

Package ‘traktok’

November 24, 2025

Type Package

Title Collecting 'TikTok' Data

Version 0.1.0

Description Getting 'TikTok' data (<<https://www.tiktok.com/>>) through the official and unofficial APIs—in other words, you can track 'TikTok'.

License GPL (>= 3)

Depends R (>= 4.2.0)

Imports askpass, cli, cookiemonster, curl, dplyr, glue, httr2, jsonlite, lobstr, methods, openssl, purrr, rlang, rvest, stats, tibble

Suggests covr, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

URL <https://github.com/JBGruber/traktok>,
<https://jbgruber.github.io/traktok/>

BugReports <https://github.com/JBGruber/traktok/issues>

Encoding UTF-8

RoxygenNote 7.3.3

Language en-GB

Config/testthat/edition 3

Config/testthat/parallel false

VignetteBuilder knitr

NeedsCompilation no

Author Johannes B. Gruber [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9177-1772>>)

Maintainer Johannes B. Gruber <JohannesB.Gruber@gmail.com>

Repository CRAN

Date/Publication 2025-11-24 18:00:02 UTC

Contents

auth_check	2
auth_hidden	3
auth_research	4
last_query	5
print.traktok_query	5
print.tt_results	6
query	6
tt_comments_api	8
tt_get_follower	9
tt_get_following_hidden	9
tt_playlist_api	10
tt_request_hidden	11
tt_search	12
tt_search_api	12
tt_search_hidden	14
tt_user_follower_api	16
tt_user_info_api	17
tt_user_info_hidden	18
tt_user_liked_videos_api	19
tt_user_pinned_videos_api	20
tt_user_reposted_api	21
tt_user_videos	22
tt_user_videos_api	23
tt_user_videos_hidden	24
tt_videos_hidden	25

Index	27
-------	----

auth_check	<i>Check whether you are authenticated</i>
------------	--

Description

[Works on: Both]

Check if the necessary token or cookies are stored on your computer already. By default, the function checks for the authentication of the research and hidden API. To learn how you can authenticate, see the [research API vignette](#) or [hidden API vignette](#). You can also view these locally with `vignette("research-api", package = "traktok")` and `vignette("unofficial-api", package = "traktok")`.

Usage

```
auth_check(research = TRUE, hidden = TRUE, silent = FALSE, fail = FALSE)
```

Arguments

research, hidden turn check on/off for the research or hidden API.

silent only return if check(s) were successful, no status on the screen

fail fail if even basic authentication for the hidden API is missing.

Value

logical vector (invisible)

Examples

```
auth_check()

au <- auth_check()
if (isTRUE(au["research"])) {
  message("Ready to use the research API!")
}
if (isTRUE(au["hidden"])) {
  message("Ready to use all function of unofficial the API!")
}
```

auth_hidden	<i>Authenticate for the hidden/unofficial API</i>
-------------	---

Description

Guides you through authentication for the hidden/unofficial API. To learn more, see the [hidden API vignette](#) or view it locally with `vignette("unofficial-api", package = "traktok")`.

Usage

```
auth_hidden(cookiefile, live = interactive())
```

Arguments

cookiefile path to your cookiefile. Usually not needed after running [auth_hidden](#) once. See `vignette("unofficial-api", package = "traktok")` for more information on authentication.

live opens Chromium browser to guide you through the auth process (experimental).

Value

nothing. Called to set up authentication

Examples

```
## Not run:  
# to run through the steps of authentication  
auth_hidden()  
# or point to a cookie file directly  
auth_hidden("www.tiktok.com_cookies.txt")  
  
## End(Not run)
```

auth_research

Authenticate for the official research API

Description

Guides you through authentication for the Research API

Usage

```
auth_research(client_key, client_secret)
```

Arguments

client_key	Client key for authentication
client_secret	Client secret for authentication

Details

You need to apply for access to the API and get the key and secret from TikTok. See <https://developers.tiktok.com/products/research-api/> for more information.

Value

An authentication token (invisible).

