

Package ‘tuber’

November 21, 2016

Title Client for the YouTube API

Version 0.7.0

Maintainer Gaurav Sood <gsood07@gmail.com>

Description Get comments posted on YouTube videos, information on how many times a video has been liked, search for videos with particular content, and much more. You can also scrape captions from a few videos. To learn more about the YouTube API, see <<https://developers.google.com/youtube/v3/>>.

URL <http://github.com/soodoku/tuber>

BugReports <http://github.com/soodoku/tuber/issues>

Depends R (>= 3.2.0)

License MIT + file LICENSE

LazyData true

Imports httr, plyr

VignetteBuilder knitr

Suggests knitr (>= 1.11), testthat, rmarkdown, xml2

RoxygenNote 5.0.1

NeedsCompilation no

Author Gaurav Sood [aut, cre]

Repository CRAN

Date/Publication 2016-11-21 21:22:50

R topics documented:

get_captions	2
get_channel_stats	3
get_comments	4
get_comment_threads	5
get_playlists	6
get_playlist_items	7
get_related_videos	8

get_stats	9
get_subscriptions	10
get_video_details	11
list_abuse_report_reasons	11
list_caption_tracks	12
list_channel_activities	13
list_channel_resources	15
list_channel_sections	16
list_channel_videos	17
list_guidecats	18
list_langs	19
list_regions	19
list_videocats	20
list_videos	21
tuber	22
tuber_check	22
tuber_GET	23
tuber_POST	23
yt_check_token	24
yt_oauth	24
yt_search	25
yt_topic_search	27

Index 28

get_captions	<i>Get Captions of a Video</i>
--------------	--------------------------------

Description

A few of Youtube videos have caption tracks available from an older YouTube API. If that caption track is available, this function returns that, Or it returns caption track specified by id resource. Check [list_caption_tracks](#) for more information.

Usage

```
get_captions(video_id = NULL, lang = "en", id = NULL, ...)
```

Arguments

video_id	ID of the video whose captions are requested. Required. No default.
lang	Language of the caption; required; default is "en" (English)
id	String. id of the caption track that is being retrieved
...	Additional arguments passed to tuber_GET .

Value

String.

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_captions(video_id="yJTXN4xrI8")  
get_captions(id="y3E1XcEME3lSISz6izkWWT5GvxjPu8pA")  
  
## End(Not run)
```

get_channel_stats	<i>Get statistics of a Channel</i>
-------------------	------------------------------------

Description

Get statistics of a Channel

Usage

```
get_channel_stats(channel_id = NULL, ...)
```

Arguments

channel_id	Character. Id of the channel
...	Additional arguments passed to tuber_GET .

Value

nested named list with top element names: kind, etag, id, snippet (list of details of the channel including title, description, etc.)
If the channel_id is mistyped or there is no information, an empty list is returned

References

<https://developers.google.com/youtube/v3/docs/channels/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_channel_stats(channel_id="UCMtFAi84ehTSYSE9XoHefig")  
get_channel_stats(channel_id="UCMtFAi84ehTSYSE9Xo") # Incorrect channel ID  
  
## End(Not run)
```

 get_comments

Get Comments

Description

Get Comments

Usage

```
get_comments(filter = NULL, part = "snippet", max_results = 100,
             text_format = "html", page_token = NULL, simplify = TRUE, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: comment_id: comment ID. parent_id: parent ID.
part	Comment resource requested. Required. Comma separated list of one or more of the following: id, snippet. e.g., "id, snippet", "id", etc. Default: snippet.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 20 and 100. Default is 100.
text_format	Data Type: Character. Default is "html". Only takes "html" or "plainText". Optional.
page_token	Specific page in the result set that should be returned. Optional.
simplify	Data Type: Boolean. Default is TRUE. If TRUE, the function returns a data frame. Else a list with all the information returned.
...	Additional arguments passed to tuber_GET .

Value

Nested named list. The entry `items` is a list of comments along with meta information. Within each of the `items` is an item `snippet` which has an item `topLevelComment$snippet$textDisplay` that contains the actual comment.

When `filter` is `comment_id`, and `simplify` is `TRUE`, and there is a correct comment id, it returns a data.frame with following cols: `id`, `authorDisplayName`, `authorProfileImageUrl`, `authorChannelUrl`, `value`, `text`, `publishedAt`, `updatedAt`

References

<https://developers.google.com/youtube/v3/docs/comments/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

get_comments(filter = c(comment_id="z13dh13j5rr0wbmq04cifrhtuypw14hsdk"))
get_comments(filter =
c(comment_id="z13dh13j5rr0wbmq04cifrhtuypw14hsdk,
  z13dh13j5rr0wbmq04cifrhtuypw14hsdk"))

## End(Not run)
```

