
clashcallerbot-reddit Documentation

Release 2.2.7

Jose A. Lerma III

Nov 15, 2018

GETTING STARTED

1	Usage Information	3
1.1	Quickstart	4
1.1.1	Prerequisites	4
1.1.2	Setup	4
1.1.3	Starting	4
1.1.4	Updating	5
1.2	Installation	5
1.2.1	Deployment Options	5
1.2.2	Building Documentation	6
1.2.3	Disclaimer	7
1.3	clashcallerbot-reddit	7
1.3.1	clashcallerbot_database module	7
1.3.2	clashcallerbot_reply module	10
1.3.3	clashcallerbot_search module	11
1.3.4	logging_conf module	12
1.4	FAQs	12
1.4.1	What is the point of ClashCallerBot?	12
1.4.2	Is it ready yet?	13
1.4.3	How do you call ClashCallerBot?	13
1.4.4	Do PMs work?	13
1.4.5	What options does it have?	13
1.4.6	Hey, why didn't it notice me?	14
1.4.7	Why was the message off by a few minutes?	14
1.4.8	What happens if it gets banned?	15
1.4.9	Can I delete my comment?	15
1.5	References	15
1.5.1	Python	15
1.5.2	PRAW	15
1.5.3	Database	15
1.5.4	Documentation	16
1.5.5	Miscellaneous	16
1.6	Index	16
	Python Module Index	17
	Index	19

ClashCallerBot was made to help [/r/ClashOfClans](#) clans coordinate attacks during [Clan Wars](#) (or [Clan War Leagues](#)) from within reddit.

For example, someone wants to attack base 1 and 7, but they haven't posted an update in over an hour and those two bases still haven't been attacked. Is it okay to attack those bases? Did your fellow clan member die? Who knows?

Well, if s/he (or someone on their behalf) had called those bases for a set period of time, you would know for certain.

Think of **ClashCallerBot** as an independent time keeper that runs entirely within reddit.

If not obvious enough, it's a fork of [Silver's RemindMeBot](#). It's a sweet (as in awesome) little program.

I plan to make a few more tweaks to make it even more useful.

Check us out on [/r/ClashCallerBot](#).

USAGE INFORMATION

Usage can be inferred from the regular expressions used to process each comment in the *clashcallerbot_search* module:

```

38 # Regular expressions
39 clashcaller_re = re.compile(r'''
40     [!|\s]?          # prefix ! or space (optional)
41     [C|c]lash[C|c]aller # lower or camelcase ClashCaller_
42     ↳(required)
43     [!|\s]          # suffix ! or space (required)
44     ''', re.VERBOSE)
45 expiration_re = re.compile(r'''
46     (?P<exp_digit>(\d){1,2}) # single or double digit_
47     ↳(required)
48     (\s)?              # space (optional)
49     (?P<exp_unit>minute(s)?\s| # minute(s) (space after_
50     ↳required)
51     min\s|             # minute abbr. (space after_
52     ↳required)
53     hour(s)?\s|        # hour(s) (space after_
54     ↳required)
55     hr\s              # hour abbr. (space after_
56     ↳required)
57     )+''', re.VERBOSE | re.IGNORECASE) # case-insensitive
58 message_re = re.compile(r'''
59     (\s)*             # space (optional)
60     "                 # opening double quote (required)
61     base(s)?          # string: base(s) (required)
62     [\W|\s]*          # non-word character or space (optional)
63     (\d){1,2}         # single or double digit (required)
64     .*               # any character after (optional)
65     "                 # closing double quote (required)
66     ''', re.VERBOSE | re.IGNORECASE) # case-insensitive

```

To sum:

- The `clashcaller` string must be present in either lower or camelcase with an exclamation point ! either before or after.
- The expiration time in minutes or hours may follow either abbreviated or with full spelling with an optional space between the number and word, but mandatory space after the word: 4min. Case is ignored. If not given, defaults to 1 hour. The expiration time is limited to within 24 hours.
- The message within quotes must follow with the singular or plural form of `base` and a required single or double digit number. Case is ignored. Maximum message length is 100 characters to save database space.