Examples

```
## Not run:  
auth_research(client_key, client_secret)  
  
## End(Not run)
```

last_query	<i>Retrieve most recent query</i>
------------	-----------------------------------

Description

If `tt_search_api` or `tt_comments_api` fail after already getting several pages, you can use this function to get all videos that have been retrieved so far from memory. Does not work when the session has crashed. In that case, look in `tempdir()` for an RDS file as a last resort.

Usage

```
last_query()
```

```
last_comments()
```

Value

a list of unparsed videos or comments.

<code>print.traktok_query</code>	<i>Print a traktok query</i>
----------------------------------	------------------------------

Description

Print a traktok query as a tree

Usage

```
## S3 method for class 'traktok_query'
print(x, ...)
```

Arguments

<code>x</code>	An object of class <code>traktok_query</code>
<code>...</code>	Additional arguments passed to <code>lobstr:::tree</code>

Value

nothing. Prints traktok query.

Examples

```
query() |>
  query_and(field_name = "hashtag_name",
            operation = "EQ",
            field_values = "rstats") |>
  print()
```

<code>print.tt_results</code>	<i>Print search result</i>
-------------------------------	----------------------------

Description

Print a traktok search results

Usage

```
## S3 method for class 'tt_results'
print(x, ...)
```

Arguments

<code>x</code>	An object of class <code>tt_results</code>
<code>...</code>	not used.

Value

nothing. Prints search results.

<code>query</code>	<i>Create a traktok query</i>
--------------------	-------------------------------

Description

Create a traktok query from the given parameters.

Usage

```
query(and = NULL, or = NULL, not = NULL)

query_and(q, field_name, operation, field_values)

query_or(q, field_name, operation, field_values)

query_not(q, field_name, operation, field_values)
```

Arguments

<code>and, or, not</code>	A list of AND/OR/NOT conditions. Must contain one or multiple lists with <code>field_name</code> , <code>operation</code> , and <code>field_values</code> each (see example).
<code>q</code>	A traktok query created with <code>query</code> .
<code>field_name</code>	The field name to query against. One of: "create_date", "username", "region_code", "video_id", "hashtag_name", "keyword", "music_id", "effect_id", "video_length".
<code>operation</code>	One of: "EQ", "IN", "GT", "GTE", "LT", "LTE".
<code>field_values</code>	A vector of values to search for.

Details

TikTok's query consists of rather complicated lists dividing query elements into AND, OR and NOT:

- **and**: The and conditions specify that all the conditions in the list must be met
- **or**: The or conditions specify that at least one of the conditions in the list must be met
- **not**: The not conditions specify that none of the conditions in the list must be met

The query can be constructed by writing the list for each entry yourself, like in the first example. Alternatively, traktok provides convenience functions to build up a query using `query_and`, `query_or`, and `query_not`, which make building a query a little easier. You can learn more at <https://developers.tiktok.com/doc/research-api-specs-query-videos#query>.

Value

A traktok query.

Examples

```
## Not run:
# using query directly and supplying the list
query(or = list(
  list(
    field_name = "hashtag_name",
    operation = "EQ",
    field_values = "rstats"
  ),
  list(
    field_name = "keyword",
    operation = "EQ",
    field_values = list("rstats", "API")
  )
))
# starting an empty query and building it up using the query_* functions
query() |>
  query_or(field_name = "hashtag_name",
           operation = "EQ",
           field_values = "rstats") |>
  query_or(field_name = "keyword",
           operation = "IN",
           field_values = c("rstats", "API"))

## End(Not run)
```

tt_comments_api	<i>Retrieve video comments</i>
-----------------	--------------------------------

Description

[Works on: Research API]

Usage

```
tt_comments_api(
    video_id,
    fields = "all",
    start_cursor = 0L,
    max_pages = 1L,
    cache = TRUE,
    verbose = interactive(),
    token = NULL
)
```

```
tt_comments(
    video_id,
    fields = "all",
    start_cursor = 0L,
    max_pages = 1L,
    cache = TRUE,
    verbose = interactive(),
    token = NULL
)
```

Arguments

video_id	The id or URL of a video
fields	The fields to be returned (defaults to all)
start_cursor	The starting cursor, i.e., how many results to skip (for picking up an old search).
max_pages	results are returned in batches/pages with 100 videos. How many should be requested before the function stops?
cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed comments.