get_comment_threads *Get Comments Threads*

Description

Get Comments Threads

Usage

```
get_comment_threads(filter = NULL, part = "snippet", text_format = "html",
  simplify = TRUE, max_results = 100, page_token = NULL, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: video_id: video ID. channel_id: channel ID. thread_id: comma-separated list of comment thread IDs threads_related_to_channel: channel ID.
part	Comment resource requested. Required. Comma separated list of one or more of the following: id, snippet. e.g., "id, snippet", "id", etc. Default: snippet.
text_format	Data Type: Character. Default is "html". Only takes "html" or "plainText". Optional.
simplify	Data Type: Boolean. Default is TRUE. If TRUE, the function returns a data frame. Else a list with all the information returned.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 20 and 100. Default is 100.
page_token	Specific page in the result set that should be returned. Optional.
...	Additional arguments passed to <code>tuber_GET</code> .

Value

Nested named list. The entry `items` is a list of comments along with meta information. Within each of the `items` is an `item snippet` which has an `item topLevelComment$snippet$textDisplay` that contains the actual comment.

If `simplify` is `TRUE`, a `data.frame` with the following columns: `authorDisplayName`, `authorProfileImageUrl`, `authorCh`

References

<https://developers.google.com/youtube/v3/docs/commentThreads/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

get_comment_threads(filter = c(video_id="N708P-A45D0"))

## End(Not run)
```

get_playlists

Get Playlists

Description

Get Playlists

Usage

```
get_playlists(filter = NULL, part = "contentDetails", max_results = 50,
             hl = NULL, page_token = NULL, simplify = TRUE, ...)
```

Arguments

<code>filter</code>	string; Required. named vector of length 1 potential names of the entry in the vector: <code>channel_id</code> : ID of the channel <code>playlist_id</code> : YouTube playlist ID.
<code>part</code>	Required. One of the following: <code>contentDetails</code> , <code>id</code> , <code>localizations</code> , <code>player</code> , <code>snippet</code> , <code>status</code> . Default: <code>contentDetails</code> .
<code>max_results</code>	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
<code>hl</code>	Language used for text values. Optional. Default is <code>en-US</code> . For other allowed language codes, see list_langs .
<code>page_token</code>	specific page in the result set that should be returned, optional
<code>simplify</code>	Data Type: Boolean. Default is <code>TRUE</code> . If <code>TRUE</code> and if <code>part</code> requested is <code>contentDetails</code> , the function returns a <code>data.frame</code> . Else a list with all the information returned.
<code>...</code>	Additional arguments passed to tuber_GET .

Value

playlists When simplify is TRUE, a data.frame with 4 columns is returned: kind, etag, id, contentDetails.itemCount

References

<https://developers.google.com/youtube/v3/docs/playlists/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

get_playlists(filter=c(channel_id="UCMtFAi84ehTSYSE9XoHefig"))
get_playlists(filter=c(channel_id="UCMtFAi84ehTSYSE9X")) # incorrect Channel ID

## End(Not run)
```

get_playlist_items *Get Playlist Items*

Description

Get Playlist Items

Usage

```
get_playlist_items(filter = NULL, part = "contentDetails",
  max_results = 50, video_id = NULL, page_token = NULL, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: item_id: comma-separated list of one or more unique playlist item IDs. playlist_id: YouTube playlist ID.
part	Required. Comma separated string including one or more of the following: contentDetails, id, snippet, status. Default: contentDetails.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
video_id	Optional. request should return only the playlist items that contain the specified video.
page_token	specific page in the result set that should be returned, optional
...	Additional arguments passed to tuber_GET .

