## 

|  |  |
| --- | --- |
| W ramkach tematu nr13. | Bot na imie „czumba”  Małe i bardzo stare marzenie które zostało zrealizowane tylko teraz.  Ostap Florchuk  Grupa 1.2/4  Wprowadzenie do informatyki |

Spis treści

[0](#_Toc62432576)

[Wstęp 1](#_Toc62432577)

[Realizacja 2](#_Toc62432578)

[Pobieranie i Podłączenie 4](#_Toc62432579)

[Blok Nr1 6](#_Toc62432580)

[Blok Nr2 7](#_Toc62432581)

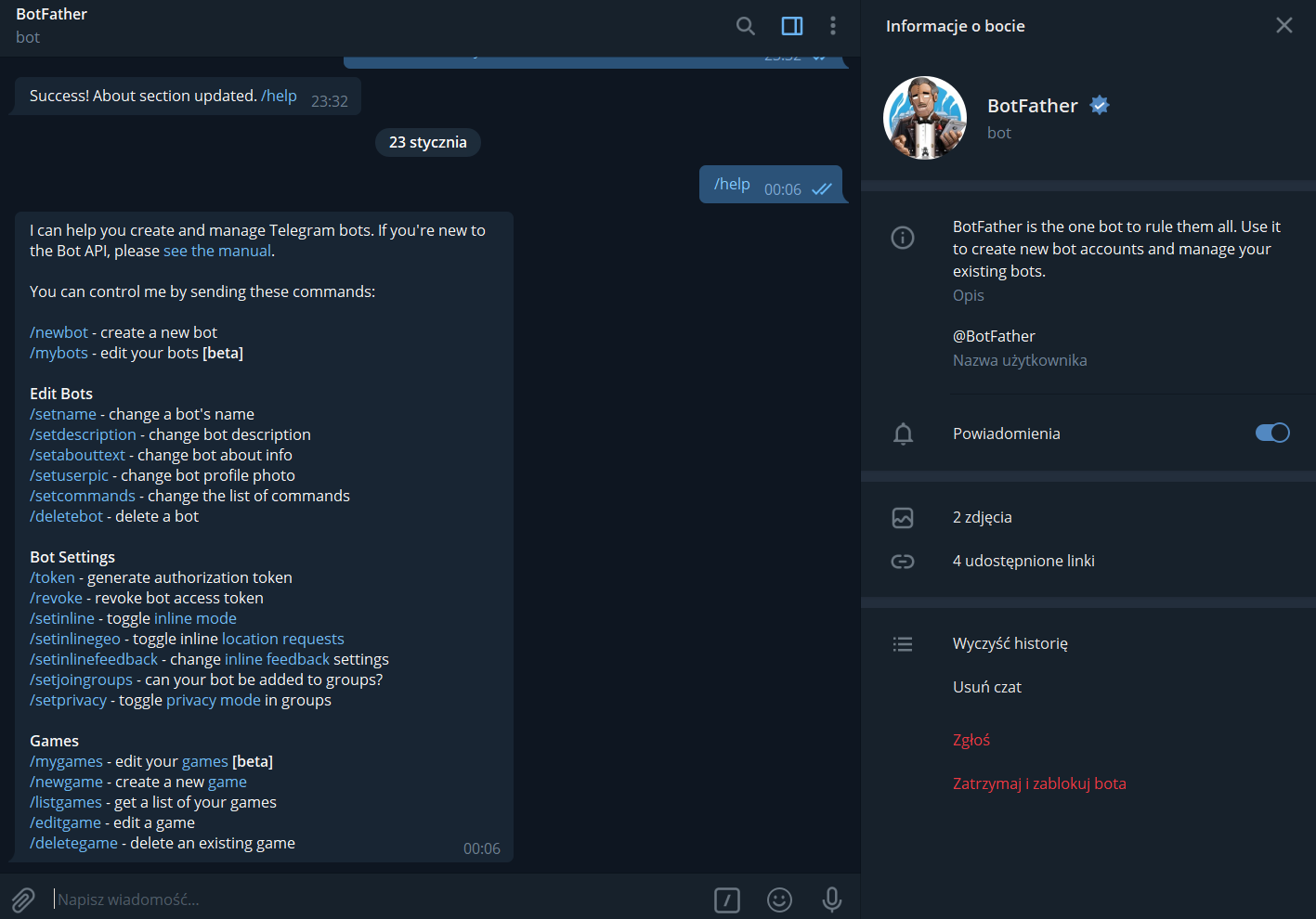
[Blok Nr3 8](#_Toc62432582)

[Feedbacks and Conclusions 12](#_Toc62432583)

# Wstęp

Wszystko się zaczęło dwa lata temu, kiedy zacząłem korzystać się z sieci „Telegram” i zobaczyłem tam takie ciekawe rzeczy które nazywane są Telegram-Boty. Dla mnie to było trochę dziwne i bardzo ciekawe, okazało się że z komputerem można rozmawiać 😉

Z tego momentu zacząłem tym interesować się ale mówiąc szczerze nie było dużo czasu na eksploracje języka ( Python ) na którym te wszystkie rzeczy są napisane (egzaminy szkolne, choroby itd.)