Examples

```
## Not run:
tt_comments("https://www.tiktok.com/@tiktok/video/7106594312292453675")
# OR
tt_comments("7106594312292453675")
# OR
tt_comments_api("7106594312292453675")

## End(Not run)
```

tt_get_follower	<i>Get followers and following of users</i>
-----------------	---

Description**[Works on: Both]**

Get usernames of users who follows a user (tt_get_follower) or get who a user is following (tt_get_following).

Usage

```
tt_get_follower(...)

tt_get_following(...)
```

Arguments

... arguments passed to [tt_user_follower_api](#) or [tt_get_follower_hidden](#). To use the research API, include token (e.g., token = NULL).

Value

a data.frame of followers and following of users

tt_get_following_hidden	<i>Get followers and following of a user from the hidden API</i>
-------------------------	--

Description**[Works on: Unofficial API]**

Get up to 5,000 accounts who follow a user or accounts a user follows.

Usage

```

tt_get_following_hidden(
  secuid,
  sleep_pool = 1:10,
  max_tries = 5L,
  cookiefile = NULL,
  verbose = interactive()
)

tt_get_follower_hidden(
  secuid,
  sleep_pool = 1:10,
  max_tries = 5L,
  cookiefile = NULL,
  verbose = interactive()
)

```

Arguments

secuid	The secuid of a user. You can get it with tt_user_info_hidden by querying an account (see example).
sleep_pool	a vector of numbers from which a waiting period is randomly drawn.
max_tries	how often to retry if a request fails.
cookiefile	path to your cookiefile. Usually not needed after running auth_hidden once. See <code>vignette("unofficial-api", package = "traktok")</code> for more information on authentication.
verbose	should the function print status updates to the screen?

Value

a data.frame of followers

Examples

```

## Not run:
df <- tt_user_info_hidden("https://www.tiktok.com/@fpoe_at")
tt_get_follower_hidden(df$secUid)

## End(Not run)

```

tt_playlist_api

Lookup TikTok playlist using the research API

Description

[Works on: Research API]

Usage

```
tt_playlist_api(playlist_id, verbose = interactive(), token = NULL)
```

```
tt_playlist(playlist_id, verbose = interactive(), token = NULL)
```

Arguments

playlist_id	playlist ID or URL to a playlist.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame video metadata.

tt_request_hidden	<i>Get json string from a TikTok URL using the hidden API</i>
-------------------	---

Description**[Works on: Unofficial API]**

Use this function in case you want to check the full data for a given TikTok video or account. In `tt_videos`, only an opinionated selection of data is included in the final object. If you want some different information, you can use this function.

Usage

```
tt_request_hidden(url, max_tries = 5L, cookiefile = NULL)
```

Arguments

url	a URL to a TikTok video or account
max_tries	how often to retry if a request fails.
cookiefile	path to your cookiefile. Usually not needed after running auth_hidden once. See <code>vignette("unofficial-api", package = "traktok")</code> for more information on authentication.

Value

a json string containing post or account data.

tt_search	<i>Search videos</i>
-----------	----------------------

Description**[Works on: Both]**

Searches videos using either the Research API (if an authentication token is present, see [auth_research](#)) or otherwise the unofficial hidden API. See [tt_search_api](#) or [tt_search_hidden](#) respectively for information about these functions.

Usage

```
tt_search(...)
```

Arguments

... arguments passed to [tt_search_api](#) or [tt_search_hidden](#). To use the research API, include token (e.g., token = NULL).

Value

a data.frame of video metadata

tt_search_api	<i>Query TikTok videos using the research API</i>
---------------	---

Description**[Works on: Research API]**

This is the version of [tt_search](#) that explicitly uses Research API. Use [tt_search_hidden](#) for the unofficial API version.

