youtube-data-api Documentation

Release 0.0.17

Leon Yin, Megan Brown

Jan 04, 2021

User Guide / Tutorial:

YouTube is a social media platform that is often overlooked in academic research. This package seeks to make this data source more accessible, while introducing new applications and methods to analyze this platform.

This client is built for GET requests from public data on YouTube. It does not work for updating data on YouTube Channels you own, or getting data from managed accounts from the Reporting API.

You can find the software on Github.

Installation

You can download the package from PyPI:

```
pip install youtube-data-api
```

Quickstart

In order to access the API, you'll need to get a service key from the Google Cloud Console.

```
from youtube_api import YouTubeDataAPI

api_key = 'AKAIXXXXXXXX'
yt = YouTubeDataAPI(api_key)

yt.search('alexandria ocasio-cortez')
```

There's a more detailed section on the *Quickstart* page.

CHAPTER 1

Installation

It is recommended to install this module by using pip:

```
pip install youtube-data-api
```

CHAPTER 2

Quickstart

```
import os
import pandas as pd
from youtube_api import YoutubeDataApi
```

In order to access the API, you'll need to get a service key from the Google Cloud Console . I like to store my API keys as environment variables in `~/.bash_profile` so that I don't have to hard code them.:

```
YT_KEY = os.environ.get('YOUTUBE_API_KEY') # you can hardcode this, too.
yt = YouTubeDataAPI(YT_KEY)
```

We now have created a `YouTubeDataAPI` class as `yt`, which can be used to make API calls, such as searching for the most relevant videos of Alexandria Ocasio-Cortez.

```
searches = yt.search(q='alexandria ocasio-cortez',
                     max_results=5)
print(searches[0])
```

```
{'video_id': 'LlillsHgcaw',
 'channel_title': 'Fox News',
 'channel_id': 'UCXIJgqnII2ZOINSWNOGFThA',
 'video_publish_date': datetime.datetime(2019, 2, 19, 4, 57, 51),
 'video_title': 'Rep. Alexandria Ocasio-Cortez taken to task by fellow progressives',
 'video_description': 'New York City Mayor Bill de Blasio criticizes Alexandria␣
↪Ocasio-Cortez over her opposition to the Amazon deal.',
 'video_category': None,
 'video_thumbnail': 'https://i.ytimg.com/vi/LlillsHgcaw/hqdefault.jpg',
 'collection_date': datetime.datetime(2019, 2, 20, 14, 48, 19, 487877)}
```

All API requests are parsed from raw JSON into dictionaries.

Typically an API call returns a list of dictionary objects. This is perfect for converting into Pandas DataFrames, or saving as JSON.

```
df_search = pd.DataFrame(searches)
df_search.head(5)
```

Aside from the default parser, the `parse` argument allows users to create custom functions to parse and process API resonses. You can also get raw JSON from the API by using the *youtube_api.parsers.raw_json()* parser, or setting parser to `None`.

```
yt.search(q='alexandria ocasio-cortez',
          max_results=1,
          parser=None)
```

```
[{'kind': 'youtube#searchResult',
  'etag': '"XpPGQXPnxQJhLgs6enD_n8JR4Qk/aGNxqVTPJsGI6aEI2tVnYhn0vS8"',
  'id': {'kind': 'youtube#video', 'videoId': 'LlillsHgcaw'},
  'snippet': {'publishedAt': '2019-02-19T04:57:51.000Z',
   'channelId': 'UCXIJgqnII2ZOINSWNOGFThA',
   'title': 'Rep. Alexandria Ocasio-Cortez taken to task by fellow progressives',
   'description': 'New York City Mayor Bill de Blasio criticizes Alexandria Ocasio-
↪Cortez over her opposition to the Amazon deal.',
   'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/LlillsHgcaw/default.jpg',
     'width': 120,
     'height': 90},
    'medium': {'url': 'https://i.ytimg.com/vi/LlillsHgcaw/mqdefault.jpg',
     'width': 320,
     'height': 180},
    'high': {'url': 'https://i.ytimg.com/vi/LlillsHgcaw/hqdefault.jpg',
     'width': 480,
     'height': 360}},
   'channelTitle': 'Fox News',
   'liveBroadcastContent': 'none'}}]
```

youtube_api.parsers are intended to allow customized data parsing for those who want it, with robust defaults for less advanced users.

CHAPTER 3

API Guide

The YouTube Data API has a single class `YouTubeDataAPI`, which authenticates your API key and provides functions to interact with the API.

