

---

# youtube-api-datakund

*Release 0.0.2*

**datakund**

**Feb 16, 2021**



## CONTENTS:

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation/Usage:</b>	<b>3</b>
<b>3</b>	<b>Import datakund</b>	<b>5</b>
<b>4</b>	<b>Creating Object</b>	<b>7</b>
<b>5</b>	<b>Functions</b>	<b>9</b>
5.1	Login . . . . .	9
5.2	Login Cookie . . . . .	9
5.3	Auto Like . . . . .	10
5.4	Auto Comment . . . . .	10
5.5	Auto Subscribe . . . . .	10
5.6	Like Comment . . . . .	11
5.7	Watch Video . . . . .	11
5.8	Upload . . . . .	11
5.9	Upload To Playlist . . . . .	12
5.10	Search . . . . .	12
5.11	Search Results . . . . .	13
5.12	Get Video Info . . . . .	13
5.13	Get Channel Info . . . . .	13
5.14	Get Transcript . . . . .	14
5.15	Get Playlist Videos . . . . .	14
5.16	Get Video Tags . . . . .	14
5.17	Get Watch History . . . . .	15
5.18	Channel Videos . . . . .	15
5.19	Watch Later Videos . . . . .	15
5.20	Check Video Exists . . . . .	16
5.21	Create Playlist . . . . .	16
5.22	Delete Video From Watch Later . . . . .	16
5.23	Find Click Video . . . . .	17
5.24	Play Pause Video . . . . .	17
5.25	Play Next Video . . . . .	17
5.26	Forward Video . . . . .	18
5.27	Set Playback Speed . . . . .	18
5.28	Replay Video . . . . .	18
5.29	Skip Ad . . . . .	19
5.30	Video Comments . . . . .	19
5.31	Get Comment Replies . . . . .	19

5.32	Click Hide Replies . . . . .	20
5.33	Click Show More Replies . . . . .	20
5.34	Click View Replies . . . . .	20
<b>6</b>	<b>Other Functions</b>	<b>21</b>
6.1	Open . . . . .	21
6.2	Get Page Title . . . . .	21
6.3	Get Page Source . . . . .	21
6.4	Get Current Url . . . . .	22
6.5	Reload . . . . .	22
6.6	Keypress . . . . .	22
6.7	Scroll . . . . .	22
6.8	End . . . . .	22
6.9	Quit . . . . .	23
<b>Index</b>		<b>25</b>

## **INTRODUCTION**

`datakund` is an automation library which can be used to automate tasks like sending mails,scraping data,auto check-out and many more. You can download the source code from here([see here](#))

It uses selenium to automate the things. You can use its inbuilt functions in a very easy way.



## INSTALLATION/USAGE:

You can find this package on Pypi (see [here](#)).

Command to install :- `pip install datakund`





## IMPORT DATAKUND

```
from datakund import *
```



## CREATING OBJECT

```
youtube=datakund.youtube ()
```

It will return the object which you can further use to call youtube functions and opens a automated browser



## FUNCTIONS

datakund provides following functions for youtube:-

### 5.1 Login

It logs in to youtube through the credentials passed in `username` and `password`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.login(username='datakund@gmail.com', password='pwd@123')
```

#### Parameters

- **username** (*str*) – Account Username
- **password** (*str*) – Account Password

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

### 5.2 Login Cookie

It logs in to youtube through the cookies passed in `cookies`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.login_cookie(cookies=[{'domain': 'youtube.com', 'expirationDate': 1676431710.556339,  
                               'hostOnly': false, 'httpOnly': false, 'name': '__Secure-3PAPISID', 'path': '/'},  
                      ...])
```

**Parameters** **cookies** (*str*) – list of cookies of youtube

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.3 Auto Like

It likes the video passed in `video_url`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.auto_like (video_url='https://www.youtube.com/watch?v=hPQ79rrkziM')
```

**Parameters** `video_url` (*str*) – video link which need to be liked

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.4 Auto Comment

It auto comments on the video passed in `video_link` with the comment passed in `comment`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.auto_comment (comment='Nice', video_link='https://www.youtube.com/watch?v=hPQ79rrkziM')
```

**Parameters**

- **comment** (*str*) – comment which need to be typed
- **video\_link** (*str*) – video url where need to post comment

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.5 Auto Subscribe

It subscribes the channel passed in `channel_url`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.auto_subscribe (channel_url='https://www.youtube.com/channel/UCM0YvsRfYfsniGAhvjvYFOSA')
```

**Parameters** `channel_url` (*str*) – Channel url which need to be subscribed

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.6 Like Comment

It likes the comment whose text is passed in `comment`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.like_comment (comment='Nice song')
```

