

## Annotation for user

Project is music bot for Discord with the ability to change roles using emoji's under the selected message. This bot can play any songs or playlists (playlists with not more then 25 songs) from YouTube and change roles on servers using emojis under selected message.

Bot commands:

- **h!play <song/playlist link or song name>** - connects to your voice chat and starts to play the song or playlist you chose (only from YouTube and playlists with not more then 25 songs).
- **h!help** - type this command in text-channel if you forgot bot commands, it will give whole list of available commands.
- **h!pause** - pauses the song.
- **h!resume** - if song is paused, starts playing again from the moment it was paused.
- **h!skip** - skips to the next song.
- **h!leave** - leaves your voice channel.

Most common problems:

1. If bot have lags during playing songs check stability of your internet connection.
2. If you asked to play playlist and bot doesn't play song immediately, just wait, bot is not almighty and can't do all your wishes at one time, it just need time to find and download all your songs.
3. If you asked to play playlist and bot can't find it, check your playlist permissions (Closed, By link or Open) and if it has permission Closed, change permission to By link or Open and type command + link again

4. If bot doesn't play song that means that it has 18+ requirement on YouTube, unfortunately it can't be fixed nowadays due to YouTube politics, so just ask bot admin to restart bot.
5. If bot plays too loud, click on the right button on bot's icon in voice channel and move the slider in user voice section to optimal for you volume.

Note: If you want to change token, emojis, roles, post, max number of roles or roles exclusions you need to do it in Config.py file using needed variables and relaunch bot

## For programmers

1. **1. Libraries used in project:** Discord.py, YouTube\_dl, Functools, Datetime.

(For more accurate library name check and version open requirements.txt which is situated in project directory).

1. **2. External tools used in project:** Ffmpeg(it's free audio extractor, you can download it by this link: <https://ffmpeg.org/download.html> and then put it in project's directory).

### 2. Global variables:

1. vc – this variable is used for setting of bot's voice channel status and is used in all functions connected to musical part (play, play\_next\_song, check\_inactivity, pause, resume, leave, skip).
2. last\_activity – this variable is used for checking bot activity, it's used in check\_inactivity function.

3. `current_song` – this variable is used for setting the song bot need to play right now, it's used in functions: `play`, `play_next_song`.

4. `queue` – this variable is used for setting queue from song that bot needs to play after current song. It's used in functions: `play`, `play_next_song`, `check_inactivity`, `leave`.

### **3. Functions:**

Note: You need to remember that nearly every function has its own decorator (`@client.event`, `@client.command` or `@loop`) that is used to set function purpose or task. Also, all the function are `async def`'s(except after function) and `ctx` is parameter required by majority of functions which use `Discord.py` and command `h!help` is in-built function in `Discord.py` that sets up automatically if `@client.command` exists and fills receives description using commentaries under function declaration.

#### **1. @loop functions**

- a. `def check_inactivity` – this function uses variables `vc`, `check_inactivity`, `queue` and is used to check that bot is used and if it is not used for more than 3 mins, bot will automatically disconnect from voice channel and clear queue, if it left.

#### **2. @Client.command and connected to them functions**

- a. `def play` – this function uses variables `vc`, `current_song`, `queue` and receives `avg` parameter which used to store link that user sent to text-channel in discord. This function is used to connect

to voice channel, extract the info using YouTube\_dl and ffmpeg, set queue and then it starts to play needed song using voice channel

- b. `def after` – this function uses parameters `ctx` and `error` (it is necessary parameter to use in this function even if it is not used). This function is used in case if song ended and is called by parameter `after` in `play` function that requires `error` parameter and this function creates task for `play_next_song_callback` function.
- c. `def play_next_song_callback` – this function is used to call `play_next_song` function
- d. `def play_next_song` – this function is used to play next song in queue if it exists and it uses variables `queue` and `current_song` to extract song info and then it calls `after` function if queue is not empty
- e. `def pause` – this function receives only `ctx` parameter and it's used to put song on pause
- f. `def resume` – this function receives only `ctx` parameter and it checks if song is pause and if yes it turns music on the moment it was stopped.
- g. `def leave` – this function receives only `ctx` parameter and it's used to leave from voice channel if bot is connected to it and clears queue.
- h. `def skip` – this function receives only `ctx` parameter and it's used to stop current song and skip to next in queue.

