

# Package ‘slackr’

March 4, 2021

**Type** Package

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

**Version** 2.2.0

**Date** 2021-03-03

**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.

**URL** <https://github.com/mrkaye97/slackr>,  
<https://mrkaye97.github.io/slackr/>

**BugReports** <https://github.com/mrkaye97/slackr/issues>

**License** MIT + file LICENSE

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

**Encoding** UTF-8

**Depends** R (>= 3.3.0)

**Imports** httr (>= 1.4.2), jsonlite, ggplot2, utils, methods, dplyr,  
graphics, grDevices, xtable (>= 1.8.4), tibble, magrittr,  
memoise (>= 2.0.0), cachem (>= 1.0.4)

**RoxygenNote** 7.1.1

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-03-04 05:30:02 UTC

## R topics documented:

auth_test . . . . .	3
call_slack_api . . . . .	3
convert_response_to_tibble . . . . .	4
dev.slackr . . . . .	5
dev_slackr . . . . .	5
ggslackr . . . . .	6
register_onexit . . . . .	7
save.slackr . . . . .	9
save_slackr . . . . .	9
slackr . . . . .	10
slackrBot . . . . .	11
slackrChannels . . . . .	12
slackrChtrans . . . . .	13
slackrIms . . . . .	13
slackrMsg . . . . .	14
slackrSetup . . . . .	14
slackrUpload . . . . .	15
slackrUsers . . . . .	15
slackr_bot . . . . .	16
slackr_census_fun . . . . .	17
slackr_channels . . . . .	18
slackr_chtrans . . . . .	18
slackr_delete . . . . .	19
slackr_dev . . . . .	19
slackr_history . . . . .	20
slackr_ims . . . . .	21
slackr_msg . . . . .	22
slackr_setup . . . . .	23
slackr_upload . . . . .	24
slackr_users . . . . .	25
textSlackr . . . . .	26
text_slackr . . . . .	26
tex_slackr . . . . .	28
with_pagination . . . . .	29

**Index**

**30**

---

auth_test	<i>Checks authentication &amp; identity against the Slack API.</i>
-----------	--

---

**Description**

Checks authentication & identity against the Slack API.

**Usage**

```
auth_test(bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"))
```

**Arguments**

bot\_user\_oauth\_token  
The Slack bot OAuth token character vector

**References**

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

**Examples**

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

---

call_slack_api	<i>A wrapper function to call the Slack API with authentication and pagination.</i>
----------------	---

---

**Description**

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

**Usage**

```
call_slack_api(  
  path,  
  ...,  
  body = NULL,  
  .method = c("GET", "POST"),  
  bot_user_oauth_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"
bot_user_oauth_token	The Slack bot OAuth token character vector
.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 <a href="#">call_slack_api</a>
element	The name of the list element to extract

**Value**

A tibble

---

dev.slackr	<i>Superseded by slackr_dev</i>
------------	---------------------------------

---

**Description**

deprecated

**Usage**

```
dev.slackr(...)
```

**Arguments**

... arguments to pass to slackr\_dev

**See Also**

[slackr\\_dev](#)

Other Deprecated functions: [dev.slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrImgs\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

dev_slackr	<i>Superseded by slackr_dev</i>
------------	---------------------------------

---

**Description**

deprecated

**Usage**

```
dev_slackr(...)
```

**Arguments**

... arguments to pass to slackr\_dev

**See Also**

[slackr\\_dev](#)

Other Deprecated functions: [dev.slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrImgs\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

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 = last_plot(),
  channels = Sys.getenv("SLACK_CHANNEL"),
  scale = 1,
  width = par("din")[1],
  height = par("din")[2],
  units = c("in", "cm", "mm"),
  dpi = 300,
  limitsize = TRUE,
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),
  file = "ggplot",
  ...
)
```

**Arguments**

<code>plot</code>	ggplot object to save, defaults to last plot displayed
<code>channels</code>	list of channels to post image to
<code>scale</code>	scaling factor
<code>width</code>	width (defaults to the width of current plotting window)
<code>height</code>	height (defaults to the height of current plotting window)
<code>units</code>	units for width and height when either one is explicitly specified (in, cm, or mm)
<code>dpi</code>	dpi to use for raster graphics
<code>limitsize</code>	when TRUE (the default), ggsave will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.
<code>bot_user_oauth_token</code>	the Slack bot user OAuth token (chr)
<code>file</code>	prefix for filenames (defaults to ggplot)
<code>...</code>	other arguments passed to graphics device

