
FJBot-DiscordBot

Release 0.0.1

Sep 30, 2020

Contents:

1	Installing FJBot	1
1.1	Clone Repository	1
1.2	Install Prerequisites	1
1.3	Setup Configs	2
1.4	Initialize Database	2
2	Setting up auto-restart using screen (Unix)	3
2.1	How it works	3
2.2	Bash script	3
2.3	Cronjob	4
3	Admin Cog Reference	5
3.1	What it does	5
3.2	Commands	5
4	Chatango Cog Reference	7
4.1	What it does	7
4.2	Commands (Discord)	7
4.3	Commands (Chatango)	7
5	FJBucks Cog Reference	9
5.1	What it does	9
5.2	Commands	9
6	Matches Cog Reference	11
6.1	What it does	11
6.2	Commands	11
7	Member Cog Reference	13
7.1	What it does	13
7.2	Commands	13
8	Scheduler Cog Reference	15
8.1	What it does	15
8.2	Command Examples	15
9	Twitter Cog Reference	17
9.1	What it does	17

9.2	Commands	17
10	Voice Cog Reference	19
10.1	What it does	19
10.2	Commands	19
11	Bot	21
12	Cogs	23
12.1	Admin	23
12.2	Chatango	24
12.3	FJBucks	26
12.4	Matches	26
12.5	Member	27
12.6	Scheduler	27
12.7	Twitter	28
12.8	Voice	29
13	Config	31
14	Utils	33
14.1	Checks	33
14.2	FJClasses	33
14.3	QuickEmbed	42
15	README	45
15.1	FJBot-DiscordBot	45
16	License	49
17	About Me	63
18	Indices and tables	65
	Python Module Index	67
	Index	69

CHAPTER 1

Installing FJBot

These are the steps needed to run your own instance of **FJBot**.

Important: If you're only looking to invite the bot to your Discord server. Use the following link: https://discordapp.com/oauth2/authorize?&client_id=364938585675137035&scope=bot&permissions=199680

1.1 Clone Repository

This downloads a copy of the [FJBot GitHub repository](#). This is the source code for the bot. You can go in to read, update, or add your own code if you'd like since you would be running your own instance.

```
$ git clone https://github.com/FancyJesse/fjbot.git
```

1.2 Install Prerequisites

This bot uses the following Python packages:

- discord.py
- mysqlclient
- youtube_dl
- tweepy

Listing 1: Installs all the above packages at once

```
$ pip install discord.py[voice] mysqlclient tweepy
```

1.3 Setup Configs

The `Config` module contains the access tokens to Discord and credentials to your database. It also has the list of `cogs` that should load upon startup.

See `Config` and just fill in the proper values.

Important:

- You must have a Discord Bot account.
 - The Twitter bot account is not required.
 - The Chatango bot uses a regular account.
-

1.4 Initialize Database

This script creates the schema and tables required for saving user information. Without this, a majority of the **FJBot**'s functionality would not work, or the bot won't start at all.

Important: TODO: I haven't created this yet.. sorry.

Setting up auto-restart using screen (Unix)

This is a quick and ugly way of ensuring the bot is running and restarts if it is not. I'm sure there are better ways of handling this out there, but this method has worked for me for over a year already.

Note: This method uses a cronjob and `screen`.

Listing 1: How to install screen

```
$ apt-get install screen
```

Listing 2: How to access cronjob list

```
$ crontab -e
```

2.1 How it works

1. A bash script is created that starts the bot in a named *screen*.
2. A cronjob is created to run the bash script every minute under the same name.
 - If the *screen* name exists, that means the bot is already running.

That's it.

2.2 Bash script

Listing 3: discordbot.sh

```
#!/bin/bash
d=`date '+%Y-%m-%d %H:%M:%S'`
if ! screen -ls | grep -q "fjbot"; then
    echo "$d: Restarting DiscordBot"
    screen -X -S fjbot quit
    screen -d -m -S fjbot python3 ./fjbot/bot.py
else
    echo "$d: Already Running"
fi
```

2.3 Cronjob

This creates the job that runs the script above every minute. Temporary logs are made to ensure they are working.

Listing 4: crontab

```
* * * * * /path/to/discordbot.sh >> /tmp/discordbot.log 2>&1 &
```


3.1 What it does

This cog provides commands that moderate users in the Server.

3.2 Commands

```
!update-prefix [character]
```

```
!clear [amount]
```

```
!say [message]
```

```
!spam
```

```
!playing [something]
```

```
!streaming [link]
```

```
!watching [something]
```

```
!listening [something]
```

```
!add-command [command] [response]
```

```
!update-command [command] [updated response]
```

```
!mute [@member]
```

```
!unmute [@member]
```

Chatango Cog Reference

4.1 What it does

This cog provides commands for managing communications with Chatango.

4.2 Commands (Discord)

Listing 1: this detail

```
!chatango-send
```

```
!chatango-users
```

4.3 Commands (Chatango)

Important: The following commands only work for the Chatango website. These commands are handled by a different bot created by this Discord bot.

All of these commands are already available under the Matches cog for Discord users.

Note: Chatango users also have access to the quick command responses.

```
!help
```

```
!register
```

```
!login
```

```
!resetpw
```

```
!stats
```

```
!rate [rating]
```

```
!bet
```

```
!rumble
```

```
!matches
```

5.1 What it does

This cog provides command for an economy.

5.2 Commands

```
!wallet
```

```
!donate [@member]
```

Matches Cog Reference

6.1 What it does

This cog provides commands that interact with a user's [Matches](#) account.

6.2 Commands

```
!current-match
```

```
!last-match
```

```
!events
```

```
!info [superstar_name]
```

```
!birthdays
```

```
!top
```

```
!bal
```

```
!bets
```

```
!match [match_id]
```

```
!matches
```

```
!bet [points] [superstar_name]
```

```
!rate [rating]
```


7.1 What it does

This cog provides every person in the Server basic command.

7.2 Commands

```
!id
```

```
!register
```

```
!login
```

```
!reset-password
```

```
!commands
```

```
!report
```

```
!joined
```

```
!invite
```

```
!roles
```

```
!uptime
```

```
!countdown
```

```
!flip
```

```
!roll
```

```
!mock [@member]
```

```
!slap [@member]
```

```
!tickle [@member]
```

```
!hug [@member]
```

```
!punch [@member]
```

Scheduler Cog Reference

8.1 What it does

This cog handles Discord alert messages.

8.2 Command Examples

```
!scheduler
```


9.1 What it does

This cog provides lookup and quick searches for Twitter accounts.

9.2 Commands

```
!send-tweet [message]
```

```
!tweets [superstar_name]
```


10.1 What it does

This cog provides commands for voice chat audio streaming.

10.2 Commands

```
!music [url]
```

```
!music-pause
```

```
!music-resume
```

```
!music-stop
```

```
!music-volume [number]
```

```
!music-join
```

```
!music-local [path]
```

```
!music-download [url]
```


The main module that runs the bot.