1.1 Quickstart

This is what you'll need to run **clashcallerbot-reddit** ASAP (As Soon As Possible). **clashcallerbot-reddit** is developed on various Linux distros, so *Installation* is assumed to be on Linux.

1.1.1 Prerequisites

Python clashcallerbot-reddit is a massive *python* script, so at least [Python 3.6](#) is needed.

Pip Configuring *python* is easiest using [pip](#). Once pip is installed, the requirements can be installed by running `pip install -r requirements.txt` from within the source code directory.

MySQL clashcallerbot-reddit uses a MySQL-compatible database to store calls and to keep track of comments that were replied to. Either [MySQL](#) or [MariaDB](#) can be used.

With these prerequisites met, **clashcallerbot-reddit** can be setup and run.

1.1.2 Setup

First, add the bot's [reddit metadata](#) to *praw-example.ini* and rename to *praw.ini*, then add the database's root and desired bot user credentials to *database-example.ini* and rename to *database.ini*.

Next, change the following line in the *clashcallerbot_database module*

```
465 # Create the clashcaller database
466 database = ClashCallerDatabase(config_file=config, root_user=False)
```

to `database = ClashCallerDatabase(config_file=config, root_user=True)`. This may get updated to be default later.

Now, the MySQL-compatible database can be setup by running the *clashcallerbot_database module* directly from within terminal:

```
python3 ./clashcallerbot_database.py
```

1.1.3 Starting

Once the database is setup, the bot can be run by calling the *clashcallerbot_search module* and the *clashcallerbot_reply module* directly from within terminal:

```
python3 ./clashcallerbot_search.py && python3 ./clashcallerbot_reply.py
```

Alternatively, by running the provided bash script from within terminal:

```
./clashcallerbot.sh
```

Note:

- Remember to set executable mode with `chmod +x ./clashcallerbot.sh`.
- Script assumes files are in same directory and is run as crontab in `no_root_setup`.
- How often to run as a crontab depends on how long you want the bot to be down/broken.
- If you have access to root, check *Installation* for info on setting up systemd instead.

- Logfile can be removed if not necessary (remove variable and `>> $logfile`).
 - `python` can be replaced with relevant python version.
 - The function in `kill` returns all script PIDs, so it must be restarted.
 - If you have access to `pidof`, you can avoid killing all script instances.
-

1.1.4 Updating

A bash script is also provided for updating and can be run directly from within terminal:

```
./redownload.sh
```

Note:

- Don't forget to set executable mode with `chmod +x ./redownload.sh`.
 - Script assumes files are all in same directory and that it is run in a `no_root_setup`.
 - The `-f` switch and `> /dev/null 2>&1` silence outputs to certain degrees.
-

1.2 Installation

This thing has to run 24/7. As a proof of concept, I set it up in an Ubuntu VM for the open alpha and a CentOS VM for the open beta. Production version was running on CloudLinux OS at one point, but has since moved to Amazon Linux AMI on AWS. If you are using a different OS, your mileage may vary.

1.2.1 Deployment Options

Believe me, the list of where I have not deployed it is shorter. I mostly went with virtual machines for the development, and tried putting the production version on a shared host. It has been migrated to an Amazon EC2 instance in hopes of minimizing costs.

Ubuntu

For the open alpha, I went with an OS that is convenient (need something? There's a package for that) as a proof of concept and debugging environment.

`ubuntu_setup`

CentOS

For the open beta, I went with an OS that is often available for production environments.

`centos_setup`

No Root

For my first attempt at a production version, I got a shared host running CloudLinux OS that does not provide root access, but has both Python and MySQL pre-installed.

no_root_setup

Amazon EC2

For what I hope is the final production version, I set up an Amazon EC2 instance running Amazon Linux AMI. I particularly like that it has pip and python pre-installed.

amazon_setup

1.2.2 Building Documentation

Note: Building the documentation is **not needed or recommended** unless contributing to the documentation. The latest version of the documentation is available at [josealermalii@github.io/clashcallerbot-reddit](https://josealermalii.github.io/clashcallerbot-reddit) or as a [PDF in the source code](#). You have been warned.