Rysunek 1. BotFather-bot za pomocą którego trzeba rejestrować swojego bota w sieci "Telegram"

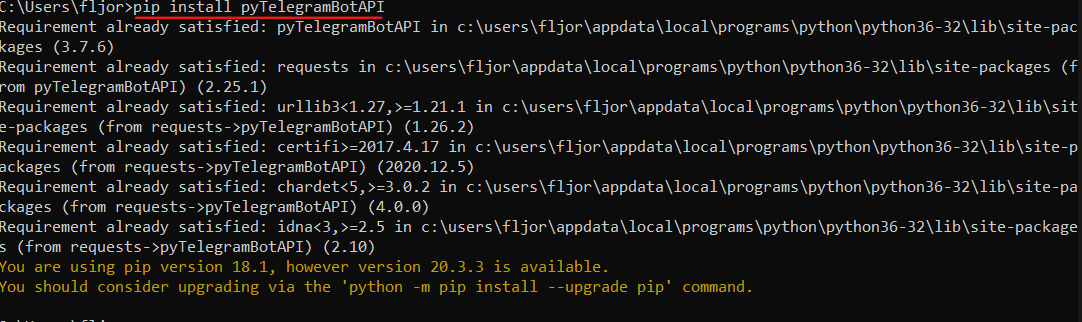
Kiedy zobaczyłem tematy dla kolokwium, zobaczyłem takie zadanie: „Na podstawie przykładów opisz metody pobierania danych z internetu przy wykorzystaniu języka Python.”, wspomniałem o moim dawnym marzeniu i zrozumiałem, że teraz mam i czas i wiedze na realizację tego. Bot działa za pomocą Internetu 🡪 Użytkownik komunikuje z nim też za pomocą Internetu 🡪 Pobieranie danych z web-stron też można realizować wykorzystując botów i różne biblioteki języka Python 🡪 W takim przypadku bot wyświetlający pogodę w wybranym mieście wydawał mi się bardzo ciekawym sposobem realizacji tematu.

# Realizacja

Zacząłem szukać łatwe i nie bardzo skomplikowane sposoby do zrealizowania mojego projektu i znalazłem bibliotekę o nazwie: pyTelegramBotAPI :

* <https://pypi.org/project/pyTelegramBotAPI/#general-api-documentation>

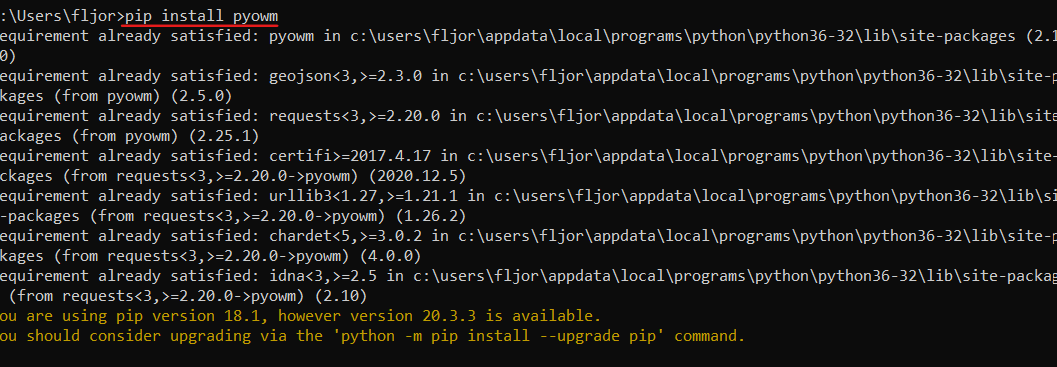
która służy do zrealizowania poprawnego działania bota i łatwego dołączenia do „Telegram”.



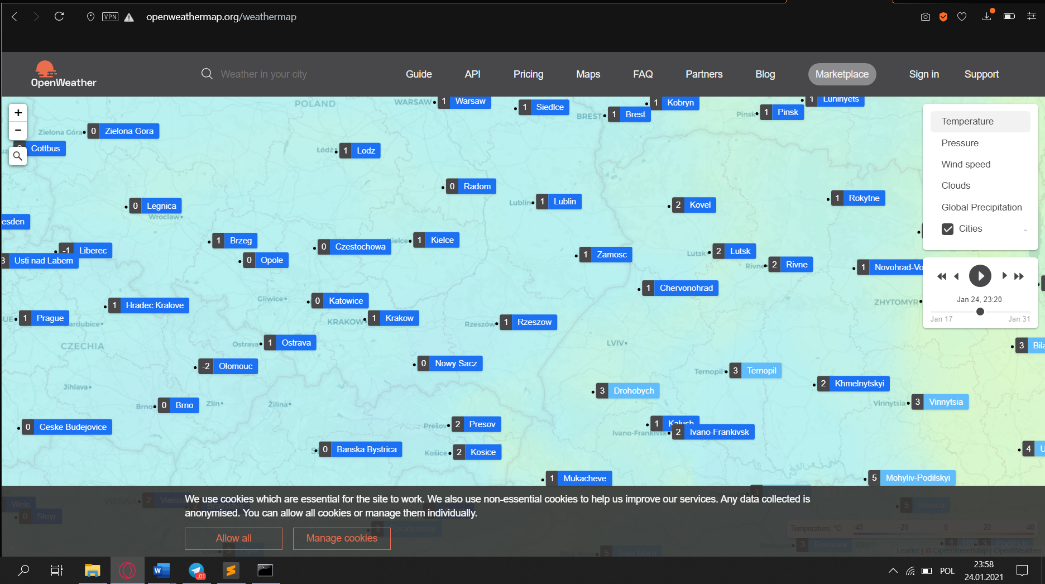
Rysunek 2.Instalacja pyTelegramBotAPI

Także bardzo długo myślałem nad zrealizowaniem pobierania danych: sposób „parsing” web stron wydawał mi się bardzo skomplikowanym i niewiarygodnym i dlatego znalazłem web stronę OpenWeatherMap.org <https://openweathermap.org/weathermap?basemap=map&cities=true&layer=temperature&lat=30&lon=-20&zoom=5> i bibliotekę pyowm:

* <https://pyowm.readthedocs.io/en/latest/>

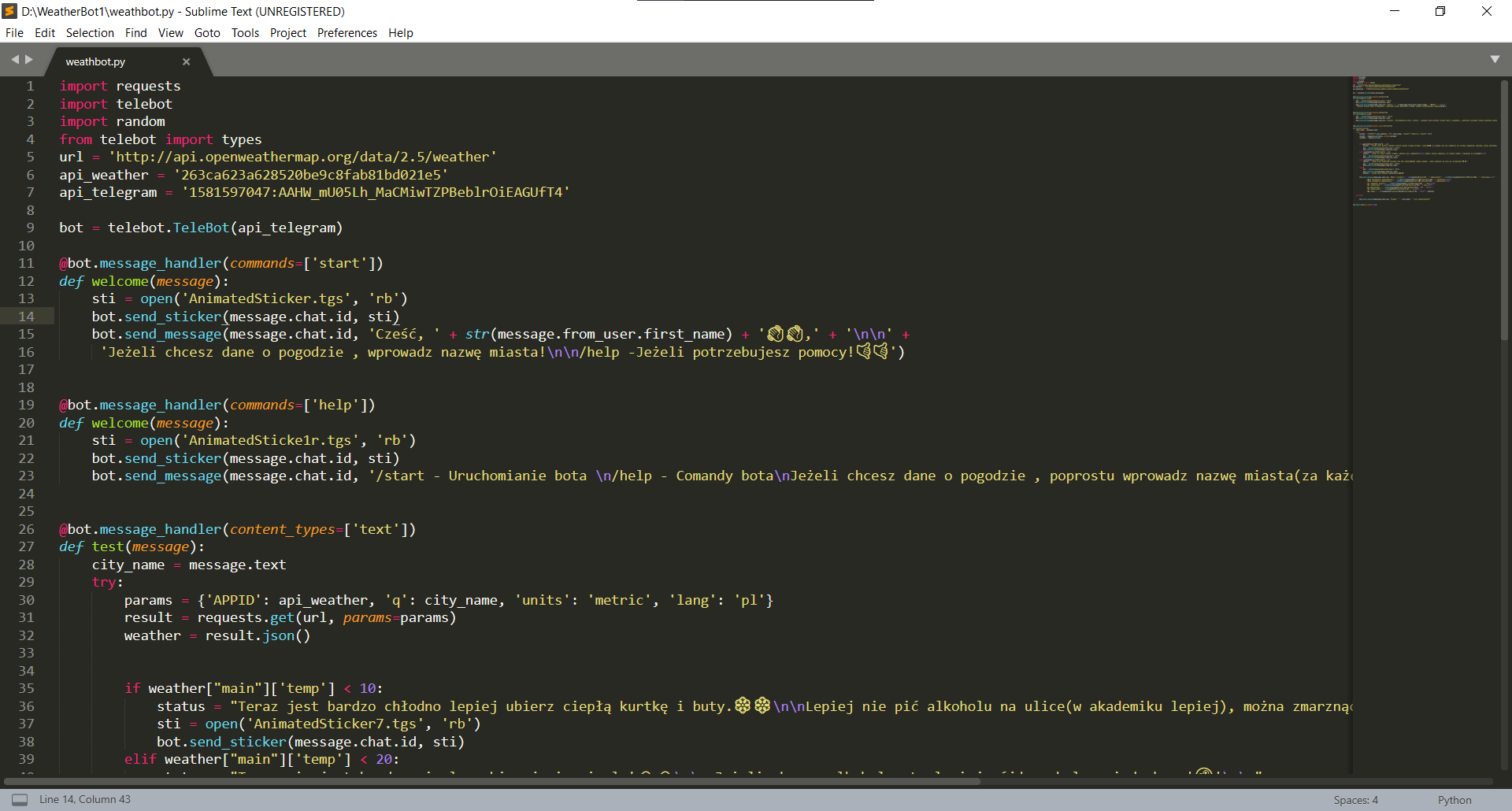


Rysunek 3.Instalacja bibioteki pyowm



Rysunek 4.WebStrona OpenWeatherMap.org

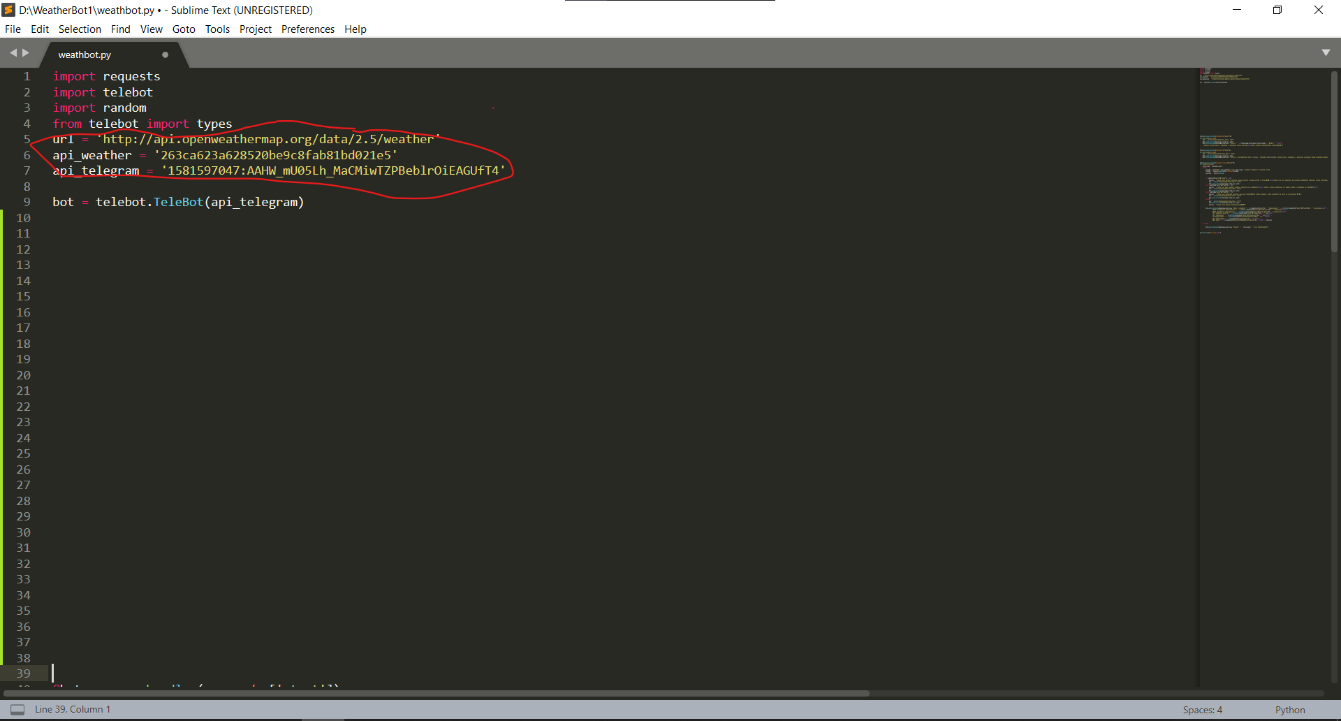
Dla wykonania kodu wykorzystałem redaktor kodu SublimeText, Pycharm dla moich celów nie był podchodzący dlatego że tam były problemy z instalacją bibliotek i po prostu chciałem na żywo zobaczyć jak wygląda taka robota bez automatycznego wyszukiwania błędów.(tak mówiąc skomplikowałem trochy sobie życie).