**Value**

httr response object (invisibly)

**Examples**

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

## End(Not run)
```

---

register_onexit	<i>Append text_slackr as on.exit to functions.</i>
-----------------	--

---

**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"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_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	which channel to post the message to (chr)
username	what user should the bot be named as (chr)
icon_emoji	what emoji to use (chr) "" will mean use the default
bot_user_oauth_token	Slack bot user OAuth token

**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**

[text\\_slackr\(\)](#)

**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
delete_slackr(count = 6,channel = '#general')

## End(Not run)
```



---

save.slackr	<i>Superseded by save_slackr</i>
-------------	----------------------------------

---

**Description**

deprecated

**Usage**

```
save.slackr(...)
```

**Arguments**

... arguments to pass to save\_slackr

**See Also**

[save\\_slackr](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

save_slackr	<i>Save R objects to an RData file on Slack</i>
-------------	---

---

**Description**

save\_slackr 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**

```
save_slackr(  
  ...,  
  channels = Sys.getenv("SLACK_CHANNEL"),  
  file = "slackr",  
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),  
  plot_text = ""  
)
```

**Arguments**

`...` objects to store in the R data file  
`channels` Slack channels to save to (optional)  
`file` filename (without extension) to use  
`bot_user_oauth_token` Slack bot user OAuth token  
`plot_text` the plot text to send with the plot (defaults to "")

**Value**

httr response object from POST call

**See Also**

[slackr\\_setup\(\)](#), [slackr\\_dev\(\)](#), [slackr\\_upload\(\)](#)

**Examples**

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

## 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"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")
)
```

**Arguments**

. . . expressions to be sent to Slack  
 channel which channel to post the message to (chr)  
 username what user should the bot be named as (chr)  
 icon\_emoji what emoji to use (chr) "" will mean use the default  
 bot\_user\_oauth\_token Slack bot user OAuth token

**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 setup an API token <https://api.slack.com/>. Also, you can pass in `as_user=TRUE`, the default, 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`). Passing `as_user=FALSE`, results in the Slack API posting the message as set in `username`.

**See Also**

[slackr\\_setup\(\)](#), [slackr\\_bot\(\)](#), [dev\\_slackr\(\)](#), [save\\_slackr\(\)](#), [slackr\\_upload\(\)](#)

**Examples**

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

## End(Not run)
```

---

 slackrBot

*Superseded by slackr\_bot*


---

**Description**

deprecated

**Usage**

```
slackrBot(...)
```

**Arguments**

... arguments to pass to `slackr_bot`

**See Also**

[slackr\\_bot](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

slackrChannels

*Superseded by `slackr_channels`*

---

**Description**

deprecated

**Usage**

```
slackrChannels(...)
```

**Arguments**

... arguments to pass to `slackr_channels`

**See Also**

[slackr\\_channels](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

slackrChtrans                    *Superseded by slackr\_chtrans*

---

**Description**

deprecated

deprecated

**Usage**

slackrChtrans(...)

slackrChTrans(...)

**Arguments**

...                    arguments to pass to slackr\_chtrans

**See Also**

[slackr\\_chtrans](#)

[slackr\\_chtrans](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

slackrIms                    *Superseded by slackr\_ims*

---

**Description**

deprecated

**Usage**

slackrIms(...)

**Arguments**

...                    arguments to pass to slackr\_ims

**See Also**[slackr\\_ims](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

`slackrMsg`*Superseded by slackr\_msg*

---

**Description**

deprecated

**Usage**`slackrMsg(...)`**Arguments**... arguments to pass to `slackr_msg`**See Also**[slackr\\_msg](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

`slackrSetup`*Superseded by slackr\_setup*

---

**Description**

deprecated

**Usage**`slackrSetup(...)`**Arguments**... arguments to pass to `slackr_setup`**See Also**[slackr\\_setup](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

slackrUpload	<i>Superseded by slackr_upload</i>
--------------	------------------------------------

---

**Description**

deprecated

**Usage**

```
slackrUpload(...)
```

**Arguments**

... arguments to pass to slackr\_upload

**See Also**

[slackr\\_upload](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUsers\(\)](#), [textSlackr\(\)](#)

---

slackrUsers	<i>Superseded by slackr_users</i>
-------------	-----------------------------------