Contains owner commands to load and unload cogs, basic logging, and command prefix checks.

`bot.discord_log` (*content=None, embed=None*)
Sends message to log channel defined in `config`.

Parameters

- **content** – The plain text to send.
- **embed** – The rich embed for the content to send.

`bot.log` (*content=None, embed=None*)
The non-async version of `bot.discord_log()`.

Parameters

- **content** – The plain text to send.
- **embed** – The rich embed for the content to send.

`bot.message_owner` (*content=None, embed=None*)
The non-async version of `bot.pm_owner()`.

Parameters

- **content** – The plain text to send.
- **embed** – The rich embed for the content to send.

`bot.on_command_error` (*ctx, error*)
Called when an error through a command occurs. Common instances are handled appropriately.
If error is not handled, the error is raised.

Parameters

- **ctx** – The invocation context.
- **error** – The command error.

`bot.on_message(ctx)`

Called whenever a message is sent to a server the bot is watching.

The message is first checked to see if a `bot.prefix()` is used. If it is, it will check the `utils.fjclasses.chatroom_command()` for the proper response. If no response is found, `process_command()` is called.

Parameters `ctx` – The invocation context.

`bot.on_ready()`

Called when the client is done preparing the data received from Discord.

Stores the datetime the bot started as an attribute.

`bot.pm_owner(content=None, embed=None)`

Sends private message to owner defined in `config`.

Parameters

- **content** – The plain text to send.
- **embed** – The rich embed for the content to send.

`bot.prefix(bot, ctx)`

Returns the command prefix defined by a guild from database.

If it does not exist, it inserts record with default command prefix defined in `config`.

Parameters

- **bot** – The Discord Bot.
- **ctx** – The invocation context.

Returns The guild's command prefix.

Important: There is full documentation available for all of these cogs, but I can't get them to import through Sphinx. I have spent too much time trying to learn about decorators and docstrings with no avail.

Please check the source files for the full documentation. If you can help with this, please reach out to me.

12.1 Admin

This cog provides administrative commands to Discord users with elevated privileges.

```
class cogs.admin.Admin (bot)
    The Admin cog class.

    add_discord_command = <discord.ext.commands.core.Command object>
    delete_messages = <discord.ext.commands.core.Command object>
    delete_spam_messages = <discord.ext.commands.core.Command object>
    mute_member = <discord.ext.commands.core.Command object>
    repeat_message = <discord.ext.commands.core.Command object>
    unmute_member = <discord.ext.commands.core.Command object>
    update_discord_command = <discord.ext.commands.core.Command object>
    update_guild_prefix = <discord.ext.commands.core.Command object>
    update_presence_listening = <discord.ext.commands.core.Command object>
    update_presence_playing = <discord.ext.commands.core.Command object>
    update_presence_streaming = <discord.ext.commands.core.Command object>
    update_presence_watching = <discord.ext.commands.core.Command object>
```

12.2 Chatango

This cog allows Chatango users to interact with their [Matches](#) account.

class `cogs.chatango.Chatango` (*bot*)

The Chatango cog class.

class `ChBot` (*name=None, password=None, pm=True*)

The Chatango bot. A hacky way of having the Discord bot interact with a different website.

bet_match (*user, args=[]*)

Submits a bet for a *Match*.

Parameters

- **user** – The user submitting the command.
- **args** – The list of arguments received. None are used.

Returns The response message for the user.

command_handler (*user, cmd, args=[]*)

Checks the command and sends it for final processing. A response is sent back to the user.

Parameters

- **user** – The user submitting the command.
- **cmd** – The command to look for.
- **args** – The parameters for the command.

join_rumble (*user, args=[]*)

Provides a quick login link to the website's Royal Rumble page.

Parameters

- **user** – The user submitting the command.
- **args** – The list of arguments received. None are used.

Returns The response message for the user.

login_link (*user*)

Sends a quick login link to the Matches website.

Parameters **user** – The user submitting the command.

Returns The response message for the user.

message_handler (*author, message*)

Reads through the original message and checks to see if a command is requested.

Parameters

- **author** – The author of the message.
- **message** – The message to handle.

onFloodBan (*room*)

Called when the bot receives a ban. Message is logged.

Parameters **room** – The room the ban was received in.

onFloodWarning (*room*)

Called when the bot receives a warning. Message is logged.

Parameters **room** – The room the warning was received in.

onInit ()

Called on init. Required.

onMessage (*room, author, message*)

Called when the bot receives a message in a room. Message is logged and handled.

Parameters

- **room** – The room the message was sent in.
- **author** – The author of the message.
- **message** – The message received.

onPMMessage (*pm, author, message*)

Called when the bot receives a private message. Message is logged and handled.

Parameters

- **pm** – The room the message was sent in.
- **author** – The author of the message.
- **message** – The message received.

open_matches (*user*)

Sends a list of open-bet Matches.

Note: A short-view in plain text is sent, as Chatango does not support rich text.

Parameters **user** – The user submitting the command.

Returns The response message for the user.

rate_match (*user, args=[]*)

Adds the user's rating to the most recently closed *Match*.

Parameters

- **user** – The user submitting the command.
- **args** – The list of arguments received. Only the first value is checked. Must cast to type *float*.

Returns The response message for the user.

register (*user*)

Registers the user.

Note: The user is registered using their Chatango name. If a user already exists with the same name, the registration will fail. This would have to be resolved manually.

Parameters **user** – The user to register.

Returns The response message for the user.

reset_pw (*user*)

Sends a quick reset password link for the Matches website.

Parameters **user** – The user submitting the command.

Returns The response message for the user.

sendRoomMessage (*room, msg*)

Sends a message to a room.

Parameters

- **room** – The room to send the message to.
- **msg** – The message received.

Returns *True* if message was sent, *False* otherwise.

sendUserMessage (*user, msg*)

Sends a message to a specific user through PM.

Parameters

- **user** – The user to send the message to.
- **msg** – The message to send.

chatango_bot_task ()

Creates a separate thread for the Chatango bot.

Note: A thread is required as the Chatango bot is not asynchronous.

chatango_log_task ()

Cycles through the Chatango bot's message buffer and sends content to a defined Discord channel.

display_users = <discord.ext.commands.core.Command object>

send_message = <discord.ext.commands.core.Command object>

start_chbot ()

Initializes and start the Chatango bot.

wait_until_chbot_running (limit=30)

Checks to see if the Chatango bot is running.

Parameters **limit** – The amount of times to check before failing. Default is 30.