3.1 youtube_api.youtube_api module

This is the client class to access the YouTube API.

class youtube_api.youtube_api.**YouTubeDataAPI**(*key*, *api_version='3'*, *verify_api_key=True*, *verbose=False*, *timeout=20*)

Bases: object

The Youtube Data API handles the keys and methods to access data from the YouTube Data API

param key YouTube Data API key. Get a YouTube Data API key here: https://console.cloud.google.com/apis/dashboard

verify_key()

Checks it the API key is valid.

Returns True if the API key is valid, False if the key is not valid.

Return type bool

get_channel_id_from_user(*username*, ***kwargs*)

Get a channel_id from a YouTube username. These are the unique identifiers for all YouTube "uers". IE. "Munchies" -> "UCaLfMkkHhSA_LaCta0BzyhQ".

Read the docs: https://developers.google.com/youtube/v3/docs/channels/list

Parameters **username** (*str*) – the username for a YouTube channel

Returns YouTube Channel ID for a given username

Return type str

get_channel_metadata_gen(*channel_id, parser=<function parse_channel_metadata>, part=['id', 'snippet', 'contentDetails', 'statistics', 'topicDetails', 'brandingSettings'], **kwargs*)

Gets a dictionary of channel metadata given a channel_id, or a list of channel_ids.

Parameters

- **channel_id** (*str or list*) – channel id(s)
- **parser** (youtube_api.parsers module) – the function to parse the json document.
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns yields the YouTube channel metadata

Return type dict

get_channel_metadata(*channel_id, parser=<function parse_channel_metadata>, part=['id', 'snippet', 'contentDetails', 'statistics', 'topicDetails', 'brandingSettings'], **kwargs*)

Gets a dictionary of channel metadata given a channel_id, or a list of channel_ids.

Read the docs: https://developers.google.com/youtube/v3/docs/channels/list

Parameters

- **channel_id** (*str or list*) – the channel id(s)
- **parser** (youtube_api.parsers module) – the function to parse the json document.
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns the YouTube channel metadata

Return type dict

get_video_metadata_gen(*video_id, parser=<function parse_video_metadata>, part=['statistics', 'snippet'], **kwargs*)

Given a *video_id* returns metrics (views, likes, comments) and metadata (description, category) as a dictionary.

Read the docs: https://developers.google.com/youtube/v3/docs/videos/list

Parameters

- **video_id** (*str or list of str*) – The ID of a video IE: "kNbhUWLH_yY", this can be found at the end of YouTube urls and by parsing links using youtube_api.youtube_api_utils.strip_youtube_id().
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns returns metadata from the inputted ‘‘video_id‘‘s.

Return type dict

get_video_metadata(*video_id, parser=<function parse_video_metadata>, part=['statistics', 'snippet'], **kwargs*)

Given a single or list of *video_id* returns metrics (views, likes, comments) and metadata (description, category) as a dictionary.

Read the docs: https://developers.google.com/youtube/v3/docs/videos/list

Parameters

- **video_id** (*str or list of str*) – the ID of a video IE: ['kNbhUWLH_yY'], this can be found at the end of YouTube urls and by parsing links using `youtube_api.youtube_api_utils.strip_youtube_id()`.
- **parser** (`youtube_api.parsers module`) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns yields a video metadata.

Return type dict

get_playlists(*channel_id, next_page_token=False, parser=<function parse_playlist_metadata>, part=['id', 'snippet', 'contentDetails'], **kwargs*)

Returns a list of playlist IDs that *channel_id* created. Note that playlists can contains videos from any users.

Read the docs: https://developers.google.com/youtube/v3/docs/playlists/list

Parameters

- **channel_id** (*str*) – a channel_id IE: "UCn8zNIfYAQNdrFRrr8oibKw"
- **next_page_token** (*str*) – a token to continue from a preciously stopped query IE: "CDIQAA"
- **parser** (`youtube_api.parsers module`) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns playlist info that `channel_id` is subscribed to.

Return type list of dict

get_videos_from_playlist_id(*playlist_id, next_page_token=None, parser=<function parse_video_url>, part=['snippet'], max_results=200000, **kwargs*)

Given a *playlist_id*, returns *video_ids* associated with that playlist.

Note that user uploads for any given channel are from a playlist named "upload playlist id". You can get this value using `youtube_api.youtube_api.get_channel_metadata()` or *youtube_api.youtube_api_utils.get_upload_playlist_id()*. The playlist ID for uploads is always the channel_id with "UU" subbed for "UC".