---

**Description**

deprecated

**Usage**

```
slackrUsers(...)
```

**Arguments**

... arguments to pass to slackr\_users

**See Also**

[slackr\\_users](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrIms\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [textSlackr\(\)](#)

---

 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(
  ...,
  channel = "",
  username = "",
  icon_emoji = "",
  incoming_webhook_url = Sys.getenv("SLACK_INCOMING_URL_PREFIX")
)
```

## Arguments

<code>...</code>	expressions to be sent to Slack
<code>channel</code>	Deprecated. will have no effect
<code>username</code>	Deprecated. will have no effect
<code>icon_emoji</code>	Deprecated. will have no effect
<code>incoming_webhook_url</code>	which <code>slack.com</code> API endpoint URL to use (see section <b>Webhook URLs</b> 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/XXXXXX/XXXXXX/XXXXXX`

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**

[slackrSetup\(\)](#), [slackr\(\)](#), [dev\\_slackr\(\)](#), [save\\_slackr\(\)](#), [slackr\\_upload\(\)](#)

**Examples**

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

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

## End(Not run)
```

---

slackr_census_fun	<i>Create a cache of the users and channels in the workspace in order to limit API requests</i>
-------------------	---

---

**Description**

Create a cache of the users and channels in the workspace in order to limit API requests

**Usage**

```
slackr_census_fun(
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")
)
```

**Arguments**

bot\_user\_oauth\_token  
the Slack bot OAuth token (chr)

**Value**

A data.frame of channels and users

---

slackr_channels	<i>Get a data frame of Slack channels</i>
-----------------	---

---

**Description**

Get a data frame of Slack channels

**Usage**

```
slackr_channels(  
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")  
)
```

**Arguments**

bot\_user\_oauth\_token  
the Slack bot OAuth token (chr)

**Value**

data.table of channels

---

slackr_chtrans	<i>Translate vector of channel names to channel IDs for API</i>
----------------	---

---

**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)
```

**Arguments**

channels          vector of channel names to parse

**Value**

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

**Author(s)**

Quinn Weber (ctb), Bob Rudis (aut)

---

slackr_delete	<i>slackr_delete</i>
---------------	----------------------

---

**Description**

Deletes the specified number of messages from the channel

**Usage**

```
slackr_delete(
  count,
  channel = Sys.getenv("SLACK_CHANNEL"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")
)
```

**Arguments**

count	the number of messages to delete
channel	the channel to delete from
bot_user_oauth_token	the Slack bot user OAuth token

---

slackr_dev	<i>Send the graphics contents of the current device to a Slack channel</i>
------------	--

---

**Description**

slackr\_dev sends the graphics contents of the current device to the specified Slack channel.

**Usage**

```
slackr_dev(
  channels = Sys.getenv("SLACK_CHANNEL"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),
  plot_text = "",
  file = "plot"
)
```

**Arguments**

channels	list of channels to post image to
bot_user_oauth_token	the Slack full bot user OAuth token (chr)
plot_text	the plot text to send with the plot (defaults to "")
file	prefix for filenames (defaults to plot)

**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\(\)](#), [save\\_slackr\(\)](#), [slackr\\_upload\(\)](#)

**Examples**

```
## Not run:
slackr_setup()

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

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

## End(Not run)
```

---

slackr_history	<i>Reads history of a channel.</i>
----------------	------------------------------------

---

**Description**

Returns a dataframe of post history in a channel.

**Usage**

```
slackr_history(
  channel = Sys.getenv("SLACK_CHANNEL"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),
  posted_to_time = as.numeric(Sys.time()),
  message_count,
  duration,
  posted_from_time,
  paginate = FALSE
)
```

**Arguments**

channel	The channel to get history from
bot_user_oauth_token	the Slack bot user OAuth token
posted_to_time	Timestamp of the last post to consider (default: current time)
message_count	The number of messages to retrieve (only when paginate = FALSE)
duration	Number of hours of history to retrieve. By default retrieves 24 hours of history.
posted_from_time	Timestamp of the first post time to consider
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

**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	<i>Get a data frame of Slack IM ids</i>
------------	---

---

**Description**

Get a data frame of Slack IM ids

**Usage**

```
slackr_ims(bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"))
```

**Arguments**

bot_user_oauth_token	the Slack bot OAuth token (chr)
----------------------	---------------------------------

**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"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),
  ...
)
```