Building the docs requires a few more pip packages:

- Sphinx
- sphinxcontrib-napoleon
- sphinx-rtd-theme

If that wasn't bad enough, a symbolic link to *praw.ini* must be made from within `~/ .config`:

```
cd ~/.config
ln -s absolute_path_here/clashcallerbot-reddit/praw.ini praw.ini
```

This has to do with how *praw* finds the *praw.ini* file. This may get fixed later much like the *database.ini* fix in *config.py*.

Now, we can build the docs in HTML format:

```
cd absolute_path_here/clashcallerbot-reddit/docs
make html
```

This will save the docs website in `../.. /clashcallerbot-reddit-docs/`.

Building the PDF is even more involved. First, LaTeX must be installed on the OS. For example, in Ubuntu 18.04:

```
sudo apt-get install texlive-latex-recommended texlive-latex-extra texlive-fonts-
↪recommended
```

Installing these dependencies is not recommended, if not needed, because they require > 300 MB of disk space.

Now, we can build the docs in PDF format:

```
cd absolute_path_here/clashcallerbot-reddit/docs
make latexpdf
```

This will save the doc's PDF in `../manual.pdf`.

1.2.3 Disclaimer

Though covered by the MIT License, I reiterate: executing code from the Internet in terminal can end up doing bad things.

Read and understand all code you copy and paste before running it.

1.3 clashcallerbot-reddit

1.3.1 clashcallerbot_database module

Setup the MySQL-compatible database.

If run directly, this module will setup the ClashCallerBot database with tables and display their format and contents. Additionally, this module provides a class with various methods for managing the MySQL-compatible database:

- Create database and tables.
- View table data and properties.
- Lock tables for reading and writing.
- Grant user permissions (if logged into database as root).
- Add rows to tables.
- Delete tables and rows.
- Convert python datetime to MySQL datetime.

class `clashcallerbot_database.ClashCallerDatabase` (*config_file=None*,
root_user=None)

Bases: `object`

Implements a class for a ClashCaller Database.

Acts as an object-relational mapper for `mysql.connector` specific to ClashCallerBot.

config_file

A `configparser` object with `database.ini` file pre-read.

Type `configparser.ConfigParser()`

root_user

Specifies whether the database will be setup as root user.

Type `bool`

mysql_connection

A `mysql.connector.connect()` object.

Type `mysql.connector.connect()`

cursor

A `mysql.connector.connect().cursor()` object.

Type `mysql.connector.connect().cursor()`

close_connections () → `bool`

Close database connections.

Method closes database cursor and connection.

Returns True if successful, False otherwise.

static convert_datetime (*dt*: <module 'datetime' from '/usr/lib/python3.6/datetime.py'>) → <module 'datetime' from '/usr/lib/python3.6/datetime.py'>
 Converts python datetime to MySQL datetime.

Method converts given python datetime object to MySQL datetime format.

Parameters **dt** – Datetime object in default format.

Returns Datetime object in MySQL format.

create_database () → bool

Create database.

Method creates database with database name.

Returns True if successful, False otherwise.

create_table (*tbl_name*: str, *cols*: str) → bool

Create table in database.

Method creates table in database with given name and specifications.

Parameters

- **tbl_name** – Name to give table.
- **cols** – Columns to put in table.

Example

```
>>> tbl_name = 'table'
>>> cols = 'id INT UNSIGNED NOT NULL AUTO_INCREMENT, '
...       'permalink VARCHAR(100), message VARCHAR(100), new_date DATETIME, '
...       'userID VARCHAR(20), PRIMARY KEY(id) '
...
>>> create_table(tbl_name, cols)
```

Returns True if successful, False otherwise.

delete_message (*tid*: str) → bool

Deletes message from message_data table.

Method deletes given table id (row) from message_data table.

Parameters **tid** – Table id from id column of message_data table.