Value

captions for the video from one of the first track

References

<https://developers.google.com/youtube/v3/docs/playlists/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_playlist_items(filter= c(playlist_id="PLrEnWoR732-CN09YykVof2lxdI3MLOZda"))  
  
## End(Not run)
```

get_related_videos	<i>Get Related Videos</i>
--------------------	---------------------------

Description

Takes a video id and returns related videos

Usage

```
get_related_videos(video_id = NULL, max_results = 50,  
  safe_search = "none", ...)
```

Arguments

video_id	Character. Required. No default.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
safe_search	Character. Optional. Takes one of three values: 'moderate', 'none' (default) or 'strict'
...	Additional arguments passed to tuber_GET .

Value

data.frame with 16 columns: video_id, publishedAt, channelId, title, description, thumbnails.default.url

References

<https://developers.google.com/youtube/v3/docs/search/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_related_videos(video_id="yJXTXN4xrI8")  
  
## End(Not run)
```

`get_stats`*Get statistics of a Video*

Description

Get statistics of a Video

Usage

```
get_stats(video_id = NULL, ...)
```

Arguments

<code>video_id</code>	Character. Id of the video. Required.
<code>...</code>	Additional arguments passed to <code>tuber_GET</code> .

Value

list with 6 elements: `id`, `viewCount`, `likeCount`, `dislikeCount`, `favoriteCount`, `commentCount`

References

<https://developers.google.com/youtube/v3/docs/videos/list#parameters>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_stats(video_id="N708P-A45D0")  
  
## End(Not run)
```

get_subscriptions *Get Subscriptions*

Description

Get Subscriptions

Usage

```
get_subscriptions(filter = NULL, part = "contentDetails",
  max_results = 50, for_channel_id = NULL, order = NULL,
  page_token = NULL, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: channel_id: ID of the channel. Required. No default. subscription_id: YouTube subscription ID
part	Part of the resource requested. Required. Character. A comma separated list of one or more of the following: contentDetails, id, snippet, subscriberSnippet. e.g. "id, snippet", "id", etc. Default: contentDetails.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
for_channel_id	Optional. String. A comma-separated list of channel IDs. Limits response to subscriptions matching those channels.
order	method that will be used to sort resources in the API response. Takes one of the following: alphabetical, relevance, unread
page_token	Specific page in the result set that should be returned. Optional. String.
...	Additional arguments passed to <code>tuber_GET</code> .

Value

named list of subscriptions

References

<https://developers.google.com/youtube/v3/docs/subscriptions/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

get_subscriptions(filter=c(channel_id="UChTJTbr5kf3hYazJZ0-euHg"))

## End(Not run)
```

get_video_details *Get Details of a Video or Videos*

Description

Get details such as when the video was published, the title, description, thumbnails, category etc.

Usage

```
get_video_details(video_id = NULL, ...)
```

Arguments

video_id	Comma separated list of IDs of the videos for which details are requested. Required.
...	Additional arguments passed to tuber_GET .

Value

list with the following elements: id (video id that was passed), publishedAt, channelId, title, description, thumbnails

References

<https://developers.google.com/youtube/v3/docs/videos/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
get_video_details(video_id="yJXTXN4xrI8")  
  
## End(Not run)
```

list_abuse_report_reasons
List reasons that can be used to report abusive videos

Description

List reasons that can be used to report abusive videos

Usage

```
list_abuse_report_reasons(part = "id, snippet", hl = "en-US", ...)
```

Arguments

part	Caption resource requested. Required. Comma separated list of one or more of the following: id, snippet. e.g., "id, snippet", "id", etc. Default: snippet.
hl	Language used for text values. Optional. Default is en-US. For other allowed language codes, see list_langs .
...	Additional arguments passed to tuber_GET .

Value

If no results, empty data.frame returned
 If part requested = "id, snippet" or "snippet", data.frame with 4 columns: etag, id, label, secReasons
 If part requested = "id", data.frame with 2 columns: etag, id

References

<https://developers.google.com/youtube/v3/docs/videoAbuseReportReasons/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_abuse_report_reasons()
list_abuse_report_reasons(part="id")
list_abuse_report_reasons(part="snippet")

