
givecookiebot-slack

Documentation

Release 1.0.0

Austin Developer Community

May 05, 2019

API REFERENCE

1	Installation	3
1.1	Slack API Tokens	3
1.2	ngrok	4
1.3	Start givecookiebot-slack	4
2	Indices and tables	9
	Python Module Index	11

givecookiebot-slack is a Slack bot that adds a cookie emoji (🍪) to a user's status for 3 hours when that user is 'given' a cookie in chat.

Something like:

@User Thanks for helping out today! 🍪

The user's status will then be updated to include the cookie emoji.

After 3 hours have passed, the cookie emoji is removed from the user's status.

INSTALLATION

givecookiebot-slack is a Python script that requires a few dependencies specified in requirements.txt.

```
pip install -r requirements.txt
```

With the Python requirements installed, we can move on to Slack API's requirements.

1.1 Slack API Tokens

First, the Slack client tokens must be added to the environment variables in ~/.bashrc:

```
export SLACK_CLIENT_ID="first_long_complicated_string_here"
export SLACK_CLIENT_SECRET="second_long_complicated_string_here"
export SLACK_SIGNING_SECRET="third_long_complicated_string_here"
export SLACK_BOT_SCOPE="channels:history users.profile:write"
```

Next, run tokens.py to get the user token that will be used to access the Slack API.

```
$ cd givecookiebot-slack # Go to source code directory (if needed)
$ source ./venv/bin/activate # Activate venv (if applicable)
(venv)$ python -m givecookiebotslack.tokens
```

Note: Only the owner can set the status of an admin user and only owners/admin users can set user statuses.

The Flask server is running at 127.0.0.1:5000, by default. Go to http://127.0.0.1:5000/begin_auth and click on Add to Slack to start the OAuth process.

Once **givecookiebot-slack** gets approval for the needed scopes and permissions, the user token should be saved to the SLACK_USER_TOKEN environment variable.

Warning: The SLACK_USER_TOKEN environment variable is lost once the terminal is closed. So, use CTRL+C to stop Flask and keep the terminal open when starting the bot.

1.2 ngrok

One can use [ngrok](#) as a quick way to access the local Flask server externally.

Note: Using ngrok requires making an account. Free accounts have dynamic domain names that change every 8 hours or when the terminal is closed.

Once the local Flask server is running, start ngrok with:

```
ngrok http 5000
```

Use either the http or https forwarding address: [xxxxxxx.ngrok.io/begin_auth](#).

In the OAuth & Permissions settings at <https://api.slack.com/apps>, make sure to set the Redirect URL to [xxxxxxx.ngrok.io/finish_auth](#).

If successful, it should render a page that says “Auth complete!”.

Warning: As mentioned in the slack

1.3 Start givecookiebot-slack

With the tokens saved, start **givecookiebot-slack** in a new terminal with:

```
$ cd givecookiebot-slack # Go to source code directory (if needed)
$ source ./venv/bin/activate # Activate venv (if applicable)
(venv)$ python -m givecookiebotslack.bot
```

The Flask server is running at 127.0.0.1:3000, by default. The bot will be waiting for events from slack at the address <http://127.0.0.1:3000/slack/events>.

Navigating there should render a page that says “These are not the slackbots you’re looking for.”

Once again, use ngrok to register the bot with Slack Events at <https://api.slack.com/apps> with:

```
ngrok http 3000
```

Provide the https forwarding link as the Request URL: <https://xxxxxxx.ngrok.io/slack/events>.

Add the bot to a channel and start passing out [].

Warning: This environment is **not** suitable for production. Tokens should be stored a secure database. Ngrok shouldn’t be used to create a tunnel. The built-in Flask server should not be used.

1.3.1 givecookiebotslack

givecookiebotslack package

Submodules

givecookiebotslack.bot module

givecookiebot-slack

Defines [Event API](#) events that the bot will respond to.

givecookiebotslack.bot.SLACK_SIGNING_SECRET

Secret identifying bot to Slack Events API. Like a password.

Type `str`

givecookiebotslack.bot.SLACK_USER_TOKEN

Token used to access Slack Web API.

Type `str`

givecookiebotslack.bot.slack_events_adapter

Instance of `SlackEventAdapter`.

Type `SlackEventAdapter`

givecookiebotslack.bot.slack_client

Instance of `SlackClient`.

Type `SlackClient`

givecookiebotslack.bot.USERMENTION_RE

Regular expression object matching a user mention pattern.

Type `re.compile`

givecookiebotslack.bot.profile

Dictionary containing template of user profile status.

Type `dict`

givecookiebotslack.bot.error_handler(*err*)

Error events

givecookiebotslack.bot.handle_message(*event_data: dict*)

Message handler

Parses message for cookie emoji (:cookie:) and a user mention, then changes mentioned user's status to include cookie emoji for 3 hours.

Returns True if successful, otherwise the error code given by the Slack API.

givecookiebotslack.tokens module

Get User and Bot tokens.

Uses Slack's [OAuth 2.0](#) to fetch the User and Bot tokens for interfacing with Slack's API.

givecookiebotslack.tokens.post_install()

Gets tokens from auth completion page.

givecookiebotslack.tokens.pre_install()

Directs user to Slack's OAuth acceptance page.

Module contents

givecookiebot-slack: Slack bot that adds a cookie emoji (🍪) to a user's status for 3 hours when that user is 'given' a cookie in chat.

GPLv3 License

Austin Developer Community

1.3.2 Contributing Guidelines

Thanks for helping out! There are plenty of typos to fix and features to add. Getting started is easy!

Installation

General installation is covered in the [README](#). Installing into a [venv](#) is recommended.

Development

Install additional requirements from requirements-dev.txt:

```
pip install -r requirements-dev.txt
```

Ensure any additional contributions follow pylint guidelines.

Building Documentation

Building documentation is not needed unless contributing to the documentation.

HTML Docs

Install requirements specified in requirements-docs.txt. Move to docs directory and build the HTML files:

```
$ make html
```

HTML files will be located in ../givecookiebot-slack-docs/html/.

LaTeX PDF

Building the PDF has more requirements that vary based on the operating system. For Ubuntu 18.04:

```
$ sudo apt-get install texlive-latex-recommended texlive-latex-extra texlive-fonts-recommended texlive-xetex
```

XeLaTeX, texlive-xetex, is needed for Unicode emojis to be inserted into the PDF. Build the PDF from within the docs directory:

```
$ make latexpdf
```

The PDF is saved to `../givecookiebot-slack.pdf`. It may have to be built multiple times to build without errors because the indexer sometimes fails.

Code of Conduct

Whether submitting Issues or Pull Requests, be sure to follow the [Code of Conduct](#).

Submit Bug/Issue/Enhancement

Open an issue in the [GitHub Issue tracker](#) following the [Issue template](#). Be as detailed as you can to help identify the bug.

Submit Pull Request

Whether adding a feature or fixing a bug, open a pull request in [GitHub](#) following the [Pull request template](#).

Contact

Our [Facebook group](#) has a link to the Slack workspace where questions can be asked in the `#general` channel or concerns can be raised by Direct Messaging the project team.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

g

givecookiebotslack, 6

givecookiebotslack.bot, 5

givecookiebotslack.tokens, 5