length of the request URI. When adding a large number of tracks it is recommended to pass them in the request body, see below.*
	Optional

`position` The position to insert the tracks, a zero-based index. For example, to insert the tracks in the first position: `position=0` ; to insert the tracks in the third position: `position=2` . If omitted, the tracks will be appended to the playlist. Tracks are added in the order they are listed in the query string or request body.	*Optional*

JSON BODY PARAMETER	REQUIRED
`uris` A JSON array of the Spotify track URIs (/documentation/web-api/#spotify-uris-and-ids) to add. For example: `{"uris": ["spotify:track:4iV5W9uYEdYUVa79Axb7Rh","spotify:track:1301WleyT98MSxVHPZCA6M"]}` A maximum of 100 tracks can be added in one request. *Note: if the* `uris` *parameter is present in the query string, any URIs listed here in the body will be ignored.*	*Optional*

`position`
The position to insert the tracks, a zero-based index. For example, to insert the tracks in the first position: `position=0` ; to insert the tracks in the third position: `position=2` . If omitted, the tracks will be appended to the playlist. Tracks are added in the order they appear in the uris array. For example:
`{"uris": ["spotify:track:4iV5W9uYEdYUVa79Axb7Rh","spotify:track:1301WleyT98MSxVHPZCA6M"], "position": 3}`
	Optional

Response

On success, the HTTP status code in the response header is `201` Created. The response body contains a `snapshot_id` in JSON format. The `snapshot_id` can be used to identify your playlist version in future requests. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to add a track when you do not have the user's authorization, or when there are more than 10.000 tracks in the playlist, returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/POST-PLAYLIST-TRACKS/)

Get a Playlist's Tracks

Get full details of the tracks of a playlist owned by a Spotify user.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Both Public and Private playlists belonging to any user are retrievable on provision of a valid access token.	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

QUERY PARAMETER	REQUIRED

`fields`
Filters for the query: a comma-separated list of the fields to return. If omitted, all fields are returned. For example, to get just the total number of tracks and the request limit:
`fields=total,limit`
A dot separator can be used to specify non-reoccurring fields, while parentheses can be used to specify reoccurring fields within objects. For example, to get just the added date and user ID of the adder:
`fields=items(added_at,added_by.id)`
Use multiple parentheses to drill down into nested objects, for example:
`fields=items(track(name,href,album(name,href)))`
Fields can be excluded by prefixing them with an exclamation mark, for example:
`fields=items.track.album(!external_urls,images)`
	Optional

`limit` The maximum number of tracks to return. Default: 100. Minimum: 1. Maximum: 100.	*Optional*
`offset` The index of the first track to return. Default: 0 (the first object).	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	**Required**

Response

On success, the response body contains an array of playlist 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 format and the HTTP status code in the response header is `200` OK. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Requesting playlists that you do not have the user's authorization to access returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-PLAYLIST-TRACKS/?PLAYLIST_ID=21THA8J9TASGUXYNBU5TSC&USER_ID=SPOTIFY_ESPA%C3%B1A)

Get a Playlist Cover Image

Get the current image associated with a specific playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. This access token must be issued on behalf of the user. Current playlist image for both Public and Private playlists of any user are retrievable on provision of a valid access token.	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

Response

On success, the response body contains a list of image objects (/documentation/web-api/reference/object-model/#image-object) in JSON format and the HTTP status code in the response header is `200` OK
On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-PLAYLIST-IMAGES?USER_ID=JMPEREZPEREZ&PLAYLIST_ID=3CEYPJA9OZ9GIPAC4ASH4N)

Remove Tracks from a Playlist

Remove one or more tracks from a user's playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Removing tracks from a user's public playlist requires authorization of the `playlist-modify-public` scope; removing tracks from a private playlist requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

JSON BODY PARAMETER	REQUIRED
`tracks` An array of objects containing Spotify URIs (https://developer.spotify.com/spotify-documentation/web-api/#spotify-uris-and-ids) of the tracks to remove. For example: `{ "tracks": [{ "uri": "spotify:track:4iV5W9uYEdYUVa79Axb7Rh" },{ "uri": "spotify:track:1301WleyT98MSxVHPZCA6M" }] }` . A maximum of 100 objects can be sent at once.	**Required**
`snapshot_id` The playlist's snapshot ID against which you want to make the changes. The API will validate that the specified tracks exist and in the specified positions and make the changes, even if more recent changes have been made to the playlist.	*Optional*

Response

On success, the response body contains a `snapshot_id` in JSON format and the HTTP status code in the response header is `200` OK. The `snapshot_id` can be used to identify your playlist version in future requests.

On error, the header status code is an error code (https://developer.spotify.com/spotify-documentation/web-api/#response-status-codes) and the response body contains an error object (https://developer.spotify.com/spotify-documentation/web-api/#response-schema). Trying to remove a track when you do not have the user's authorization returns error `403` Forbidden. Attempting to use several different ways to remove tracks returns `400` Bad Request. Other client errors returning `400` Bad Request include specifying invalid positions.

Notes

### Frequently Asked Questions:

- **Is it possible to delete a playlist?** No, it isn't. The reason there is no endpoint for this is explained in our Working With Playlists Guide in the section Following and Unfollowing a Playlist (/documentation/general/guides/working-with-playlists/#following-and-unfollowing-a-playlist).
- **Can I use `X-HTTP-Method-Override` or similar to send a DELETE request overriding the HTTP verb?** Not at the moment, the delete operation needs to be specified through a DELETE request.

TRY IN OUR WEB CONSOLE (/CONSOLE/DELETE-PLAYLIST-TRACKS/)

Get a List of Current User's Playlists

Get a list of the playlists owned or followed by the current Spotify user.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Private playlists are only retrievable *for the current user* and requires the `playlist-read-private` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) to have been authorized by the user. Note that this scope alone will not return collaborative playlists, even though they are always private. Collaborative playlists are only retrievable *for the current user* and requires the `playlist-read-collaborative` scope (/documentation/general/guides/authorization-guide/#list-of-scopes) to have been authorized by the user.	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of playlists to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first playlist to return. Default: 0 (the first object). Maximum offset: 100.000. Use with `limit` to get the next set of playlists.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of simplified playlist objects (/documentation/web-api/reference/object-model/#playlist-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Please note that the access token has to be tied to a user.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER-PLAYLISTS/)

Get a Playlist

Get a playlist owned by a Spotify user.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. Both Public and Private playlists belonging to any user are retrievable on provision of a valid access token.	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	**Required**

QUERY PARAMETER	REQUIRED
`fields` Filters for the query: a comma-separated list of the fields to return. If omitted, all fields are returned. For example, to get just the playlist's description and URI: `fields=description,uri` . A dot separator can be used to specify non-reoccurring fields, while parentheses can be used to specify reoccurring fields within objects. For example, to get just the added date and user ID of the adder: `fields=tracks.items(added_at,added_by.id)` . Use multiple parentheses to drill down into nested objects, for example: `fields=tracks.items(track(name,href,album(name,href)))` . Fields can be excluded by prefixing them with an exclamation mark, for example: `fields=tracks.items(track(name,href,album(!name,href)))`	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the response body contains a playlist object (/documentation/web-api/reference/object-model/#playlist-object-full) in JSON format and the HTTP status code in the response header is `200` OK. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Requesting playlists that you do not have the user's authorization to access returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-PLAYLIST/?PLAYLIST_ID=59ZBFPES4DQWEJBPWHZRTC&USER_ID=SPOTIFY)

Create a Playlist

Create a playlist for a Spotify user. (The playlist will be empty until you add tracks (/documentation/web-api/reference/playlists/add-tracks-to-playlist/).)