Rysunek 5.Printscreen kodu napisanego w SublimeText

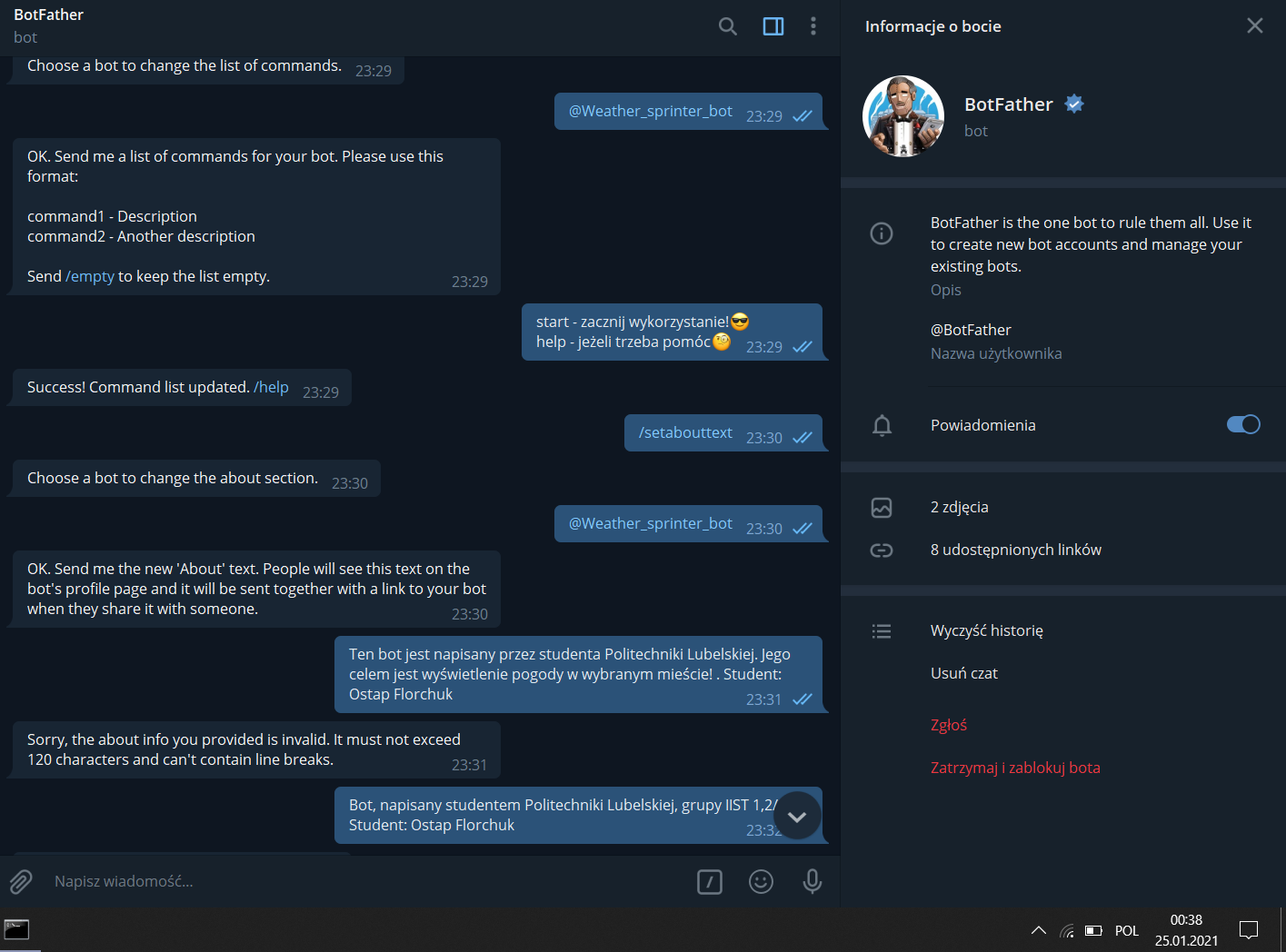
# Pobieranie i Podłączenie

Bardzo wielu mi to czasu zajęło, problem był taki, że web-strona z której pobieram dane zmieniła swój regulamin, i teraz wiele komand są płatne i bardzo długo się męczyłem z podłączeniem i pobieraniem danych przez klucz API ale w ogóle mi się to udało(dzięki web-stronie GitHub):

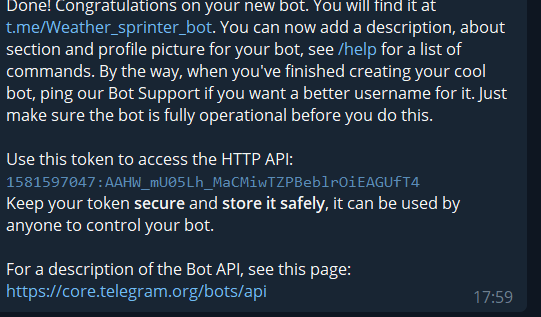


Rysunek 6.Linki i klucze API