Returns *True* if the Chatango bot is running, *False* otherwise

12.3 FJBucks

This cog provides a useless economy.

class `cogs.fjbucks.FJBucks (bot)`

The FJBucks cog class.

donate_bucks = <discord.ext.commands.core.Command object>

fjbucks_balance = <discord.ext.commands.core.Command object>

12.4 Matches

This cog allows users to interact with their [Matches](#) account.

class `cogs.matches.Matches (bot)`

The Matches cog class.

current_champions = <discord.ext.commands.core.Command object>

current_match_info = <discord.ext.commands.core.Command object>

leaderboard_season1 = <discord.ext.commands.core.Command object>

leaderboard_season2 = <discord.ext.commands.core.Command object>

leaderboard_season3 = <discord.ext.commands.core.Command object>

match_info = <discord.ext.commands.core.Command object>

open_matches = <discord.ext.commands.core.Command object>

place_match_bet = <discord.ext.commands.core.Command object>

rate_match = <discord.ext.commands.core.Command object>

recent_match_info = <discord.ext.commands.core.Command object>

royalrumble_info = <discord.ext.commands.core.Command object>

superstar_birthdays = <discord.ext.commands.core.Command object>

superstar_info = <discord.ext.commands.core.Command object>

upcoming_events = <discord.ext.commands.core.Command object>

```

user_current_bets = <discord.ext.commands.core.Command object>
user_join_royalrumble = <discord.ext.commands.core.Command object>
user_stats_season1 = <discord.ext.commands.core.Command object>
user_stats_season2 = <discord.ext.commands.core.Command object>
user_stats_season3 = <discord.ext.commands.core.Command object>

```

12.5 Member

This cog provides basic bot interactions to Discord members.

```

class cogs.member.Member(bot)
    The Member class for the cog.

    countdown = <discord.ext.commands.core.Command object>
    flip_coin = <discord.ext.commands.core.Command object>
    hug_member = <discord.ext.commands.core.Command object>
    invite = <discord.ext.commands.core.Command object>
    joined = <discord.ext.commands.core.Command object>
    member_roles = <discord.ext.commands.core.Command object>
    misc_commands = <discord.ext.commands.core.Command object>
    mock_member = <discord.ext.commands.core.Command object>
    punch_member = <discord.ext.commands.core.Command object>
    register_user = <discord.ext.commands.core.Command object>
    report = <discord.ext.commands.core.Command object>
    roll_dice = <discord.ext.commands.core.Command object>
    send_discord_id = <discord.ext.commands.core.Command object>
    slap_member = <discord.ext.commands.core.Command object>
    tickle_member = <discord.ext.commands.core.Command object>
    uptime = <discord.ext.commands.core.Command object>
    user_login_link = <discord.ext.commands.core.Command object>
    user_reset_password_link = <discord.ext.commands.core.Command object>

```

12.6 Scheduler

This cog handles scheduled messages to a channel.

```

class cogs.scheduler.Scheduler(bot)
    The Scheduler cog class.

    check_scheduler
        A background task helper that abstracts the loop and reconnection logic for you.

        The main interface to create this is through loop().

```

`scheduler_pending = <discord.ext.commands.core.Command object>`

`scheduler_task (payload)`

Handles a single weekly scheduled message by creating the message and sleeping until the appropriate time defined in the message.

Important: The channel the message is sent to is defined in `config`. Modify where appropriate for your own Discord server.

Parameters `payload` – The event details to deliver. See `utils.fjclasses.DbHelper.chatroom_scheduler_list()`.

`showtime_schedule_task ()`

Retrieves the closest event from the database and creates a scheduled message to post to a channel.

Important: The channel the message is sent to is defined in `config`. Modify where appropriate for your own Discord server.

12.7 Twitter

This cog communicates with Twitter.

`class cogs.twitter.Twitter (bot)`

The Twitter cog class.

`class MyStreamListener (api=None)`

A streaming class that listens to live Twitter activity.

Important: This is a work in progress. Twitter has updated their API and Tweepy support hasn't caught up.

`on_data (data)`

Called when a Twitter activity occurs.

Parameters `data` – The Twitter activity.

Returns `True` by default.

`on_error (status)`

Called when a general error occurs.

Parameters `status` – The error status.

`latest_tweets (twitter_id, count=1)`

Fetches the latest tweets from a Twitter user's timeline.

Parameters

- `twitter_id` – The Twitter account's id to get the timeline from.
- `count` – The amount of tweets to get from the timeline. Default is 1.

Returns A list of tweets.

`live_tweet (msg)`

Tweets out a message.

Parameters `msg` – The message to tweet out.

Returns A hyperlink to the tweet after it's posted.

`send_tweet = <discord.ext.commands.core.Command object>`

`stream_test ()`

Testing the listener.

`superstar_birthday_task ()`

Sends our a Tweet when it is a *Superstars* birthday.

`superstar_tweets = <discord.ext.commands.core.Command object>`

`tweet_log (message)`

Sends a message to a Discord channel.

Important: The channel the message is sent to is defined in `config`. Modify where appropriate for your own Discord server.

Parameters `message` –

12.8 Voice

This cog plays audio in voice channels on request.

`class cogs.voice.YTDLSource (source, *, data, volume=0.5)`

Creates a player from a YouTube URL.

Note: Based off the [discord voice example](#).

`classmethod from_url (url, *, loop=None, stream=False)`

Prepares the audio from the URL.

Note: If the URL points to a playlist, it takes the first item from the list.

Parameters

- `url` – The YouTube URL for the audio to play.
- `loop` – The event loop to use. Default is *None*.
- `stream` – Whether to download or stream the source. Default is *False*.

Returns The audio player.

`class cogs.voice.Voice (bot)`

The Voice cog class.

`change_volume = <discord.ext.commands.core.Command object>`

`ensure_voice (ctx)`

Checks if the voice channel can be started.

Note:

- The requester must be in a voice channel.
 - Stops a current player if it currently exists.
-

Parameters `ctx` – The invocation context.

```
join = <discord.ext.commands.core.Command object>
pause_audio = <discord.ext.commands.core.Command object>
play_local = <discord.ext.commands.core.Command object>
play_yt = <discord.ext.commands.core.Command object>
resume_audio = <discord.ext.commands.core.Command object>
stop_audio = <discord.ext.commands.core.Command object>
stream_yt = <discord.ext.commands.core.Command object>
```

CHAPTER 13

Config

```
"""This module contains all account and database credentials."""

#####
# General Bot Author Information #
# Be sure to update this if      #
# you create own instance.      #
#####
base = {
    'description': 'FJBot is a Discord Bot written in Python by FancyJesse',
    'default_prefix': '!', # The default command prefix
    'invite': { # The invite links for the guild and the bot
        'guild': 'https://discord.gg/Q9mX5hQ',
        'bot': 'https://discordapp.com/oauth2/authorize?&client_id=364938585675137035&
↪scope=bot&permissions=199680',
    },
    'owner_id': 192077042722799616, # The author's Discord ID
    'guild_id': 361689774723170304, # The originating Discord Server ID
    'startup_cogs': [ # List of cogs to load on startup
        'cogs.admin',
        'cogs.member',
        'cogs.scheduler',
        'cogs.matches',
        'cogs.chatango',
        'cogs.voice',
        'cogs.twitter',
        'cogs.fjbucks',
    ],
    'channel': { # Dictionary of Discord Server Channels - Must update IDs
        'log': 0, # This is a custom channel used by the bot to log messages
        'general': 0, # The default channel when first creating a Server
        'voice': 0, # The general voice channel
        'chatango': 0, # This is a custom channel used by Chatango cog
        'twitter': 0, # This is a custom channel used by Twitter cog
        'ppv': 0, # This is a custom channel used by Matches cog
    }
}
```