Read the docs: https://developers.google.com/youtube/v3/docs/playlistItems

Parameters

- **playlist_id** – the playlist_id IE: "UUaLfMkkHhSA_LaCta0BzyhQ"

- **next_page_token** (*str*) – a token to continue from a preciously stopped query IE: "CDIQAA"
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.
- **max_results** – How many video IDs should returned? Contrary to the name, this is actually the minimum number of results to be returned.

Returns video ids associated with playlist_id.

Return type list of dict

get_subscriptions(*channel_id, next_page_token=False, parser=<function parse_subscription_descriptive>, part=['id', 'snippet'], **kwargs*)

Returns a list of channel IDs that *channel_id* is subscribed to.

Read the docs: https://developers.google.com/youtube/v3/docs/subscriptions

Parameters

- **channel_id** (*str*) – a channel_id IE: "UCn8zNIfYAQNdrFRrr8oibKw"
- **next_page_token** (*str*) – a token to continue from a preciously stopped query IE: "CDIQAA"
- **stop_after_n_iteration** (*int*) – stops the API calls after N API calls
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns channel IDs that channel_id is subscribed to.

Return type list

get_featured_channels_gen(*channel_id, parser=<function parse_featured_channels>, part=['id', 'brandingSettings'], **kwargs*)

Given a *channel_id* returns a dictionary {channel_id : [list, of, channel_ids]} of featured channels.

Optionally, can take a list of channel IDS, and returns a list of dictionaries.

Read the docs: https://developers.google.com/youtube/v3/docs/channels/list

Parameters

- **channel_id** (*str of list of str*) – channel_ids IE: ['UCn8zNIfYAQNdrFRrr8oibKw']
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns yields metadata for featured channels

Return type dict

get_featured_channels(*channel_id*, *parser=<function parse_featured_channels>*, ***kwargs*)
Given a *channel_id* returns a dictionary {channel_id : [list, of, channel_ids]} of featured channels.

Optionally, can take a list of channel IDs, and returns a list of dictionaries.

Read the docs: https://developers.google.com/youtube/v3/docs/channels/list

Parameters

- **channel_id** (*str or list of str*) – channel_ids IE:['UCn8zNIfYAQNdrFRrr8oibKw']
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns metadata for featured channels from channel_id.

Return type list of dict

get_video_comments(*video_id*, *get_replies=True*, *max_results=None*, *next_page_token=False*, *parser=<function parse_comment_metadata>*, *part=['snippet']*, ***kwargs*)
Returns comments and replies to comments for a given video.

Read the docs: https://developers.google.com/youtube/v3/docs/commentThreads/list

Parameters

- **video_id** (*str*) – a video_id IE: "eqwPlwHSL_M"
- **get_replies** (*bool*) – whether or not to get replies to comments
- **parser** (youtube_api.parsers module) – the function to parse the json document
- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.

Returns comments and responses to comments of the given video_id.

Return type list of dict

search(*q=None*, *channel_id=None*, *max_results=5*, *order_by='relevance'*, *next_page_token=None*, *published_after=946684800.0*, *published_before=32503680000.0*, *location=None*, *location_radius='1km'*, *region_code=None*, *safe_search=None*, *relevance_language=None*, *event_type=None*, *topic_id=None*, *video_duration=None*, *search_type='video'*, *parser=<function parse_rec_video_metadata>*, *part=['snippet']*, ***kwargs*)
Search YouTube for either videos, channels for keywords. Only returns up to 500 videos per search. For an exhaustive search, take advantage of the published_after and published_before params. Note the docstring needs to be updated to account for all the arguments this function takes.

Read the docs: https://developers.google.com/youtube/v3/docs/search/list

Parameters

- **q** (*list or str*) – regex pattern to search using | for or, && for and, and - for not. IE boat|fishing is boat or fishing
- **max_results** (*int*) – max number of videos returned by a search query.
- **parser** (youtube_api.parsers module) – the function to parse the json document