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Creating a public playlist for a user requires authorization of the `playlist-modify-public` scope; creating a private playlist requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**
`Content-Type` The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{user_id}` The user's Spotify user ID (/documentation/web-api/#spotify-uris-and-ids).	**Required**

JSON BODY PARAMETER	REQUIRED
`name` The name for the new playlist, for example `"Your Coolest Playlist"` . This name does not need to be unique; a user may have several playlists with the same name.	**Required**
`public` Defaults to `true` . If `true` the playlist will be public, if `false` it will be private. To be able to create private playlists, the user must have granted the `playlist-modify-private` scope (/documentation/general/guides/authorization-guide/#list-of-scopes)	*Optional*
`collaborative` Defaults to `false` . If `true` the playlist will be collaborative. Note that to create a collaborative playlist you must also set `public` to `false` . To create collaborative playlists you must have granted `playlist-modify-private` and `playlist-modify-public` scopes (/documentation/general/guides/authorization-guide/#list-of-scopes) .	*Optional*
`description` value for playlist description as displayed in Spotify Clients and in the Web API.	*Optional*

Response

On success, the response body contains the created playlist object (/documentation/web-api/reference/object-model/#playlist-object-full) in JSON format and the HTTP status code in the response header is `200` OK or `201` Created. There is also a `Location` response header giving the Web API endpoint for the new playlist.

On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). Trying to create a playlist when you do not have the user's authorization returns error `403` Forbidden.

TRY IN OUR WEB CONSOLE (/CONSOLE/POST-PLAYLISTS/)

Albums API

↑ Back to top

Get Multiple Albums

Get Spotify catalog information for multiple albums identified by their Spotify IDs.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the albums. Maximum: 20 IDs.	**Required**
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string from_token. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object whose key is `"albums"` and whose value is an array of album objects (/documentation/web-api/reference/object-model/#album-object-full) in JSON format.

Objects are returned in the order requested. If an object is not found, a `null` value is returned in the appropriate position. Duplicate `ids` in the query will result in duplicate objects in the response. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-SEVERAL-ALBUMS/)

Get an Album

Get Spotify catalog information for a single album.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID of the album.	**Required**

QUERY PARAMETER	REQUIRED
`market` The market you'd like to request. Synonym for `country` .	*Optional*

Response

On success, the HTTP status code in the response header is 200 OK and the response body contains an album object in JSON format. On error, the header status code is an error code and the response body contains an error object.

**TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ALBUM/?
ID=0SNOF9WDWHWUNNAHPD3BAJ)**

Get an Album's Tracks

Get Spotify catalog information about an album's tracks. Optional parameters can be used to limit the number of tracks returned.

Request

HEADER	REQUIRED

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID of the album.	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of tracks to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first track to return. Default: 0 (the first object). Use with limit to get the next set of tracks.	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string from_token. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is 200 OK and the response body contains an album object in JSON format. On error, the header status code is an error code and the response body contains an error object.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ALBUM-TRACKS/)

Tracks API

↑ Back to top

Get Several Tracks

Get Spotify catalog information for multiple tracks based on their Spotify IDs.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the tracks. Maximum: 50 IDs.	**Required**
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object whose key is `tracks` and whose value is an array of track objects (/documentation/web-api/reference/object-model/#track-object-full) in JSON format.

Objects are returned in the order requested. If an object is not found, a `null` value is returned in the appropriate position. Duplicate `ids` in the query will result in duplicate objects in the response. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-SEVERAL-TRACKS/?IDS=3N3PPAM7VGAVA1IARUC9LP,3TWNVMDTFQTAD5GMKEDHLD)

Get Audio Analysis for a Track

Get a detailed audio analysis for a single track identified by its unique Spotify ID.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.	**Required**

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains an audio analysis object in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-AUDIO-ANALYSIS-TRACK/?ID=06AKEBRKUCKW0KREUWRNVT)

Get Audio Features for a Track

Get audio feature information for a single track identified by its unique Spotify ID.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.	**Required**

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains an audio features object (/documentation/web-api/reference/object-model/#audio-features-object) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-AUDIO-FEATURES-TRACK/?ID=06AKEBRKUCKW0KREUWRNVT)

Get a Track

Get Spotify catalog information for a single track identified by its unique Spotify ID.

Request

HEADER	REQUIRED

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.	**Required**

QUERY PARAMETER	REQUIRED
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a track object (/documentation/web-api/reference/object-model/#track-object-full) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-TRACK/?ID=3N3PPAM7VGAVA1IARUC9LP)

Get Audio Features for Several Tracks

Get audio features for multiple tracks based on their Spotify IDs.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the tracks. Maximum: 100 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains an object whose key is `"audio_features"` and whose value is an array of audio features objects in JSON format.

Objects are returned in the order requested. If an object is not found, a `null` value is returned in the appropriate position. Duplicate `ids` in the query will result in duplicate objects in the response. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-AUDIO-FEATURES-SEVERAL-TRACKS/?IDS=4JPKVNYNVCJ8TUMKJAJ50A,2NRANZE9UCMPAS5XVBXL40,24JYGZOLM0EMRQEGTFCICG)

Player API

↑ Back to top

Skip User’s Playback To Next Track

Skips to next track in the user’s queue.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/POST-NEXT/)

Set Repeat Mode On User's Playback

Set the repeat mode for the user's playback. Options are repeat-track, repeat-context, and off.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`state` **track**, **context** or **off**. **track** will repeat the current track. **context** will repeat the current context. **off** will turn repeat off.	**Required**
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-REPEAT/)

Transfer a User's Playback

Transfer playback to a new device and determine if it should start playing.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

JSON BODY PARAMETER	REQUIRED
`device_ids` A JSON array containing the ID of the device on which playback should be started/transferred. For example: `{device_ids:["74ASZWbe4lXaubB36ztrGX"]}` Note: Although an array is accepted, only a single device_id is currently supported. Supplying more than one will return `400 Bad Request`	**Required**
`play` **true**: ensure playback happens on new device. **false** or not provided: keep the current playback state.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-USER-PLAYER)

Get the User's Currently Playing Track

Get the object currently being played on the user's Spotify account.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-read-currently-playing` and/or `user-read-playback-state` scope authorized in order to read information.	**Required**

QUERY PARAMETER	REQUIRED
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token`. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	**Required**

Response

A successful request will return a `200 OK` response code with a json payload that contains information about the currently playing track and context (see below). The information returned is for the last known state, which means an inactive device could be returned if it was the last one to execute playback.

When no available devices are found, the request will return a `200 OK` response but with no data populated.

When no track is currently playing, the request will return a `204 NO CONTENT` response with no payload.

