YouTube API

getting youtube to the terminal !

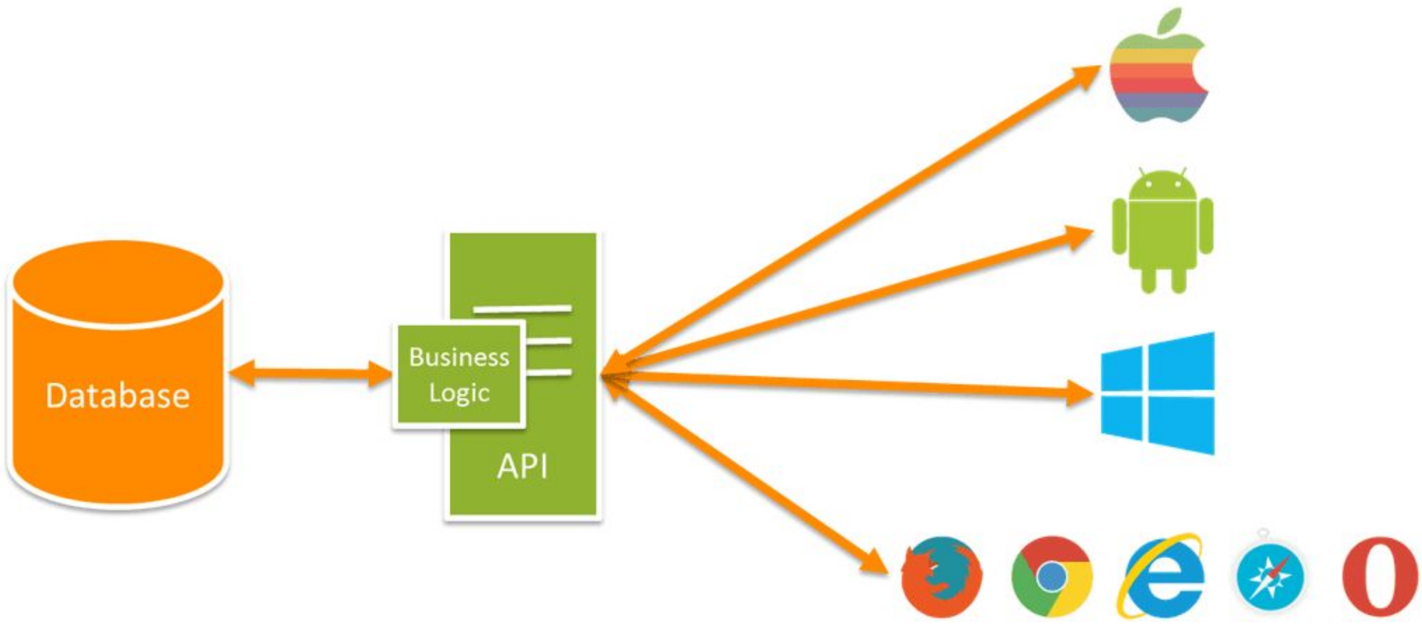
Team details

Harsh Raj	(180010017)
Manjeet Kapil	(180010021)
Rupesh Kalantre	(180010029)
Shriram Ghadge	(180010015)

Contents :

1. API Definition:

An application programming interface is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc



2. YouTube API :

The youtube APIs and Tools let you bring the Youtube experience to your webpage, application , or device.It is possible to search for videos, retrieve standard feeds, and see related content. A program can also authenticate as a user to upload videos, modify user playlists, and more.

Mainly two APIs :

- Data and analysis API

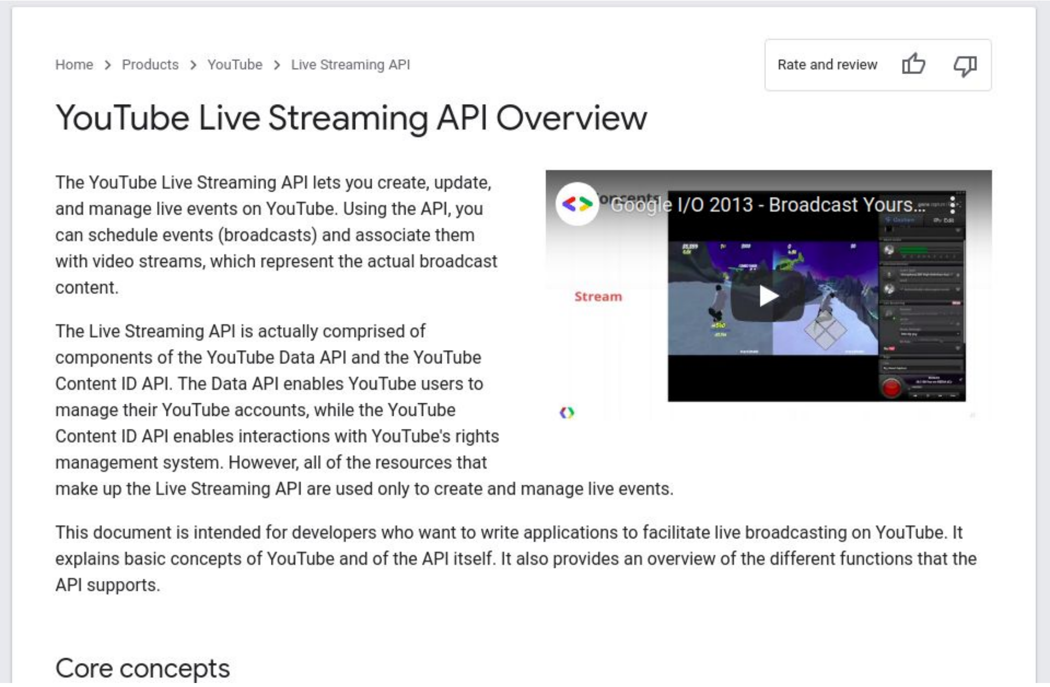
  The Data and Analytics APIs section briefly describes APIs that let your application access features and data normally available on the YouTube website : such as no. of likes on a video, no. of subs to a channel.