Usage

```
tt_search_api(
  query,
  start_date = Sys.Date() - 1,
  end_date = Sys.Date(),
  fields = "all",
  start_cursor = 0L,
  search_id = NULL,
  is_random = FALSE,
  max_pages = 1,
  parse = TRUE,
  cache = TRUE,
```

```

    verbose = interactive(),
    token = NULL
)

tt_query_videos(
  query,
  start_date = Sys.Date() - 1,
  end_date = Sys.Date(),
  fields = "all",
  start_cursor = 0L,
  search_id = NULL,
  is_random = FALSE,
  max_pages = 1,
  parse = TRUE,
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)

```

Arguments

query	A query string or object (see query).
start_date, end_date	A start and end date to narrow the search (required; can be a maximum of 30 days apart).
fields	The fields to be returned (defaults to all)
start_cursor	The starting cursor, i.e., how many results to skip (for picking up an old search).
search_id	The search id (for picking up an old search).
is_random	Whether the query is random (defaults to FALSE).
max_pages	results are returned in batches/pages with 100 videos. How many should be requested before the function stops?
parse	Should the results be parsed? Otherwise, the original JSON object is returned as a nested list.
cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed TikTok videos (or a nested list).

Examples

```
## Not run:
# look for a keyword or hashtag by default
tt_search_api("rstats")

# or build a more elaborate query
query() |>
  query_and(field_name = "region_code",
            operation = "IN",
            field_values = c("JP", "US")) |>
  query_or(field_name = "hashtag_name",
            operation = "EQ", # rstats is the only hashtag
            field_values = "rstats") |>
  query_or(field_name = "keyword",
            operation = "IN", # rstats is one of the keywords
            field_values = "rstats") |>
  query_not(operation = "EQ",
            field_name = "video_length",
            field_values = "SHORT") |>
  tt_search_api()

# when a search fails after a while, get the results and pick it back up
# (only work with same parameters)
last_pull <- last_query()
query() |>
  query_and(field_name = "region_code",
            operation = "IN",
            field_values = c("JP", "US")) |>
  query_or(field_name = "hashtag_name",
            operation = "EQ", # rstats is the only hashtag
            field_values = "rstats") |>
  query_or(field_name = "keyword",
            operation = "IN", # rstats is one of the keywords
            field_values = "rstats") |>
  query_not(operation = "EQ",
            field_name = "video_length",
            field_values = "SHORT") |>
  tt_search_api(start_cursor = length(last_pull) + 1,
                search_id = attr(last_pull, "search_id"))

## End(Not run)
```

tt_search_hidden

Search videos

Description

[Works on: Unofficial API]

This is the version of [tt_search](#) that explicitly uses the unofficial API. Use [tt_search_api](#) for the Research API version.

Usage

```

tt_search_hidden(
  query,
  solve_captchas = FALSE,
  timeout = 5L,
  scroll = "5m",
  return_urls = FALSE,
  save_video = FALSE,
  verbose = interactive(),
  headless = TRUE,
  ...
)

```

Arguments

query	query as one string.
solve_captchas	open browser to solve appearing captchas manually.
timeout	time (in seconds) to wait between scrolling and solving captchas.
scroll	how long to keep scrolling before returning results. Can be a numeric value of seconds or a string with seconds, minutes, hours or days (see examples).
return_urls	return video URLs instead of downloading the vidoes.
save_video	passed to tt_videos_hidden if return_urls = FALSE.
verbose	should the function print status updates to the screen?
headless	open the browser to show the scrolling.
...	Additional arguments to be passed to the tt_videos_hidden function.

Details

The function will wait between scraping search results. To get more than 6 videos, you need to provide cookies of a logged in account. For more details see the unofficial-api vignette: `vignette("unofficial-api", package = "traktok")`

Value

a data.frame containing metadata searched posts or character vector of URLs.