If private session is enabled the response will be a `204 NO CONTENT` with an empty payload.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-USERS-CURRENTLY-PLAYING-TRACK/)

Get Information About The User's Current Playback

Get information about the user's current playback state, including track, track progress, and active device.

Request

QUERY PARAMETER	REQUIRED
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token`. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide/).	*Optional*

Response

A successful request will return a `200 OK` response code with a json payload that contains information about the current playback. The information returned is for the last known state, which means an inactive device could be returned if it was the last one to execute playback. When no available devices are found, the request will return a `200 OK` response but with no data populated.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-USER-PLAYER/)

Seek To Position In Currently Playing Track

Seeks to the given position in the user's currently playing track.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback	**Required**

QUERY PARAMETER	REQUIRED
`position_ms` The position in milliseconds to seek to. Must be a positive number. Passing in a position that is greater than the length of the track will cause the player to start playing the next song.	**Required**
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-SEEK/)

Skip User's Playback To Previous Track

Skips to previous track in the user's queue.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/POST-PREVIOUS/)

Start/Resume a User's Playback

Start a new context or resume current playback on the user's active device.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

JSON BODY PARAMETER	REQUIRED
`context_uri` string	*Optional*
`uris` Array of URIs	*Optional*
`offset` Object	*Optional*
`position_ms` integer	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PLAY/)

Pause a User's Playback

Pause playback on the user's account.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PAUSE/)

Set Volume For User's Playback

Set the volume for the user's current playback device.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`volume_percent` The volume to set. Must be a value from 0 to 100 inclusive.	**Required**
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-VOLUME/)

Get Current User's Recently Played Tracks

Get tracks from the current user's recently played tracks.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-read-recently-played` scope authorized in order to read the user''s recently played track.'	**Required**

QUERY PARAMETER	REQUIRED
`limit` The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`after` A Unix timestamp in milliseconds. Returns all items after (but not including) this cursor position. If `after` is specified, `before` must not be specified.	*Optional*
`before` A Unix timestamp in milliseconds. Returns all items before (but not including) this cursor position. If `before` is specified, `after` must not be specified.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of play history objects (wrapped in a cursor-based paging object) in JSON format. The play history items each contain the context the track was played from (e.g. playlist, album), the date and time the track was played, and a track object (simplified) (/documentation/web-api/reference/object-model/#track-object-simplified). On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

If private session is enabled the response will be a `204 NO CONTENT` with an empty payload.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-RECENTLY-PLAYED/)

Get a User's Available Devices

Get information about a user's available devices.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-read-playback-state` scope authorized in order to read information.	**Required**

Response

A successful request will return a `200 OK` response code with a json payload that contains the device objects (see below). When no available devices are found, the request will return a 200 OK response with an empty devices list.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-USERS-AVAILABLE-DEVICES/)

Toggle Shuffle For User's Playback

Toggle shuffle on or off for user's playback.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of a user. The access token must have the `user-modify-playback-state` scope authorized in order to control playback.	**Required**

QUERY PARAMETER	REQUIRED
`state` **true** : Shuffle user's playback **false** : Do not shuffle user's playback.	**Required**
`device_id` The id of the device this command is targeting. If not supplied, the user's currently active device is the target.	*Optional*

Response

A completed request will return a `204 NO CONTENT` response code, and then issue the command to the player. Due to the asynchronous nature of the issuance of the command, you should use the Get Information About The User's Current Playback (/documentation/web-api/reference/player/get-information-about-the-users-current-playback) endpoint to check that your issued command was handled correctly by the player.

If the device is not found, the request will return `404 NOT FOUND` response code.

If the user making the request is non-premium, a `403 FORBIDDEN` response code will be returned.

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-SHUFFLE/?STATE=TRUE)

Browse API

↑ Back to top

Get All Categories

Get a list of categories used to tag items in Spotify (on, for example, the Spotify player's "Browse" tab).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`country` A country: an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Provide this parameter if you want to narrow the list of returned categories to those relevant to a particular country. If omitted, the returned items will be globally relevant.	*Optional*

QUERY PARAMETER	REQUIRED
`locale` The desired language, consisting of an ISO 639-1 (http://en.wikipedia.org/wiki/ISO_639-1) language code and an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2), joined by an underscore. For example: `es_MX` , meaning "Spanish (Mexico)". Provide this parameter if you want the category metadata returned in a particular language. Note that, if `locale` is not supplied, or if the specified language is not available, all strings will be returned in the Spotify default language (American English). The `locale` parameter, combined with the `country` parameter, may give odd results if not carefully matched. For example `country=SE&locale=de_DE` will return a list of categories relevant to Sweden but as German language strings.	*Optional*
`limit` The maximum number of categories to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first item to return. Default: 0 (the first object). Use with `limit` to get the next set of categories.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object with a `categories` field, with an array of category objects (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

Once you have retrieved the list, you can use Get a Category (https://developer.spotify.com/web-api/get-category/) to drill down further.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-BROWSE-CATEGORIES/)

Get a Category

Get a single category used to tag items in Spotify (on, for example, the Spotify player's "Browse" tab).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{category_id}` The Spotify category ID (/documentation/web-api/#spotify-uris-and-ids) for the category.	**Required**

QUERY PARAMETER	REQUIRED
`country` A country: an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Provide this parameter to ensure that the category exists for a particular country.	*Optional*
`locale` The desired language, consisting of an ISO 639-1 (http://en.wikipedia.org/wiki/ISO_639-1) language code and an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2), joined by an underscore. For example: `es_MX` , meaning "Spanish (Mexico)". Provide this parameter if you want the category strings returned in a particular language. Note that, if `locale` is not supplied, or if the specified language is not available, the category strings returned will be in the Spotify default language (American English).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a category object in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

Once you have retrieved the category, you can use Get a Category's Playlists (https://developer.spotify.com/web-api/get-categorys-playlists/) to drill down further.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-BROWSE-CATEGORY/)

Get a Category's Playlists

Get a list of Spotify playlists tagged with a particular category.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{category_id}` The Spotify category ID (/documentation/web-api/#spotify-uris-and-ids) for the category.	**Required**

QUERY PARAMETER	REQUIRED
`country` A country: an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Provide this parameter to ensure that the category exists for a particular country.	*Optional*
`limit` The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first item to return. Default: 0 (the first object). Use with `limit` to get the next set of items.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of simplified playlist objects (/documentation/web-api/reference/object-model/#playlist-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

Once you have retrieved the list, you can use Get a Playlist (https://developer.spotify.com/web-api/get-playlist/) and Get a Playlist's Tracks (https://developer.spotify.com/web-api/get-playlists-tracks/) to drill down further.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CATEGORY-PLAYLISTS/?COUNTRY=BR&CATEGORY_ID=PARTY&LIMIT=2)

Get Recommendations

Recommendations are generated based on the available information for a given seed entity and matched against similar artists and tracks. If there is sufficient information about the provided seeds, a list of tracks will be returned together with pool size details.