Returns True for success, False otherwise.

describe_table (*tbl_name*: str) → list

Gets description of table.

Method returns a list describing the structure of the given table.

Parameters **tbl_name** – Name of table to describe

Returns List with table description, empty list otherwise.

drop_table (*tbl_name*: str) → bool

Drop table from database.

Function drops given table from given database.

Parameters **tbl_name** – Table to drop.

Returns True if successful, False otherwise.

find_comment_id (*cid: str*) → bool

Check comment_list table for comment id.

Method checks comment_list table for given comment id.

Parameters **cid** – Comment id to search for.

Returns True for success, false otherwise.

get_messages (*time_now: datetime.datetime*) → list

Retrieves list of messages that have expired.

Method returns list of messages whose expiration times are before current datetime.

Parameters **time_now** – Current datetime.

Returns List containing results of query.

get_rows (*tbl_name: str*) → tuple

Fetch table rows.

Method gets rows of given table by order of id in a tuple.

Parameters **tbl_name** – Name of table to get rows from.

Returns Tuple containing each row's data, empty tuple otherwise.

get_tables () → list

Return table list of database.

Method returns a list with the names of the tables.

Returns List of table names.

grant_permissions () → bool

Grants bot user permissions to database.

Method grants bot user permissions to database.

Returns True if successful, False otherwise.

Notes

Only database root user can grant database permissions.

lock_read (*tbl_name: str*) → bool

Locks table for reading.

Method locks a given table for read access.

Parameters **tbl_name** – Name of table to lock.

Returns True if successful, False otherwise.

Notes

- Any previous locks are [implicitly released](#).
- Read locks have lower priority than write locks.

lock_write (*tbl_name: str*) → bool

Locks table for writing.

Method locks a given table for write access.

Parameters **tbl_name** – Name of table to lock.

Returns True if successful, False otherwise.

Notes

- Any previous locks are *implicitly released*.
- Write locks have higher priority than read locks.

open_connections () → bool

Open database connections.

Method makes database connection and cursor.

Returns True if successful, False otherwise.

save_comment_id (*cid: str*) → bool

Saves comment id into comment_list table.

Method saves given comment id into the comment_list table.

Parameters **cid** – Comment id to save.

Returns True for success, false otherwise.

save_message (*link: str, msg: str, exp: <module 'datetime' from '/usr/lib/python3.6/datetime.py'>, usr_name: str*) → bool

Saves given comment data into message_data table.

Method saves given inputs in message_date table as a row.

Parameters

- **link** – Comment permalink.
- **msg** – Comment message.
- **exp** – Expiration datetime object.
- **usr_name** – Comment author username.

Returns True for success, false otherwise.

select_database () → bool

Select database for command execution.

Method selects database within MySQL for command execution.

Returns True if successful, False otherwise.

clashcallerbot_database.**main**()

1.3.2 clashcallerbot_reply module

Checks messages in database and sends PM if expiration time passed.

This module checks messages saved in a MySQL-compatible database and sends a reminder via PM if the expiration time has passed. If so, the message is removed from the database.

`clashcallerbot_reply.get_parent(link: str) → str`

Fetch parent comment or submission.

Function gets parent comment of given permalink or submission if top level comment.

Parameters `link` – Permalink to get parent of.

Returns Parent comment link or default string.

`clashcallerbot_reply.main()`

`clashcallerbot_reply.send_reminder(link: str, msg: str, usr: str) → bool`

Sends reminder PM to username.

Function sends given permalink and message to given username.

Parameters

- `link` – Permalink to comment.
- `msg` – Message in comment that was saved.
- `usr` – User to send reminder to.

Returns True if successful, False otherwise.

1.3.3 clashcallerbot_search module

Searches recent reddit comments for ClashCaller! string and saves to database.

This module uses the Python Reddit API Wrapper (PRAW) to search recent reddit comments for the ClashCaller! string.

If found, the userID, permalink, comment time, message, and expiration time (if any) are parsed. The default, or provided, expiration time is applied, then all the comment data is saved to a MySQL-compatible database.