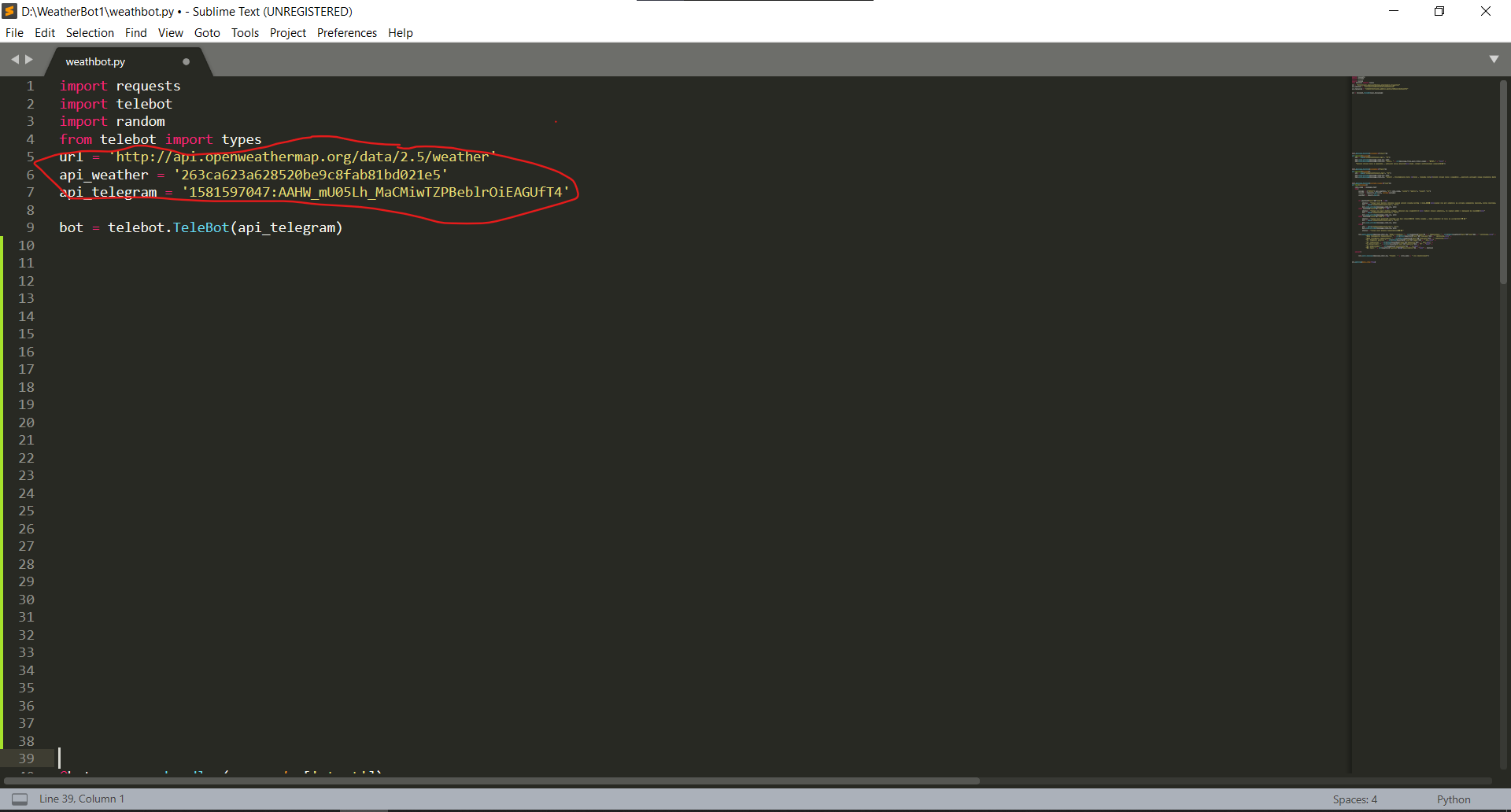
Żeby zaczął działać w „Telegram”, mu trzeba udostępnić klucz API który jest nadany przez głównego bota w sieci @BotFather. Dlatego trzeba go znaleźć🡪 napisać mu że chcesz utworzyć swego bota🡪nadać mu imię🡪nadać mu user-name za którym go można znaleźć u mnie to @Weather\_sprinter\_bot🡪Nadać mu opis za pomocą /setdescription🡪nadać mu główne zdjęcie /setuserpic🡪nadać dwie standardowe komendy /setcommands🡪otrzymać klucz API