(continues on next page)

(continued from previous page)

```

        'wwe': 0, # This is a custom channel used by Matches cog
        'aew': 0, # This is a custom channel used by Matches cog
    },
}

#####
# Discord Bot Credentials #
#####
discord = {
    'access_token': '',
}

#####
# MySQL Database Credentials #
#####
mysql = {
    'host': '', # The host of the database
    'db': '', # The database/schema name
    'user': '', # The database user to login with (should have write permissions)
    'secret': '', # The database user's password
}

#####
# Chatango Bot Credentials and Rooms #
#####
chatango = {
    'username': '', # The Chatango account username
    'secret': '', # The Chatango account password
    'rooms': [], # The list of rooms to have the bot join
}

#####
# Twitter Bot Credentials #
#####
twitter = {
    'consumer_key': '',
    'consumer_secret': '',
    'access_token': '',
    'access_token_secret': '',
}

#####
# Logging Settings #
#####
logging = {
    'debug': 0, # Levels - 0=Info, 1=Debug
    'format': '%(asctime)s - %(levelname)s - %(name)s - %(message)s', # The logging_
↪format
}

```

14.1 Checks

This module provides basic checks to common procedures.

`utils.checks.confirm(author)`

Parameters `author` – The `ctx.author` attribute value.

Returns `True` if the author matches the context and properly responds, `False` otherwise.

`utils.checks.is_dm(ctx)`

Check to see if the message received is a direct message.

Parameters `ctx` – The invocation context.

Returns `True` if the context is a direct message, `False` otherwise.

`utils.checks.is_number(author)`

Parameters `author` – The `ctx.author` attribute value.

Returns `True` if the author matches the context and properly responds, `False` otherwise.

`utils.checks.is_registered()`

Check to see if the user is registered in database.

Returns `True` if the `utils.fjclasses.DiscordUser` exists.

Raises `utils.fjclasses.UserNotRegisteredError` if User is not found in database.

14.2 FJClasses

This module contains all classes and models used throughout the bot.

14.2.1 `_Abstract`

`_Base`

class `utils.fjclasses._Base` (*id=None, name=None*)

A generic class with only `id` and `name` attribute.

fetch_info ()

Abstract function for fetching object's attribute values from database.

fill_info (*row*)

Abstract function for applying object's attribute values based on query result.

Parameters `row` – The SQL result.

id

The object's unique identifier value.

Type return

name

The name of the object.

Type return

`_Database`

class `utils.fjclasses._Database`

A generic database handler.

close ()

Closes the database connection and cursor. Fails gracefully.

connection

The database connection.

Type return

cursor

The database cursor.

Type return

db

The database object (itself).

Type return

execute (*sql, params=None*)

Executes query with or without params.

Parameters

- **sql** – The SQL query.
- **params** – The parameters for the query.

fetchall ()

Grabs all values from query result.

Returns All query results.

fetchone()

Grabs single value from query result.

Returns A single query result.

query (*sql*, *params=None*)

Executes a query and returns all results in a single call.

Parameters

- **sql** – The SQL query.
- **params** – The parameters for the query.

Returns All query results.

_User**class** `utils.fjclasses._User`

A generic class used by different user types.

current_bets ()

Return the list of currently placed bets.

Returns Query result of currently placed bets. Error message if none.

fetch_info ()

Abstract function to fetch the user info.

fill_info (*row*)

Applies attributes based on query result.

Parameters **row** – The query result.

fjbucks_check (*amount*)

Checks whether the FJBucks transaction is valid to make.

Parameters **amount** – The amount of FJBucks for the transaction.

Returns Query result with success or failure message.

fjbucks_transaction (*amount*, *note*)

Calls stored procedure to create a FJBucks transaction.

Parameters

- **amount** – The amount of FJBucks for the transaction.
- **note** – A memo for the transaction.

Returns Query result with success or failure message.

fjbucks_wallet ()

Returns The user's FJBucks wallet info.

is_registered ()

Checks if the user is registered.

Returns *True* if id and username values exist, *False* otherwise.

join_royalrumble ()

TODO

mention

Abstract function to notify the user inside the chat.

place_bet (*match_id, team, bet*)

Places a bet for match.

Parameters

- **match_id** – The Match id to bet for.
- **team** – The Match team id to bet for.
- **bet** – The amount betting.

Returns Query result with success or failure message.

rate_match (*match_id, rating*)

Places a rating for a match.

Parameters

- **match_id** – The Match id to rate for.
- **rating** – The rating for the match.

Returns Query result with success or failure message.

refresh ()

Re-initializes the user's attribute values.

register ()

Abstract function to register the user.

request_login_link ()

Creates hyperlink to auto-login user to the website.

This creates a randomized token that allows the user to bypass initial login screen.

Important: Do not post this link in a public chat. It must be DMed to the user requesting it only. Once the hyperlink can only be used once and within a short time frame.

Returns The hyperlink to auto-login a user to the website.

request_reset_password_link ()

Creates hyperlink to reset the user's password.

Once called a temporary password is set for the user to bypass their original password. This is required for users that have not yet set up a password for their account; users that registered through the bot and not the website.

Important: Do not post this link in a public chat. It must be DMed to the user requesting it only. Once the hyperlink can only be used once and within a short time frame.

Returns The hyperlink to reset the user's password for the website.

royalrumble_info ()

TODO

stats (*season*)

Return user's stats for the Matches season.

Parameters **season** – The Match season to fetch stats for.

Returns The user stats for the Matches season.

username

The user's username.

Type return

validate_bet (*match_id, team, bet*)

Checks to see if user is able to place the bet through a stored procedure.

Parameters

- **match_id** – The Match id to bet for.
- **team** – The Match team id to bet for.
- **bet** – The amount betting.

Returns Query result with success or failure message.

14.2.2 _Exceptions

GuildNotOriginError

class `utils.fjclasses.GuildNotOriginError` (*message=None, *args*)

Custom exception class. Thrown when command requires origin guild.

UserNotRegisteredError

class `utils.fjclasses.UserNotRegisteredError` (*message=None, *args*)

Custom exception class. Thrown when command requires registered user.

14.2.3 ChatangoUser

class `utils.fjclasses.ChatangoUser` (*author*)

This class is used to refer to individual Chatango authors.

chatango

The author. (itself)

Type return

fetch_info ()

Fetches attribute values from database using current Chatango author unique identifier (name).

Note: If no results are found with id, no attributes are set.

join_royalrumble ()

TODO

mention

The author's mention.

Type return

name

The author's name.

Type return

register ()

Inserts the author as a User in the database using Discord profile.