The comment is replied to, then the userID is PM'ed confirmation.

`clashcallerbot_search.have_replied(cid: str, bot_name: str) → bool`

Checks if bot user has replied to a comment.

Function checks reply authors for bot user.

Parameters

- `cid` – Comment ID to get replies of.
- `bot_name` – Name of bot to check for.

Returns True if successful, False otherwise.

`clashcallerbot_search.is_recent(cmnt: praw.models.reddit.comment.Comment) → bool`

Checks if comment is a recent comment.

Function compares comment time with program start time.

Parameters `cmnt` – praw comment object.

Returns True if comment time is after start time, False otherwise.

`clashcallerbot_search.main()`

`clashcallerbot_search.send_confirmation(u_name: str, link: str, exp: datetime.datetime) → bool`

Send confirmation to reddit user.

Function sends given user confirmation of given expiration time with given link.

Parameters

- **u_name** – username of user.
- **link** – Permalink of comment.
- **exp** – Expiration datetime of call.

Returns True if successful, False otherwise.

`clashcallerbot_search.send_confirmation_reply` (*cid: str, link: str, exp: datetime.datetime*)
→ str

Replies to a comment.

Function replies to a given comment ID with a given message.

Parameters

- **cid** – Comment ID to reply to.
- **link** – Permalink of comment.
- **exp** – Expiration datetime of call.

Returns id of new comment if successful, None otherwise

`clashcallerbot_search.send_error_message` (*u_name: str, link: str, error: str*) → bool
Send error message to reddit user.

Function sends given error to given user.

Parameters

- **u_name** – username of user.
- **link** – Permalink of comment.
- **error** – Error to send to user.

Returns True if successful, False otherwise.

1.3.4 logging_conf module

Defines logging dictionary.

Module defines dictionary for `logging.config.dictConfig()`

`logging_conf.LOGGING`

Dictionary containing definitions for the loggers, handlers, and formatters.

Type dict

1.4 FAQs

No one really asked them, but these are the anticipated FAQs (Frequently Asked Questions).

1.4.1 What is the point of ClashCallerBot?

ClashCallerBot was made to help [/r/ClashOfClans](#) clans coordinate attacks during [Clan Wars](#) (or [Clan War Leagues](#)) from within reddit.

For example, someone wants to attack base 1 and 7, but they haven't posted an update in over an hour and those two bases still haven't been attacked. Is it okay to attack those bases? Did your fellow clan member die? Who knows?

Well, if s/he (or someone on their behalf) had called those bases for a set period of time, you would know for certain.

Think of **ClashCallerBot** as an independent time keeper that runs entirely within reddit.

If not obvious enough, it's a fork of [Silver's RemindMeBot](#). It's a sweet (as in awesome) little program.

I plan to make a few more tweaks to make it even more useful.

1.4.2 Is it ready yet?

Closed Beta until I add a few more features. Let me know in [/r/ClashCallerBot](#) or via [PM](#) if you'd like your subreddit to be added to the closed beta.

1.4.3 How do you call ClashCallerBot?

ClashCaller! TIME OPTION "REQUIRED MESSAGE" (with quotes)

Time is optional (defaults to an hour); however, a message is required. Meaning if you do a simple ClashCaller! the bot won't do anything, but you will contribute to spam. Ideally, the message will be something like "Base #3" or something to indicate the base being called. Everything before ClashCaller! is not caught and everything after is. So feel free to use it in long winded posts but make sure it's after to avoid problems.

The PM will give you the permalink to your original comment.

1.4.4 Do PMs work?

No, they don't. Everyone needs to see what you called so they know not to attack it.

