Package 'rtgstat'

May 9, 2026

Title Client for 'TGStat API'

Version 0.3.6

Description Allow function for using 'TGStat Stat API' and 'TGStat Search API', for more details see <https://api.tgstat.ru/docs/ru/start/intro.html>. 'TGStat' provide telegram channel analytics data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

URL https://selesnow.github.io/rtgstat/

BugReports https://github.com/selesnow/rtgstat/issues

Imports cli (>= 3.0.0), dplyr (>= 1.0.0), httr2 (>= 0.2.0), purrr, snakecase, stringr, tidyr (>= 1.0.0), utils

Language en-US

Suggests httptest2, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Alexey Seleznev [aut, cre] (ORCID: <https://orcid.org/0000-0003-0410-7385>)

Maintainer Alexey Seleznev <selesnow@gmail.com>

Repository CRAN

Date/Publication 2026-04-17 14:40:02 UTC

Contents

`rtgstat-package` *rtgstat: Client for 'TGStat API'*

Description

Allow function for using 'TGStat Stat API' and 'TGStat Search API', for more details see https://api.tgstat.ru/docs/ru/start/intro.html. 'TGStat' provide telegram channel analytics data.

Author(s)

Maintainer: Alexey Seleznev <selesnow@gmail.com> (ORCID)

See Also

Useful links:

- https://selesnow.github.io/rtgstat/
- Report bugs at https://github.com/selesnow/rtgstat/issues

tg_api_usage *API request statistics*

Description

API request statistics

Usage

```
tg_api_usage()
```

Value

tibble with API quote stat

tg_auth *Set API Token of 'TgStat'*

Description

Set API Token of 'TgStat'

Usage

```
tg_auth(token)
```

Arguments

token	Your API token.

Value

Use only for set token. No return value.

References

See also TGStat API Documentation of Authorization

tg_categories *Category list*

Description

List of 'TGStat' channel categories

Usage

```
tg_categories(lang = NULL)
```

Arguments

lang	Response language

Value

tibble with categories

References

See also TGStat API Documentation of metrod database/categories

tg_channel *Get chennel info*

Description

Get general information about the channel - link to the channel, name, description, avatar, number of subscribers at the moment.

Usage

```
tg_channel(channel_id = tg_get_channel_id())
```

Arguments

channel_id	hannel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')

Value

tibble with channel metadata

References

See also TGStat API Documentation of metrod channels/get

Examples

```
## Not run:
channel <- tg_channel(channel_id = "R4marketing")

## End(Not run)
```

`tg_channels_search` *Channel search*

Description

The method allows you to search for channels by keyword or get a list of channels in a category.

Usage

```
tg_channels_search(
  query = NULL,
  search_by_description = FALSE,
  country = "ru",
  language = "russian",
  category = NULL,
  limit = 100
)
```

Arguments

`query`	Search keyword
`search_by_description`	Search in channel description?
`country`	Channel geography (country). Use `tg_countries` for get countries dictionary.
`language`	Channel content language. Use `tg_languages` for get languages dictionary.
`category`	Channel category. Use `tg_categories` for get categories dictionary.
`limit`	Maximum number of channels in a response, no more than 100.

Value

tibble with channels

References

See also TGStat API Documentation of metrod channels/search

Examples

```
## Not run:
channels <- tg_channels_search(
   query    = "data",
   country  = "ru",
   category = "tech"
)

## End(Not run)
```

`tg_channel_avg_posts_reach` *Getting the average coverage of channel publications over time*

Description

Allows you to get the indicator "average coverage of publications" in dynamics by days, weeks, months.

Usage

```
tg_channel_avg_posts_reach(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  group = c("day", "week", "month")
)
```

Arguments

`channel_id`	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
`start_date`	Start date of report period
`end_date`	End date of report period
`group`	Time group: day, week, month

Details

For the `group = 'day'` grouping, the value for the "average coverage of publications" as of the end of the day will be returned. For groupings `group = 'week'` and `group = 'month'`, the value of the indicator "average coverage of publications" at the end of the last day of the period (week or month) will be returned. By default, the result will be returned for the last 10 days. However, you can specify the required period using the `start_date` and `end_date` parameters, while observing the restrictions on your tariff. Depending on the requested grouping type group - the `period` field will take one of the following formats:

- day: Y-m-d
- week: Y-W
- month: Y-m

Value

tibble with post reach dinamics

References

See also TGStat API Documentation of metrod channels/avg-posts-reach

Examples

```
## Not run:
tg_set_channel_id('R4marketing')
post_reach <- tg_channel_avg_posts_reach()

## End(Not run)
```

---

tg_channel_err *Obtaining an ERR indicator for a channel in dynamics*

Description

Allows you to get the "ERR" indicator in dynamics by day, week, month.

Usage

```
tg_channel_err(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  group = c("day", "week", "month")
)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
start_date	Start date of report period
end_date	End date of report period
group	Time group: day, week, month

Details

For the `group = 'day'` grouping, the value for the "average coverage of publications" as of the end of the day will be returned. For groupings `group = 'week'` and `group = 'month'`, the value of the indicator "average coverage of publications" at the end of the last day of the period (week or month) will be returned. By default, the result will be returned for the last 10 days. However, you can specify the required period using the `start_date` and `end_date` parameters, while observing the restrictions on your tariff. Depending on the requested grouping type group - the `period` field will take one of the following formats:

- day: Y-m-d
- week: Y-W
- month: Y-m

Value

tibble with channel ERR dinamics

References

See also TGStat API Documentation of metrod channels/err

Examples

```
## Not run:
tg_set_channel_id('R4marketing')
err <- tg_channel_err()

## End(Not run)
```

`tg_channel_forwards` *Getting a list of reposts from a channel*

Description

Allows you to get a list of reposts of publications from a channel to other channels.

Usage

```
tg_channel_forwards(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date()
)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
start_date	Date forwards from
end_date	Date forwards to

Value

tibble with forwards

References

See also TGStat API Documentation of metrod channels/forwards

Examples

```
## Not run:
forwards <- tg_channel_forwards(
    channel_id = 'R4marketing',
    start_date = '2021-01-01',
    end_date   = '2021-09-30'
)

## End(Not run)
```

---

tg_channel_mentions *Getting a list of mentions*

Description

The method allows you to get a list of mentions of a channel in other channels.

Usage

```
tg_channel_mentions(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date()
)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
start_date	Date mentioned from
end_date	Date mentioned to

Details

A publication is considered a mention if it contains a link to a channel like username, t.me/username, t.me/username/1234. In case of mentioning a channel, the `mention_type` parameter will contain the value 'channel'. If a specific publication of the channel is mentioned, then the parameter will contain the value 'post'.

Value

tibble with mention data

References

See also TGStat API Documentation of metrod channels/mentions

Examples

```
## Not run:
mentions <- tg_channel_mentions(
    channel_id = 'R4marketing',
    start_date = '2021-10-01',
    end_date = '2021-10-31'
 )

## End(Not run)
```

---

tg_channel_posts *Retrieving a list of publications*

Description

The method allows you to get channel publications according to the specified parameters. Returns channel messages sorted in reverse chronological order (most recent from the top).

Usage

```
tg_channel_posts(
  channel_id = tg_get_channel_id(),
  start_time = Sys.Date() - 15,
  end_time = Sys.Date(),
  hide_forwards = 0,
  hide_deleted = 0
)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
start_time	Date of publication from
end_time	Date of publication to
hide_forwards	Hide reposts from search results
hide_deleted	Hide deleted posts

Value

tibble with channel posts

References

See also TGStat API Documentation of metrod channels/posts

Examples

```
## Not run:
posts <- tg_channel_posts(
   channel_id = "R4marketing",
   start_time = "2021-11-01 00:00:00",
   end_time = "2021-11-30 23:59:59"
)

## End(Not run)
```

---

tg_channel_stat *Get channel stat*

Description

The method allows you to obtain basic statistics - the number of participants, the average coverage of the publication, the percentage of engagement of subscribers (ERR), the total daily coverage, the citation index (CI)

Usage

```
tg_channel_stat(channel_id = tg_get_channel_id())
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')

Value

tibble with channel stat

References

See also TGStat API Documentation of metrod channels/stat

Examples

```
## Not run:
channel_stat <- tg_channel_stat(channel_id = "R4marketing")