Returns Query result with success or failure message.

royalrumble_info ()

TODO

stats_text (*season*)

Formats the user's stats for the season as plain text.

Parameters **season** – The season id.

Returns Formatted text message.

14.2.4 DbHelper

class `utils.fjclasses.DbHelper`

This class handles the most common and generic queries.

add_chatroom_command (*command*, *response*)

TODO

Parameters

- **command** – The chatroom command to insert.
- **response** – The response for the chatroom command.

chatroom_command (*command*)

Fetches response to a quick chatroom command.

Parameters **command** – The command to fetch the response for.

Returns The query result if found, *False* otherwise.

chatroom_command_list ()

Fetches the complete list of available chatroom commands.

Returns The query result of chatroom commands ordered by alpha.

chatroom_scheduler_list ()

Fetches list of weekly alert scheduler.

Returns The query result for the weekly schedule.

future_events (*ppv_check=0*)

Fetches list of upcoming PPV events.

Parameters **ppv_check** – The PPV flag check. Default is 0.

Returns The query result of upcoming events.

guild_info (*guild_id*)

Fetches Discord guild information by id.

Parameters **guild_id** – The guild id.

Returns The command prefix for the guild, *False* otherwise.

leaderboard (*season*)

Fetches the leaderboard for the season.

Parameters **season** – The season id.

Returns Query result of User stats ordered by total points for the season.

search_match_by_current ()

Fetches info for the current Match.

Returns A list of :class:'utils.fjclasses._Base'.

search_match_by_id (*id*)

Fetches Match info by id.

Parameters *id* – The Match id.

Returns A list of :class:'utils.fjclasses._Base'.

search_match_by_open_bets ()

Fetches info for Matches that are available to bet on.

Returns A list of :class:'utils.fjclasses._Base'.

search_match_by_open_bets_and_superstar_name (*name*)

Fetches info for Matches that are available to bet on and contain the Superstar.

Parameters *name* – The Superstar name.

Returns A list of :class:'utils.fjclasses._Base'.

search_match_by_recent_completed ()

Fetches info for the most recently completed Match.

Returns A list of :class:'utils.fjclasses._Base'.

search_superstar_by_name (*name*)

Fetches Superstar info by name.

Parameters *name* – The Superstar's name to query for.

Returns A list of :class:'utils.fjclasses._Base'.

search_user_by_name (*name*)

Fetches User info by name.

Parameters *name* – The User's name to query for.

Returns A list of :class:'utils.fjclasses._Base'.

superstar_birthday_upcoming ()

Fetches Superstar info who's birthday is coming up.

Returns The query result of Superstars.

update_chatroom_command (*command*, *response*)

TODO

Parameters

- **command** – The chatroom command to update.
- **response** – The updated response for the chatroom command.

update_guild_info (*guild*, *prefix*)

Updates the Discord guild's information and command prefix.

Note: If an entry for the guild does not exist, a new entry for the guild is inserted.

Parameters

- **guild** – The guild object.
- **prefix** – The guild’s command prefix to update to.

Returns TODO

14.2.5 DiscordUser

class `utils.fjclasses.DiscordUser` (*author*)

This class is used to refer to individual Discord authors.

discord

The author object. (itself)

Type return

fetch_info ()

Fetches attribute values from database using current Discord author unique identifier (id).

Note: If no results are found with id, no attributes are set.

fjbucks_wallet_embed ()

Creates rich content of the User’s FJBucks wallet.

Returns Rich content.

join_royalrumble ()

TODO

mention

The author’s mention value.

Type return

name

The author’s name value.

Type return

register ()

Inserts the author as a User in the database using Discord profile.

Returns Query result with success or failure message.

royalrumble_info ()

TODO

stats_embed (*season*)

Creates rich content of the User’s stats for the season.

Parameters **season** – The season id.

Returns Rich content.

14.2.6 Match

class `utils.fjclasses.Match` (*id=None*)

The class is used to refer to individual Matches.

contains_contestant (*name*)

Checks if Superstar is a Match contestant.

Parameters **name** – The Superstar name.

Returns *True* if the Superstar is a contestant, *False* otherwise.

contestants_by_team (*team_id*)

Returns the list of Superstars in the Match based on the team id.

Parameters **team_id** – The team id.

Returns The list Superstars in the team, *False* otherwise.

fetch_info ()

Fetches attribute values from database using current Match unique identifier (*id*).

Note: If no results are found with *id*, no attributes are set.

fill_info (*row*)

Applies attributes based on query result.

Parameters **row** – The query result.

info_embed ()

Creates rich content of the Match's info.

Returns Rich content.

info_text ()

Formatted text of a Match's info. (long version)

Returns Formatted text.

info_text_short ()

Formatted text of a Match's info. (short version)

Returns Formatted text.

set_teams (*rows*)

Initializes teams attribute with query result.

Parameters **rows** – The list of contestants in the Match.

team_by_contestant (*name*)

Returns the team id of Superstar in the Match based on the Superstar's name.

Parameters **name** – The Superstar's name.

Returns The Superstar's team, *False* otherwise.

14.2.7 Superstar

class `utils.fjclasses.Superstar` (*id=None*)

This class is used to refer to individual Superstars.

fetch_info ()

Fetches attribute values from database using current Chatango author unique identifier (*id*).

Note: If no results are found with *id*, no attributes are set.

`fill_info(row)`

Applies attributes based on query result.

Parameters `row` – The query result.

`info_embed()`

Creates rich content of the Superstar's info.

Returns Rich content.

14.3 QuickEmbed

This module provides a quick way to personalize messages back to the user.

Instead of flat text messages or going through all the steps to setup and personalize rich content, this module is used.

`utils.quickembed.error(desc, user=None)`

Styles the rich content for a failed response. Colored red.

Parameters

- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

`utils.quickembed.filler(embed, desc, user)`

Creates the rich content and fills in all `user` info.

Parameters

- **embed** – The base rich content.
- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

`utils.quickembed.general(desc, user=None)`

Styles the rich content for a generic response. Colored blue.

Parameters

- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

`utils.quickembed.info(desc, user=None)`

Styles the rich content for an informative response. Colored white.

Parameters

- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

`utils.quickembed.question(desc, user=None)`

Styles the rich content for a question response. Colored yellow.

Parameters

- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

`utils.quickembed.success(desc, user=None)`

Styles the rich content for a successful response. Colored green.

Parameters

- **desc** – The title of the rich content.
- **user** – The `utils.fjclasses.DiscordUser` responding to.

Returns The personalized rich content.

15.1 FJBOT-DISCORDBOT



A Discord bot with focus on wrestling events and [Matches](#).

My Discord Server: [WatchWrestling](#)

Invite FJBOT to your Discord Server: [FJBOT](#)

15.1.1 Introduction

This Discord Bot was originally created back in 2017 to integrate [Matches](#) into the server. This allows users to bet and rate matches, look up Superstar biographies, share gifs, and be alerted of upcoming wrestling events.