**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	which channel to post the message to (chr)
username	what user should the bot be named as (chr)
icon_emoji	what emoji to use (chr) "" will mean use the default
bot_user_oauth_token	Slack bot user OAuth token
...	other arguments passed to the Slack API <code>chat.postMessage</code> 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\(\)](#), [dev\\_slackr\(\)](#), [save\\_slackr\(\)](#), [slackr\\_upload\(\)](#)

**Examples**

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

## 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 = "",
  bot_user_oauth_token = "",
  config_file = "~/slackr",
  echo = FALSE,
  cacheChannels = TRUE,
  cache_dir = ""
)
```

**Arguments**

<code>channel</code>	default channel to send the output to (chr) defaults to #general
<code>username</code>	the username output will appear from (chr) defaults to slackr
<code>icon_emoji</code>	which emoji picture to use (chr) defaults to none (can be left blank in config file as well)
<code>incoming_webhook_url</code>	the Slack URL prefix to use (chr) defaults to none
<code>bot_user_oauth_token</code>	the Slack full bot user OAuth token (chr)
<code>config_file</code>	a configuration file (DCF) - see <a href="#">read.dcf</a> - format with the config values.
<code>echo</code>	display the configuration variables (bool) initially FALSE
<code>cacheChannels</code>	a boolean for whether or not you want to cache channels to limit API requests (deprecated)
<code>cache_dir</code>	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 location pointed to by `config_file`, the values there will be used. The fields should be specified as such in the file:

```
bot_user_oauth_token: SLACK_BOT_USER_OAUTH_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\(\)](#), [dev\\_slackr\(\)](#), [save\\_slackr\(\)](#), [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\_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 = basename(filename),
  initial_comment = basename(filename),
  channels = Sys.getenv("SLACK_CHANNEL"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")
)
```

**Arguments**

filename	path to file
title	title on Slack (optional - defaults to filename)
initial_comment	comment for file on slack (optional - defaults to filename)
channels	Slack channels to save to (optional)
bot_user_oauth_token	Slack bot user OAuth token

**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\(\)](#), [dev\\_slackr\(\)](#), [save\\_slackr\(\)](#)

---

 slackr\_users

*Get a data frame of Slack users*

---

**Description**

Get a data frame of Slack users

**Usage**

```
slackr_users(bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"))
```

**Arguments**

bot\_user\_oauth\_token  
the Slack bot OAuth token (chr)

**Value**

data.frame of users

---

textSlackr	<i>Superseded by text_slackr</i>
------------	----------------------------------

---

**Description**

deprecated

**Usage**

```
textSlackr(...)
```

**Arguments**

... arguments to pass to text\_slackr

**See Also**

[slackr\\_msg](#)

Other Deprecated functions: [dev.slackr\(\)](#), [dev\\_slackr\(\)](#), [save.slackr\(\)](#), [slackrBot\(\)](#), [slackrChannels\(\)](#), [slackrChtrans\(\)](#), [slackrImgs\(\)](#), [slackrMsg\(\)](#), [slackrSetup\(\)](#), [slackrUpload\(\)](#), [slackrUsers\(\)](#)

---

text_slackr	<i>text_slackr</i>
-------------	--------------------

---

**Description**

Sends basic text to a slack channel. Calls the chat.postMessage method on the Slack Web API. Information on this method can be found here: <https://api.slack.com/methods/chat.postMessage>

**Usage**

```
text_slackr(
  text,
  ...,
  preformatted = TRUE,
  channel = Sys.getenv("SLACK_CHANNEL"),
  username = Sys.getenv("SLACK_USERNAME"),
  icon_emoji = Sys.getenv("SLACK_ICON_EMOJI"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN")
)
```

**Arguments**

text	The character vector to be posted
...	Optional arguments such as: parse, unfurl_links, etc.
preformatted	Should the text be sent as preformatted text. Defaults to TRUE
channel	The name of the channels to which the DataTable should be sent. Prepend channel names with a hashtag. Prepend private-groups with nothing. Prepend direct messages with an @
username	what user should the bot be named as (chr)
icon_emoji	what emoji to use (chr) "" will mean use the default
bot_user_oauth_token	your full Slack bot user OAuth token

**Value**

httr response object (invisibly)

**Note**

You can pass in `as_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)**

Quinn Weber (aut), Bob Rudis (ctb)

**References**

<https://github.com/mrkaye97/slackr/pull/11>

**See Also**

<https://api.slack.com/methods/chat.postMessage>

**Examples**

```
## Not run:  
slackr_setup()  
text_slackr('hello world', as_user=TRUE)  
  
## End(Not run)
```

---

 tex\_slackr

*Post a tex output to a Slack channel*


---

### Description

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

### Usage

```
tex_slackr(
  obj,
  channels = Sys.getenv("SLACK_CHANNEL"),
  bot_user_oauth_token = Sys.getenv("SLACK_BOT_USER_OAUTH_TOKEN"),
  ext = "png",
  path = NULL,
  ...
)
```

### Arguments

<code>obj</code>	character object containing tex to compile
<code>channels</code>	list of channels to post image to
<code>bot_user_oauth_token</code>	the Slack full bot user OAuth token (chr)
<code>ext</code>	character, type of format to return, can be tex, pdf, or any image device, Default: 'png'
<code>path</code>	character, path to save tex_preview outputs, if NULL then tempdir is used, Default: NULL
<code>...</code>	other arguments passed to <code>texPreview::tex_preview()</code> , 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\(\)](#) [xtable::print.xtable\(\)](#)

**Examples**

```
## Not run:
slackr_setup()
obj=xtable::xtable(mtcars)
tex_slackr(obj,print.xtable.opts=list(scalebox=getOption("xtable.scalebox", 0.8)))

tex_slackr(obj,ext = 'pdf',print.xtable.opts=list(scalebox=getOption("xtable.scalebox", 0.8)))

tex_slackr(obj,ext = 'tex',print.xtable.opts=list(scalebox=getOption("xtable.scalebox", 0.8)))

tex_slackr(obj,path='testdir',print.xtable.opts=list(scalebox=getOption("xtable.scalebox", 0.8)))

## End(Not run)
```

---

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 an 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

## \* **Deprecated functions**

dev.slackr, 5  
dev\_slackr, 5  
save.slackr, 9  
slackrBot, 11  
slackrChannels, 12  
slackrChtrans, 13  
slackrIms, 13  
slackrMsg, 14  
slackrSetup, 14  
slackrUpload, 15  
slackrUsers, 15  
textSlackr, 26

## \* **Internal**

convert\_response\_to\_tibble, 4

auth\_test, 3

call\_slack\_api, 3, 4

convert\_response\_to\_tibble, 4

convert\_response\_to\_tibble(), 29

dev.slackr, 5, 5, 9, 12–15, 26

dev\_slackr, 5, 5, 9, 12–15, 26

dev\_slackr(), 11, 17, 23–25, 28

dplyr::bind\_rows(), 29

ggslackr, 6

read.dcf, 23

register\_onexit, 7

save.slackr, 5, 9, 12–15, 26

save\_slackr, 9, 9

save\_slackr(), 11, 17, 20, 23–25

slackr, 10

slackr(), 17, 23, 24

slackr\_bot, 12, 16

slackr\_bot(), 11, 23

slackr\_census\_fun, 17

slackr\_channels, 12, 18

slackr\_chtrans, 13, 18

slackr\_delete, 19

slackr\_dev, 5, 19

slackr\_dev(), 6, 10

slackr\_history, 20

slackr\_ims, 14, 21

slackr\_msg, 14, 22, 26

slackr\_setup, 14, 23

slackr\_setup(), 10, 11, 16, 20, 23, 25

slackr\_upload, 15, 24

slackr\_upload(), 10, 11, 17, 20, 23, 24

slackr\_users, 15, 25

slackrBot, 5, 9, 11, 12–15, 26

slackrChannels, 5, 9, 12, 12, 13–15, 26

slackrChTrans (slackrChtrans), 13

slackrChtrans, 5, 9, 12, 13, 14, 15, 26

slackrIms, 5, 9, 12, 13, 13, 14, 15, 26

slackrMsg, 5, 9, 12–14, 14, 15, 26

slackrSetup, 5, 9, 12–14, 14, 15, 26

slackrSetup(), 17

slackrUpload, 5, 9, 12–15, 15, 26

slackrUsers, 5, 9, 12–15, 15, 26

tex\_slackr, 28

texPreview::tex\_preview(), 28, 29

text\_slackr, 26

text\_slackr(), 8

textSlackr, 5, 9, 12–15, 26

with\_pagination, 29

xtable::print.xtable(), 29