Examples

```

## Not run:
# search videos with hashtag #rstats for default time
tt_search_hidden("#rstats")

# search videos for 10 seconds
tt_search_hidden("#rstats", scroll = "10s")
tt_search_hidden("#rstats", scroll = 10)

# search videos for 10 minutes

```

```

tt_search_hidden("#rstats", scroll = "10m")
tt_search_hidden("#rstats", scroll = "10mins")

# search videos for 10 hours
tt_search_hidden("#rstats", scroll = "10h")
tt_search_hidden("#rstats", scroll = "10hours")

# search videos until all are found
tt_search_hidden("#rstats", scroll = Inf)
# the functions runs until the end of all search results, which can take a
# long time. You can cancel the search and retrieve all collected results
# with last_query though!
last_query()

## End(Not run)

```

tt_user_follower_api *Get followers and following of users from the research API*

Description

[Works on: Research API]

Usage

```

tt_user_follower_api(
  username,
  max_pages = 1,
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)

tt_user_following_api(
  username,
  max_pages = 1,
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)

```

Arguments

username	name(s) of the user(s) to be queried
max_pages	results are returned in batches/pages with 100 videos. How many should be requested before the function stops?
cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.

verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame containing follower of following account information.

Examples

```
## Not run:
tt_user_follower_api("jbgruber")
# OR
tt_user_following_api("https://www.tiktok.com/@tiktok")
# OR
tt_get_follower("https://www.tiktok.com/@tiktok")

## End(Not run)
```

tt_user_info_api	<i>Lookup TikTok information about a user using the research API</i>
------------------	--

Description

[Works on: Research API]

Usage

```
tt_user_info_api(
  username,
  fields = "all",
  verbose = interactive(),
  token = NULL
)

tt_user_info(username, fields = "all", verbose = interactive(), token = NULL)
```

Arguments

username	name(s) of the user(s) to be queried
fields	The fields to be returned (defaults to all)
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed TikTok videos the user has posted.

Examples

```
## Not run:
tt_user_info_api("jbgruber")
# OR
tt_user_info_api("https://www.tiktok.com/@tiktok")
# OR
tt_user_info("https://www.tiktok.com/@tiktok")

## End(Not run)
```

tt_user_info_hidden	<i>Get infos about a user from the hidden API</i>
---------------------	---

Description

[Works on: Unofficial API]

Access the publicly available information about a user.

Usage

```
tt_user_info_hidden(username, parse = TRUE)
```

Arguments

username	A URL to a video or username.
parse	Whether to parse the data into a data.frame (set to FALSE to get the full list).

Value

A data.frame or list of user info.

Examples

```
## Not run:
df <- tt_user_info_hidden("https://www.tiktok.com/@fpoe_at")

## End(Not run)
```

`tt_user_liked_videos_api`*Lookup which videos were liked by a user using the research API*

Description

[Works on: Research API]

Usage

```
tt_user_liked_videos_api(  
    username,  
    fields = "all",  
    max_pages = 1,  
    cache = TRUE,  
    verbose = interactive(),  
    token = NULL  
)
```

```
tt_get_liked(  
    username,  
    fields = "all",  
    max_pages = 1,  
    cache = TRUE,  
    verbose = interactive(),  
    token = NULL  
)
```

Arguments

username	name(s) of the user(s) to be queried
fields	The fields to be returned (defaults to all)
max_pages	results are returned in batches/pages with 100 videos. How many should be requested before the function stops?
cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed TikTok videos the user has posted.

Examples

```
## Not run:
tt_get_liked("jbgruber")
# OR
tt_user_liked_videos_api("https://www.tiktok.com/@tiktok")
# OR
tt_user_liked_videos_api("https://www.tiktok.com/@tiktok")

# note: none of these work because I could not find any account that
# has likes public!

## End(Not run)
```

tt_user_pinned_videos_api

Lookup which videos were pinned by a user using the research API

Description

[Works on: Research API]

Usage

```
tt_user_pinned_videos_api(
  username,
  fields = "all",
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)

tt_get_pinned(
  username,
  fields = "all",
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)
```

Arguments

username	vector of user names (handles) or URLs to users' pages.
fields	The fields to be returned (defaults to all)
cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed TikTok videos the user has posted.

