Package 'slackr'

February 26, 2023

Type Package

Title Send Messages, Images, R Objects and Files to 'Slack' Channels/Users

Version 3.3.1

Author Bob Rudis [aut, cre], Jay Jacobs [ctb], David Severski [ctb], Quinn Weber [ctb], Konrad Karczewski [ctb], Shinya Uryu [ctb], Gregory Jefferis [ctb], Ed Niles [ctb], Rick Saporta [ctb], Jonathan Sidi [aut, ctb], Matt Kaye [aut, ctb], Xinye Li [ctb], Andrie de Vries [aut, ctb]

Maintainer Matt Kaye <mrkaye97@gmail.com>

Description 'Slack' <https://slack.com/> provides a service for teams to collaborate by sharing messages, images, links, files and more. Functions are provided that make it possible to interact with the 'Slack' platform 'API'. When you need to share information or data from R, rather than resort to copy/ paste in e-mails or other services like 'Skype' <https://www.skype.com/en/>, you can use this package to send well-formatted output from multiple R objects and expressions to all teammates at the same time with little effort. You can also send images from the current graphics device, R objects, and upload files.

License MIT + file LICENSE

URL https://github.com/mrkaye97/slackr,
https://mrkaye97.github.io/slackr/

BugReports https://github.com/mrkaye97/slackr/issues

Depends R (>= 3.3.0)

Imports cachem (>= 1.0.4), dplyr, grDevices, httr (>= 1.4.2), jsonlite, magrittr, memoise (>= 2.0.0), rlang, tibble, utils, withr

Suggests covr, ggplot2, knitr, rmarkdown, svglite, testthat (>= 3.0.0), texPreview

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-02-26 01:12:06 UTC

R topics documented:

`auth_test` *Checks authentication & identity against the Slack API.*

Description

Checks authentication & identity against the Slack API.

Usage

```
auth_test(token = Sys.getenv("SLACK_TOKEN"))
```

Arguments

token	Authentication token bearing required scopes.

References

https://api.slack.com/methods/auth.test

Examples

```
if (Sys.getenv("SLACK_TOKEN") != "") {
  auth_test()
}
```

---

call_slack_api — *A wrapper function to call the Slack API with authentication and pagination.*

Description

A wrapper function to call the Slack API with authentication and pagination.

Usage

```
call_slack_api(
  path,
  ...,
  body = NULL,
  .method = c("GET", "POST"),
  token,
  .verbose = Sys.getenv("SLACKR_VERBOSE", "FALSE"),
  .next_cursor = ""
)
```

Arguments

path	The API definition path, e.g. `/api/auth.test`.
...	These arguments must be named and will be added to the API query string.
body	If `.method` = `POST` the `body` gets passed to the API body.
.method	Either "GET" or "POST".
token	Authentication token bearing required scopes.
.verbose	If TRUE, prints `httr` verbose messages. Useful for debugging.
.next_cursor	The value of the next cursor, when using pagination.

Value

The API response (a named list)

`convert_response_to_tibble` *Convert Slack API json response to tibble.*

Description

Convert Slack API json response to tibble.

Usage

```
convert_response_to_tibble(x, element)
```

Arguments

`x`	The Slack API response object, returned from call_slack_api.
`element`	The name of the list element to extract.

Value

A tibble

`create_config_file` *Create the config file used in* `slackr_setup()`

Description

Create the config file used in `slackr_setup()`

Usage

```
create_config_file(
  filename = "~/.slackr",
  token = Sys.getenv("SLACK_TOKEN"),
  incoming_webhook_url = Sys.getenv("SLACK_INCOMING_WEBHOOK_URL"),
  icon_emoji = Sys.getenv("SLACK_ICON_EMOJI"),
  username = Sys.getenv("SLACK_USERNAME"),
  channel = Sys.getenv("SLACK_CHANNEL")
)
```

Arguments

`filename` the name of the config file to save. We recommend using a hidden file (starting with '.')

`token` Authentication token bearing required scopes.

`incoming_webhook_url` the incoming webhook URL (Default: whatever is set as an env var).

`icon_emoji` the icon emoji to use as the default.

`username` the username to send messages from (will default to "slackr" if no username is set).

`channel` Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See the chat.postMessage endpoint documentation for details.

Value

TRUE if successful (invisibly)

See Also

`slackr_setup()`

Examples

```
## Not run:
# using `create_config_file()` after `slackr_setup()`
create_config_file()

# using `create_config_file()` before `slackr_setup()`
create_config_file(
  token = "xox-",
  incoming_webhook_url = "https://hooks-",
  channel = "#general",
  username = "slackr",
  icon_emoji = "tada"
)

slackr_setup()

## End(Not run)
```

`ggslackr` *Post a ggplot to a Slack channel*

Description

Unlike the `slackr_dev()` function, this one takes a `ggplot` object, eliminating the need to have a graphics device (think use in scripts).

Usage

```
ggslackr(
  plot = ggplot2::last_plot(),
  channels = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  file = "ggplot.png",
  initial_comment = NULL,
  thread_ts = NULL,
  title = NULL,
  ...
)
```

Arguments

`plot` ggplot object to save, defaults to last plot displayed.

`channels` Comma-separated list of channel names or IDs where the file will be shared.

`token` Authentication token bearing required scopes.

`file` Prefix for filenames (defaults to `ggplot`).

`initial_comment` The message text introducing the file in specified channels.

`thread_ts` Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

`title` Title of file.

`...` other arguments passed to ggsave

Value

`httr` response object (invisibly)

Examples

```
## Not run:
slackr_setup()
ggslackr(qplot(mpg, wt, data = mtcars))

## End(Not run)
```

`register_onexit` *Append slackr_msg as on.exit to functions.*

Description

Appends to the body of a function an on.exit call to run at the end of the call.

Usage

```
register_onexit(
  f,
  ...,
  header_msg = NULL,
  use_device = FALSE,
  env = parent.frame(2),
  channel = Sys.getenv("SLACK_CHANNEL"),
  username = Sys.getenv("SLACK_USERNAME"),
  icon_emoji = Sys.getenv("SLACK_ICON_EMOJI"),
  token = Sys.getenv("SLACK_TOKEN")
)
```

Arguments

`f`	function or character.
`...`	expressions to be sent to Slack.
`header_msg`	boolean, message to append to start of Slack output, Default: NULL.
`use_device`	boolean, passes current image in the graphics device to Slack as part of f, Default: FALSE.
`env`	environment to assign appended function to with relation to the function environment, Default: parent.frame(2) (global environment).
`channel`	Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See the chat.postMessage endpoint documentation for details.
`username`	what user should the bot be named as (chr).
`icon_emoji`	what emoji to use (chr) `""` will mean use the default.
`token`	Authentication token bearing required scopes.

Details

If a character is passed to f then it will evaluate internally to a function.

Value

function

Author(s)

Jonathan Sidi (aut)

See Also

`slackr_msg()`

Examples

```
## Not run:
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl", "Trt"))
weight <- c(ctl, trt)

# pass a message to Slack channel 'general'
register_onexit(lm, "bazinga!", channel = "#general")

lm.D9 <- slack_lm(weight ~ group)

# test that output keeps inheritance
summary(lm.D9)

# pass a message to Slack channel 'general' with a header message to begin output
register_onexit(
  lm,
  "bazinga!",
  channel = "#general",
  header_msg = "This is a message to begin"
)

lm.D9 <- slack_lm(weight ~ group)

# onexit with an expression that calls lm.plot
register_onexit(
  lm,
  {
    par(mfrow = c(2, 2), oma = c(0, 0, 2, 0))
    plot(z)
  },
  channel = "#general",
  header_msg = "This is a plot just for this output",
  use_device = TRUE
)

lm.D9 <- slack_lm(weight ~ group)

# clean up slack channel from examples
slackr_delete(count = 6, channel = "#general")

## End(Not run)
```

slackr *Output R expressions to a Slack channel/user*

Description

Takes an expr, evaluates it and sends the output to a Slack chat destination. Useful for logging, messaging on long compute tasks or general information sharing.

Usage

```
slackr(
  ...,
  channel = Sys.getenv("SLACK_CHANNEL"),
  username = Sys.getenv("SLACK_USERNAME"),
  icon_emoji = Sys.getenv("SLACK_ICON_EMOJI"),
  token = Sys.getenv("SLACK_TOKEN"),
  thread_ts = NULL,
  reply_broadcast = FALSE
)
```

Arguments

`...`	expressions to be sent to Slack.
`channel`	Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See the chat.postMessage endpoint documentation for details.
`username`	what user should the bot be named as (chr).
`icon_emoji`	what emoji to use (chr) `""` will mean use the default.
`token`	Authentication token bearing required scopes.
`thread_ts`	Provide another message's ts value to make this message a reply. Avoid using a reply's ts value; use its parent instead.
`reply_broadcast`	Used in conjunction with thread_ts and indicates whether reply should be made visible to everyone in the channel or conversation. Defaults to FALSE.

Details

By default, everything but `expr` will be looked for in a `"SLACK_"` environment variable. You can override or just specify these values directly instead, but it's probably better to call `slackr_setup()` first.

Value

the response (invisibly)

Note

You need a `https://www.slack.com` account and will also need to set up an API token `https://api.slack.com/`

See Also

`slackr_setup()`, `slackr_bot()`, `slackr_dev()`, `slackr_save()`, `slackr_upload()`

Examples

```
## Not run:
slackr_setup()
slackr("iris info", head(iris), str(iris))

## End(Not run)
```

---

slackr_bot *Send result of R expressions to a Slack channel via webhook API*

Description

Takes an `expr`, evaluates it and sends the output to a Slack chat destination via the webhook API. Useful for logging, messaging on long compute tasks or general information sharing.

Usage

```
slackr_bot(
  ...,
  incoming_webhook_url = Sys.getenv("SLACK_INCOMING_WEBHOOK_URL")
)
```

Arguments

`...` expressions to be sent to Slack.

`incoming_webhook_url`
which `slack.com` API endpoint URL to use (see section **Webhook URLs** for details).

Details

By default, everything but `expr` will be looked for in a "`SLACK_`" environment variable. You can override or just specify these values directly instead, but it's probably better to call `slackr_setup()` first.

Webhook URLs

Webhook URLs look like:

- `https://hooks.slack.com/services/XXXXX/XXXXX/XXXXX`

OLD STYLE WEBHOOKS ARE NO LONGER SUPPORTED

Note

You need a `https://www.slack.com` account and will also need to setup an incoming webhook: `https://api.slack.com/`. Old style webhooks are no longer supported.

See Also

slackr_setup(), slackr(), slackr_dev(), slackr_save(), slackr_upload()

Examples

```
## Not run:
slackr_setup()
slackr_bot("iris info", head(iris), str(iris))

# or directly
slackr_bot(
  "Test message",
  incoming_webhook_url = "https://hooks.slack.com/services/XXXXX/XXXXX/XXXXX"
)

## End(Not run)
```

---

slackr_channels *Get a data frame of Slack channels*

Description

Get a data frame of Slack channels

Usage

```
slackr_channels(token = Sys.getenv("SLACK_TOKEN"), exclude_archived = TRUE)
```

Arguments

token	Authentication token bearing required scopes.
exclude_archived	A boolean indicating whether or not to exclude archived channels.

Value

A data.frame of channels

slackr_chtrans *Translate vector of channel names to channel IDs for API*

Description

Given a vector of one or more channel names, retrieve list of active channels and try to replace channels that begin with "#" or "@" with the channel ID for that channel.

Usage

```
slackr_chtrans(channels, token = Sys.getenv("SLACK_TOKEN"))
```

Arguments

channels	Comma-separated list of channel names or IDs where the file will be shared.
token	Authentication token bearing required scopes.

Value

character vector - original channel list with # or @ channels replaced with ID's.

Author(s)

Quinn Weber (ctb), Bob Rudis (aut)

slackr_csv *Send a data frame to Slack as a CSV*

Description

slackr_csv simplifies the process of sending a data frame to Slack as a CSV file. It is highly recommended to leave the filename argument as the default (tempfile), as changing it will persist a csv file in your working directory.

Usage

```
slackr_csv(
  data,
  filename = tempfile(fileext = ".csv"),
  title = NULL,
  initial_comment = NULL,
  channels = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  thread_ts = NULL,
  ...
)
```

Arguments

`data` the data frame or tibble to upload.