Rysunek 7.Uzupelnianie danych/rejestracja bota

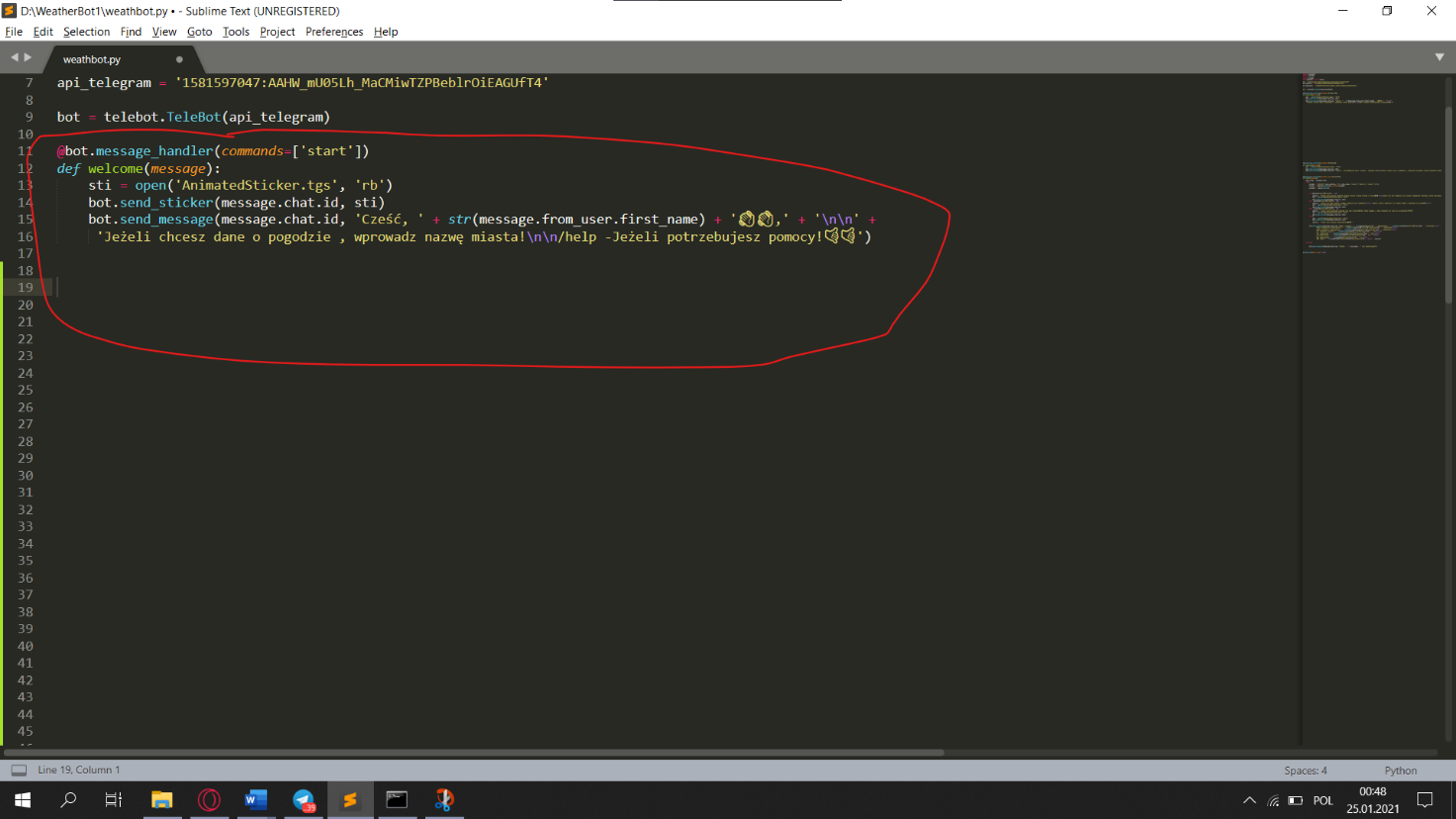


Rysunek 8.Otrzymanie klucza



Rysunek 9.Wpisanie klucza w polu api\_telegram

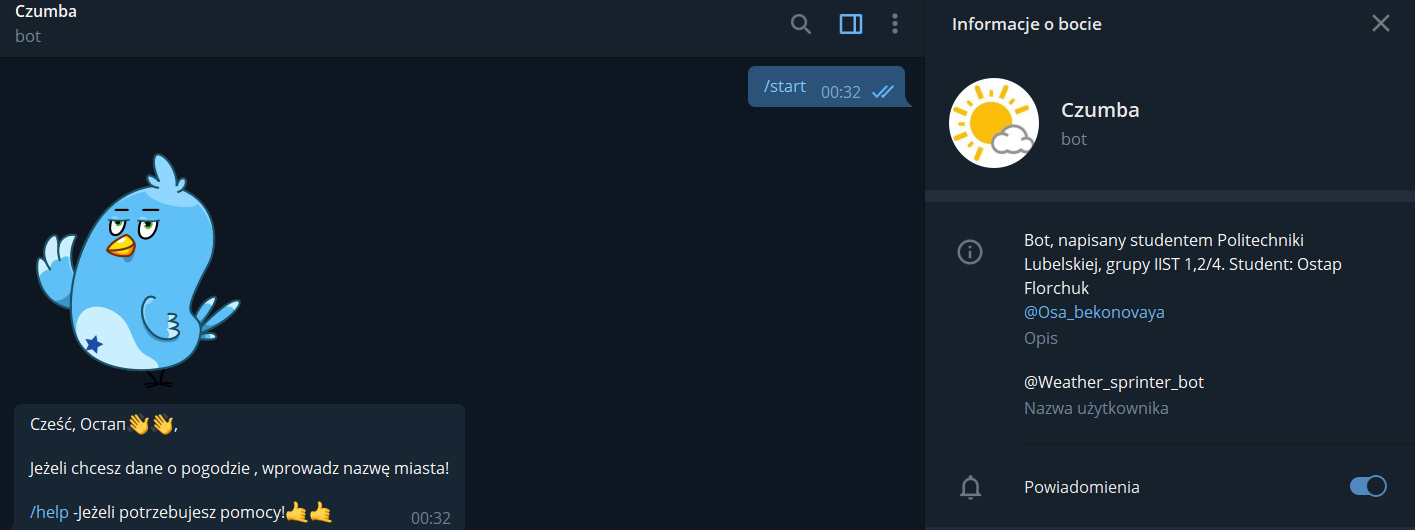
# Blok Nr1



Rysunek 10.Blok Nr1. Blok w którym jest wyświetlana 1 wiadomość z opsiem Bota i miłym ptakiem)

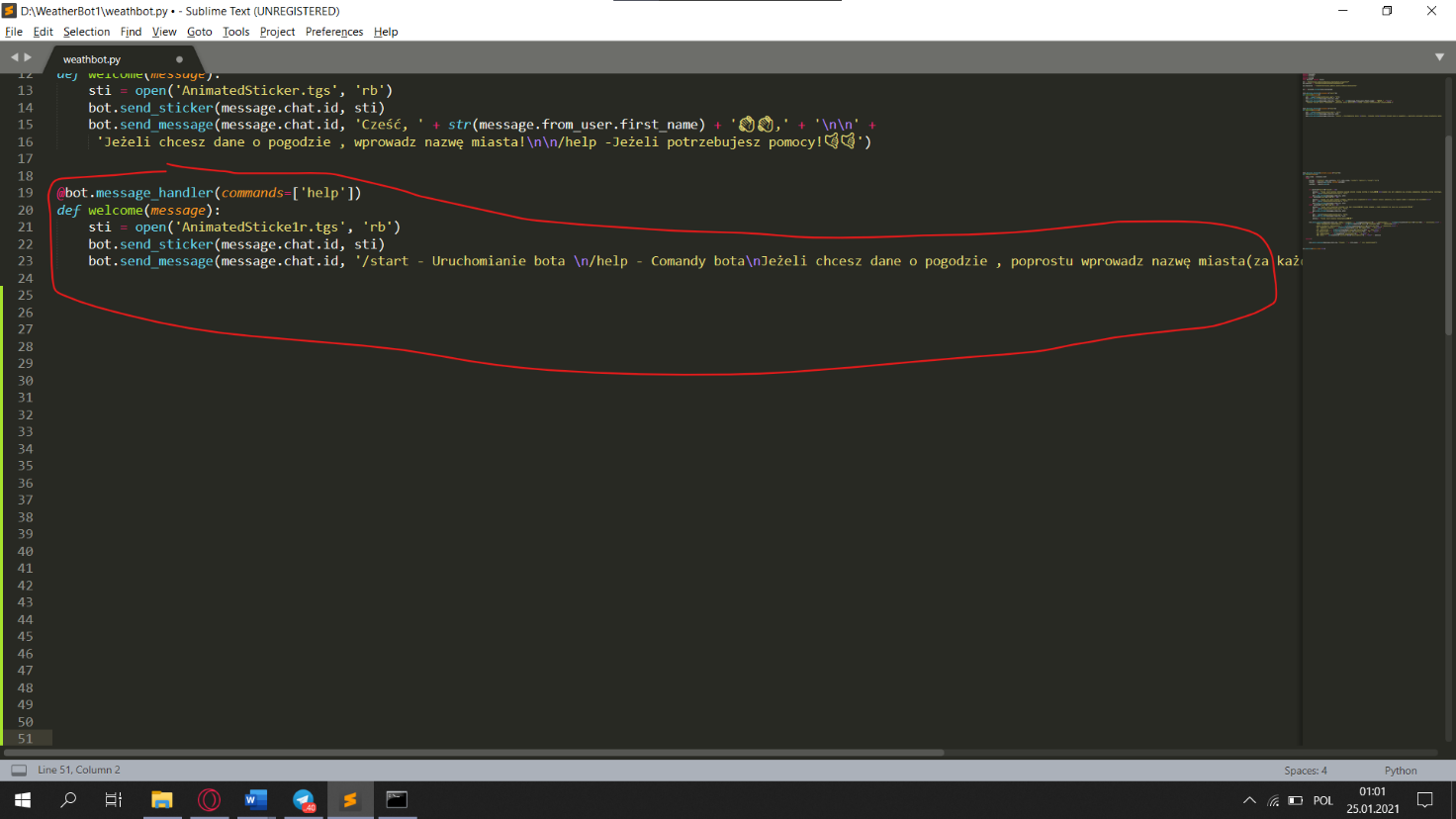
Tutaj są wykorzystane metody:

* sti = open('AnimatedSticker.tgs', 'rb')- wysyła wiadomość z miłym ptakiem w rozszerzeniu .tgs.
* bot.send\_message(message.chat.id,…)- wysyła wiadomość z opisem bota i poradami do wykorzystania.
* @bot.message\_handler(commands=['start'])- metoda która wywoła wykonanie bloku przy wprowadzeniu komendy /start na czacie z botem.



