

---

# **ytmusicapi**

***Release 0.4.3***

**Mar 23, 2020**



---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contents</b>	<b>7</b>
3.1	Setup . . . . .	7
3.2	Usage . . . . .	8
3.3	Reference . . . . .	8
	<b>Index</b>	<b>13</b>



The purpose of this library is to automate interactions with [YouTube Music](#), such as retrieving your library content or creating large playlists.

**This project is not supported nor endorsed by Google**



# CHAPTER 1

---

## Features

---

- Library management: list, create, delete, and modify playlists and playlist items
- Search: Search for songs on YouTube Music





## CHAPTER 2

---

### Usage Example

---

For a complete documentation of available functions, refer to the *Reference*

```
from ytmusicapi import YTMusic

ytmusic = YTMusic('headers_auth.json')
playlistId = ytmusic.create_playlist("test", "test description")
search_results = ytmusic.search("Oasis Wonderwall")
ytmusic.add_playlist_items(playlistId, [search_results[0]['videoId']])
```



### 3.1 Setup

```
pip install ytmusicapi
```

#### Authenticated requests

To run authenticated requests you need to copy your request headers from a POST request in your YTMusic Web Client.

To do so, follow these steps:

- Copy `headers_auth.json.example` (see below) to `headers_auth.json`
- Open YTMusic in Firefox
- Go to the developer tools (Ctrl-Shift-I) and right click a POST request.
- Copy the request headers (right click > copy > copy request headers)
- Paste the three missing items to `headers_auth.json`

#### `headers_auth.json.example`

```
{
  "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101_
↪Firefox/72.0",
  "Accept": "*/*",
  "Accept-Language": "en-US,en;q=0.5",
  "Content-Type": "application/json",
  "X-Goog-AuthUser": "0",
  "x-origin": "https://music.youtube.com",
  "Authorization": "PASTE_AUTHORIZATION",
  "X-Youtube-Identity-Token": "PASTE_IDENTITY",
  "Cookie" : "PASTE_COOKIE"
}
```

## 3.2 Usage

**Unauthenticated** requests for retrieving playlist content or searching:

```
from ytmusicapi import YTMusic

ytmusic = YTMusic()
```

If an endpoint requires authentication you will receive an error: Please provide authentication before using this function

For authenticated requests you need to set up your credentials first: [Setup](#)

**Authenticated** request to interact with your playlist:

```
from ytmusicapi import YTMusic

ytmusic = YTMusic('headers_auth.json')
```

Detailed example with authenticated requests:

```
from ytmusicapi import YTMusic

ytmusic = YTMusic('headers_auth.json')
playlistId = ytmusic.create_playlist("test", "test description")
search_results = ytmusic.search("Oasis Wonderwall")
ytmusic.add_playlist_items(playlistId, [search_results[0]['videoId']])
```

## 3.3 Reference

Reference for the YTMusic class.

**class** ytmusicapi.YTMusic(auth="")

Allows automated interactions with YouTube Music by emulating the YouTube web client's requests. Permits both authenticated and non-authenticated requests. Authentication header data must be provided on initialization.

YTMusic.\_\_init\_\_(auth="")

Create a new instance to interact with YouTube Music.

**Parameters** **auth** – Optional. Provide authentication credentials to manage your library. Should be an adjusted version of *headers\_auth.json.example* in the project root. Default: A default header is used without authentication.

YTMusic.search(query, filter=None)

Search YouTube music Returns up to 20 results within the provided category. By default only songs (audio-only) are returned

**Parameters**

- **query** – Query string, i.e. 'Oasis Wonderwall'
- **filter** – Filter for item types. Allowed values: 'songs', 'videos', 'albums', 'artists', 'playlists'. Default: Default search, including all types of items.

**Returns**

List of results depending on filter. `resultType` specifies the type of item (important for default search). albums, artists and playlists additionally contain a `browseId`, corresponding to `albumId`, `channelId` and `playlistId` (`browseId`='VL'+`playlistId`)

Example list:

```
[
  {
    'videoId': 'ZrOKjDZotkA',
    'artist': 'Oasis',
    'title': 'Wonderwall (Remastered)',
    'resultType': 'song'
  },
  {
    'videoId': 'Gvfgut8nAgw',
    'artist': 'Oasis',
    'title': 'Wonderwall',
    'resultType': 'song'
  }
]
```

**YTMusic.get\_playlists()**

Retrieves the content of the ‘Library’ page

**Returns** List of owned playlists.

Each item is in the following format:

```
{
  'playlistId': 'PLQwVilKxHM6rz0fDJVv_0U1XGEWf-bFys',
  'title': 'Playlist title'
}
```

**YTMusic.get\_liked\_songs(limit=1000)**

Gets playlist items for the ‘Liked Songs’ playlist

**Parameters** `limit` – How many items to return. Default: 1000

**Returns** List of `playlistItem` dictionaries. See `get_playlist_items()`

**YTMusic.get\_history()**

Gets your play history in reverse chronological order

**Returns** List of `playlistItems`, see `get_playlist_items()` The additional property ‘played’ indicates when the `playlistItem` was played

**YTMusic.rate\_song(videoId, rating='INDIFFERENT')**

Rates a song (“thumbs up”/“thumbs down” interactions on YouTube Music)

**Parameters**

- **videoId** – Video id
- **rating** – One of ‘LIKE’, ‘DISLIKE’, ‘INDIFFERENT’

‘INDIFFERENT’ removes the previous rating and assigns no rating

**YTMusic.get\_playlist\_items(playlistId, limit=1000)**

Returns a list of playlist items

**Parameters**

- **playlistId** – Playlist id
- **limit** – How many songs to return. Default: 1000

**Returns** List of playlistItem dictionaries

Each item is in the following format:

```
{
  'videoId': 'PLQwVilKxHM6rz0fDJVv_0U1XGEWf-bFys',
  'artist': 'Artist',
  'title': 'Song Title',
  'setVideoId': '56B44F6D10557CC6'
}
```

The setVideoId is the unique id of this playlist item and needed for moving/removing playlist items

`YTMusic.create_playlist` (*title, description, privacy\_status='PRIVATE'*)

Creates a new empty playlist and returns its id.

**Parameters**

- **title** – Playlist title
- **description** – Playlist description
- **privacy\_status** – Playlists can be 'PUBLIC', 'PRIVATE', or 'UNLISTED'. Default: 'PRIVATE'

**Returns** ID of the YouTube playlist

`YTMusic.edit_playlist` (*playlistId, title=None, description=None, privacyStatus=None*)

Edit title, description or privacyStatus of a playlist.

**Parameters**

- **playlistId** – Playlist id
- **title** – Optional. New title for the playlist
- **description** – Optional. New description for the playlist
- **privacyStatus** – Optional. New privacy status for the playlist

**Returns** Status String or full response

`YTMusic.delete_playlist` (*playlistId*)

Delete a playlist.

**Parameters** **playlistId** – Playlist id

**Returns** Status String or full response

`YTMusic.add_playlist_items` (*playlistId, videoIds*)

Add songs to an existing playlist

**Parameters**

- **playlistId** – Playlist id
- **videoIds** – List of Video ids

**Returns** Status String or full response

`YTMusic.remove_playlist_items` (*playlistId, videos*)

Remove songs from an existing playlist

**Parameters**

- **playlistId** – Playlist id
- **videos** – List of Dictionaries containing video information. Must contain videoId and setVideoId

**Returns** Status String or full response





## Symbols

`__init__()` (*ytmusicapi.YTMusic method*), 8

## A

`add_playlist_items()` (*ytmusicapi.YTMusic method*), 10

## C

`create_playlist()` (*ytmusicapi.YTMusic method*), 10

## D

`delete_playlist()` (*ytmusicapi.YTMusic method*), 10

## E

`edit_playlist()` (*ytmusicapi.YTMusic method*), 10

## G

`get_history()` (*ytmusicapi.YTMusic method*), 9

`get_liked_songs()` (*ytmusicapi.YTMusic method*), 9

`get_playlist_items()` (*ytmusicapi.YTMusic method*), 9

`get_playlists()` (*ytmusicapi.YTMusic method*), 9

## R

`rate_song()` (*ytmusicapi.YTMusic method*), 9

`remove_playlist_items()` (*ytmusicapi.YTMusic method*), 10

## S

`search()` (*ytmusicapi.YTMusic method*), 8

## Y

`YTMusic` (*class in ytmusicapi*), 8