Examples

```
## Not run:
tt_get_pinned("jbgruber")
# OR
tt_user_pinned_videos_api("https://www.tiktok.com/@tiktok")
# OR
tt_user_pinned_videos_api("https://www.tiktok.com/@tiktok")

## End(Not run)
```

tt_user_reposted_api *Lookup which videos were liked by a user using the research API*

Description

[Works on: Research API]

Usage

```
tt_user_reposted_api(
  username,
  fields = "all",
  max_pages = 1,
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)

tt_get_reposted(
  username,
  fields = "all",
  max_pages = 1,
  cache = TRUE,
  verbose = interactive(),
  token = NULL
)
```

Arguments

username	name(s) of the user(s) to be queried
fields	The fields to be returned (defaults to all)
max_pages	results are returned in batches/pages with 100 videos. How many should be requested before the function stops?

cache	should progress be saved in the current session? It can then be retrieved with <code>last_query()</code> if an error occurs. But the function will use extra memory.
verbose	should the function print status updates to the screen?
token	The authentication token (usually supplied automatically after running auth_research once).

Value

A data.frame of parsed TikTok videos the user has posted.

Examples

```
## Not run:
tt_get_reposted("jbgruber")
# OR
tt_user_reposted_api("https://www.tiktok.com/@tiktok")
# OR
tt_user_reposted_api("https://www.tiktok.com/@tiktok")

# note: none of these work because nobody has this enabled!

## End(Not run)
```

tt_user_videos	<i>Get videos from a TikTok user's profile</i>
----------------	--

Description**[Works on: Both]**

Get all videos posted by a user (or multiple user's for the Research API). Searches videos using either the Research API (if an authentication token is present, see [auth_research](#)) or otherwise the unofficial hidden API. See [tt_user_videos_api](#) or [tt_user_videos_hidden](#) respectively for information about these functions.

Usage

```
tt_user_videos(username, ...)
```

Arguments

username	The username or usernames whose videos you want to retrieve.
...	Additional arguments to be passed to the tt_user_videos_hidden or tt_user_videos_api function.

Value

a data.frame containing metadata of user posts.

Examples

```
## Not run:
# Get hidden videos from the user "fpoe_at"
tt_user_videos("fpoe_at")

## End(Not run)
```

tt_user_videos_api	<i>Get videos from a TikTok user's profile</i>
--------------------	--

Description**[Works on: Research API]**

Get all videos posted by a user or multiple user's. This is a convenience wrapper around [tt_search_api](#) that takes care of moving time windows (search is limited to 30 days). This is the version of [tt_user_videos](#) that explicitly uses Research API. Use [tt_user_videos_hidden](#) for the unofficial API version.

Usage

```
tt_user_videos_api(
  username,
  since = "2020-01-01",
  to = Sys.Date(),
  verbose = interactive(),
  ...
)
```

Arguments

username	The username or usernames whose videos you want to retrieve.
since, to	limits from/to when to go through the account in 30 day windows.
verbose	should the function print status updates to the screen?
...	Additional arguments to be passed to the tt_search_api function.

Value

a data.frame containing metadata of user posts.

Examples

```
## Not run:
# Get videos from the user "fpoe_at" since October 2024
tt_user_videos_api("fpoe_at", since = "2024-10-01")

# often makes sense to combine this with the account creation time from the
# hidden URL
```

```
fpoet_at_info <- tt_user_info_hidden(username = "fpoet_at")
tt_user_videos_api("fpoet_at", since = fpoet_at_info$create_time)

## End(Not run)
```

tt_user_videos_hidden *Get videos from a TikTok user's profile*

Description

[Works on: Unofficial API]

Get all videos posted by a TikTok user.

Usage

```
tt_user_videos_hidden(
  username,
  solve_captchas = FALSE,
  return_urls = FALSE,
  save_video = FALSE,
  timeout = 5L,
  scroll = "5m",
  verbose = interactive(),
  ...
)
```

Arguments

username	The username of the TikTok user whose hidden videos you want to retrieve.
solve_captchas	open browser to solve appearing captchas manually.
return_urls	return video URLs instead of downloading the videos.
save_video	passed to tt_videos_hidden if return_urls = FALSE.
timeout	time (in seconds) to wait between scrolling and solving captchas.
scroll	how long to keep scrolling before returning results. Can be a numeric value of seconds or a string with seconds, minutes, hours or days (see examples).
verbose	should the function print status updates to the screen?
...	Additional arguments to be passed to the tt_videos_hidden function.

