youtube-api-datakund

Release 0.0.2

datakund

Feb 11, 2022

CONTENTS:

CHAPTER

ONE

INTRODUCTION

`bot-studio` is an automation library which can be used to automate tasks like sending mails,scraping data,auto checkout 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.

CHAPTER

TWO

INSTALLATION/USAGE:

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

Command to install :- `pip install bot-studio`

CHAPTER

THREE

IMPORT BOT-STUDIO

```
from bot_studio import *
```

CHAPTER

FOUR

CREATING OBJECT

```
youtube=bot_studio.youtube()
or
youtube=bot_studio.youtube(headless=True,....)
```

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

4.1 Browser Options

Option	Default Value	Description
headless	False	Can set it to True if wants headless
proxy	No proxy	Pass proxy value e.g 98.0.2.5:4000
profile_path	creates temporary profile	Pass profile path e.g C:\Users\username\AppData\Local\Google\Chrome\User Data\
user_agent	No user agent	Pass user agent e.g python 2.7", "platform":"Windows
download_folder	Downloads in default folder	If want to set download directory to custom e.g E:files\

CHAPTER

FIVE

FUNCTIONS

`bot-studio` provides following functions for youtube:-

5.1 Login

It logins 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 logins 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/UCM0YvsRfYfsniGAhjvYFOSA'*`)`

> **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=PLsuCfYXzi5DJfjxOmPRJIS4KLlJhAlr8

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/UCM0YvsRfYfsniGAhjvYFOSA'*`)`

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=PLsuCfYXzi5DJfjxOmPRJIS4KLlJhAlr8P'*`)`

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=PLsuCfYXzi5DJfjxOmPRJIS4KLlJ*

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'}], "success_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

CHAPTER

SIX

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 bot-studio application runing in background.

Note: You will need to import bot-studio library again to start application.

youtube.**quit**()

Returns {}

Return type dict

INDEX

B

Y