Originally, this bot ran through a single Python file. I had no idea that I would still be supporting the bot this long or even begin scaling it to include additional features and commands. Because of the expansion of the bot, I decided to transition this Discord Bot project to use cogs. Made code management a whole lot easier. Now I continue support on the bot as a hobby and development experience.

15.1.2 Sample

Register an account

```
!register
```

Get a login link via PM

```
!login
```

Display current stats

```
!stats
```

Display current leaderboard

```
!top
```

Display Superstar/Wrestler's info

```
!bio Alexa
```

Display current matches

```
!matches
```

Bet on a match

```
!bet 100 Brock
```

Display current bets

```
!bets
```

many more commands available. see the [documentation](#)

15.1.3 Prerequisites (Development)

- Python 3
- MySQL
- Discord Server

15.1.4 Setup & Deploy (Development)

Clone repository

```
git clone https://github.com/idle-user/fjbot.git
```

Install requirements

```
pip install discord.py[voice] mysqlclient youtube_dl tweepy m2r
```

Create config file. See `config.py`

```
./fjbot/config.py
```

Run MySQL Database setup *There are many tables, procedures, and triggers required to run accounts properly*

TODO

Run bot

```
python3 bot.py
```

15.1.5 Update History

2020-03-19

- Fixed prefix issue on PM
- Updated Start and End bot.py messages

15.1.6 License

This project is licensed under the GNU GPLv3 License - see the [LICENSE](#) file for details

15.1.7 Authors

idle-user

CHAPTER 16

License

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether

(continues on next page)

(continued from previous page)

gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

(continues on next page)

(continued from previous page)

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding

(continues on next page)

Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,

(continues on next page)

(continued from previous page)

and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the

(continues on next page)

(continued from previous page)

product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as

(continues on next page)

(continued from previous page)

part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or

(continues on next page)

(continued from previous page)

authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

(continues on next page)

(continued from previous page)

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

(continues on next page)

(continued from previous page)

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you

(continues on next page)

(continued from previous page)

to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

(continues on next page)

(continued from previous page)

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands

(continues on next page)

(continued from previous page)

might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

CHAPTER 17

About Me

First and foremost, thank you for visiting this page. I have spent countless hours ↵
↵working
on this bot. I've done entire rewrites until I created something I was satisfied with ↵
↵stable
to work on. I also did not document my code until it was too late. Then I had to ↵
↵spend a couple
days on how to properly document to make it compatible with Sphinx. So, again, thank ↵
↵you for spending
time for visiting this documentation.

I first learned how to program in college. There the only language taught was Java, ↵
↵but that was
enough to help me understand syntax and how languages differ from one another. After ↵
↵Java, I learned
C (which I'm sure I forgot how to use). Then I taught myself JavaScript, PHP, SQL and ↵
↵Python - along
with various markup languages.

After learning a handful of languages, I decided to create my own website. Not ↵
↵through a website
builder, but from scratch. I was way too stubborn and lazy to look into website ↵
↵builders and what is
out in the market at the time. I figured the best way to create my website is to open ↵
↵up a file with
`nano` and started building. I finally got to put one of my many RaspberryPis to work.
↵ What came out
of this was: <https://fancyjesse.com>

I did not advertise or promote my website much because everything was always a `work-
↵in-progress`.
It was just something for myself, to prove that I can program and create a website if ↵
↵I wanted to. I
also did not wanted others to get the wrong impression that I can `design` websites. ↵
↵I am not a

(continues on next page)

(continued from previous page)

web-designer at all and have very little interest in learning how to make websites.
↳ look pretty
(thank you bootstrap for existing).

Through the years I somehow got back into watching wrestling. WWE, NJPW, AEW, Indies -
↳ loved all of
it. I would watch PPVs through streaming websites. The website I would watch my
↳ stream in had a
Chatango chatroom on the side. That chat was pretty annoying, full of spam, and very
↳ very VERY
toxic. It came to the point where that regular folks that just wanted to watch the
↳ stream and have
small talk about they were watching were just bombarded with racist and pornographic
↳ gifs. This is
what brought the creation of the WatchWrestling Discord Server: [https://discord.gg/
↳ Q9mX5hQ](https://discord.gg/Q9mX5hQ)

With the addition to the Discord Server, I wanted to learn about user management at
↳ the website level,
so I started what would be my largest web project today, Matches. In short, it lets
↳ you bet and rate
on wrestling matches that are added for points. Each season starts and ends on WWE's
↳ WrestleMania,
and a leaderboard is kept to show the user's rankings and stats. I created various
↳ scripts that
scraped webpages to help populate wrestler data I was storing. After spending time
↳ creating a
database with various tables and procedures that worked with each other, I wanted
↳ other to join the
site. I didn't want to advertise my website because I felt it was crappy and I didn't
↳ want to be
that type of person. So I starting reading about how to create a Discord bot.

The WatchWrestling Discord Server is still alive with a small group of casual members
↳ that have stuck
around through it all. It was because of these members that I stuck around and
↳ decided to create the
Discord bot - again, because I was too lazy and stubborn to use any bot that was out
↳ there. Plus I
really wanted to learn how it all worked. Honestly, it was a fun learning experience,
↳ but there came
many times where I figured I should just throw it all away. I just had to remind
↳ myself that this
should be a fun project and nothing more.

There is still much I need to improve on when it comes to programming. But overall,
↳ this was a great
learning experience. Thank you for your time if you made it this far.

CHAPTER 18

Indices and tables

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

b

bot, 21

c

cogs.admin, 23
cogs.chatango, 24
cogs.fjbucks, 26
cogs.matches, 26
cogs.member, 27
cogs.scheduler, 27
cogs.twitter, 28
cogs.voice, 29