- Player and Players API

  The Data and Analytics APIs section briefly describes APIs that let your application access features and data normally available on the YouTube website : such as no. of likes on a video, no. of subs to a channel.

3. Advantages :

1. A great advantage of using YouTube's API is that 3rd party companies are able to arrange pay per click coupled with advertising for your videos. This allows the video uploader to receive money for the work they have put into their videos.
2. YouTube API's also allow a tighter integration of the video service into a 3rd party website. This essentially means that the user does not need to click a link to go to the YouTube video. Instead, they are able to view the video on the same page that they are visiting.



Home > Products > YouTube > Live Streaming API
Rate and review
Google I/O 2013 - Broadcast Yours...
Stream

YouTube Live Streaming API Overview

The YouTube Live Streaming API lets you create, update, and manage live events on YouTube. Using the API, you can schedule events (broadcasts) and associate them with video streams, which represent the actual broadcast content.

The Live Streaming API is actually comprised of components of the YouTube Data API and the YouTube Content ID API. The Data API enables YouTube users to manage their YouTube accounts, while the YouTube Content ID API enables interactions with YouTube's rights management system. However, all of the resources that make up the Live Streaming API are used only to create and manage live events.

This document is intended for developers who want to write applications to facilitate live broadcasting on YouTube. It explains basic concepts of YouTube and of the API itself. It also provides an overview of the different functions that the API supports.

Core concepts

4. Accessing The API:

In order to use the publicly available youtube api, one must:

1. Generate an API key
2. Read and understand documentation to know about various methods provided.

4.1 Generating The API Key :

- Sign in to Google Console Cloud
- Create Project
- Go to API and Services
- Select your project and go to library
- Choose Youtube API V3
- Click on enable and fill the details

Alternatively, follow this youtube tutorial.

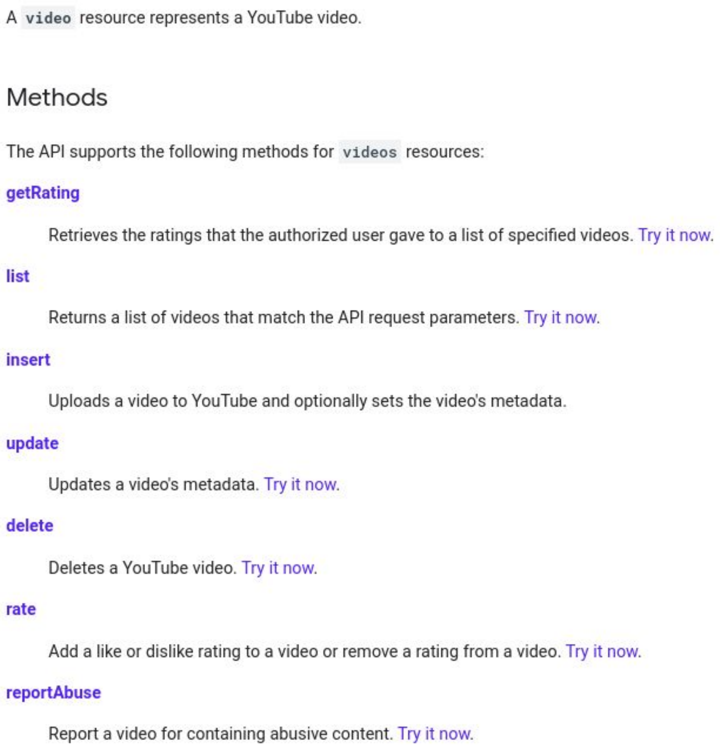
5. Methods that YouTube API provides

The YouTube api provides numerous methods to access various kinds of public and private data from the youtube. Users can get straightforward details like channel details, comments on the video, etc to more narrow details like the geographic diversity of viewers.

1. Video
2. Captions
3. Comments
4. CommentsThread
5. VideoCategories
6. VideoAbuseReportReasons
7. Playlist
8. PlaylistItems
9. Thumbnails
10. Subscribers
11. Watermarks
12. Channels
13. Activities
14. ChannelBanners
15. ChannelSection
16. Search
17. I18nLanguages
18. I18nRegions
19. Members
20. MembershipLevels

Fig. List of various methods provided by the youtube api

- [Documentation]



A `video` resource represents a YouTube video.

Methods

The API supports the following methods for `videos` resources:

getRating

Retrieves the ratings that the authorized user gave to a list of specified videos. Try it now.

list

Returns a list of videos that match the API request parameters. Try it now.

insert

Uploads a video to YouTube and optionally sets the video's metadata.

update

Updates a video's metadata. Try it now.

delete

Deletes a YouTube video. Try it now.

rate

Add a like or dislike rating to a video or remove a rating from a video. Try it now.

reportAbuse

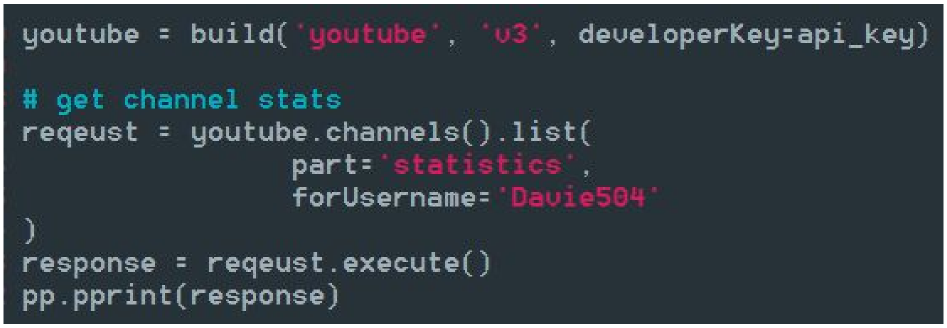
Report a video for containing abusive content. Try it now.

6. Some Illustrations :

1. Channel Statistics : (Documentation)

With the youtube api, getting the youtube channel statistics is as easy as calling a function with channel id/username as given parameters.

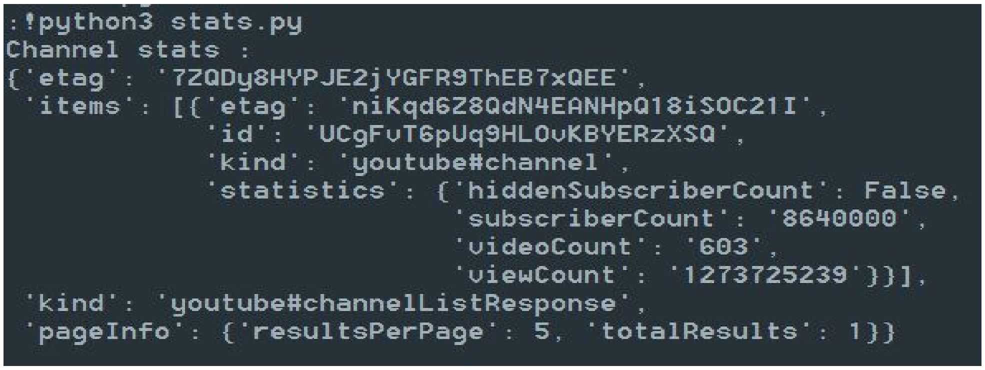
Python Code :



```
youtube = build('youtube', 'v3', developerKey=api_key)

# get channel stats
reqeust = youtube.channels().list(
                part='statistics',
                forUsername='Davie504'
)
response = reqeust.execute()
pp.pprint(response)
```

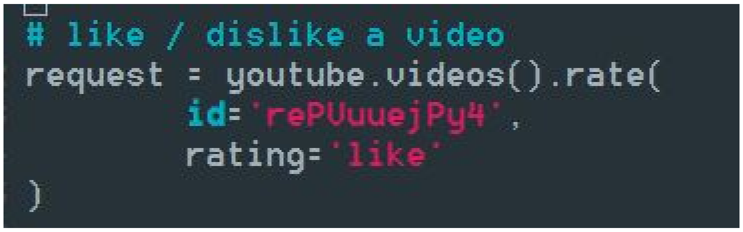
Output :



```
:!python3 stats.py
Channel stats :
{'etag': '7ZQDy8HYPJE2jYGFR9ThEB7xQEE',
 'items': [{'etag': 'niKqd6Z8QdN4EANHpQ18iSOC21I',
            'id': 'UCgFvT6pUq9HLOvKBYERzXSQ',
            'kind': 'youtube#channel',
            'statistics': {'hiddenSubscriberCount': False,
                           'subscriberCount': '8640000',
                           'videoCount': '603',
                           'viewCount': '1273725239'}}],
 'kind': 'youtube#channelListResponse',
 'pageInfo': {'resultsPerPage': 5, 'totalResults': 1}}
```

2. Liking or disliking a video : (Documentation)

We can like or dislike a video on behalf of your channel.It needs authentication to be done beforehand. Here **id** is unique video id of the video we want to like/dislike.



```
# like / dislike a video
request = youtube.videos().rate(
        id='rePUuuejPy4',
        rating='like'
)
```

Bibliography

➔ Video tutorial 1

➔ Video tutorial 2

➔ Youtube-api documentation

➔ References

➔ wiki