## End(Not run)
```

list_caption_tracks *Get Captions of a Video*

Description

Get Captions of a Video

Usage

```
list_caption_tracks(part = "snippet", video_id = NULL, lang = "en",
  id = NULL, simplify = TRUE, ...)
```

Arguments

part	Caption resource requested. Required. Comma separated list of one or more of the following: id, snippet. e.g., "id, snippet", "id" Default: snippet.
video_id	ID of the video whose captions are requested. Required. No default.
lang	Language of the caption; required; default is english ("en")
id	comma-separated list of IDs that identify the caption resources that should be retrieved; optional; string
simplify	Boolean. Default is TRUE. When TRUE, and part is snippet, a data.frame is returned
...	Additional arguments passed to <code>tuber_GET</code> .

Value

list of caption tracks. When `simplify` is TRUE, a data.frame is returned with following columns: videoId, lastUpdated, trackKind, language, name, audioTrackType, isCC, isLarge, isEasyReader, isDraft, (caption id)

References

<https://developers.google.com/youtube/v3/docs/captions/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_caption_tracks(video_id="yJXTXN4xrI8")

## End(Not run)
```

```
list_channel_activities
```

List Channel Activity

Description

Returns a list of channel events that match the request criteria.

Usage

```
list_channel_activities(filter = NULL, part = "snippet", max_results = 50,
  page_token = NULL, published_after = NULL, published_before = NULL,
  region_code = NULL, simplify = TRUE, ...)
```

Arguments

<code>filter</code>	string; Required. named vector of length 1 potential names of the entry in the vector: <code>channel_id</code> : ID of the channel. Required. No default.
<code>part</code>	specify which part do you want. It can only be one of the three: <code>contentDetails</code> , <code>id</code> , <code>snippet</code> . Default is <code>snippet</code> .
<code>max_results</code>	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
<code>page_token</code>	specific page in the result set that should be returned, optional
<code>published_after</code>	Character. Optional. RFC 339 Format. For instance, "1970-01-01T00:00:00Z"
<code>published_before</code>	Character. Optional. RFC 339 Format. For instance, "1970-01-01T00:00:00Z"
<code>region_code</code>	ISO 3166-1 alpha-2 country code, optional, see also list_regions
<code>simplify</code>	Data Type: Boolean. Default is TRUE. If TRUE and if part requested is <code>contentDetails</code> , the function returns a <code>data.frame</code> . Else a list with all the information returned.
<code>...</code>	Additional arguments passed to tuber_GET .

Value

named list If `simplify` is TRUE, a `data.frame` is returned with 18 columns: `publishedAt`, `channelId`, `title`, `description`

References

<https://developers.google.com/youtube/v3/docs/activities/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_channel_activities(filter = c(channel_id="UCRw8bIz2wMLmfgAgWm903cA"))
list_channel_activities(filter = c(channel_id="UCRw8bIz2wMLmfgAgWm903cA", regionCode="US"))
list_channel_activities(filter = c(channel_id="UCMtFAi84ehTSYSE9XoHefig"),
                        published_before = "2016-02-10T00:00:00Z",
                        published_after="2016-01-01T00:00:00Z")

## End(Not run)
```

`list_channel_resources`*Returns List of Requested Channel Resources*

Description

Returns List of Requested Channel Resources

Usage

```
list_channel_resources(filter = NULL, part = "contentDetails",  
  max_results = 50, page_token = NULL, hl = NULL, ...)
```

Arguments

<code>filter</code>	string; Required. named vector of length 1 potential names of the entry in the vector: <code>category_id</code> : YouTube guide category that returns channels associated with that category <code>username</code> : YouTube username that returns channel associated with that username. <code>channel_id</code> : a comma-separated list of the YouTube channel ID(s) for the resource(s) that are being retrieved
<code>part</code>	a comma separated list of channel resource properties that response will include string. Required. One of the following: <code>auditDetails</code> , <code>brandingSettings</code> , <code>contentDetails</code> , <code>contentOwnerDetails</code> . Default is <code>contentDetails</code> .
<code>max_results</code>	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
<code>page_token</code>	specific page in the result set that should be returned, optional
<code>hl</code>	Language used for text values. Optional. Default is <code>en-US</code> . For other allowed language codes, see list_langs .
<code>...</code>	Additional arguments passed to tuber_GET .

Value

list

References

<https://developers.google.com/youtube/v3/docs/channels/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
list_channel_resources(filter = c(channel_id = "UCT5Cx114IS3wHkJXNyuj4TA"))
```

```
list_channel_resources(filter = c(username = "latenight"), part="id, contentDetails")
list_channel_resources(filter = c(username = "latenight"), part="id, contentDetails",
max_results=10)

## End(Not run)
```

list_channel_sections *List Channel Sections*

Description

Returns list of channel sections that channel id belongs to.