For artists and tracks that are very new or obscure there might not be enough data to generate a list of tracks.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`seed_artists` A comma separated list of Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for seed artists. Up to 5 seed values may be provided in any combination of `seed_artists` , `seed_tracks` and `seed_genres` .	**Required**

QUERY PARAMETER	REQUIRED
`seed_genres` A comma separated list of any genres in the set of available genre seeds. Up to 5 seed values may be provided in any combination of `seed_artists`, `seed_tracks` and `seed_genres`.	**Required**
`seed_tracks` A comma separated list of Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for a seed track. Up to 5 seed values may be provided in any combination of `seed_artists`, `seed_tracks` and `seed_genres`.	**Required**
`limit` The target size of the list of recommended tracks. For seeds with unusually small pools or when highly restrictive filtering is applied, it may be impossible to generate the requested number of recommended tracks. Debugging information for such cases is available in the response. Default: 20. Minimum: 1. Maximum: 100.	*Optional*
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token`. Provide this parameter if you want to apply Track Relinking (/documentation/general/guides/track-relinking-guide). Because `min_*`, `max_*` and `target_*` are applied to pools before relinking, the generated results may not precisely match the filters applied. Original, non-relinked tracks are available via the `linked_from` attribute of the relinked track response (/documentation/general/guides/track-relinking-guide).	*Optional*
`min_*` *Multiple values.* For each tunable track attribute, a hard floor on the selected track attribute's value can be provided. See tunable track attributes below for the list of available options. For example, `min_tempo=140` would restrict results to only those tracks with a tempo of greater than 140 beats per minute.	*Optional*
`max_*` *Multiple values.* For each tunable track attribute, a hard ceiling on the selected track attribute's value can be provided. See tunable track attributes below for the list of available options. For example, `max_instrumentalness=0.35` would filter out most tracks that are likely to be instrumental.	*Optional*
`target_*` *Multiple values.* For each of the tunable track attributes (below) a target value may be provided. Tracks with the attribute values nearest to the target values will be preferred. For example, you might request `target_energy=0.6` and `target_danceability=0.8`. All target values will be weighed equally in ranking results.	*Optional*

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains a recommendations response object in JSON format.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-RECOMMENDATIONS/?SEED_ARTISTS=4NHQUGZHTTLFVGF5SZESLK&SEED_TRACKS=0C6XIDDPZE81M2Q797ORDA&MIN_ENERGY=0.4&MIN_POPULARITY=50&M

Get Recommendation Genres

Retrieve a list of available genres seed parameter values for recommendations (/documentation/web-api/reference/browse/get-recommendations/).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains a recommendations response object in JSON format.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-AVAILABLE-GENRE-SEEDS/)

Get All New Releases

Get a list of new album releases featured in Spotify (shown, for example, on a Spotify player's "Browse" tab).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`country` A country: an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Provide this parameter if you want the list of returned items to be relevant to a particular country. If omitted, the returned items will be relevant to all countries.	*Optional*
`limit` The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`offset` The index of the first item to return. Default: 0 (the first object). Use with `limit` to get the next set of items.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a `message` and an `albums` object. The `albums` object contains an array of simplified album objects (/documentation/web-api/reference/object-model/#album-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

Once you have retrieved the list, you can use Get an Album's Tracks (/documentation/web-api/reference/albums/get-albums-tracks/) to drill down further.

The results are returned in an order reflected within the Spotify clients, and therefore may not be ordered by date.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-NEW-RELEASES/?COUNTRY=SE)

Get All Featured Playlists

Get a list of Spotify featured playlists (shown, for example, on a Spotify player's 'Browse' tab).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`country` A country: an ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). Provide this parameter if you want the list of returned items to be relevant to a particular country. If omitted, the returned items will be relevant to all countries.	*Optional*
`locale` The desired language, consisting of a lowercase ISO 639-1 language code (http://en.wikipedia.org/wiki/ISO_639-1) and an uppercase ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2), joined by an underscore. For example: `es_MX` , meaning "Spanish (Mexico)". Provide this parameter if you want the results returned in a particular language (where available). Note that, if `locale` is not supplied, or if the specified language is not available, all strings will be returned in the Spotify default language (American English). The `locale` parameter, combined with the `country` parameter, may give odd results if not carefully matched. For example `country=SE&locale=de_DE` will return a list of categories relevant to Sweden but as German language strings.	*Optional*
`timestamp` A timestamp in ISO 8601 format (http://en.wikipedia.org/wiki/ISO_8601): `yyyy-MM-ddTHH:mm:ss` . Use this parameter to specify the user's local time to get results tailored for that specific date and time in the day. If not provided, the response defaults to the current UTC time. Example: "2014-10-23T09:00:00" for a user whose local time is 9AM. If there were no featured playlists (or there is no data) at the specified time, the response will revert to the current UTC time.	*Optional*
`limit` The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*

QUERY PARAMETER	REQUIRED
`offset` The index of the first item to return. Default: 0 (the first object). Use with `limit` to get the next set of items.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a `message` and a `playlists` object. The `playlists` object contains an array of simplified playlist objects (/documentation/web-api/reference/object-model/#playlist-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

Once you have retrieved the list of playlist objects, you can use Get a Playlist (https://developer.spotify.com/web-api/get-playlist/) and Get a Playlist's Tracks (https://developer.spotify.com/web-api/get-playlists-tracks/) to drill down further.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-FEATURED-PLAYLISTS/?COUNTRY=SE&LIMIT=2)

User Profile API

↑ Back to top

Get a User's Profile

Get public profile information about a Spotify user.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

PATH PARAMETER	REQUIRED
`{user_id}` The user's Spotify user ID (/documentation/web-api/#spotify-uris-and-ids).	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a user object (/documentation/web-api/reference/object-model/#user-object-public) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). If a user with that user_id doesn't exist, the status code is `404` NOT FOUND.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-USERS-PROFILE/?USER_ID=WIZZLER)

Get Current User's Profile

Get detailed profile information about the current user (including the current user's username).

Request

HEADER	REQUIRED

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the current user. Reading the user's email address requires the `user-read-email` scope; reading country and product subscription level requires the `user-read-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a user object (/documentation/web-api/reference/object-model/#user-object-private) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema). When requesting fields that you don't have the user's authorization to access, it will return error `403` Forbidden.

Important! If the `user-read-email` scope is authorized, the returned JSON will include the email address that was entered when the user created their Spotify account. **This email address is unverified**; do not assume that the email address belongs to the user.

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-CURRENT-USER/)

Follow API

↑ Back to top

Get Following State for Artists/Users

Check to see if the current user is following one or more artists or other Spotify users.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `user-follow-read` scope.	**Required**

QUERY PARAMETER	REQUIRED
`type` The ID type: either `artist` or `user` .	**Required**
`ids` A comma-separated list of the artist or the user Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) to check. For example: `ids=74ASZWbe4lXaubB36ztrGX,08td7MxkoHQkXnWAYD8d6Q` . A maximum of 50 IDs can be sent in one request.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a JSON array of `true` or `false` values, in the same order in which the `ids` were specified. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-FOLLOWING-CONTAINS/?TYPE=USER&IDS=EXAMPLEUSER01)

Check if Users Follow a Playlist

Check to see if one or more Spotify users are following a specified playlist.

