Robbot project documentation

Roberto Antoniello

February 19, 2022

In this file I resume how Robbot works with more details than the README file you can find in the repository. Robbot is a Telegram bot with a bunch of different features. Some of them extend the use of Telegram you usually do, some of them are just funny features.
The bot is written in Python and it is running in Python3.8 at the moment.

1 Configuration file

The configuration file is the first thing you must check because without it well compiled, the bot won't run at all. This file is also well explained in README file so I won't spend a lot of words on it.

$$config - example.json \Rightarrow config.json$$

1.1 DB path

The .db file will be saved in the directory you choose and set in config.json file

$$"path - db" \Rightarrow config.json$$

1.2 Super admin data

In this section we put our Telegram data as super admin(the only one) of the bot. The super admin is able to manage the db changes such as adding users, promote to admin and deleting users/revoke admin powers.

$$"id - super - admin" \Rightarrow config.json$$

1.3 commands,admin commands and super

Here we put the name of commands the bot is able to recognize and execute. For example if we don't want to execute the weather we just don't put it on user-commands field. It's the same for admin and super commands.

$$"user - commands" \Rightarrow config.json$$
$$"admin - commands" \Rightarrow config.json$$
$$"super - admin - commands" \Rightarrow config.json$$

2 main

app.py is the main source file. It creates the session, it connects with Telegram API using Pyrogram Client class and it's in constant wait for arriving messages. First of all it fetches the information from configuration file, then it starts the connection. While waiting, the bot will ignore any messages it doesn't recognize as a known command. Otherwise if it's a known command, it checks for permission to the user launched that command. If it's a recognized user, the entire message will be managed by a parser function and passed to a specific fetch function written inside controller.py.

$$app.py \xrightarrow{\text{command}} controller.py$$

2.1 parser function