Details

This function uses rvest to scrape a TikTok user's profile and retrieve any hidden videos.

Value

A list of video data or URLs, depending on the value of return_urls.

a data.frame containing metadata of user posts or character vector of URLs.

Examples

```
## Not run:
# Get hidden videos from the user "fpoe_at"
tt_user_videos_hidden("fpoe_at")

## End(Not run)
```

tt_videos_hidden	<i>Get video metadata and video files from URLs</i>
------------------	---

Description

[Works on: Unofficial API]

Usage

```
tt_videos_hidden(
  video_urls,
  save_video = TRUE,
  overwrite = FALSE,
  dir = ".",
  cache_dir = NULL,
  sleep_pool = 1:10,
  max_tries = 5L,
  cookiefile = NULL,
  verbose = interactive(),
  ...
)

tt_videos(...)
```

Arguments

video_urls	vector of URLs or IDs to TikTok videos.
save_video	logical. Should the videos be downloaded.
overwrite	logical. If save_video=TRUE and the file already exists, should it be overwritten?
dir	directory to save videos files to.
cache_dir	if set to a path, one RDS file with metadata will be written to disk for each video. This is useful if you have many videos and want to pick up where you left if something goes wrong.

sleep_pool	a vector of numbers from which a waiting period is randomly drawn.
max_tries	how often to retry if a request fails.
cookiefile	path to your cookiefile. Usually not needed after running auth_hidden once. See <code>vignette("unofficial-api", package = "traktok")</code> for more information on authentication.
verbose	should the function print status updates to the screen?
...	handed to <code>tt_videos_hidden</code> (for <code>tt_videos</code>) and (further) to tt_request_hidden .

Details

The function will wait between scraping two videos to make it less obvious that a scraper is accessing the site. The period is drawn randomly from the `sleep_pool` and multiplied by a random fraction.

Note that the video file has to be requested in the same session as the metadata. So while the URL to the video file is included in the metadata, this link will not work in most cases.

Value

a data.frame containing post metadata.

Examples

```
## Not run:
tt_videos("https://www.tiktok.com/@tiktok/video/7106594312292453675")

## End(Not run)
```

Index

auth_check, 2
auth_hidden, 3, 3, 10, 11, 26
auth_research, 4, 8, 11–13, 17, 19, 20, 22

last_comments (last_query), 5
last_query, 5

print.traktok_query, 5
print.tt_results, 6

query, 6, 13
query_and (query), 6
query_not (query), 6
query_or (query), 6

tt_comments (tt_comments_api), 8
tt_comments_api, 8
tt_get_follower, 9
tt_get_follower_hidden, 9
tt_get_follower_hidden
 (tt_get_following_hidden), 9
tt_get_following (tt_get_follower), 9
tt_get_following_hidden, 9
tt_get_liked
 (tt_user_liked_videos_api), 19
tt_get_pinned
 (tt_user_pinned_videos_api), 20
tt_get_reposted (tt_user_reposted_api),
 21
tt_playlist (tt_playlist_api), 10
tt_playlist_api, 10
tt_query_videos (tt_search_api), 12
tt_request_hidden, 11, 26
tt_search, 12, 12, 14
tt_search_api, 12, 12, 14, 23
tt_search_hidden, 12, 14
tt_user_follower_api, 9, 16
tt_user_following_api
 (tt_user_follower_api), 16
tt_user_info (tt_user_info_api), 17

tt_user_info_api, 17
tt_user_info_hidden, 10, 18
tt_user_liked_videos_api, 19
tt_user_pinned_videos_api, 20
tt_user_reposted_api, 21
tt_user_videos, 22, 23
tt_user_videos_api, 22, 23
tt_user_videos_hidden, 22, 23, 24
tt_videos (tt_videos_hidden), 25
tt_videos_hidden, 15, 24, 25