- **part** (*list*) – The part parameter specifies a comma-separated list of one or more resource properties that the API response will include. Different parameters cost different quota costs from the API.
- **order_by** (*str*) – Return search results ordered by either `relevance`, `date`, `rating`, `title`, `videoCount`, `viewCount`.
- **next_page_token** (*str*) – A token to continue from a preciously stopped query IE:CDIQAA
- **published_after** (*datetime*) – Only show videos uploaded after datetime
- **published_before** (*datetime*) – Only show videos uploaded before datetime
- **location** (*tuple*) – Coodinates of video uploaded in location.
- **location_radius** (*str*) – The radius from the `location` param to include in the search.
- **region_code** (*str*) – search results for videos that can be viewed in the specified country. The parameter value is an ISO 3166-1 alpha-2 country code.
- **safe_search** (*str or None*) – whether or not to include restricted content, options are "moderate", "strict", None.
- **relevance_language** (*str*) – Instructs the API to return search results that are most relevant to the specified language.
- **event_type** (*str*) – whether the video is "live", "completed", or "upcoming".
- **topic_id** (*str*) – only contain resources associated with the specified topic. The value identifies a Freebase topic ID.
- **video_duration** (*str*) – filter on video durations "any", "long", "medium", "short".
- **search_type** – return results on a "video", "channel", or "playlist" search.

Returns incomplete video metadata of videos returned by search query.

Return type list of dict

get_recommended_videos(*video_id*, *max_results=5*, *parser=<function parse_rec_video_metadata>*, ***kwargs*)

Get recommended videos given a video ID. This extends the search API. Note that search history does not influence results.

Read the docs: https://developers.google.com/youtube/v3/docs/search/list

Parameters

- **video_id** – (str) a video_id IE: "eqwPlwHSL_M"
- **max_results** – (int) max number of recommended vids
- **parser** (`youtube_api.parsers module`) – the function to parse the json document

Returns incomplete video metadata from recommended videos of `video_id`.

Return type list of dict

class youtube_api.youtube_api.**YoutubeDataApi**(*key*, ***kwargs*)

Bases: *youtube_api.youtube_api.YouTubeDataAPI*

Variant case of the main YouTubeDataAPI class. This class will de depricated by version 0.0.21

3.2 youtube_api.parsers module

Every function from the *youtube_api.youtube_api* class has an argument for `parser`. `parser` can be any function that takes a dictionary as input. Here are the default parser fucntions for each function. Use these as templates to build your own custom parsers, or use the *youtube_api.parsers.raw_json()* or `None` as the `parser` argument for the raw API response.

youtube_api.parsers.**raw_json**(*item*)
Returns the raw json output from the API.

youtube_api.parsers.**parse_video_metadata**(*item*)
Parses and processes raw output and returns video_id, channel_title, channel_id, video_publish_date, video_title, video_description, video_category, video_view_count, video_comment_count, video_like_count, video_dislike_count, video_thumbnail, video_tags, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_video_url**(*item*)
Parses and processes raw output and returns publish_date, video_id, channel_id, collection_date

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_channel_metadata**(*item*)
Parses and processes raw output and returns channel_id, title, account_creatation_date, keywords, description, view_count, video_count, subscription_count, playlist_id_likes, playlist_id_uploads, topic_ids, country, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_subscription_descriptive**(*item*)
Parses and processes raw output and returns subscription_title, subscription_channel_id, subscription_kind, subscription_publish_date, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_featured_channels**(*item*)
Parses and processes raw output and returns a dictionary where the key is the channel_id and the key is a list of channel URLs.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_playlist_metadata**(*item*)
Parses and processes raw output and returns playlist_name, playlist_id, playlist_publish_date, playlist_n_videos, channel_id, channel_name, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_comment_metadata**(*item*)
Parses and processes raw output and returns video_id, commenter_channel_url, commenter_channel_display_name, comment_id, comment_like_count, comment_publish_date, text, commenter_rating, comment_parent_id, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_rec_video_metadata**(*item*)
Parses and processes raw output and returns video_id, channel_title, channel_id, video_publish_date, video_title, video_description, video_category, video_thumbnail, collection_date.

Params item json document

Returns parsed dictionary

Return type dict

youtube_api.parsers.**parse_caption_track**(*item*)
Returns the video_id, captions and collection_date.

Params item json document

Returns parsed dictionary

Return type dict

3.3 youtube_api.youtube_api_utils module

These are utils used by the client class, with some additional functions for analysis.

youtube_api.youtube_api_utils.**parse_yt_datetime**(*date_str*)
Parses a date string returned from YouTube's API into a Python datetime.

youtube_api.youtube_api_utils.**get_upload_playlist_id**(*channel_id*)
Given a channel_id, returns the user uploaded playlist id.

youtube_api.youtube_api_utils.**get_liked_playlist_id**(*channel_id*)
Given a channel_id, returns the user liked playlist id.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

y

Index