**Parameters** `comment` (*str*) – comment text which need to be liked

**Returns** {"body": [{}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.7 Watch Video

It opens the video passed in `video_url` and then waits for the time passed in `time`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.watch_video (time='120', video_url='https://www.youtube.com/watch?v=eLrJUdBHiXA&list=PLsuCfYXzi5DJfjxOmPI')
```

**Parameters**

- **time** (*str*) – amount of time in seconds to wait
- **video\_url** (*str*) – video which need to be opened

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.8 Upload

It uploads the video to Youtube.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.upload (title='Testing Upload', video_path='C:/Users/video.mp4', kid_type="Yes, it's made for kids", description='I am testing', type='Public')
```

**Parameters**

- **title** (*str*) – title of video

- **video\_path** (*str*) – local file path of video
- **kid\_type** (*str*) – e.g. Yes, it's made for kids
- **description** (*str*) – description of video
- **type** (*str*) – e.g. Public or Private

**Returns** {"body": {"VideoLink": "VideoLink"}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.9 Upload To Playlist

It uploads the video to youtube.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.upload_to_playlist (title='Testing upload', video_path='C:/users/video.mp4',  
                             kid_type="Yes, it's made for kids", description='I am testing',  
                             playlist='DataKund', type='Public')
```

### Parameters

- **title** (*str*) – title of video
- **video\_path** (*str*) – local file path of video
- **kid\_type** (*str*) – e.g Yes, it's made for kids
- **description** (*str*) – description of video
- **playlist** (*str*) – playlist name
- **type** (*str*) – e.g. Private or Public

**Returns** {"body": {"VideoLink": "VideoLink"}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.10 Search

It searches the keyword passed in keyword on youtube.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.search (keyword='latest movies')
```

**Parameters** **keyword** (*str*) – keyword need to be searched on youtube

**Returns** {"body": {}}, {"success\_score": "100", "errors": []}



**Return type** dict

## 5.11 Search Results

It fetches the results from the youtube search results like title, link, channelname etc.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.search_results()
```

**Returns** {"body": [{"viewsandtime": 'viewsandtime', 'channel': 'channel', 'title': 'title', 'link': 'link'}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.12 Get Video Info

It fetches the video info whose link is passed in video\_url.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_video_info(video_url='https://www.youtube.com/watch?v=UF8uR6Z6KLc')
```

**Parameters** video\_url (*str*) – video url

**Returns** {"body": {'DisLikes': 'DisLikes', 'Title': 'Title', 'Subscribers': 'Subscribers', 'Comments': 'Comments', 'ChannelLink': 'ChannelLink', 'ChannelName': 'ChannelName', 'Desc': 'Desc', 'Views': 'Views', 'Duration': 'Duration', 'Publish\_Date': 'Publish\_Date', 'Likes': 'Likes'}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.13 Get Channel Info

It fetches the info of channel whose link is passed in channel\_link.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_channel_info(channel_link='https://www.youtube.com/channel/UCMOYvsRfYfsniGAhJvYFOSA')
```

**Parameters** channel\_link (*str*) – channel link whose info need to be fetched

**Returns** {"body": {"title": "title", "links": "links", "views": "views", "joined": "joined", "description": "description", "subscribers": "subscribers", "location": "location"}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.14 Get Transcript

It fetches the transcript of the video whose link is passed in `video_url`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_transcript (video_url='https://www.youtube.com/watch?v=UF8uR6Z6KLc')
```

**Parameters** `video_url` (*str*) – video link whose transcript need to be fetched

**Returns** {"body": {"Transcript": "Transcript"}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.15 Get Playlist Videos

It fetches link and title of the videos in the playlist passed in `playlist_link`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_playlist_videos (playlist_link='https://www.youtube.com/playlist?list=PLsuCfYXzi5DJfjxOmPRJIS4KLLJhAlr3')
```

**Parameters** `playlist_link` (*str*) – playlist link whose videos need to be fetched

**Returns** {"body": [{"Title": "Title", "Video\_Link": "Video\_Link"}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.16 Get Video Tags

It fetches the video tags and tag links whose link is passed in `video_link`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_video_tags (video_link='https://www.youtube.com/watch?v=eLrJUdBHiXA&list=PLsuCfYXzi5DJfjxOmPRJIS4K')
```

**Parameters** `video_link` (*str*) – video link whose tags need to be fetched

**Returns** {"body": {'tags': 'tags', 'taglinks': 'taglinks'}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.17 Get Watch History

It fetches the title and link of the videos in watch history which is currently opened in browser

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_watch_history()
```

**Returns** {"body": [{'title': 'title', 'link': 'link'}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.18 Channel Videos

It fetches the channel videos from page opened in browser.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.channel_videos()
```

**Returns** {"body": [{'Title': 'Title', 'Video\_Link': 'Video\_Link'}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.19 Watch Later Videos