Rysunek 11.Wynik działania 1 bloku

# Blok Nr2



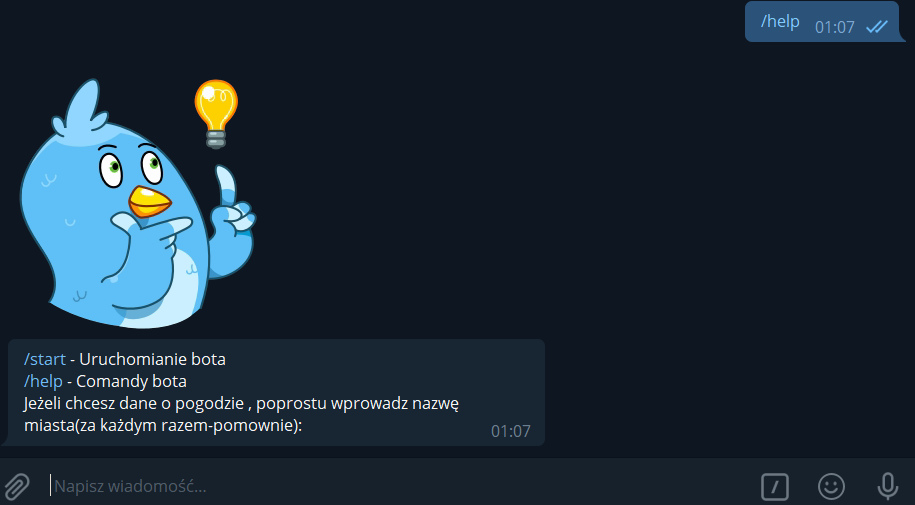
Rysunek 12.Blok2

Blok Nr2 mało-czym różnią się od pierwszego, on zostanie wywołany po napisaniu na czacie komendy /help

I wyświetli wiadomość w której można uzyskać pomóc.

Tutaj są wykorzystane metody:

* sti = open('AnimatedSticker.tgs', 'rb')- wysyła wiadomość z miłym ptakiem w rozszerzeniu .tgs.
* bot.send\_message(message.chat.id,…)- wysyła wiadomość z pomocniczą informacją.
* @bot.message\_handler(commands=['start'])- metoda która wywoła wykonanie bloku przy wprowadzeniu komendy /help na czacie z botem.



Rysunek 13.Wynik działania bloku Nr2

# Blok Nr3

Blok Nr3 jest największy i najbardziej skomplikowany 😊



Rysunek 14.Blok Nr3

Tutaj są wiele komand i metod pobranych z biblioteki pyowm:

* Try/except standardowe bloki języka Python
* params = {'APPID': api\_weather, 'q': city\_name, 'units': 'metric', 'lang': 'pl'} - pobiera pogodę na moment konwersacji z botem w zadanej lokalizacji(mieście)
* result = requests.get(url, params=params)
* weather = result.json() – jeżeli dobrze zrozumiałem z dokumentacji to pobiera dane z biblioteki JSON(otrzymuje odpowiedź).
* @bot.message\_handler(content\_types=['text']) – wywołuje cały blok kiedy wprowadzona jest nazwa dowolnego miasta na dowolnym języku.
* If \ elif- w tych blokach przemiennej „status” są nadane różne wartości tekstowe w zależności od parametrów temperatury.(Dodałem trochy humoru pro alkohol, proszę nie osądzać surowo) 😊
* bot.send\_message(message.chat.id,..) – wyświetla wszystkie dane , pobrane z web-strony : Temperaturę(min/max) , wilgotność, prędkość wiatru, widoczność i nawet ciśnienie, także wyświetla zawartość przemiennej „status”.
* W bloku „except” jest wyświetlona wiadomość jeżeli nie było znalezione wprowadzone miasto.



Rysunek 15.Wynik działania przy poprawnie wprowadzonym mieście



Rysunek 16.Wynik niepopranego miasta



Rysunek 17.Bot działa nawet dla wprowadzonych nazw państw

# Feedbacks and Conclusions

Moi koledzy 4 dziennym testowaniu powiedziały, że działa poprawnie bez wielkich problemów.

Głównym zauważeniem jest to, że nie zawsze znajdzie niektóre miasta i trzeba po 3 razy ich wpisywać, no to już problem web-strony.

Dobrym punktem oni mówią jest humor i możliwość wpisywania nazw na różnych językach.

Także komfortowym jest menu stworzone za pomocą @BotFather, ułatwia wprowadzenie komand przez użytkownika.



W końcu , po wszystkich działaniach, nauczyłem się korzystać się z różnych bibliotek, instalować ich, korzystać się z nich , także nauczyłem się szukać informacji w internecie i po prostu spełniłem swoje dawnie marzenie 😊.

Na przyszłość: myślę, że będę rozwijać temat Telegram- Botów w swoim życiu i w ogóle będę dalej uczyć się języka Python, dlatego że poznałem go ze strony kształcenia i ze strony zabawy i funkcjalności.

\*komentarz: bot działa tylko wtedy, kiedy mój komputer jest włączony i działa jako serwer. Dla go uruchomiania przez „cmd” trzeba w wierszu polecenia(w moim przypadku) wpisać takie komendy:

cd /d D:\WeatherBot1

python weathbot.py

Jeżeli pan doctor zachce, to proszę mi napisać na teamsa, żeby pan mógł przetestować bota jako zwykły użytkownik.