u

utils.checks, 33
utils.fjclasses, 33
utils.quickembed, 42

Symbols

`_Base` (class in `utils.fjclasses`), 34

`_Database` (class in `utils.fjclasses`), 34

`_User` (class in `utils.fjclasses`), 35

A

`add_chatroom_command()`
(`utils.fjclasses.DbHelper` method), 38

`add_discord_command` (`cogs.admin.Admin` attribute), 23

`Admin` (class in `cogs.admin`), 23

B

`bet_match()` (`cogs.chatango.Chatango.ChBot` method), 24

`bot` (module), 21

C

`change_volume` (`cogs.voice.Voice` attribute), 29

`Chatango` (class in `cogs.chatango`), 24

`chatango` (`utils.fjclasses.ChatangoUser` attribute), 37

`Chatango.ChBot` (class in `cogs.chatango`), 24

`chatango_bot_task()` (`cogs.chatango.Chatango` method), 25

`chatango_log_task()` (`cogs.chatango.Chatango` method), 25

`ChatangoUser` (class in `utils.fjclasses`), 37

`chatroom_command()` (`utils.fjclasses.DbHelper` method), 38

`chatroom_command_list()`
(`utils.fjclasses.DbHelper` method), 38

`chatroom_scheduler_list()`
(`utils.fjclasses.DbHelper` method), 38

`check_scheduler` (`cogs.scheduler.Scheduler` attribute), 27

`close()` (`utils.fjclasses._Database` method), 34

`cogs.admin` (module), 23

`cogs.chatango` (module), 24

`cogs.fjbucks` (module), 26

`cogs.matches` (module), 26

`cogs.member` (module), 27

`cogs.scheduler` (module), 27

`cogs.twitter` (module), 28

`cogs.voice` (module), 29

`command_handler()`
(`cogs.chatango.Chatango.ChBot` method), 24

`confirm()` (in module `utils.checks`), 33

`connection` (`utils.fjclasses._Database` attribute), 34

`contains_contestant()` (`utils.fjclasses.Match` method), 40

`contestants_by_team()` (`utils.fjclasses.Match` method), 41

`countdown` (`cogs.member.Member` attribute), 27

`current_bets()` (`utils.fjclasses._User` method), 35

`current_champions` (`cogs.matches.Matches` attribute), 26

`current_match_info` (`cogs.matches.Matches` attribute), 26

`cursor` (`utils.fjclasses._Database` attribute), 34

D

`db` (`utils.fjclasses._Database` attribute), 34

`DbHelper` (class in `utils.fjclasses`), 38

`delete_messages` (`cogs.admin.Admin` attribute), 23

`delete_spam_messages` (`cogs.admin.Admin` attribute), 23

`discord` (`utils.fjclasses.DiscordUser` attribute), 40

`discord_log()` (in module `bot`), 21

`DiscordUser` (class in `utils.fjclasses`), 40

`display_users` (`cogs.chatango.Chatango` attribute), 26

`donate_bucks` (`cogs.fjbucks.FJBucks` attribute), 26

E

`ensure_voice()` (`cogs.voice.Voice` method), 29

`error()` (in module `utils.quickembed`), 42

`execute()` (`utils.fjclasses._Database` method), 34

F

[fetch_info\(\) \(utils.fjclasses._Base method\)](#), 34
[fetch_info\(\) \(utils.fjclasses._User method\)](#), 35
[fetch_info\(\) \(utils.fjclasses.ChatangoUser method\)](#), 37
[fetch_info\(\) \(utils.fjclasses.DiscordUser method\)](#), 40
[fetch_info\(\) \(utils.fjclasses.Match method\)](#), 41
[fetch_info\(\) \(utils.fjclasses.Superstar method\)](#), 41
[fetchall\(\) \(utils.fjclasses._Database method\)](#), 34
[fetchone\(\) \(utils.fjclasses._Database method\)](#), 34
[fill_info\(\) \(utils.fjclasses._Base method\)](#), 34
[fill_info\(\) \(utils.fjclasses._User method\)](#), 35
[fill_info\(\) \(utils.fjclasses.Match method\)](#), 41
[fill_info\(\) \(utils.fjclasses.Superstar method\)](#), 41
[filler\(\) \(in module utils.quickembed\)](#), 42
[FJBucks \(class in cogs.fjbucks\)](#), 26
[fjbucks_balance \(cogs.fjbucks.FJBucks attribute\)](#), 26
[fjbucks_check\(\) \(utils.fjclasses._User method\)](#), 35
[fjbucks_transaction\(\) \(utils.fjclasses._User method\)](#), 35
[fjbucks_wallet\(\) \(utils.fjclasses._User method\)](#), 35
[fjbucks_wallet_embed\(\) \(utils.fjclasses.DiscordUser method\)](#), 40
[flip_coin \(cogs.member.Member attribute\)](#), 27
[from_url\(\) \(cogs.voice.YTDLSource class method\)](#), 29
[future_events\(\) \(utils.fjclasses.DbHelper method\)](#), 38

G

[general\(\) \(in module utils.quickembed\)](#), 42
[guild_info\(\) \(utils.fjclasses.DbHelper method\)](#), 38
[GuildNotOriginError \(class in utils.fjclasses\)](#), 37

H

[hug_member \(cogs.member.Member attribute\)](#), 27

I

[id \(utils.fjclasses._Base attribute\)](#), 34
[info\(\) \(in module utils.quickembed\)](#), 42
[info_embed\(\) \(utils.fjclasses.Match method\)](#), 41
[info_embed\(\) \(utils.fjclasses.Superstar method\)](#), 42
[info_text\(\) \(utils.fjclasses.Match method\)](#), 41
[info_text_short\(\) \(utils.fjclasses.Match method\)](#), 41
[invite \(cogs.member.Member attribute\)](#), 27
[is_dm\(\) \(in module utils.checks\)](#), 33
[is_number\(\) \(in module utils.checks\)](#), 33
[is_registered\(\) \(in module utils.checks\)](#), 33
[is_registered\(\) \(utils.fjclasses._User method\)](#), 35

J

[join \(cogs.voice.Voice attribute\)](#), 30
[join_royalrumble\(\) \(utils.fjclasses._User method\)](#), 35
[join_royalrumble\(\) \(utils.fjclasses.ChatangoUser method\)](#), 37
[join_royalrumble\(\) \(utils.fjclasses.DiscordUser method\)](#), 40
[join_rumble\(\) \(cogs.chatango.Chatango.ChBot method\)](#), 24
[joined \(cogs.member.Member attribute\)](#), 27

L

[latest_tweets\(\) \(cogs.twitter.Twitter method\)](#), 28
[leaderboard\(\) \(utils.fjclasses.DbHelper method\)](#), 38
[leaderboard_season1 \(cogs.matches.Matches attribute\)](#), 26
[leaderboard_season2 \(cogs.matches.Matches attribute\)](#), 26
[leaderboard_season3 \(cogs.matches.Matches attribute\)](#), 26
[live_tweet\(\) \(cogs.twitter.Twitter method\)](#), 28
[log\(\) \(in module bot\)](#), 21
[login_link\(\) \(cogs.chatango.Chatango.ChBot method\)](#), 24

M

[Match \(class in utils.fjclasses\)](#), 40
[match_info \(cogs.matches.Matches attribute\)](#), 26
[Matches \(class in cogs.matches\)](#), 26
[Member \(class in cogs.member\)](#), 27
[member_roles \(cogs.member.Member attribute\)](#), 27
[mention \(utils.fjclasses._User attribute\)](#), 35
[mention \(utils.fjclasses.ChatangoUser attribute\)](#), 37
[mention \(utils.fjclasses.DiscordUser attribute\)](#), 40
[message_handler\(\) \(cogs.chatango.Chatango.ChBot method\)](#), 24
[message_owner\(\) \(in module bot\)](#), 21
[misc_commands \(cogs.member.Member attribute\)](#), 27
[mock_member \(cogs.member.Member attribute\)](#), 27
[mute_member \(cogs.admin.Admin attribute\)](#), 23

N

[name \(utils.fjclasses._Base attribute\)](#), 34
[name \(utils.fjclasses.ChatangoUser attribute\)](#), 37
[name \(utils.fjclasses.DiscordUser attribute\)](#), 40