Request

HEADER	REQUIRED

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `playlist-read-private` scope if a private playlist is requested.	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) of the playlist.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of Spotify User IDs (/documentation/web-api/#spotify-uris-and-ids) ; the ids of the users that you want to check to see if they follow the playlist. Maximum: 5 ids.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains a JSON array of `true` or `false` values, in the same order in which the `ids` were specified. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-PLAYLIST-FOLLOWERS-CONTAINS/?IDS=POSSAN,ELOGAIN&USER_ID=JMPEREZPEREZ&PLAYLIST_ID=2V3INVBX8AY1GT2UXTUKUT)

Follow Artists or Users

Add the current user as a follower of one or more artists or other Spotify users.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `user-follow-modify` scope.	**Required**
`Content-Type` *Required if IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json`	*Optional*

QUERY PARAMETER	REQUIRED
`type` The ID type: either `artist` or `user` .	**Required**
`ids` A comma-separated list of the artist or the user Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=74ASZWbe4lXaubB36ztrGX,08td7MxkoHQkXnWAYD8d6Q` . A maximum of 50 IDs can be sent in one request.	**Required**

JSON BODY PARAMETER	REQUIRED
`ids` A JSON array of the artist or user Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `{ids:["74ASZWbe4lXaubB36ztrGX", "08td7MxkoHQkXnWAYD8d6Q"]}` . A maximum of 50 IDs can be sent in one request. *Note: if the* `ids` *parameter is present in the query string, any IDs listed here in the body will be ignored.*	**Required**

Response

On success, the HTTP status code in the response header is `204` No Content and the response body is empty. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-FOLLOWING/?TYPE=USER&IDS=EXAMPLEUSER01)

Follow a Playlist

Add the current user as a follower of a playlist.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `user-follow-modify` scope.	**Required**
`Content-Type` The content type of the request body: `application/json`	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) of the playlist. Any playlist can be followed, regardless of its public/private status (/documentation/general/guides/working-with-playlists/#public-private-and-collaborative-status), as long as you know its playlist ID.	**Required**

JSON BODY PARAMETER	REQUIRED
`public` Defaults to `true` . If `true` the playlist will be included in user's public playlists, if `false` it will remain private. To be able to follow playlists privately, the user must have granted the `playlist-modify-private` scope (/documentation/general/guides/authorization-guide/#list-of-scopes).	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body is empty. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-PLAYLIST-FOLLOWERS/?PLAYLIST_ID=2V3INVBX8AY1GT2UXTUKUT&BODY-JSON=%7B%0D%0A++%22PUBLIC%22%3A+TRUE%0D%0A%7D&USER_ID=JMPEREZPEREZ)

Get User's Followed Artists

Get the current user's followed artists.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `user-follow-modify` scope.	**Required**

QUERY PARAMETER	REQUIRED
`type` The ID type: currently only `artist` is supported.	**Required**
`limit` The maximum number of items to return. Default: 20. Minimum: 1. Maximum: 50.	*Optional*
`after` The last artist ID retrieved from the previous request.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an `artists` object. The `artists` object in turn contains a cursor-based paging object (/documentation/web-api/reference/object-model/#cursor-based-paging-object) of Artists (/documentation/web-api/reference/object-model/#artist-object-full). On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-FOLLOWING/?TYPE=ARTIST&LIMIT=20)

Unfollow Artists or Users

Remove the current user as a follower of one or more artists or other Spotify users.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials. Requires the `user-follow-modify` scope.	**Required**
`Content-Type` *Required if IDs are passed in the request body, otherwise ignored.* The content type of the request body: `application/json` .	*Optional*

QUERY PARAMETER	REQUIRED
`type` The ID type: either `artist` or `user` .	**Required**
`ids` A comma-separated list of the artist or the user Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `ids=74ASZWbe4lXaubB36ztrGX,08td7MxkoHQkXnWAYD8d6Q` . A maximum of 50 IDs can be sent in one request.	**Required**

JSON BODY PARAMETER	REQUIRED
`ids` A JSON array of the artist or user Spotify IDs (/documentation/web-api/#spotify-uris-and-ids). For example: `{ids:["74ASZWbe4lXaubB36ztrGX", "08td7MxkoHQkXnWAYD8d6Q"]}` . A maximum of 50 IDs can be sent in one request. *Note: if the* `ids` *parameter is present in the query string, any IDs listed here in the body will be ignored.*	*Optional*

Response

On success, the HTTP status code in the response header is `204` No Content and the response body is empty. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/DELETE-FOLLOWING/?TYPE=USER&IDS=EXAMPLEUSER01)

Unfollow Playlist

Remove the current user as a follower of a playlist.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the user. Unfollowing a publicly followed playlist for a user requires authorization of the `playlist-modify-public` scope; unfollowing a privately followed playlist requires the `playlist-modify-private` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes). *Note that the scopes you provide relate only to whether the current user is following the playlist publicly or privately (i.e. showing others what they are following), not whether the playlist itself is public or private.*	**Required**

PATH PARAMETER	REQUIRED
`{playlist_id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) of the playlist that is to be no longer followed.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body is empty. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/DELETE-PLAYLIST-FOLLOWERS/?PLAYLIST_ID=2V3INVBX8AY1GT2UXTUKUT&USER_ID=JMPEREZPEREZ)

Artists API

↑ Back to top

Get Multiple Artists

Get Spotify catalog information for several artists based on their Spotify IDs.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

QUERY PARAMETER	REQUIRED
`ids` A comma-separated list of the Spotify IDs (/documentation/web-api/#spotify-uris-and-ids) for the artists. Maximum: 50 IDs.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object whose key is `"artists"` and whose value is an array of artist objects (/documentation/web-api/reference/object-model/#artist-object-full) in JSON format.

Objects are returned in the order requested. If an object is not found, a `null` value is returned in the appropriate position. Duplicate `ids` in the query will result in duplicate objects in the response. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-SEVERAL-ARTISTS/?IDS=0OSGXFWSNNOXHD2FKUZ2GY,3DBVYJ7JUOMT4GE9607QIN)

Get an Artist

Get Spotify catalog information for a single artist identified by their unique Spotify ID.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID of the artist.	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an artist object (/documentation/web-api/reference/object-model/#artist-object-full) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ARTIST/?ID=0ODUWJ0SBJDRQHYGGUXECF)

Get an Artist's Albums

Get Spotify catalog information about an artist's albums. Optional parameters can be specified in the query string to filter and sort the response.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the artist.	**Required**

QUERY PARAMETER	REQUIRED
`include_groups` A comma-separated list of keywords that will be used to filter the response. If not supplied, all album types will be returned. Valid values are: - `album` - `single` - `appears_on` - `compilation` For example: `include_groups=album,single` .	*Optional*
`market` Synonym for `country` . An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Supply this parameter to limit the response to one particular geographical market. For example, for albums available in Sweden: `market=SE` . *If not given, results will be returned for all markets and you are likely to get duplicate results per album, one for each market in which the album is available!*	*Optional*
`limit` The number of album objects to return. Default: 20. Minimum: 1. Maximum: 50. For example: `limit=2`	*Optional*
`offset` The index of the first album to return. Default: 0 (i.e., the first album). Use with `limit` to get the next set of albums.	*Optional*

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an array of simplified album objects (/documentation/web-api/reference/object-model/#album-object-simplified) (wrapped in a paging object (/documentation/web-api/reference/object-model/#paging-object)) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ARTIST-ALBUMS/?ALBUM_TYPE=SINGLE&LIMIT=2&MARKET=ES&ID=1VCWHAC5F2US3YHPWWBIA6)

Get an Artist's Top Tracks

Get Spotify catalog information about an artist's top tracks by country.