`filename` the file to save to. Defaults to a tempfile. Using the default is *highly* advised, as using a non-tempfile will write a file that persists on the disk (either in the working directory, or at the location specified).

`title` Title of file.

`initial_comment` The message text introducing the file in specified channels.

`channels` Comma-separated list of channel names or IDs where the file will be shared.

`token` Authentication token bearing required scopes.

`thread_ts` Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

`...` additional arguments to be passed to `write.csv()`.

Value

`httr` response object from `POST` call (invisibly)

`httr` response object from `POST` call (invisibly)

Author(s)

Matt Kaye (aut)

See Also

`slackr_upload()`

`slackr_delete` *slackr_delete*

Description

Deletes the specified number of messages from the channel

Usage

```
slackr_delete(
  count,
  channel = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN")
)
```

Arguments

`count` The number of messages to delete.

`channel` Channel, private group, or IM channel to delete messages from. Can be an encoded ID, or a name. See the chat.postMessage endpoint documentation for details.

`token` Authentication token bearing required scopes.

slackr_dev *Send the graphics contents of the current device to a Slack channel*

Description

`slackr_dev` sends the graphics contents of the current device to the specified Slack channel.

Usage

```
slackr_dev(
  channels = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  file = "plot",
  initial_comment = NULL,
  title = NULL,
  thread_ts = NULL
)
```

Arguments

`channels` Comma-separated list of channel names or IDs where the file will be shared.

`token` Authentication token bearing required scopes.

`file` prefix for filenames (defaults to `plot`).

`initial_comment` The message text introducing the file in specified channels.

`title` Title of file.

`thread_ts` Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

Value

`httr` response object from `POST` call

Author(s)

Konrad Karczewski (ctb), Bob Rudis (aut)

References

https://github.com/mrkaye97/slackr/pull/12/files

See Also

slackr_setup(), slackr_save(), slackr_upload()

Examples

```
## Not run:
slackr_setup()

# base
library(maps)
map("usa")
slackr_dev("#results", file = "map")

# base
barplot(VADeaths)
slackr_dev("@jayjacobs")

## End(Not run)
```

slackr_history *Reads history of a channel.*

Description

Returns a dataframe of post history in a channel.

Usage

```
slackr_history(
  message_count = 100,
  channel = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  posted_to_time = as.numeric(Sys.time()),
  duration = NULL,
  posted_from_time = 0,
  paginate = FALSE,
  inclusive = TRUE
)
```

Arguments

message_count The number of messages to retrieve (only when paginate = FALSE). Corresponds to limit in the conversations.history docs. Note: If using pagination, setting a value of (e.g.) 1 will result in paginating through the channel's history one message at a time. Slack recommends using a value of 200. (Default: 100) `channel` Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name.

`token` Authentication token bearing required scopes.

`posted_to_time` Timestamp of the last post to consider. Corresponds to `latest` in the conversations.history docs. Default: current time.

`duration` Number of hours of history to retrieve. If neither `duration` nor `posted_from_time` is specified, there is no time limit on the retrieved history. (default: `NULL`)

`posted_from_time`
Timestamp of the first post time to consider. If both posted_to_time and duration is specifed, they take precedence. Corresponds to `oldest` in the conversations.history docs. Default: 0

`paginate` If TRUE, uses the Slack API pagination mechanism, and will retrieve all history inside the timeframe. Otherwise, makes a single call to the API and retrieves a maximum of `message_count` messages.

`inclusive` Include messages with oldest or latest timestamps in results. Ignored unless either timestamp is specified.

Value

A `tibble` with message metadata

Scopes

You need one or more of these scopes enabled in your slack app: * channels:history * groups:history * im:history * mpim:history

References

`https://api.slack.com/methods/conversations.history`

`slackr_ims` *Get a data frame of Slack IM ids*

Description

Get a data frame of Slack IM ids

Usage

```
slackr_ims(token = Sys.getenv("SLACK_TOKEN"))
```

Arguments

`token` Authentication token bearing required scopes.

Value

`data.frame` of im ids and user names

Author(s)

Quinn Weber (aut), Bob Rudis (ctb)

References

https://github.com/mrkaye97/slackr/pull/13

slackr_msg *Sends a message to a slack channel.*

Description

Sends a message to a slack channel.