1.4.5 What options does it have?

Usage can be inferred from the regular expressions used to process each comment in the *clashcallerbot_search* module:

```

38 # Regular expressions
39 clashcaller_re = re.compile(r'''
40     [!|\s]?                # prefix ! or space (optional)
41     [C|c]lash[C|c]aller    # lower or camelcase ClashCaller
42     ↪(required)
43     [!|\s]                 # suffix ! or space (required)
44     ''', re.VERBOSE)
45 expiration_re = re.compile(r'''
46     (?P<exp_digit>(\d){1,2}) # single or double digit
47     ↪(required)
48     (\s)?                   # space (optional)
49     (?P<exp_unit>minute(s)?\s| # minute(s) (space after
50     ↪required)
51     min\s|                  # minute abbr. (space after
52     ↪required)
53     hour(s)?\s|             # hour(s) (space after
54     ↪required)
55     hr\s                    # hour abbr. (space after
56     ↪required)
57     )+''', re.VERBOSE | re.IGNORECASE) # case-insensitive

```

(continues on next page)

(continued from previous page)

```

52 message_re = re.compile(r'''
53     (\s)*          # space (optional)
54     "              # opening double quote (required)
55     base(s)?       # string: base(s) (required)
56     [\W|\s]*       # non-word character or space (optional)
57     (\d){1,2}      # single or double digit (required)
58     .*             # any character after (optional)
59     "              # closing double quote (required)
60     ''', re.VERBOSE | re.IGNORECASE) # case-insensitive

```

To sum:

- The `clashcaller` string must be present in either lower or camelcase with an exclamation point ! either before or after.
- The expiration time in minutes or hours may follow either abbreviated or with full spelling with an optional space between the number and word, but mandatory space after the word: 4min. Case is ignored. If not given, defaults to 1 hour. The expiration time is limited to within 24 hours.
- The message within quotes must follow with the singular or plural form of `base` and a required single or double digit number. Case is ignored. Maximum message length is 100 characters to save database space.

ClashCallerBot will then make this comment and will message you at the given time:

```

ClashCallerBot here!
I will be messaging you on Nov. 10, 2018 at 10:30:29 AM (UTC) to remind you of **this_
↳call.**

Thank you for entrusting us with your warring needs!

[^ (More info)] (https://www.reddit.com/r/ClashCallerBot/comments/4e9vo7/clashcallerbot_
↳info/)

```

1.4.6 Hey, why didn't it notice me?

Whoops! You should delete your comment (cut down on spam) and try again. Sometimes the bot misses a comment because there are too many comments happening at once on reddit, connection issues, my PC restarted, my Internet is down/too slow, or I'm watching Netflix. Do not worry about the reminder though, as long you got the confirmation PM/comment, the reminder PM will come. Worst case scenario is that it will come late if there are connection issues.

Also, make sure you are calling the bot correctly. Like so:

```
ClashCaller! OPTIONAL TIME (default is an hour without it) "REQUIRED MESSAGE".
```

You must do it as `ClashCaller!` exactly because it is case sensitive.

1.4.7 Why was the message off by a few minutes?

The bot currently goes to sleep every 2 minutes to save on resources. Meaning your message can be as late as 2 minutes + any connection issues it is having with reddit.

1.4.8 What happens if it gets banned?

As long as it doesn't receive a global ban on reddit, it will get your message. For example, the bot is currently banned on /r/AskReddit (all bots are). Although it won't send the message to your comment, it will PM you to confirm.

If you see others doing ClashCaller! and no response from the bot, no worries! PMs are private.

1.4.9 Can I delete my comment?

Yes. The bot sends a PM to the username that called it. As long as you don't delete your account you'll get your message. Note that if you delete your message – unless it's a top level comment – it won't be able to return the original parent comment.

1.5 References

clashcallerbot-reddit wasn't built with inherent knowledge. These are all the reference materials used to help make the bot run properly. Truth be told, these are all the bookmarks in my `clashcallerbot` folder.