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the artist	**Required**

QUERY PARAMETER	REQUIRED
`market` An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . Synonym for `country` .	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object whose key is `"tracks"` and whose value is an array of up to 10 track objects (/documentation/web-api/reference/object-model/#track-object-full) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ARTIST-TOP-TRACKS/?COUNTRY=SE&ID=43ZHCT0CAZBISJO8DG9PNE)

Get an Artist's Related Artists

Get Spotify catalog information about artists similar to a given artist. Similarity is based on analysis of the Spotify community's listening history (http://news.spotify.com/se/2010/02/03/related-artists/).

Request

HEADER	REQUIRED
`Authorization` A valid user access token or your client credentials.	**Required**

PATH PARAMETER	REQUIRED
`{id}` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the artist	**Required**

Response

On success, the HTTP status code in the response header is `200` OK and the response body contains an object whose key is `"artists"` and whose value is an array of up to 20 artist objects (/documentation/web-api/reference/object-model/#artist-object-full) in JSON format. On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/GET-ARTIST-RELATED-ARTISTS/?ID=43ZHCT0CAZBISJO8DG9PNE)

Search API

↑ Back to top

Search for an Item

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

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details.	**Required**

QUERY PARAMETER	REQUIRED

QUERY PARAMETER	REQUIRED
`q` Search query keywords and optional field filters and operators. For example: `q=roadhouse%20blues` .	**Required**
`type` A comma-separated list of item types to search across. Valid types are: `album` , `artist` , `playlist` , and `track` . Search results include hits from all the specified item types. For example: `q=name:abacab&type=album,track` returns both albums ***and*** tracks with "abacab" included in their name.	**Required**
`market` An ISO 3166-1 alpha-2 country code (https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) or the string `from_token` . If a country code is specified, only artists, albums, and tracks with content that is playable in that market is returned. **Note**: - 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.	*Optional*
`limit` Maximum number of results to return. Default: 20 Minimum: 1 Maximum: 50 **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.	*Optional*
`offset` The index of the first result to return. Default: 0 (the first result). Maximum offset (including limit): 10,000. Use with limit to get the next page of search results.	*Optional*
`include_external` Possible values: *audio* If *include_external=audio* is specified the response will include any relevant audio content that is hosted externally. By default external content is filtered out from responses.	*Optional*

Response

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).

Notes

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.

Notes

- 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.

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

Personalization API

↑ Back to top

Get a User's Top Artists and Tracks

Get the current user’s top artists or tracks based on calculated affinity.

Request

HEADER	REQUIRED
`Authorization` A valid access token from the Spotify Accounts service: see the Web API Authorization Guide (/documentation/general/guides/authorization-guide/) for details. The access token must have been issued on behalf of the current user. Getting details of a user's top artists and tracks requires authorization of the `user-top-read` scope. See Using Scopes (/documentation/general/guides/authorization-guide/#list-of-scopes).	**Required**

PATH PARAMETER	REQUIRED
`{type}` The type of entity to return. Valid values: `artists` or `tracks`	**Required**

QUERY PARAMETER	REQUIRED
`type` The number of entities to return. Default: 20. Minimum: 1. Maximum: 50. For example: `limit=2`	*Optional*
`offset` The index of the first entity to return. Default: 0 (i.e., the first track). Use with limit to get the next set of entities.	*Optional*
`time_range` Over what time frame the affinities are computed. Valid values: `long_term` (calculated from several years of data and including all new data as it becomes available), `medium_term` (approximately last 6 months), `short_term` (approximately last 4 weeks). Default: `medium_term`	*Optional*

Response

On success, the HTTP status code in the response header is `200 OK` and the response body contains a paging object (/documentation/web-api/reference/object-model/#paging-object) of Artists (/documentation/web-api/reference/object-model/#artist-object-full) or Tracks (/documentation/web-api/reference/object-model/#track-object-full). On error, the header status code is an error code (/documentation/web-api/#response-status-codes) and the response body contains an error object (/documentation/web-api/#response-schema).

TRY IN OUR WEB CONSOLE (/CONSOLE/PUT-CURRENT-USER-SAVED-TRACKS/?IDS=7OUMYWPWJ422JRCDASZB7P%2C4VQPORUHP5EDPBER92T6LQ%2C2TAKCWOAAZWIXQIJPHIX7B)

Objects Index

↑ Back to top

AlbumObject

KEY	TYPE
`album_type` The type of the album: `album` , `single` , or `compilation` .	String
`artists` The artists of the album. Each artist object includes a link in `href` to more detailed information about the artist.	Array[ArtistObject]
`available_markets` The markets in which the album is available: ISO 3166-1 alpha-2 country codes (http://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.	Array[String]
`copyrights` The copyright statements of the album.	Array[CopyrightObject]
`external_ids` Known external IDs for the album.	ExternalIdObject
`external_urls` Known external URLs for this album.	ExternalUrlObject

KEY	TYPE
`genres` A list of the genres used to classify the album. For example: "Prog Rock" , "Post-Grunge". (If not yet classified, the array is empty.)	Array[String]
`href` A link to the Web API endpoint providing full details of the album.	String
`id` The Spotify ID for the album.	String
`images` The cover art for the album in various sizes, widest first.	Array[ImageObject]
`label` The label for the album.	String
`name` The name of the album. In case of an album takedown, the value may be an empty string.	String
`popularity` The popularity of the album. The value will be between 0 and 100, with 100 being the most popular. The popularity is calculated from the popularity of the album's individual tracks.	Integer
`release_date` The date the album was first released, for example "1981-12-15". Depending on the precision, it might be shown as "1981" or "1981-12".	String
`release_date_precision` The precision with which release_date value is known: "year" , "month" , or "day".	String
`tracks` The tracks of the album.	Array[SimplifiedTrackObject]
`type` The object type: "album"	String
`uri` The Spotify URI for the album.	String

ArtistObject

KEY	TYPE
`external_urls` Known external URLs for this artist.	ExternalUrlObject
`followers` Information about the followers of the artist.	FollowersObject
`genres` A list of the genres the artist is associated with. For example: `"Prog Rock"` , `"Post-Grunge"` . (If not yet classified, the array is empty.)	Array[String]
`href` A link to the Web API endpoint providing full details of the artist.	String
`id` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the artist.	String
`images` Images of the artist in various sizes, widest first.	Array[ImageObject]
`name` The name of the artist.	String
`popularity` 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.	Integer
`type` The object type: `"artist"`	String