Usage

```
slackr_msg(
  txt = "",
  channel = Sys.getenv("SLACK_CHANNEL"),
  username = Sys.getenv("SLACK_USERNAME"),
  icon_emoji = Sys.getenv("SLACK_ICON_EMOJI"),
  token = Sys.getenv("SLACK_TOKEN"),
  thread_ts = NULL,
  reply_broadcast = FALSE,
  ...
)
```

Arguments

`txt` text message to send to Slack. If a character vector of length > 1 is passed in, they will be combined and separated by newlines.

`channel` Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See the chat.postMessage endpoint documentation for details.

`username` what user should the bot be named as (chr).

`icon_emoji` what emoji to use (chr) `""` will mean use the default.

`token` Authentication token bearing required scopes.

`thread_ts` Provide another message's ts value to make this message a reply. Avoid using a reply's ts value; use its parent instead.

`reply_broadcast` Used in conjunction with thread_ts and indicates whether reply should be made visible to everyone in the channel or conversation. Defaults to FALSE.

`...` other arguments passed to the Slack API `chat.postMessage` call

Value

the response (invisibly)

Note

You need a https://www.slack.com account and will also need to setup an API token https://api.slack.com/ Also, you can pass in add_user=TRUE as part of the ... parameters and the Slack API will post the message as your logged-in user account (this will override anything set in username)

See Also

slackr_setup(), slackr_bot(), slackr_dev(), slackr_save(), slackr_upload()

Examples

```
## Not run:
slackr_setup()
slackr_msg("Hi")

## End(Not run)
```

---

slackr_save *Save R objects to an RData file on Slack*

Description

slackr_save enables you upload R objects (as an R data file) to Slack and (optionally) post them to one or more channels (if channels is not empty).

Usage

```
slackr_save(
  ...,
  channels = Sys.getenv("SLACK_CHANNEL"),
  file = "slackr",
  token = Sys.getenv("SLACK_TOKEN"),
  initial_comment = NULL,
  title = NULL,
  thread_ts = NULL
)
```

Arguments

`...` objects to store in the R data file.

`channels` Comma-separated list of channel names or IDs where the file will be shared.

`file` filename (without extension) to use.

`token` Authentication token bearing required scopes.

`initial_comment` The message text introducing the file in specified channels.

`title` Title of file.

`thread_ts` Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

Value

`httr` response object from `POST` call

See Also

`slackr_setup()`, `slackr_dev()`, `slackr_upload()`

Examples

```
## Not run:
slackr_setup()
slackr_save(mtcars, channels = "#slackr", file = "mtcars")

## End(Not run)
```

`slackr_setup` *Setup environment variables for Slack API access*

Description

Initialize all the environment variables `slackr()` will need to use to work properly.

Usage

```
slackr_setup(
  channel = "#general",
  username = "slackr",
  icon_emoji = "",
  incoming_webhook_url = "",
  token = "",
  config_file = "~/.slackr",
  echo = FALSE,
  cache_dir = ""
)
```

Arguments

channel default channel to send the output to (chr) defaults to #general.

username the username output will appear from (chr) defaults to slackr.

icon_emoji which emoji picture to use (chr) defaults to none (can be left blank in config file as well).

incoming_webhook_url the Slack URL prefix to use (chr) defaults to none.

token Authentication token bearing required scopes.

config_file a configuration file (DCF) - see read.dcf - format with the config values.

echo display the configuration variables (bool) initially FALSE.

cache_dir the location for an on-disk cache. defaults to an in-memory cache if no location is specified.

Details

By default, slackr() (and other functions) will use the #general room and a username of slackr() with no emoji.

If a valid file is found at the locaiton pointed to by config_file, the values there will be used. The fields should be specified as such in the file:

```
token: SLACK_TOKEN
channel: #general
username: slackr
incoming_webhook_url: https://hooks.slack.com/services/XXXXX/XXXXX/XXXXX
```

Value

"Successfully connected to Slack"

Note

You need a Slack account and all your API URLs & tokens setup to use this package.