Usage

```
list_channel_sections(filter = NULL, part = "snippet", hl = NULL, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: channel_id: Channel ID id: Section ID
part	specify which part do you want. It can only be one of the following: contentDetails, id, localization. Default is snippet.
hl	language that will be used for text values, optional, default is en-US. See also list_langs
...	Additional arguments passed to tuber_GET .

Value

captions for the video from one of the first track

References

<https://developers.google.com/youtube/v3/docs/activities/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_channel_sections(c(channel_id="UCRw8bIz2wMLmfgAgWm903cA"))

## End(Not run)
```

list_channel_videos *Returns List of Requested Channel Videos*

Description

Returns List of Requested Channel Videos

Usage

```
list_channel_videos(channel_id = NULL, max_results = 50, ...)
```

Arguments

channel_id	String. ID of the channel. Required.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
...	Additional arguments passed to tuber_GET .

Value

character vector of video IDs

References

<https://developers.google.com/youtube/v3/docs/channels/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
list_channel_videos(channel_id = "UCT5Cx114IS3wHkJXNyuj4TA")  
list_channel_videos(channel_id = "UCT5Cx114IS3wHkJXNyuj4TA", max_results=10)  
  
## End(Not run)
```

list_guidecats	<i>Get list of categories that can be associated with YouTube channels</i>
----------------	--

Description

Get list of categories that can be associated with YouTube channels

Usage

```
list_guidecats(filter = NULL, hl = NULL, ...)
```

Arguments

filter	string; Required. named vector of length 1 potential names of the entry in the vector: region_code: Character. Required. Has to be a ISO 3166-1 alpha-2 code (see https://www.iso.org/obp/ui/#search) category_id: YouTube channel category ID
hl	Language used for text values. Optional. Default is en-US. For other allowed language codes, see list_langs .
...	Additional arguments passed to tuber_GET .

Value

data.frame with 5 columns: region_code, channelId, title, etag, id

References

<https://developers.google.com/youtube/v3/docs/guideCategories/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
list_guidecats(c(region_code = "JP"))  
  
## End(Not run)
```

list_langs	<i>List Languages That Youtube Currently Supports</i>
------------	---

Description

List Languages That Youtube Currently Supports

Usage

```
list_langs(hl = NULL, ...)
```

Arguments

hl	Language used for text values. Optional. Default is en-US. For other allowed language codes, see list_langs .
...	Additional arguments passed to tuber_GET .

Value

data.frame with 3 columns: hl (two letter abbreviation), name (of the language), etag

References

<https://developers.google.com/youtube/v3/docs/i18nLanguages/list>

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
list_langs()  
  
## End(Not run)
```

list_regions	<i>List Content Regions That Youtube Currently Supports</i>
--------------	---

Description

List Content Regions That Youtube Currently Supports

Usage

```
list_regions(hl = NULL, ...)
```

Arguments

hl Language used for text values. Optional. Default is en-US. For other allowed language codes, see [list_langs](#).

... Additional arguments passed to [tuber_GET](#).

Value

data.frame with 3 columns: gl (two letter abbreviation), name (of the region), etag

References

<https://developers.google.com/youtube/v3/docs/i18nRegions/list>

Examples

```
## Not run:
# Set API token via yt_oauth() first
list_regions()
## End(Not run)
```

list_videocats	<i>List of Categories That Can be Associated with Videos</i>
----------------	--

Description

List of Categories That Can be Associated with Videos

Usage

```
list_videocats(filter = NULL, ...)
```

Arguments

filter string; Required. named vector of length 1 potential names of the entry in the vector: region_code: Character. Required. Has to be a ISO 3166-1 alpha-2 code (see <https://www.iso.org/obp/ui/#search>) category_id: video category ID

... Additional arguments passed to [tuber_GET](#).

Value

data.frame with 6 columns: region_code, channelId, title, assignable, etag, id

References

<https://developers.google.com/youtube/v3/docs/videoCategories/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_videocats(c(region_code = "JP"))
list_videocats() # Will throw an error asking for a valid filter with valid region_code

## End(Not run)
```

list_videos	<i>List (Most Popular) Videos</i>
-------------	-----------------------------------