KEY	TYPE
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the artist.	String

AudioFeaturesObject

KEY	TYPE
`acousticness` A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	Float
`analysis_url` An HTTP URL to access the full audio analysis of this track. An access token is required to access this data.	String
`danceability` Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.	Float
`duration_ms` The duration of the track in milliseconds.	Integer
`energy` Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.	Float
`id` The Spotify ID for the track.	String
`instrumentalness` Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.	Float
`key` The key the track is in. Integers map to pitches using standard Pitch Class notation (https://en.wikipedia.org/wiki/Pitch_class) . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.	Integer
`liveness` Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.	Float
`loudness` The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.	Float
`mode` Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.	Integer
`speechiness` Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.	Float
`tempo` The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.	Float
`time_signature` An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).	Integer
`track_href` A link to the Web API endpoint providing full details of the track.	String

KEY	TYPE
`type` The object type: "audio_features"	String
`uri` The Spotify URI for the track.	String
`valence` A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).	Float

CategoryObject

KEY	TYPE
`href` A link to the Web API endpoint returning full details of the category.	String
`icons` The category icon, in various sizes.	Array[ImageObject]
`id` The Spotify category ID (/documentation/web-api/#spotify-uris-and-ids) of the category.	String
`name` The name of the category.	String

ContextObject

KEY	TYPE
`type` The object type, e.g. "artist", "playlist", "album".	String
`href` A link to the Web API endpoint providing full details of the track.	String
`external_urls` External URLs for this context.	ExternalUrlObject
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the context.	String

CurrentlyPlayingObject

KEY	TYPE
`context` A Context Object. Can be `null` .	ContextObject
`timestamp` Unix Millisecond Timestamp when data was fetched	Integer
`progress_ms` Progress into the currently playing track. Can be `null` .	Integer
`is_playing` If something is currently playing, return `true` .	Boolean
`item` The currently playing track. Can be `null` .	TrackObject

KEY	TYPE
`currently_playing_type` The object type of the currently playing item. Can be one of `track` , `episode` , `ad` or `unknown` .	String

CursorObject

KEY	TYPE
`after` The cursor to use as key to find the next page of items.	String

CursorPagingObject

KEY	TYPE
`href` A link to the Web API endpoint returning the full result of the request.	String
`items` The requested data.	Array[Object]
`limit` The maximum number of items in the response (as set in the query or by default).	Integer
`next` URL to the next page of items. (`null` if none)	String
`cursors` The cursors used to find the next set of items.	CursorObject
`total` The total number of items available to return.	Integer

DeviceObject

KEY	TYPE
`id` The device ID. This may be `null` .	String
`is_active` If this device is the currently active device.	Boolean
`is_private_session` If this device is currently in a private session.	Boolean
`is_restricted` Whether controlling this device is restricted. At present if this is “true” then no Web API commands will be accepted by this device.	Boolean
`name` The name of the device.	String
`type` Device type, such as “computer”, “smartphone” or “speaker”.	String
`volume_percent` The current volume in percent. This may be null.	Integer

DevicesObject

KEY	TYPE

KEY	TYPE
`devices` A list of 0..n Device objects	Array[DeviceObject]

ExternalIdObject

KEY	TYPE
`isrc` International Standard Recording Code (http://en.wikipedia.org/wiki/International_Standard_Recording_Code)	String
`ean` International Article Number (http://en.wikipedia.org/wiki/International_Article_Number_%28EAN%29)	String
`upc` Universal Product Code (http://en.wikipedia.org/wiki/Universal_Product_Code)	String

PagingObject

KEY	TYPE
`href` A link to the Web API endpoint returning the full result of the request	String
`items` The requested data.	Array[Object]
`limit` The maximum number of items in the response (as set in the query or by default).	Integer
`next` URL to the next page of items. (`null` if none)	String
`offset` The offset of the items returned (as set in the query or by default)	Integer
`previous` URL to the previous page of items. (`null` if none)	String
`total` The total number of items available to return.	Integer

PlayHistoryObject

KEY	TYPE
`track` The track the user listened to.	SimplifiedTrackObject
`played_at` The date and time the track was played.	Timestamp
`context` The context the track was played from.	ContextObject

PlaylistObject

KEY	TYPE
`collaborative` `true` if the owner allows other users to modify the playlist.	Boolean

KEY	TYPE
`external_urls` Known external URLs for this playlist.	ExternalUrlObject
`href` A link to the Web API endpoint providing full details of the playlist.	String
`id` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	String
`images` 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/). *Note: If returned, the source URL for the image (`url`) is temporary and will expire in less than a day.*	Array[ImageObject]
`name` The name of the playlist.	String
`owner` The user who owns the playlist	PublicUserObject
`public` The playlist's public/private status: `true` the playlist is public, `false` the playlist is private, `null` the playlist status is not relevant. For more about public/private status, see Working with Playlists (/documentation/general/guides/working-with-playlists/)	Boolean
`snapshot_id` The version identifier for the current playlist. Can be supplied in other requests to target a specific playlist version	String
`tracks` A collection containing a link (`href`) 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.	Array[Tracks]
`type` The object type: "playlist"	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the playlist.	String

PlaylistTrackObject

KEY	TYPE
`added_at` The date and time the track was added. *Note that some very old playlists may return `null` in this field.*	Timestamp
`added_by` The Spotify user who added the track. *Note that some very old playlists may return `null` in this field.*	PublicUserObject
`is_local` Whether this track is a local file (https://developer.spotify.com/web-api/local-files-spotify-playlists/) or not.	Boolean
`track` Information about the track.	TrackObject

PrivateUserObject

KEY	TYPE
`country` The country of the user, as set in the user's account profile. An ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2). *This field is only available when the current user has granted access to the user-read-private (/documentation/general/guides/authorization-guide/#list-of-scopes) scope.*	String
`display_name` The name displayed on the user's profile. `null` if not available.	String

KEY	TYPE
`email` The user's email address, as entered by the user when creating their account. ***Important!*** *This email address is unverified; there is no proof that it actually belongs to the user. This field is only available when the current user has granted access to the user-read-email (/documentation/general/guides/authorization-guide/#list-of-scopes) scope.*	String
`external_urls` Known external URLs for this user.	ExternalUrlObject
`followers` Information about the followers of the user.	FollowersObject
`href` A link to the Web API endpoint for this user.	String
`id` The Spotify user ID (/documentation/web-api/#spotify-uris-and-ids) for the user.	String
`images` The user's profile image.	Array[ImageObject]
`product` The user's Spotify subscription level: "premium", "free", etc. (The subscription level "open" can be considered the same as "free".) *This field is only available when the current user has granted access to the user-read-private (/documentation/general/guides/authorization-guide/#list-of-scopes) scope.*	String
`type` The object type: "user"	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the user.	String

PublicUserObject

KEY	TYPE
`display_name` The name displayed on the user's profile. `null` if not available.	String
`external_urls` Known public external URLs for this user.	ExternalUrlObject
`followers` Information about the followers of this user.	FollowersObject
`href` A link to the Web API endpoint for this user.	String
`id` The Spotify user ID (/documentation/web-api/#spotify-uris-and-ids) for this user.	String
`images` The user's profile image.	Array[ImageObject]
`type` The object type: "user"	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for this user.	String

RecommendationSeedObject

KEY	TYPE
`afterFilteringSize` The number of tracks available after min_* and max_* filters have been applied.	Integer