## End(Not run)
```

`tg_channel_subscribers` *Get channel subscribers nunmber by day*

Description

The method allows you to get the number of channel subscribers in dynamics by hours, days, weeks, months.

Usage

```
tg_channel_subscribers(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  group = c("day", "hour", "week", "month")
)
```

Arguments

`channel_id`	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
`start_date`	Start date of report period
`end_date`	End date of report period
`group`	Time group: hour, day, week, month

Details

For grouping group = day, the number of subscribers as of the end of the day will be returned.

For groupings group = week and group = month, the number of subscribers at the end of the last day of the period (week or month) will be returned.

Depending on the requested grouping type group - the period field will take one of the following formats:

- hour: Y-m-d H:00
- day: Y-m-d
- week: Y-W
- month: Y-m

Value

tibble with subscribers stat

References

See also TGStat API Documentation of metrod channels/subscribers

Examples

```
## Not run:
channel_subscribers <- tg_channel_subscribers(
    channel_id = "R4marketing",
    start_date = "2021-06-01",
    end_date = "2021-10-31",
    group = "month"
 )

## End(Not run)
```

---

tg_channel_views *Getting the number of views in dynamics*

Description

Getting the number of views in dynamics

Usage

```
tg_channel_views(
  channel_id = tg_get_channel_id(),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  group = c("day", "hour", "week", "month")
)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')
start_date	Start date of report period
end_date	End date of report period
group	Time group: hour, day, week, month

Details

The method allows you to get the total number of views per day on the channel, in dynamics by days, weeks, months.

Value

tibble with channel views

Examples

```
## Not run:
tg_auth('Your token')
tg_set_channel_id('R4marketing')

views <- tg_channel_views(
  start_date = '2021-09-01',
  end_date = '2021-09-30',
  group = "day"
)

## End(Not run)
```

---

tg_countries *List of countries*

Description

List of countries 'TGStat'

Usage

```
tg_countries(lang = NULL)
```

Arguments

lang	Response language

Value

tibble with countries

References

See also TGStat API Documentation of metrod database/countries

tg_get_channel_id *Get defaukt channel ID*

Description

Get defaukt channel ID

Usage

```
tg_get_channel_id()
```

Value

character, default session channel id

tg_get_token *Get API Token of 'TgStat'*

Description

Get API Token of 'TgStat'

Usage

```
tg_get_token()
```

Value

Api token

tg_languages *List of languages*

Description

List of available languages for 'TGStat' channels

Usage

```
tg_languages(lang = NULL)
```

Arguments

lang	Response language

Value

tibble

References

See also TGStat API Documentation of metrod database/languages

---

`tg_mentions_by_channels` *Keyword mentions by channel*

Description

A method for obtaining data on the mentions of a keyword / phrase grouped by channel. Suitable for tracking channels that often write on a given topic, mention a brand or person in Telegram publications. Returns information about the channel, the number of mentions, reach, and the date of the last mention of the keyword in the channel.

Usage

```
tg_mentions_by_channels(
  query,
  peer_type = c("all", "channel", "chat"),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  hide_forwards = 0,
  strong_search = 0,
  minus_mords = NULL,
  extended_syntax = 0
)
```

Arguments

`query`	Search query
`peer_type`	Source type (channel, chat, all)
`start_date`	Published date from (timestamp)
`end_date`	Date published to (timestamp)
`hide_forwards`	Hide reposts from search results
`strong_search`	Enable strict search (disables morphology and search by part of a word)
`minus_mords`	List of negative words (separator - space)
`extended_syntax`	Whether the request uses extended query syntax, see details

Details

Keyword / phrase search methods support extended query syntax. You must pass the extendedSyntax parameter (or extended_syntax in newer API methods) to indicate to the parser that the search query contains statements from the extended query language.

Morphology:

Regardless of the form in which you used a word in a query, by default all its morphological forms are taken into account (in any case, singular and plural). That is, by request mom will also find publications in which mom, mom, mom, mom, etc. are found. To change this behavior, you must use the = operator.

Exact occurrence of the word. Operator =

The = operator in front of a word tells the analyzer that the given word should be searched for in an exact match with the transmitted one. The query `=mom` will only find posts with the word mom. Publications containing the words mum, mum, mum, mum in the text will NOT be found.

Search by multiple words

When transferring several words separated by spaces to a search query, publications will be found in which each of these words occurs at the same time. The request mom dad will find publications in the text of which both of these words appear simultaneously in any order and case, at any distance from each other.

OR operator |

If you need to find publications in which at least one of the words occurs, you must use the OR operator `|`. Request `Mom | dad` will find publications in the text of which at least one of these words is found.

Search for a phrase. Operator ""

The query `mama washed the frame`, enclosed in double quotes, sets a strict word order, explaining to the analyzer that it needs to find the entire phrase passed. Only those publications will be found in which these three words appear side by side in the same order as specified in the request. Publications containing these words in word forms other than those submitted will also be found. To change this behavior, you must use the = operator. The `query = "mama soap frame"` will only find publications in which these three words appear side by side in the same order and in the same case as indicated in the query.

Using negative keywords. Operator -

Using the operator `-` you can specify which words should not appear in the text of the publication. The query `"mama soap" -frame` will show publications that contain the phrases mummy soap, mummy washed, ..., but do not contain the words frame, frame, etc.

Grouping words. Operator ()

Using parentheses in a search query allows you to group parts of a query and make more complex combinations using the operators described above. The query `(mom | dad | brother | sister) (soap | painted) (frame | door)` will find publications in the text of which at least one word from each word group is necessarily found. Those. publications will be found containing: mom washed the frame, dad washed the frame, sister painted the door, etc. The query `(mom | dad) (dyed) -(frame | door | hair)` will find publications, the text of which must contain at least one of the words of the first group mom, dad, it must contain a word from the second group painted, dyed, dyed, but not contains words from the last group frame, door, hair.

You can practice writing search queries in our publication search tool (do not forget to check the "Advanced language" checkbox to enable the advanced query syntax mode).

Value

list

References

See also TGStat API Documentation of metrod words/mentions-by-period

Examples

```
## Not run:
mentions_data <- tg_mentions_by_channels(
    query = 'Alexey Seleznev',
    start_date = '2021-09-01',
    end_date = '2021-09-30'
)

mentions <- mentions_data$items
channels <- mentions_data$channels

## End(Not run)
```

---

tg_mentions_by_period *Dynamics of the keyword mentions by period*

Description

A method to track the dynamics of mentions and reach of keywords or phrases. Suitable for monitoring the mention of a brand or person in Telegram publications. Returns the number of mentions and reach of a keyword for each day of the requested period.

Usage

```
tg_mentions_by_period(
  query,
  peer_type = c("all", "channel", "chat"),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  group = c("day", "week", "month"),
  hide_forwards = 0,
  strong_search = 0,
  minus_mords = NULL,
  extended_syntax = 0
)
```

Arguments

query	Search query
peer_type	Source type (channel, chat, all)
start_date	Published date from (timestamp)
end_date	Date published to (timestamp)
group	Time group: day, week, month
hide_forwards	Hide reposts from search results
strong_search	Enable strict search (disables morphology and search by part of a word)
minus_mords	List of negative words (separator - space)
extended_syntax	Whether the request uses extended query syntax, see details

Details

Keyword / phrase search methods support extended query syntax. You must pass the extendedSyntax parameter (or extended_syntax in newer API methods) to indicate to the parser that the search query contains statements from the extended query language.

Morphology:

Regardless of the form in which you used a word in a query, by default all its morphological forms are taken into account (in any case, singular and plural). That is, by request mom will also find publications in which mom, mom, mom, mom, etc. are found. To change this behavior, you must use the = operator.

Exact occurrence of the word. Operator =

The = operator in front of a word tells the analyzer that the given word should be searched for in an exact match with the transmitted one. The query `=mom` will only find posts with the word mom. Publications containing the words mum, mum, mum, mum in the text will NOT be found.

Search by multiple words

When transferring several words separated by spaces to a search query, publications will be found in which each of these words occurs at the same time. The request mom dad will find publications in the text of which both of these words appear simultaneously in any order and case, at any distance from each other.

OR operator |

If you need to find publications in which at least one of the words occurs, you must use the OR operator `|`. Request `Mom | dad` will find publications in the text of which at least one of these words is found.

Search for a phrase. Operator ""

The query `mama washed the frame`, enclosed in double quotes, sets a strict word order, explaining to the analyzer that it needs to find the entire phrase passed. Only those publications will be found in which these three words appear side by side in the same order as specified in the request. Publications containing these words in word forms other than those submitted will also be found. To change this behavior, you must use the = operator. The `query = "mama soap frame"` will only find publications in which these three words appear side by side in the same order and in the same case as indicated in the query.

Using negative keywords. Operator -

Using the operator - you can specify which words should not appear in the text of the publication. The query "mama soap" -frame will show publications that contain the phrases mummy soap, mummy washed, ..., but do not contain the words frame, frame, etc.

Grouping words. Operator ()

Using parentheses in a search query allows you to group parts of a query and make more complex combinations using the operators described above. The query (mom | dad | brother | sister) (soap | painted) (frame | door) will find publications in the text of which at least one word from each word group is necessarily found. Those. publications will be found containing: mom washed the frame, dad washed the frame, sister painted the door, etc. The query (mom | dad) (dyed) - (frame | door | hair) will find publications, the text of which must contain at least one of the words of the first group mom, dad, it must contain a word from the second group painted, dyed, dyed, but not contains words from the last group frame, door, hair.

You can practice writing search queries in our publication search tool (do not forget to check the "Advanced language" checkbox to enable the advanced query syntax mode).

Value

tibble with mention statistics

References

See also TGStat API Documentation of metrod words/mentions-by-period

Examples

```
## Not run:
mentions <- tg_mentions_by_period(
    query = 'Alexey Seleznev',
    start_date = '2021-09-01',
    end_date = '2021-09-30'
)

## End(Not run)
```

---

tg_options *Get rtgstat option values*

Description

Get rtgstat option values

Usage

```
tg_options()
```

Value

no return data, using for side effect

`tg_post` *Retrieving publication data*

Description

Retrieving publication data

Usage

```
tg_post(post_id)
```

Arguments

`post_id`	Post ID (t.me/username/123, t.me/c/1256804429/1230 or post ID in TGStat)

Details

Get information and publications in Telegram - number of views, publication date, content, ...

Value

tibble with post data

References

See also TGStat API Documentation of metrod posts/get

Examples

```
## Not run:
post <- tg_post(
    post_id = 'https://t.me/R4marketing/887'
)

## End(Not run)
```

`tg_posts_search` *Search publications*

Description

Method for searching publications by keyword. Returns publications, sorted in reverse chronological order (most recent from the top), in which the search text was found.

Usage

```
tg_posts_search(
  query,
  peer_type = c("all", "channel", "chat"),
  start_date = Sys.Date() - 15,
  end_date = Sys.Date(),
  hide_forwards = 0,
  hide_deleted = 0,
  strong_search = 0,
  minus_mords = NULL,
  extended_syntax = 0
)
```

Arguments

`query`	Search query
`peer_type`	Source type (channel, chat, all)
`start_date`	Published date from (timestamp)
`end_date`	Date published to (timestamp)
`hide_forwards`	Hide reposts from search results
`hide_deleted`	Hide deleted posts
`strong_search`	Enable strict search (disables morphology and search by part of a word)
`minus_mords`	List of negative words (separator - space)
`extended_syntax`	Whether the request uses extended query syntax, see details

Details

Keyword / phrase search methods support extended query syntax. You must pass the extendedSyntax parameter (or extended_syntax in newer API methods) to indicate to the parser that the search query contains statements from the extended query language.

Morphology:

Regardless of the form in which you used a word in a query, by default all its morphological forms are taken into account (in any case, singular and plural). That is, by request mom will also find

publications in which mom, mom, mom, mom, etc. are found. To change this behavior, you must use the = operator.

Exact occurrence of the word. Operator =

The = operator in front of a word tells the analyzer that the given word should be searched for in an exact match with the transmitted one. The query `=mom` will only find posts with the word mom. Publications containing the words mum, mum, mum, mum in the text will NOT be found.

Search by multiple words

When transferring several words separated by spaces to a search query, publications will be found in which each of these words occurs at the same time. The request mom dad will find publications in the text of which both of these words appear simultaneously in any order and case, at any distance from each other.