Description

List (Most Popular) Videos

Usage

```
list_videos(part = "contentDetails", max_results = 50, page_token = NULL,
            hl = NULL, region_code = NULL, video_category_id = NULL, ...)
```

Arguments

part	Required. Comma separated string including one or more of the following: contentDetails, fileDetails, id, liveStreamingDetails, localizations, player, processingDetails. Default: contentDetails.
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50.
page_token	specific page in the result set that should be returned, optional
hl	Language used for text values. Optional. Default is en-US. For other allowed language codes, see list_langs .
region_code	Character. Required. Has to be a ISO 3166-1 alpha-2 code (see https://www.iso.org/obp/ui/#search).
video_category_id	the video category for which the chart should be retrieved. See also list_videocats .
...	Additional arguments passed to tuber_GET .

Value

data.frame with 5 columns: channelId, title, assignable, etag, id

References

<https://developers.google.com/youtube/v3/docs/search/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

list_videos()

## End(Not run)
```

tuber	<i>tuber provides access to the YouTube API V3.</i>
-------	---

Description

tuber provides access to the YouTube API V3.

tuber_check	<i>Request Response Verification</i>
-------------	--------------------------------------

Description

Request Response Verification

Usage

```
tuber_check(req)
```

Arguments

req	request
-----	---------

Value

in case of failure, a message

tuber_GET	<i>Base POST AND GET functions. Not exported.</i>
-----------	---

Description

GET

Usage

tuber_GET(path, query, ...)

Arguments

path	path to specific API request URL
query	query list
...	Additional arguments passed to GET .

Value

list

tuber_POST	<i>POST</i>
------------	-------------

Description

POST

Usage

tuber_POST(path, query, body = "", ...)

Arguments

path	path to specific API request URL
query	query list
body	passing image through body
...	Additional arguments passed to GET .

Value

list

yt_check_token	<i>Check if authentication token is in options</i>
----------------	--

Description

Check if authentication token is in options

Usage

```
yt_check_token()
```

yt_oauth	<i>Set up Authorization</i>
----------	-----------------------------

Description

The function looks for `.httr-oauth` in the working directory. If it doesn't find it, it expects an application ID and a secret. If you want to remove the existing `.httr-oauth`, set `remove_old_oauth` to `TRUE`. By default, it is set to `FALSE`. The function launches a browser to allow you to authorize the application

Usage

```
yt_oauth(app_id = NULL, app_secret = NULL, scope = "ssl",
         token = ".httr-oauth", ...)
```

Arguments

app_id	client id; required; no default
app_secret	client secret; required; no default
scope	Character. <code>ssl</code> , <code>basic</code> , <code>own_account_readonly</code> , <code>upload_and_manage_own_videos</code> and <code>partner_audit</code> . Required. <code>ssl</code> and <code>basic</code> are basically interchangeable. Default is <code>ssl</code> .
token	path to file containing the token. If a path is given, the function will first try to read from it. Default is <code>.httr-oauth</code> in the local directory. So if there is such a file, the function will first try to read from it.
...	Additional arguments passed to <code>oauth2.0_token</code>

Value

sets the `google_token` option and also saves `.httr_oauth` in the working directory (find out the working directory via `getwd()`)

References

<https://developers.google.com/youtube/v3/docs/>

<https://developers.google.com/youtube/v3/guides/auth/client-side-web-apps> for different scopes

Examples

```
## Not run:
yt_oauth("998136489867-5t3tq1g7hbovoj46dreqd6k5kd35ctjn.apps.googleusercontent.com",
        "Mb0St6cQhhFkwETXKur-L9rN")

## End(Not run)
```

yt_search

Search YouTube

Description

Search for videos, channels and playlists. (By default, the function searches for videos.)

Usage