O

[on_command_error\(\) \(in module bot\)](#), 21
[on_data\(\) \(cogs.twitter.Twitter.MyStreamListener method\)](#), 28

- on_error() (cogs.twitter.Twitter.MyStreamListener method), 28
 on_message() (in module bot), 21
 on_ready() (in module bot), 22
 onFloodBan() (cogs.chatango.Chatango.ChBot method), 24
 onFloodWarning() (cogs.chatango.Chatango.ChBot method), 24
 onInit() (cogs.chatango.Chatango.ChBot method), 24
 onMessage() (cogs.chatango.Chatango.ChBot method), 24
 onPMMessage() (cogs.chatango.Chatango.ChBot method), 24
 open_matches (cogs.matches.Matches attribute), 26
 open_matches() (cogs.chatango.Chatango.ChBot method), 25
- ## P
- pause_audio (cogs.voice.Voice attribute), 30
 place_bet() (utils.fjclasses._User method), 36
 place_match_bet (cogs.matches.Matches attribute), 26
 play_local (cogs.voice.Voice attribute), 30
 play_yt (cogs.voice.Voice attribute), 30
 pm_owner() (in module bot), 22
 prefix() (in module bot), 22
 punch_member (cogs.member.Member attribute), 27
- ## Q
- query() (utils.fjclasses._Database method), 35
 question() (in module utils.quickembed), 42
- ## R
- rate_match (cogs.matches.Matches attribute), 26
 rate_match() (cogs.chatango.Chatango.ChBot method), 25
 rate_match() (utils.fjclasses._User method), 36
 recent_match_info (cogs.matches.Matches attribute), 26
 refresh() (utils.fjclasses._User method), 36
 register() (cogs.chatango.Chatango.ChBot method), 25
 register() (utils.fjclasses._User method), 36
 register() (utils.fjclasses.ChatangoUser method), 38
 register() (utils.fjclasses.DiscordUser method), 40
 register_user (cogs.member.Member attribute), 27
 repeat_message (cogs.admin.Admin attribute), 23
 report (cogs.member.Member attribute), 27
 request_login_link() (utils.fjclasses._User method), 36
 request_reset_password_link() (utils.fjclasses._User method), 36
 reset_pw() (cogs.chatango.Chatango.ChBot method), 25
 resume_audio (cogs.voice.Voice attribute), 30
 roll_dice (cogs.member.Member attribute), 27
 royalrumble_info (cogs.matches.Matches attribute), 26
 royalrumble_info() (utils.fjclasses._User method), 36
 royalrumble_info() (utils.fjclasses.ChatangoUser method), 38
 royalrumble_info() (utils.fjclasses.DiscordUser method), 40
- ## S
- Scheduler (class in cogs.scheduler), 27
 scheduler_pending (cogs.scheduler.Scheduler attribute), 28
 scheduler_task() (cogs.scheduler.Scheduler method), 28
 search_match_by_current() (utils.fjclasses.DbHelper method), 39
 search_match_by_id() (utils.fjclasses.DbHelper method), 39
 search_match_by_open_bets() (utils.fjclasses.DbHelper method), 39
 search_match_by_open_bets_and_superstar_name() (utils.fjclasses.DbHelper method), 39
 search_match_by_recent_completed() (utils.fjclasses.DbHelper method), 39
 search_superstar_by_name() (utils.fjclasses.DbHelper method), 39
 search_user_by_name() (utils.fjclasses.DbHelper method), 39
 send_discord_id (cogs.member.Member attribute), 27
 send_message (cogs.chatango.Chatango attribute), 26
 send_tweet (cogs.twitter.Twitter attribute), 29
 sendRoomMessage() (cogs.chatango.Chatango.ChBot method), 25
 sendUserMessage() (cogs.chatango.Chatango.ChBot method), 25
 set_teams() (utils.fjclasses.Match method), 41
 showtime_schedule_task() (cogs.scheduler.Scheduler method), 28
 slap_member (cogs.member.Member attribute), 27
 start_chbot() (cogs.chatango.Chatango method), 26
 stats() (utils.fjclasses._User method), 36
 stats_embed() (utils.fjclasses.DiscordUser method), 40

stats_text() (*utils.fjclasses.ChatangoUser method*), 38
 stop_audio (*cogs.voice.Voice attribute*), 30
 stream_test() (*cogs.twitter.Twitter method*), 29
 stream_yt (*cogs.voice.Voice attribute*), 30
 success() (*in module utils.quickembed*), 43
 Superstar (*class in utils.fjclasses*), 41
 superstar_birthday_task() (*cogs.twitter.Twitter method*), 29
 superstar_birthday_upcoming() (*utils.fjclasses.DbHelper method*), 39
 superstar_birthdays (*cogs.matches.Matches attribute*), 26
 superstar_info (*cogs.matches.Matches attribute*), 26
 superstar_tweets (*cogs.twitter.Twitter attribute*), 29

T

team_by_contestant() (*utils.fjclasses.Match method*), 41
 tickle_member (*cogs.member.Member attribute*), 27
 tweet_log() (*cogs.twitter.Twitter method*), 29
 Twitter (*class in cogs.twitter*), 28
 Twitter.MyStreamListener (*class in cogs.twitter*), 28

U

unmute_member (*cogs.admin.Admin attribute*), 23
 upcoming_events (*cogs.matches.Matches attribute*), 26
 update_chatroom_command() (*utils.fjclasses.DbHelper method*), 39
 update_discord_command (*cogs.admin.Admin attribute*), 23
 update_guild_info() (*utils.fjclasses.DbHelper method*), 39
 update_guild_prefix (*cogs.admin.Admin attribute*), 23
 update_presence_listening (*cogs.admin.Admin attribute*), 23
 update_presence_playing (*cogs.admin.Admin attribute*), 23
 update_presence_streaming (*cogs.admin.Admin attribute*), 23
 update_presence_watching (*cogs.admin.Admin attribute*), 23
 uptime (*cogs.member.Member attribute*), 27
 user_current_bets (*cogs.matches.Matches attribute*), 26
 user_join_royalrumble (*cogs.matches.Matches attribute*), 27
 user_login_link (*cogs.member.Member attribute*), 27

user_reset_password_link (*cogs.member.Member attribute*), 27
 user_stats_season1 (*cogs.matches.Matches attribute*), 27
 user_stats_season2 (*cogs.matches.Matches attribute*), 27
 user_stats_season3 (*cogs.matches.Matches attribute*), 27
 username (*utils.fjclasses._User attribute*), 37
 UserNotRegisteredError (*class in utils.fjclasses*), 37
 utils.checks (*module*), 33
 utils.fjclasses (*module*), 33
 utils.quickembed (*module*), 42

V

validate_bet() (*utils.fjclasses._User method*), 37
 Voice (*class in cogs.voice*), 29

W

wait_until_chbot_running() (*cogs.chatango.Chatango method*), 26

Y

YTDLSource (*class in cogs.voice*), 29