It fetches link and title of the videos in watch later list.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.watch_later_videos()
```

**Returns** {"body": [{'Title': 'Title', 'Video\_Link': 'Video\_Link'}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.20 Check Video Exists

It returns title of video passed in `video_url` if exists, otherwise returns `None` in `title`

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.check_video_exists (video_url='https://www.youtube.com/watch?v=hPQ79rrkziM')
```

**Parameters** `video_url` (*str*) – video url which need to be checked for existence

**Returns** {"body": {"Title": 'Title'}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.21 Create Playlist

It creates new playlist with the name passed in `playlist_name` and returns playlist like in `playlist_link`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.create_playlist (playlist_name='DataKund')
```

**Parameters** `playlist_name` (*str*) – name of the new playlist to create

**Returns** {"body": {"playlist\_link": 'playlist\_link'}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.22 Delete Video From Watch Later

It deletes the video whose title is passed in `title` from the watch later videos.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.delete_video_from_watch_later (title='How to insert data to mysql?')
```

**Parameters** `title` (*str*) – title of video which needs to be deleted

**Returns** {"body": [{}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.23 Find Click Video

It will click on the video whose title is passed in `searchtext`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.find_click_video (searchtext='MySql DataBase Tutorials')
```

**Parameters** `searchtext` (*str*) – title or keyword which is present in video title or description

**Returns** {"body": [{}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.24 Play Pause Video

It play or pause the video playing currently on browser.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.play_pause_video ()
```

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.25 Play Next Video

It clicks on next video to play next video.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.play_next_video ()
```

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.26 Forward Video

It forwards the video currently playing on browser with 5 seconds.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.forward_video()
```

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.27 Set Playback Speed

It sets the playback speed passed in *speed* in the video currently playing on browser.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.set_playback_speed(speed='1.25')
```

**Parameters** **speed** (*str*) – speed to set e.g. 1.25

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.28 Replay Video

It replays the video currently playing on browser.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.replay_video()
```

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.29 Skip Ad

It clicks on skip ad button if available.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.skip_ad()
    Returns {"body": {}, "success_score": "100", "errors": []}
    Return type dict
```

## 5.30 Video Comments

It fetches the comments data like comment text, username ,userlink.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.video_comments()
    Returns {"body": [{"Comment": 'Comment', 'UserLink': 'UserLink', 'user': 'user', 'Time':
    'Time', 'Likes': 'Likes'}], "success_score": "100", "errors": []}
    Return type dict
```

## 5.31 Get Comment Replies

It fetches the comments replies text along with userlink and username currently opened in browser.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.get_comment_replies()
    Returns {"body": [{"userlink": 'userlink', 'replytext': 'replytext', 'user': 'user'}], "suc-
    cess_score": "100", "errors": []}
    Return type dict
```

## 5.32 Click Hide Replies

It will click on hide replies of the comment whose text is passed in `comment`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.click_hide_replies (comment='Nice song')
```

**Parameters** `comment` (*str*) – comment text whose replies needs to be hide

**Returns** {"body": [{}], "success\_score": "100", "errors": []}

**Return type** dict

## 5.33 Click Show More Replies

It will click on show more replies button in comments section.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.click_show_more_replies ()
```

**Returns** {"body": {}, "success\_score": "100", "errors": []}

**Return type** dict

## 5.34 Click View Replies

It will click on view replies button of the comment whose text is passed in `comment`.

**body:** returns data

**success\_score:** api success rate

**errors:** errors encountered in api

Here is the code:-

```
youtube.click_view_replies (comment='Nice song')
```

**Parameters** `comment` (*str*) – comment text whose replies need to be viewed

**Returns** {"body": [{}], "success\_score": "100", "errors": []}

**Return type** dict



## OTHER FUNCTIONS

You can use basic functions which selenium provides with this library like opening a url, get pagesource, get current url etc. These are the functions:-

### 6.1 Open

It will open the url provided in the argument.

`youtube.open(url)`

**Parameters** `url` (*str*) – Link which need to be opened

**Returns** {}

**Return type** dict

### 6.2 Get Page Title

It returns the title of page opened.

`youtube.get_page_title()`

**Returns** {"pagetitle": "youtube"}

**Return type** dict

### 6.3 Get Page Source

It returns the pagesource of page opened.

`youtube.get_page_source()`

**Returns** {"pagesource": "pagesource"}

**Return type** dict

## 6.4 Get Current Url

It returns the pagesource of page opened.

```
youtube.get_current_url()
```

**Returns** {"url": "url"}

**Return type** dict

## 6.5 Reload

It reloads the page opened.

```
youtube.reload()
```

**Returns** {}

**Return type** dict

## 6.6 Keypress

It perform the keypress passed.

```
youtube.keypress(key)
```

**Parameters** **key** (*str*) – Key which need to be pressed, e.g pagedown,arrowleft,enter

**Returns** {}

**Return type** dict

## 6.7 Scroll

It scrolls to the end of page.