See Also

slackr(), slackr_dev(), slackr_save(), slackr_upload()

Examples

```
## Not run:
# reads from default file (i.e. ~/.slackr)
slackr_setup()

# reads from alternate config
slackr_setup(config_file = "/path/to/my/slackrconfig")

# the hard way
```

```
slackr_setup(
  channel = "#code",
  incoming_webhook_url = "https://hooks.slack.com/services/XXXXX/XXXXX/XXXXX"
)

## End(Not run)
```

slackr_teardown *Unset env vars created by* slackr_setup()

Description

Unset env vars created by `slackr_setup()`

Usage

```
slackr_teardown()
```

Value

TRUE if successful (invisibly)

See Also

`slackr_setup()`

Examples

```
## Not run:
slackr_teardown()

## End(Not run)
```

slackr_tex *Post a tex output to a Slack channel*

Description

Unlike the `slackr_dev()` function, this one takes a `tex` object, eliminating the need write to pdf and convert to png to pass to slack.

Usage

```
slackr_tex(
  obj,
  channels = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  ext = "png",
  path = NULL,
  title = NULL,
  initial_comment = NULL,
  thread_ts = NULL,
  ...
)
```

Arguments

`obj` character object containing tex to compile.

`channels` Comma-separated list of channel names or IDs where the file will be shared.

`token` Authentication token bearing required scopes.

`ext` character, type of format to return, can be tex, pdf, or any image device, Default: 'png'.

`path` character, path to save tex_preview outputs, if NULL then tempdir is used, Default: NULL.

`title` Title of file.

`initial_comment` The message text introducing the file in specified channels.

`thread_ts` Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

`...` other arguments passed to `texPreview::tex_preview()`, see Details

Details

Please make sure `texPreview` package is installed before running this function. For TeX setup refer to the Setup notes on `LaTeX`.

Value

`httr` response object (invisibly)

Note

You need to setup a full API token (i.e. not a webhook & not OAuth) for this to work Also, you can pass in `add_user=TRUE` as part of the `...` parameters and the Slack API will post the message as your logged-in user account (this will override anything set in `username`)

Author(s)

Jonathan Sidi (aut)

See Also

texPreview::tex_preview()

---

slackr_upload *Send a file to Slack*

Description

slackr_upload enables you upload files to Slack and (optionally) post them to one or more channels (if channels is not empty).

Usage

```
slackr_upload(
  filename,
  title = NULL,
  initial_comment = NULL,
  channels = Sys.getenv("SLACK_CHANNEL"),
  token = Sys.getenv("SLACK_TOKEN"),
  thread_ts = NULL
)
```

Arguments

filename	path to file.
title	Title of file.
initial_comment	The message text introducing the file in specified channels.
channels	Comma-separated list of channel names or IDs where the file will be shared.
token	Authentication token bearing required scopes.
thread_ts	Provide another message's ts value to upload this file as a reply. Never use a reply's ts value; use its parent instead.

Value

httr response object from POST call (invisibly)

httr response object from POST call (invisibly)

Author(s)

Quinn Weber (ctb), Bob Rudis (aut)

References

https://github.com/mrkaye97/slackr/pull/15/files

See Also

`slackr_setup()`, `slackr_dev()`, `slackr_save()`

`slackr_users` *Get a data frame of Slack users*

Description

Get a data frame of Slack users

Usage

```
slackr_users(token = Sys.getenv("SLACK_TOKEN"))
```

Arguments

`token`	Authentication token bearing required scopes.

Value

`data.frame` of users

`warn_for_args` *Check for token-parameter mismatches*

Description

Check for token-parameter mismatches

Usage

```
warn_for_args(token, ...)
```

Arguments

`token`	Authentication token bearing required scopes.
`...`	Additiional arguments passed to the function called.

Value

No return value. Called for side effects

with_pagination *Calls the slack API with pagination using cursors.*

Description

This loops over fun, extracts the next_cursor from the API response, and injects this into the next loop. At the completion of each loop, the function convert_response_to_tibble() is run with extract as and argument. The results are combined with dplyr::bind_rows()

Usage

```
with_pagination(fun, extract)
```

Arguments

fun	A function that calls the slack API.
extract	The name of the element to extract from the API response.

Value

A tibble

See Also

call_slack_api

Index