OR operator |

If you need to find publications in which at least one of the words occurs, you must use the OR operator `|`. Request `Mom | dad` will find publications in the text of which at least one of these words is found.

Search for a phrase. Operator ""

The query `mama washed the frame`, enclosed in double quotes, sets a strict word order, explaining to the analyzer that it needs to find the entire phrase passed. Only those publications will be found in which these three words appear side by side in the same order as specified in the request. Publications containing these words in word forms other than those submitted will also be found. To change this behavior, you must use the = operator. The `query = "mama soap frame"` will only find publications in which these three words appear side by side in the same order and in the same case as indicated in the query.

Using negative keywords. Operator -

Using the operator `-` you can specify which words should not appear in the text of the publication. The query `"mama soap" -frame` will show publications that contain the phrases mummy soap, mummy washed, ..., but do not contain the words frame, frame, etc.

Grouping words. Operator ()

Using parentheses in a search query allows you to group parts of a query and make more complex combinations using the operators described above. The query `(mom | dad | brother | sister) (soap | painted) (frame | door)` will find publications in the text of which at least one word from each word group is necessarily found. Those. publications will be found containing: mom washed the frame, dad washed the frame, sister painted the door, etc. The query `(mom | dad) (dyed) - (frame | door | hair)` will find publications, the text of which must contain at least one of the words of the first group mom, dad, it must contain a word from the second group painted, dyed, dyed, but not contains words from the last group frame, door, hair.

You can practice writing search queries in our publication search tool (do not forget to check the "Advanced language" checkbox to enable the advanced query syntax mode).

Value

list with two tibbles

Examples

```
## Not run:
post_search <- tg_posts_search(
    query = 'rtgstat package',
    peer_type = 'channel',
    start_date = '2021-11-01',
    end_date = '2021-11-31'
)

search_result <- post_search$items
channels <- post_search$channels


## End(Not run)
```

`tg_post_stat` *Getting publication statistics*

Description

Getting publication statistics

Usage

```
tg_post_stat(post_id, group = c("day", "hour"))
```

Arguments

`post_id`	Post ID (t.me/username/123, t.me/c/1256804429/1230 or post ID in TGStat)
`group`	Grouping results (hour, day)

Details

Obtaining publication statistics - the number of views at the moment, the list of reposts and mentions, the dynamics of the growth of views by hours / days.

Value

list with tibbles

References

See also TGStat API Documentation of metrod posts/stat

Examples

```
## Not run:
post_stat <- tg_post_stat(
    post_id = 'https://t.me/R4marketing/887',
    group = 'day'
)

views <- post_stat$views
forwards <- post_stat$forwards
mentions <- post_stat$mentions

## End(Not run)
```

---

tg_set_api_quote_alert_rate *Set API limit alert rate*

Description

Set API limit alert rate

Usage

```
tg_set_api_quote_alert_rate(api_quote_alert_rate)
```

Arguments

api_quote_alert_rate	Max reach of API limit to alert

Value

using for side effect, no return value

---

tg_set_channel_id *Set session default channel id*

Description

Set session default channel id

Usage

```
tg_set_channel_id(channel_id)
```

Arguments

channel_id	Channel ID (@username, t.me/username, t.me/joinchat/AAAAABbbbbcccc ... or channel ID in 'TGStat')

Value

Using for side effect, no return data

Examples

```
## Not run:
tg_set_channel_id('R4marketing')
stat <- tg_channel_stat()

## End(Not run)
```

---

tg_set_check_api_quote

Disable or enable API limit alert

Description

Disable or enable API limit alert

Usage

```
tg_set_check_api_quote(check_api_quote)
```

Arguments

check_api_quote	Logical, disable (or enable) API limit alerts

Value

using for side effect, no return value

tg_set_interval *Set time interval in seconds between tries of HTTP queries*

Description

Set time interval in seconds between tries of HTTP queries

Usage

```
tg_set_interval(interval)
```

Arguments

interval	delay between retries

Value

using for side effect, no return value

tg_set_max_tries *Set max tries of HTTP queries*

Description

Set max tries of HTTP queries

Usage

```
tg_set_max_tries(max_tries)
```

Arguments

max_tries	integer, maximum number of attempts

Value

using for side effect, no return value

Index