```
youtube.scroll()
```

**Returns** {}

**Return type** dict

## 6.8 End

It ends the youtube session and close the automated chromedriver.

---

**Note:** You will need to create youtube object again after `end()`.

---

```
youtube.end()
```

**Returns** {}

**Return type** dict

## 6.9 Quit

It quits the datakund application running in background.

---

**Note:** You will need to import datakund library again to start datakund application.

---

`youtube.quit()`

**Returns** {}

**Return type** dict



## B

## built-in function

- youtube.auto\_comment(), 10
- youtube.auto\_like(), 10
- youtube.auto\_subscribe(), 10
- youtube.channel\_videos(), 15
- youtube.check\_video\_exists(), 16
- youtube.click\_hide\_replies(), 20
- youtube.click\_show\_more\_replies(), 20
- youtube.click\_view\_replies(), 20
- youtube.create\_playlist(), 16
- youtube.delete\_video\_from\_watch\_later(), built-in function, 16
- youtube.end(), 22
- youtube.find\_click\_video(), 17
- youtube.forward\_video(), 18
- youtube.get\_channel\_info(), 13
- youtube.get\_comment\_replies(), 19
- youtube.get\_current\_url(), 22
- youtube.get\_page\_source(), 21
- youtube.get\_page\_title(), 21
- youtube.get\_playlist\_videos(), 14
- youtube.get\_transcript(), 14
- youtube.get\_video\_info(), 13
- youtube.get\_video\_tags(), 14
- youtube.get\_watch\_history(), 15
- youtube.keypress(), 22
- youtube.like\_comment(), 11
- youtube.login(), 9
- youtube.login\_cookie(), 9
- youtube.open(), 21
- youtube.play\_next\_video(), 17
- youtube.play\_pause\_video(), 17
- youtube.quit(), 23
- youtube.reload(), 22
- youtube.replay\_video(), 18
- youtube.scroll(), 22
- youtube.search(), 12
- youtube.search\_results(), 13
- youtube.set\_playback\_speed(), 18
- youtube.skip\_ad(), 19

- youtube.upload(), 11
- youtube.upload\_to\_playlist(), 12
- youtube.video\_comments(), 19
- youtube.watch\_later\_videos(), 15
- youtube.watch\_video(), 11

## Y

- youtube.auto\_comment()
  - built-in function, 10
- youtube.auto\_like()
  - built-in function, 10
- youtube.auto\_subscribe()
  - built-in function, 10
- youtube.channel\_videos()
  - built-in function, 15
- youtube.check\_video\_exists()
  - built-in function, 16
- youtube.click\_hide\_replies()
  - built-in function, 20
- youtube.click\_show\_more\_replies()
  - built-in function, 20
- youtube.click\_view\_replies()
  - built-in function, 20
- youtube.create\_playlist()
  - built-in function, 16
- youtube.delete\_video\_from\_watch\_later()
  - built-in function, 16
- youtube.end()
  - built-in function, 22
- youtube.find\_click\_video()
  - built-in function, 17
- youtube.forward\_video()
  - built-in function, 18
- youtube.get\_channel\_info()
  - built-in function, 13
- youtube.get\_comment\_replies()
  - built-in function, 19
- youtube.get\_current\_url()
  - built-in function, 22
- youtube.get\_page\_source()
  - built-in function, 21
- youtube.get\_page\_title()
  - built-in function, 21

- built-in function, 21
- youtube.get\_playlist\_videos()
  - built-in function, 14
- youtube.get\_transcript()
  - built-in function, 14
- youtube.get\_video\_info()
  - built-in function, 13
- youtube.get\_video\_tags()
  - built-in function, 14
- youtube.get\_watch\_history()
  - built-in function, 15
- youtube.keypress()
  - built-in function, 22
- youtube.like\_comment()
  - built-in function, 11
- youtube.login()
  - built-in function, 9
- youtube.login\_cookie()
  - built-in function, 9
- youtube.open()
  - built-in function, 21
- youtube.play\_next\_video()
  - built-in function, 17
- youtube.play\_pause\_video()
  - built-in function, 17
- youtube.quit()
  - built-in function, 23
- youtube.reload()
  - built-in function, 22
- youtube.replay\_video()
  - built-in function, 18
- youtube.scroll()
  - built-in function, 22
- youtube.search()
  - built-in function, 12
- youtube.search\_results()
  - built-in function, 13
- youtube.set\_playback\_speed()
  - built-in function, 18
- youtube.skip\_ad()
  - built-in function, 19
- youtube.upload()
  - built-in function, 11
- youtube.upload\_to\_playlist()
  - built-in function, 12
- youtube.video\_comments()
  - built-in function, 19
- youtube.watch\_later\_videos()
  - built-in function, 15
- youtube.watch\_video()
  - built-in function, 11