1.5.1 Python

- `logging.config` - Logging configuration
- `datetime` - Basic date and time types
 - `strptime()` and `strftime()` Behavior
- `re` - Regular expression operations
- `configparser` - Configuration file parser

1.5.2 PRAW

- Submission Stream Reply Bot
- Comment Extraction and Parsing
- `SubredditStream`
- `Comment`
- `Reddit`
- `praw/config.py` at v6.0.0

1.5.3 Database

- MySQL 8.0 Reference Manual
 - Internal Locking Methods
- How to connect Python programs to MariaDB
- MySQL Connector/Python Developer Guide - Connector/Python Coding Examples
- MySQL Connector/Python Developer Guide - Guidelines for Python Developers

1.5.4 Documentation

- Setting up Sphinx for generating documentation from DocStrings, leveraging the Napoleon extension
 - Sphinx for Python documentation - Giselle Zeno
 - * Publishing sphinx-generated docs on github - sphinxdoc-test v0.1 documentation
 - Example Google Style Python Docstrings - napoleon 0.7 documentation
 - reStructuredText Primer - Sphinx 2.0.0+ documentation
- Semantic Versioning 2.0.0
- LaTeX error attempting to build PDF on Ubuntu 11.04

1.5.5 Miscellaneous

- JoseALermaIII/clashcallerbot-reddit
- Regex Tester
- Testing Thread Trois: ClashCallerBot
- commenting - reddit.com
- Logging module is causing Sphinx to crash. - Google Groups

1.6 Index

PYTHON MODULE INDEX

C

clashcallerbot_database, [7](#)
clashcallerbot_reply, [10](#)
clashcallerbot_search, [11](#)

I

logging_conf, [12](#)

INDEX

C

clashcallerbot_database (module), 7
clashcallerbot_reply (module), 10
clashcallerbot_search (module), 11
ClashCallerDatabase (class in clashcallerbot_database), 7
close_connections() (clashcallerbot_database.ClashCallerDatabase method), 7
config_file (clashcallerbot_database.ClashCallerDatabase attribute), 7
convert_datetime() (clashcallerbot_database.ClashCallerDatabase static method), 7
create_database() (clashcallerbot_database.ClashCallerDatabase method), 8
create_table() (clashcallerbot_database.ClashCallerDatabase method), 8
cursor (clashcallerbot_database.ClashCallerDatabase attribute), 7

D

delete_message() (clashcallerbot_database.ClashCallerDatabase method), 8
describe_table() (clashcallerbot_database.ClashCallerDatabase method), 8
drop_table() (clashcallerbot_database.ClashCallerDatabase method), 8

F

find_comment_id() (clashcallerbot_database.ClashCallerDatabase method), 9

G

get_messages() (clashcallerbot_database.ClashCallerDatabase method), 9

get_parent() (in module clashcallerbot_reply), 10
get_rows() (clashcallerbot_database.ClashCallerDatabase method), 9
get_tables() (clashcallerbot_database.ClashCallerDatabase method), 9
grant_permissions() (clashcallerbot_database.ClashCallerDatabase method), 9

H

have_replied() (in module clashcallerbot_search), 11

I

is_recent() (in module clashcallerbot_search), 11

L

lock_read() (clashcallerbot_database.ClashCallerDatabase method), 9
lock_write() (clashcallerbot_database.ClashCallerDatabase method), 9

LOGGING (in module logging_conf), 12
logging_conf (module), 12

M

main() (in module clashcallerbot_database), 10
main() (in module clashcallerbot_reply), 11
main() (in module clashcallerbot_search), 11
mysql_connection (clashcallerbot_database.ClashCallerDatabase attribute), 7

O

open_connections() (clashcallerbot_database.ClashCallerDatabase method), 10

R

root_user (clashcallerbot_database.ClashCallerDatabase attribute), 7

S

`save_comment_id()` (clashcallerbot_database.ClashCallerDatabase method), [10](#)

`save_message()` (clashcallerbot_database.ClashCallerDatabase method), [10](#)

`select_database()` (clashcallerbot_database.ClashCallerDatabase method), [10](#)

`send_confirmation()` (in module `clashcallerbot_search`), [11](#)

`send_confirmation_reply()` (in module `clashcallerbot_search`), [12](#)

`send_error_message()` (in module `clashcallerbot_search`), [12](#)

`send_reminder()` (in module `clashcallerbot_reply`), [11](#)