slackr : : CHEAT SHEET

Send files, messages, R objects, and images to Slack directly from R

Installation

CRAN version

```
install.packages("slackr")
```

Development version

```
devtools::install_github("mrkaye97/slackr")
```

Setup

i. Go to https://api.slack.com/apps
ii. Click “**Create New App**” and then follow the setup instructions

Multi-channel bot

1. Click “**OAuth & Permissions**” under “**Features**”
2. Enable the following scopes in order to get all of the functionality:
   - `channels:read`
   - `chat:write`
   - `users:read`
   - `chat:write.customize`
   - `files:read`
   - `chat:write.public`
   - `groups:read`
   - `im:write`
   - `groups:write`
   - `incoming-webhook`
   - `channels:history`
3. Click “**Install to Workspace**”
4. Select a channel (for webhook messages)
5. Copy the Bot User OAuth Access Token
6. Click “**Incoming Webhooks**” under “**Features**”
7. Copy the Webhook URL

Single-channel bot

1. Click “**Incoming “Webhooks**” under “**Features**”
2. Turn the “**Activate Incoming Webhooks**” switch on
3. Click “**Add New Webhook to Workspace**”
4. Select the channel you’d like the bot to post to
5. Copy the Webhook URL

Setup environment variables for Slack API access

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

Generate Config Text (.txt) File (optional)

```
bot_user_oauth_token: xoxb-[…]
channel: #yourchannel
username: yourusername
incoming_webhook_url: https://
hooks.slack.com/services/XXXXX/XXXXX/XXXXX
```

Common Functions

Post a ggplot to a Slack channel

```
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",
  ...
)
```

Save R objects to an RData file on Slack

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

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

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

Delete the specified number of messages from the channel

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

Send a message to a Slack channel

```
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"),
  ...
)
```

Send a file to Slack

```
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")
)
```

Other Usages

`auth_test()`	Checks authentication & identity against the Slack API
`call_slack_api()`	A wrapper function to call the Slack API with authentication and pagination
`convert_response_to_tibble()`	Convert Slack API json response to tibble
`register_onexit()`	Append text_slackr as on.exit to functions
`slackr_census_fun()`	Create a cache of the users and channels in the workspace in order to limit API requests
`slackr_channels()`	Get a data frame of Slack channels
`slackr_chtrans()`	Translate vector of channel names to channel IDs for API
`slackr_dev()`	Send the graphics contents of the current device to a Slack channel
`slackr_history()`	Reads history of a channel
`slackr_ims()`	Get a data frame of Slack IM IDs
`slackr_users()`	Get a data frame of Slack users
`text_slackr()`	Sends basic text to a slack channel
`tex_slackr()`	Post a tex output to a Slack channel
`with_pagination()`	Calls the slack API with pagination using cursors

Vignettes

Creating a single-channel bot

```
vignette('webhook–setup', package = ‘slackr’)
```

Creating a fully-functional multi-channel bot

```
vignette('scoped–bot–setup', package = ‘slackr’)
```

Usage

```
vignette('using–slackr', package = ‘slackr')
```

RStudio® is a trademark of RStudio, Inc. • slackr (2021) by Bob Rudis, et al. • CC BY SA Daniel M. Villarreal • dan.m.villarreal@gmail.com • Learn more at https://github.com/mrkaye97/slackr • package version 2.2.0 • Updated: 2021-03-03