vidyo

Release 0.1.0

Ethan Henderson

Jul 23, 2021

GETTING STARTED

A simple way to get information on YouTube videos.

CHAPTER

ONE

FEATURES

- Pythonic syntax lets you feel right at home
- Lightweight nature means you don't have to worry about excessive dependencies
- Simple to use objects take the hassle out of API inconsistencies

1.1 Creating a Google Developers application

This section will run you through the process of creating an application on the Google Developers console. You will need one of these in order to use vidyo.

1.1.1 Creating a project

To create a project:

1. Sign into your Google account. This does not need to be the same account as the one you are retrieving data from.
2. On the navbar, click on the dropdown that says "Select a project".
3. Click on "NEW PROJECT".
4. Enter your project name, then click "CREATE". You do not need to set an organisation.

After this, you'll be redirected back to the dashboard. It may take some time to create the project.

1.1.2 Setting up the YouTube Data API

To enable the API:

1. Click on "ENABLE APIS AND SERVICES".
2. Search for "youtube", then click "YouTube Data API v3".
3. Click "ENABLE".

After this, you now need to create an API Key. To do this:

1. Click "Credentials" on the side bar.
2. Click "CREATE CREDENTIALS".
3. Click "API Key".
4. Copy the key to your clipboard.

5. Ideally, store this key in a separate file.

Warning: You should **never** share this API key with anyone.

You're all done! You can now start working with vidyo.

1.2 Installing vidyo

This section will go through how to install vidyo on your system.

Important: Python 3.7.0 or greater is required.

1.2.1 Creating a virtual environment

A virtual environment allows you to contain a project's dependencies in a way that means no dependencies conflict with other projects. It is recommended that you create a virtual environment when working with vidyo. To do this, run the following commands:

Linux/macOS/UNIX

```
python3 -m venv .venv
source ./.venv/bin/activate
```

Windows

```
py -3 -m venv .venv
.venv\Scripts\activate
```

1.2.2 Installing the lastest stable version

You can install the latest version of vidyo by using the following command:

```
pip install vidyo
```

1.2.3 Installing other versions

To install a specific version (for example, if you want to try the newest development releases), use the following command:

```
pip install vidyo==0.1.0
# or...
pip install vidyo==0.1.0.dev1
# etc.
```

> **Warning:** Development versions may be unstable, and thus unsuitable for production environments.

1.2.4 Installing from source

You can also install vidyo from source if you want to. Using this method, you can use a specific branch – such as `develop` – to install a version different from the latest stable one. You can do this using the following command:

```
git clone https://github.com/parafoxia/vidyo
cd vidyo
git checkout develop  # ...or any branch you want
python3 -m pip install -U .
```

1.3 API reference

This section will detail how vidyo can be used.

1.3.1 Version info

`vidyo.`**`__version__`**

The currently installed vidyo version, represented in the **PEP 440** format.

1.3.2 Exceptions

`class vidyo.errors.`**`VidyoError`**

The base exception class.

`class vidyo.errors.`**`ResponseNotOK`**

Exception thrown when a request returned a non-OK status code.

1.3.3 Classes

`class vidyo.client.`**`Client`**(*key: str*)

The client class.

Parameters **key** (*str*) – The API key for the YouTube Data API.

`get_video`(*id_: str*) → *vidyo.video.Video*

Executes an API request to get video information.

Parameters **id** (*str*) – The video ID to get information for.

Returns An object containing video information.

Return type *Video*

Raises ***ResponseNotOK*** – The request returned a non-OK status code.

`class vidyo.video.`**`Video`**(*data: dict*)

An object containing video information.

Parameters **data** (*dict*) – Response data from the YouTube Data API. This should be the first element in the `items` list.

id
The video ID.

Type str

published
The date and time and the video was published.

Type datetime.datetime

channel
The YouTube channel to which the video was published.

Type *PartialChannel*

title
The video title.

Type str

description
The video's description.

Type str

tags
A list of the video's tags. If there are none, this is an empty list.

Type list[str]

category_id
The video's category's ID.

Type int

live
Whether the video is currently live. This could mean it is a premiere or a live stream.

Type bool

duration
The video's duration.

Type datetime.timedelta

is_3d
Whether the video was uploaded in 3D.

Type bool

is_hd
Whether there is a HD version of the video available (720p or above).

Type bool

captioned
Whether there are captions available on the video.

Type bool

is_360
Whether the video was recorded in 360 degrees.

Type bool

`privacy`
The video's privacy status. Can be "public", "unlisted", or "private".

Type str

`license`
The video's license. Can be either "youtube" or "creativeCommon".

Type str

`embeddable`
Whether the video is embeddable.

Type bool

`for_kids`
Whether the video is marked as "Made For Kids".

Type bool

`views`
The number of views the video has. If this is not available, this will be -1.

Type int

`likes`
The number of likes the video has. If ratings are disabled, this will be -1.

Type int

`dislikes`
The number of dislikes the video has. If ratings are disabled, this will be -1.

Type int

`favourites`
The number of favourites the video has. If this is not available, this will be -1.

Type int

`comments`
The number of comments the video has. If comments are disabled, this will be -1.

Type int

`get_thumbnail()` → PIL.Image.Image
Gets the highest resolution thumbnail available.

Returns A Pillow image.

Return type PIL.Image.Image

Raises *ResponseNotOK* – The request returned a non-OK status code.

`class vidyo.channel.PartialChannel`(*id_: str*, *name: str*)
An object with limited channel information.

Parameters

- **id** (*str*) – The channel ID.
- **name** (*str*) – The channel name.

`id`
The channel ID.

Type str

name

The channel name.

Type str

1.4 Changelog

1.4.1 v0.1.0

- Add `Client` class.
- Add `Video` class.
- Add `PartialChannel` class.

INDEX