KEY	TYPE
`afterRelinkingSize` The number of tracks available after relinking for regional availability.	Integer
`href` A link to the full track or artist data for this seed. For tracks this will be a link to a Track Object (/documentation/web-api/reference/object-model/#track-object-full). For artists a link to an Artist Object (/documentation/web-api/reference/object-model/#artist-object-full). For genre seeds, this value will be `null` .	String
`id` The id used to select this seed. This will be the same as the string used in the `seed_artists` , `seed_tracks` or `seed_genres` parameter.	String
`initialPoolSize` The number of recommended tracks available for this seed.	Integer
`type` The entity type of this seed. One of `artist` , `track` or `genre` .	String

RecommendationsResponseObject

KEY	TYPE
`seeds` An array of recommendation seed objects (/documentation/web-api/reference/object-model/#recommendations-seed-object).	Array[RecommendationSeedObject]
`tracks` An array of track object (simplified) (/documentation/web-api/reference/object-model/#track-object-simplified) ordered according to the parameters supplied.	Array[SimplifiedTrackObject]

SavedAlbumObject

KEY	TYPE
`added_at` The date and time the album was saved Timestamps are returned in ISO 8601 format as Coordinated Universal Time (UTC) with a zero offset: YYYY-MM-DDTHH:MM:SSZ. If the time is imprecise (for example, the date/time of an album release), an additional field indicates the precision; see for example, release_date in an album object.	Timestamp
`album` Information about the album.	AlbumObject

SavedTrackObject

KEY	TYPE
`added_at` The date and time the track was saved. Timestamps are returned in ISO 8601 format as Coordinated Universal Time (UTC) with a zero offset: YYYY-MM-DDTHH:MM:SSZ. If the time is imprecise (for example, the date/time of an album release), an additional field indicates the precision; see for example, release_date in an album object.	Timestamp
`track` Information about the track.	TrackObject

SimplifiedAlbumObject

KEY	TYPE

KEY	TYPE
`album_group` The field is present when getting an artist's albums. Possible values are "album", "single", "compilation", "appears_on". Compare to album_type this field represents relationship between the artist and the album.	String
`album_type` The type of the album: one of "album", "single", or "compilation".	String
`artists` The artists of the album. Each artist object includes a link in `href` to more detailed information about the artist.	Array[SimplifiedArtistObject]
`available_markets` The markets in which the album is available: ISO 3166-1 alpha-2 country codes (http://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.	Array[String]
`external_urls` Known external URLs for this album.	ExternalUrlObject
`href` A link to the Web API endpoint providing full details of the album.	String
`id` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the album.	String
`images` The cover art for the album in various sizes, widest first.	Array[ImageObject]
`name` The name of the album. In case of an album takedown, the value may be an empty string.	String
`type` The object type: "album"	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the album.	String

SimplifiedTrackObject

KEY	TYPE
`artists` The artists who performed the track. Each artist object includes a link in `href` to more detailed information about the artist.	Array[SimplifiedArtistObject]
`available_markets` 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.	Array[String]
`disc_number` The disc number (usually `1` unless the album consists of more than one disc).	Integer
`duration_ms` The track length in milliseconds.	Integer
`explicit` Whether or not the track has explicit lyrics (`true` = yes it does; `false` = no it does not OR unknown).	Boolean
`external_urls` External URLs for this track.	ExternalUrlObject
`href` A link to the Web API endpoint providing full details of the track.	String
`id` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.	String
`is_playable` 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` .	Boolean

KEY	TYPE
`linked_from` Part of the response when Track Relinking (/documentation/general/guides/track-relinking-guide/) is applied and is only part of the response if the track linking, in fact, exists. The requested track has been replaced with a different track. The track in the `linked_from` object contains information about the originally requested track.	LinkedTrackObject
`name` The name of the track.	String
`preview_url` A URL to a 30 second preview (MP3 format) of the track.	String
`track_number` The number of the track. If an album has several discs, the track number is the number on the specified disc.	Integer
`type` The object type: "track".	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the track.	String

TrackObject

KEY	TYPE
`album` The album on which the track appears. The album object includes a link in `href` to full information about the album.	SimplifiedAlbumObject
`artists` The artists who performed the track. Each artist object includes a link in `href` to more detailed information about the artist.	Array[ArtistObject]
`available_markets` 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.	Array[String]
`disc_number` The disc number (usually `1` unless the album consists of more than one disc).	Integer
`duration_ms` The track length in milliseconds.	Integer
`explicit` Whether or not the track has explicit lyrics (`true` = yes it does; `false` = no it does not OR unknown).	Boolean
`external_ids` Known external IDs for the track.	ExternalIdObject
`external_urls` Known external URLs for this track.	ExternalUrlObject
`href` A link to the Web API endpoint providing full details of the track.	String
`id` The Spotify ID (/documentation/web-api/#spotify-uris-and-ids) for the track.	String
`is_playable` 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` .	Boolean
`linked_from` 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.	TrackObject
`restrictions` Part of the response when Track Relinking (/documentation/general/guides/track-relinking-guide/) is applied, the original track is not available in the given market, and Spotify did not have any tracks to relink it with. The track response will still contain metadata for the original track, and a restrictions object containing the reason why the track is not available: `"restrictions" : {"reason" : "market"}`	Array[TrackRestrictionObject]

KEY	TYPE
`name` The name of the track.	String
`popularity` The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity. Note that the popularity value may lag actual popularity by a few days: the value is not updated in real time.	Integer
`preview_url` A link to a 30 second preview (MP3 format) of the track. Can be `null`	String
`track_number` The number of the track. If an album has several discs, the track number is the number on the specified disc.	Integer
`type` The object type: “track”.	String
`uri` The Spotify URI (/documentation/web-api/#spotify-uris-and-ids) for the track.	String

TuneableTrackObject

KEY	TYPE
`acousticness` A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	Float
`danceability` Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.	Float
`duration_ms` The duration of the track in milliseconds.	Integer
`energy` Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.	Float
`instrumentalness` Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.	Float
`key` The key the track is in. Integers map to pitches using standard Pitch Class notation (https://en.wikipedia.org/wiki/Pitch_class). E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.	Integer
`liveness` Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.	Float
`loudness` The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.	Float
`mode` Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.	Integer

KEY	TYPE
`popularity` The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. *Note: When applying track relinking via the* `market` *parameter, it is expected to find relinked tracks with popularities that do not match* `min_*` *,* `max_*` *and* `target_*` *popularities. These relinked tracks are accurate replacements for unplayable tracks with the expected popularity scores. Original, non-relinked tracks are available via the* `linked_from` *attribute of the relinked track response (/documentation/general/guides/track-relinking-guide).*	Float
`speechiness` Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.	Float
`tempo` The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.	Float
`time_signature` An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).	Integer
`valence` A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).	Float

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/)

DISCOVER (/DISCOVER/)

CONSOLE (/CONSOLE/)

DASHBOARD (/DASHBOARD/)

USE CASES (/USE-CASES)

DESIGN & BRANDING GUIDELINES (/BRANDING-GUIDELINES)

LEGAL (/TERMS/)

Terms of Service (/terms/)

Third Party Licenses (/legal/third-party-licenses/)

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