```
yt_search(term = NULL, max_results = 50, channel_id = NULL,
          channel_type = NULL, type = "video", event_type = NULL,
          location = NULL, location_radius = NULL, published_after = NULL,
          published_before = NULL, video_definition = "any",
          video_caption = "any", video_license = "any", video_syndicated = "any",
          video_type = "any", simplify = TRUE, page_token = NULL, ...)
```

Arguments

term	Character. Search term; required; no default
max_results	Maximum number of items that should be returned. Integer. Optional. Can be between 0 and 50. Default is 50. Search results are constrained to a maximum of 500 videos if type is video and we have a value of channel_id.
channel_id	Character. Only return search results from this channel; Optional.
channel_type	Character. Optional. Takes one of two values: 'any' or 'show'. Default is 'any'
type	Character. Optional. Takes one of three values: 'video', 'channel', or 'playlist'. Default is 'video'.
event_type	Character. Optional. Takes one of three values: 'completed', 'live' or 'upcoming'
location	Character. Optional. Latitude and Longitude within parentheses, e.g. "(37.42307,-122.08427)"
location_radius	Character. Optional. e.g. "1500m", "5km", "10000ft", "0.75mi"

published_after	Character. Optional. RFC 339 Format. For instance, "1970-01-01T00:00:00Z"
published_before	Character. Optional. RFC 339 Format. For instance, "1970-01-01T00:00:00Z"
video_definition	Character. Optional. Takes one of three values: 'any' (return all videos; Default), 'high' and 'standard'
video_caption	Character. Optional. Takes one of three values: 'any' (return all videos; Default), 'closedCaption', 'none'. Type must be set to video.
video_license	Character. Optional. Takes one of three values: 'any' (return all videos; Default), 'creativeCommon' (return videos with Creative Commons license), 'youtube' (return videos with standard YouTube license).
video_syndicated	Character. Optional. Takes one of two values: 'any' (return all videos; Default), 'true' (return only syndicated videos)
video_type	Character. Optional. Takes one of three values: 'any' (return all videos; Default), 'episode' (return episode of shows), 'movie' (return movies)
simplify	Boolean. Return a data.frame if TRUE. Default is TRUE. If FALSE, it returns a list that carries additional information.
page_token	specific page in the result set that should be returned, optional
...	Additional arguments passed to <code>tuber_GET</code> .

Value

data.frame with 15 elements: publishedAt, channelId, title, description, thumbnails.default.url, thumbnail

References

<https://developers.google.com/youtube/v3/docs/search/list>

Examples

```
## Not run:

# Set API token via yt_oauth() first

yt_search(term="Barack Obama")
yt_search(term="Barack Obama", published_after="2016-10-01T00:00:00Z")
yt_search(term="Barack Obama", published_before="2016-09-01T00:00:00Z")
yt_search(term="Barack Obama", published_before="2016-03-01T00:00:00Z",
          published_after="2016-02-01T00:00:00Z")
yt_search(term="Barack Obama", published_before = "2016-02-10T00:00:00Z",
          published_after="2016-01-01T00:00:00Z")

## End(Not run)
```

yt_topic_search	<i>Search YouTube by Topic It uses the Freebase list of topics</i>
-----------------	--

Description

Search YouTube by Topic It uses the Freebase list of topics

Usage

```
yt_topic_search(topic = NULL, ...)
```

Arguments

topic	topic being searched for; required; no default
...	Additional arguments passed to tuber_GET .

Value

a list

Examples

```
## Not run:  
  
# Set API token via yt_oauth() first  
  
yt_topic_search(topic="Barack Obama")  
  
## End(Not run)
```

Index

GET, [23](#)
get_captions, [2](#)
get_channel_stats, [3](#)
get_comment_threads, [5](#)
get_comments, [4](#)
get_playlist_items, [7](#)
get_playlists, [6](#)
get_related_videos, [8](#)
get_stats, [9](#)
get_subscriptions, [10](#)
get_video_details, [11](#)

list_abuse_report_reasons, [11](#)
list_caption_tracks, [2](#), [12](#)
list_channel_activities, [13](#)
list_channel_resources, [15](#)
list_channel_sections, [16](#)
list_channel_videos, [17](#)
list_guidcats, [18](#)
list_langs, [6](#), [12](#), [15](#), [16](#), [18](#), [19](#), [19](#), [20](#), [21](#)
list_regions, [14](#), [19](#)
list_videocats, [20](#), [21](#)
list_videos, [21](#)

oauth2.0_token, [24](#)

tuber, [22](#)
tuber-package (tuber), [22](#)
tuber_check, [22](#)
tuber_GET, [2-21](#), [23](#), [26](#), [27](#)
tuber_POST, [23](#)

yt_check_token, [24](#)
yt_oauth, [24](#)
yt_search, [25](#)
yt_topic_search, [27](#)