### 3. @Client.events

- a. `def on_ready` – this function is used to let programmer know that bot is ready and it launches loop on `check_inactivity` function.
- b. `def on_raw_reaction_add` – this function receives payload parameter and is used to change user role who put emoji on selected message. This function receives list of emojis selected as alternative of role, post ID and roles ID. It checks user ID and looking if it's valid on server and gives role if bot founds emoji in config file and gives corresponding role (of course if user don't have more role that it was set up).
- c. `def on_raw_reaction_remove` - this function receives payload parameter and is used to change user role who took off emoji from selected message. It checks user ID and looking if it's valid on server and removes role if user took off emoji from selected message.

### 4. Config file – this file stores variables and dictionaries like:

1. Token – this variable is used to store token given by discord to connect your code with bot profile set up on discord page.
2. Post\_ID – this variable is used to store ID from selected message on server that is used as indicator that there you can put emoji and you will receive role
3. Roles – this dictionary is used to store emojis used to change role and corresponding to them role id's

4. Max\_rolex\_per\_user – this variable is used to indicate in code how many roles can have every user on server
5. Excroles – this tuple is used in case if user have some roles that mustn't be count during giving role

## **5. Analogs of this bot and libraries:**

1. Discord.py - in Python language this library don't have any analogs, because it's official API made and published by Discord official developers
2. YouTube\_DL – it has analog caled **yt\_dlp**, but the reason why I choose YouTube\_DL, even if it can't turn on 18+ music(yt\_dlp can do this) that it more stable and doesn't require additional soft such as PhantomJS etc.
3. Off course my bot has analogs that have bigger functional or ability to use not only YouTube, but the main advantage of my bot is that it can turn on any video on any theme that many popular music bot banned.

## **6. Representation of received by bot data**

Bot requires that all commands must be entered correctly, because bot will simply ignore your input. For example, if you will write h!go bot will simply ignore your command, because you entered command that not in command list, same thing will be if you will type wrong prefix, because prefix is the main thing by which bot identifies that this message was sent exactly to him and only him

and this feature helps every bot on big server with many other musical bot distinguish that user is asking this bot, but not another.

## **7. Representation of output data by bot**

If you entered command correctly bot will output you exact things

provided by the developers that respond to this command. For example, if you type h!play bot will immediately connect to your voice channel and will start playing song you asked him to play.

## **8. Progress of work**

During this project I learned 2 new libraries and understood how to

make AI and learned how to use async functions and new Python syntaxis. Before, start I thought that it would be good Idea to use JSON, but during making of project, I realized that I can do exact same things, but using Python which made work simpler in terms of data import and storing. Also, before I started project, I didn't exactly understood what libraries apart from Discord.py I will need, so it was interesting for me to understand that I can use any library and make my bot whatever I like him to be.

## **9. Unfinished parts**

Unfortunately, it didn't succeed everything I expected in the very

beginning, especially because I made this bot not only as my half year

project, but a bot that I can use on my server and something that can be useful to me and my friends.

1. Optimization of playlist download – right now this bot can normally use playlist with +-25 songs, with some errors in terminal due too in-built restrictions by API on wait and list size from 50 to 75 songs, but not more. In future I want to optimize this moment for more comfortable usage. I have an idea how to make it, but I don't know will it help or make it worse, but I want to try to firstly download first song from playlist and turn it on and behind the scene during it bot will install others and add them to queue.
2. Repeat command – this function wasn't made, because it is least used personally and corresponding on my friend's answer command, but anyway I want to implement it. I have an idea how I can make it, if it is possible, I will take information from `current_song` variable and again add it to que, as first-time solution it can work, but as long-term solution it has a lot of disadvantages and problems and I will think more accurate about this feature in future.

## **10. Conclusion**

This bot has all the most necessary commands for Discord bots of this type and an additional function (role change). I personally think that I did a great job developing this bot - I learned how to use a couple of libraries, learned how to use asynchronous functions and understood how bot development works.

## **Few examples for testing**



Link on test server where you can test this bot:

<https://discord.gg/BT6VfRrTTG>

Few command examples:

h!play Bazzi Young & Alive

h!play <https://www.youtube.com/watch?v=RLnA25dVzrQ>

h!play <https://www.youtube.com/playlistlist=PLUmTiGOxKUOhdwifnVGxaMCbDvCVdIyyU>

Other commands don't need any extra input except h!<command>, but they will work only if bot is playing music at this moment.

All features with roles you can test under selected message you can see on screenshot, just click on emojis that was already put under this message:



