
ytmusicapi

Release 0.5.1

Apr 12, 2020

Contents

1	Features	3
2	Usage Example	5
3	Contents	7
3.1	Setup	7
3.2	Usage	8
3.3	Reference	8
	Index	13

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
- Uploads: Upload songs, list uploaded songs and delete uploaded songs

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']])
```


CHAPTER 3

Contents

3.1 Setup

3.1.1 Installation

```
pip install ytmusicapi
```

3.1.2 Authenticated requests

To run authenticated requests you need to set up you need to copy your request headers from a POST request in your YTMusic Web Client. To do so, follow these steps:

- Open <https://music.youtube.com> in Firefox
- Go to the developer tools (Ctrl-Shift-I) and find an authenticated POST request. You can filter for /browse to easily find a suitable request.
- Copy the request headers (right click > copy > copy request headers)

Now call `YTMusic.setup()` and paste the request headers and it will create configuration file in the correct format in the current directory.

3.1.3 Manual file creation

Alternatively, you can paste the three missing items to *headers_auth.json* below and create your own file:

```
{
  "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",
```

(continues on next page)

(continued from previous page)

```
"Content-Type": "application/json",
"X-Goog-AuthUser": "0",
"x-origin": "https://music.youtube.com",
"Authorization": "PASTE_AUTHORIZATION",
"Cookie" : "PASTE_COOKIE"
}
```

3.2 Usage

3.2.1 Unauthenticated

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

3.2.2 Authenticated

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

After you have created the authentication JSON, you can instantiate the class:

```
from ytmusicapi import YTMusic
ytmusic = YTMusic('headers_auth.json')
```

With the `ytmusic` instance you can now perform authenticated requests:

```
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.

3.3.1 Setup

See also the [Setup](#) page

classmethod `YTMusic.setup()`

Requests browser headers from the user and stores a configuration JSON file in the correct format in the current directory

3.3.2 Search

`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'
  }
]
```

3.3.3 Library

`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

3.3.4 Playlists

`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_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 PlaylistItems, see `get_playlist_items()`. Must contain videoId and setVideoId

Returns Status String or full response

3.3.5 Uploads

`YTMusic.get_uploaded_songs (limit=25)`

Returns a list of uploaded songs

Parameters **limit** – How many songs to return. Default: 25

Returns List of uploaded songs.

Each item is in the following format:

```
{
  "entityId": "t_po_CICr2crg7OWpchDpjPjrBA",
  "videoId": "Uise6RPKoek",
  "artist": "Coldplay",
  "title": "A Sky Full Of Stars",
  "album": "Ghost Stories"
}
```

`YTMusic.upload_song(filepath)`

Uploads a song to YouTube Music

Parameters `filepath` – Path to the music file

Returns Status String or full response

`YTMusic.delete_uploaded_song(uploaded_song)`

Deletes a previously uploaded song

Parameters `uploaded_song` – The uploaded song to delete, e.g. `get_uploaded_songs()`

Returns Status String or error

Symbols

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

A

`add_playlist_items()` (*ytmusicapi.YTMusic method*), 11

C

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

D

`delete_playlist()` (*ytmusicapi.YTMusic method*), 11

`delete_uploaded_song()` (*ytmusicapi.YTMusic method*), 12

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*), 10

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

`get_uploaded_songs()` (*ytmusicapi.YTMusic method*), 11

R

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

`remove_playlist_items()` (*ytmusicapi.YTMusic method*), 11

S

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

`setup()` (*ytmusicapi.YTMusic class method*), 9

U

`upload_song()` (*ytmusicapi.YTMusic method*), 11

Y

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