
discord_cakebot Documentation

Release 1.0.2

leagueofcake

Oct 03, 2020

Contents

1	Contents	1
1.1	Command List	1
1.2	Developing Modules	7

1.1 Command List

1.1.1 General

!del

Deletes your previous message. Searches up to the previous 500 messages in the channel.

Usage

`!del`

!google

Generates a Google search link for a keyword. For lazy people like me.

Usage

- `!google <keyword>`
-

!hello

cakebot says hello! Use to check if cakebot is online.

Usage

- `!hello`
-

!bye

Turn off cakebot.

Usage

- `!bye`
-

!invite

Generates a link to invite cakebot to your server

Usage

- `!invite`
-

!redirect

Redirects a message to another channel.

Usage

- `!redirect <channel mention> <message>`

Examples

- Redirects message to #alt: `!redirect #alt Hi guys, from the main channel!`
-

!say

Says a message in another channel. Requires *owner* permission.

Usage

- `!say <channel mention> <message>`
-

Examples

- Redirects message to #alt: `!say #alt Hi guys!`
-

1.1.2 Music

!musicprefix

Sets the prefix for queuing music for your server's music bot.

Usage

- `!musicprefix` - displays the current music prefix set for the server
- `!musicprefix <prefix>` - sets the music prefix to `<prefix>`. Requires `manage_guild` or `musicprefix` permission.

The prefix can consist of multiple words.

Examples

- Set music prefix to `~play`: `!musicprefix ~play`
 - Set music prefix to `! lm` `play`: `!musicprefix ! lm play`
-

!play

Queues music using the musicprefix for the channel (check with `!musicprefix`)

Usage

- `!play <keyword/title/alias>`

Not case sensitive. If multiple matches are found, cakebot will display 13 possible matches and prompt the user to `!playid`.

Variants

- `!play` - Queues a song by name or alias
- `!playid` - Queues a song by id
- `!playalbum` - Queues all the songs in an album

Examples

- Will have multiple matches: `!play snow`
 - Exact match, play song: `!play sound of silence`
 - Exact match for alias: `!play haifuriop`
-

!playalbum

Queues an entire album - variant of `!play`. The songs to be queued are displayed in the same format as `!search` - use `!page <number>` to examine. If the songs to be queued are correct, use `!yes` to confirm and queue.

Usage

- `!playalbum <name/keyword>`

Name/keyword is not case sensitive.

Examples

- Play album named *snow halation*: `!playalbum snow halation`
-

!playid

Queues a song by id - variant of `!play`

Usage

- `!playid <id number>`

A song's id can be found with `!search`

Examples

- Play song with id 316: `!playid 316`
-

!search

Searches the song database for a song with a matching alias/song/artist/album name.

Usage

- `!search <keyword>`

Displays up to 13 results at a time. Not case sensitive. If there are more than 13 results, use `!page <number>` to access the required page.

Examples

- Search for songs with the keyword snow: `!search snow`
-

!reqsong

Shows links to forms for requesting songs to be added to the database.

Usage

- `!reqsong`
-

1.1.3 Modtools

!logchannel

Gets or sets the channel for logging messages.

Usage

- `!logchannel` - displays the current channel for logging messages
 - `!logchannel set` - sets the current channel as the logging channel. Requires *manage_guild* or *logchannel* permission.
-

!purge

Purges a given amount of messages from the current channel. Can specify a user to purge only their messages.

Usage

- `!purge <number>` - purges `<number>` of messages in the current channel. Requires *manage_guild* permission.
- `!purge <user mention> <number>` - purges `<number>` of messages by `<user mention>` within the last 500 messages. Max `<number>` is 100. Requires *manage_guild* permission.

Examples

- Purge last 5 messages: `!purge 5`
 - Purge Clyde's last 10 messages: `!purge @Clyde#1234 10`
-

1.1.4 Permissions

`!permissions`

Gets or sets the cakebot permissions for a given user. This does **NOT** set server permissions, but rather permissions for cakebot commands.

Permissions are required for: `* !musicprefix (set) * !permissions (set) * !logchannel (set)`

Usage

- `!permissions` - displays your current cakebot permissions
- `!permissions <user mention>` - displays current cakebot permissions for the mentioned user
- `permissions <user mention> <command|commands>` - add permissions for the given user. Requires `manage_guild` permission.

Examples

- Give Clyde musicprefix permissions: `!permissions @Clyde#1234 musicprefix`
 - Give Clyde musicprefix and logchannel permissions; `!permissions @Clyde#1234 musicprefix logchannel`
-

1.1.5 Miscellaneous

`!timedcats`

Sends random cat images in timed intervals :3

Usage

- `!timedcats <number> <interval>`

The interval can be m (minutes) or h (hours). Default number and interval is 5 m.

Examples

- Send cat images every minute for 3 minutes: `!timedcats 3 m`
- Send cat images every hour for 10 hours: `!timedcats 10 h`

!trollurl

Replaces characters in a URL to make a similar looking one.

Usage * `!trollurl <url>`

Examples

- Troll a Google link: `!trollurl https://www.google.com`

1.2 Developing Modules

Custom modules are all built off the `ModuleInterface`, which specifies two main things:

1. Common state that must be provided by your module to function properly (e.g. `command_handlers` and `help_entries`), and;
2. Methods that each module is provided access to (e.g. `say()` and `auth_function()`).

`command_handlers` provides a mapping between a command said to cakebot in Discord, and a function which handles the command. For example, in the sample module below, the command `!foo` is mapped to the asynchronous function `foo()`.

1.2.1 Sample Custom Module

Here we define a simple custom module with a single command: `!foo`, along with its corresponding help entry file.

In `mymodule_help.py`:

```
from modules.HelpEntry import HelpEntry

_foo_desc = 'cakebot says foo! '
_foo_usage = '!foo'

help_entries = {
    'foo': HelpEntry('!foo', _foo_desc, _foo_usage, 'miscellaneous')
}
```

In `MyModule.py`:

```
from modules.ModuleInterface import ModuleInterface

class MyModule(ModuleInterface):
    async def foo(self, message):
        await self.say(message.channel, "Foo")
```

(continues on next page)

(continued from previous page)

```
command_handlers = {
    "!foo": foo
}

help_entries = mymodule_help.help_entries
```

1.2.2 Loading Custom Modules into cakebot

1. Add the import for your module into `Bot.py`
2. Add the module into `Bot._modules`, with a name to identify it for loading. Note this identifier cannot be a duplicate of any existing identifiers.
3. Load the module in `run.py` using `bot.load_module(identifier)`, with the identifier you chose above.

In `Bot.py`:

```
from modules.core.Core import Core
# Other module imports...
from modules.mymodule.MyModule import MyModule

class Bot:
    _modules = {
        'core': Core,
        # Other modules...
        'mymodule': MyModule
    }
```

In `run.py`:

```
bot = Bot(client, logger)
bot.load_module('core')
# Other module loaders...
bot.load_module('mymodule')
```