



W ramach 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

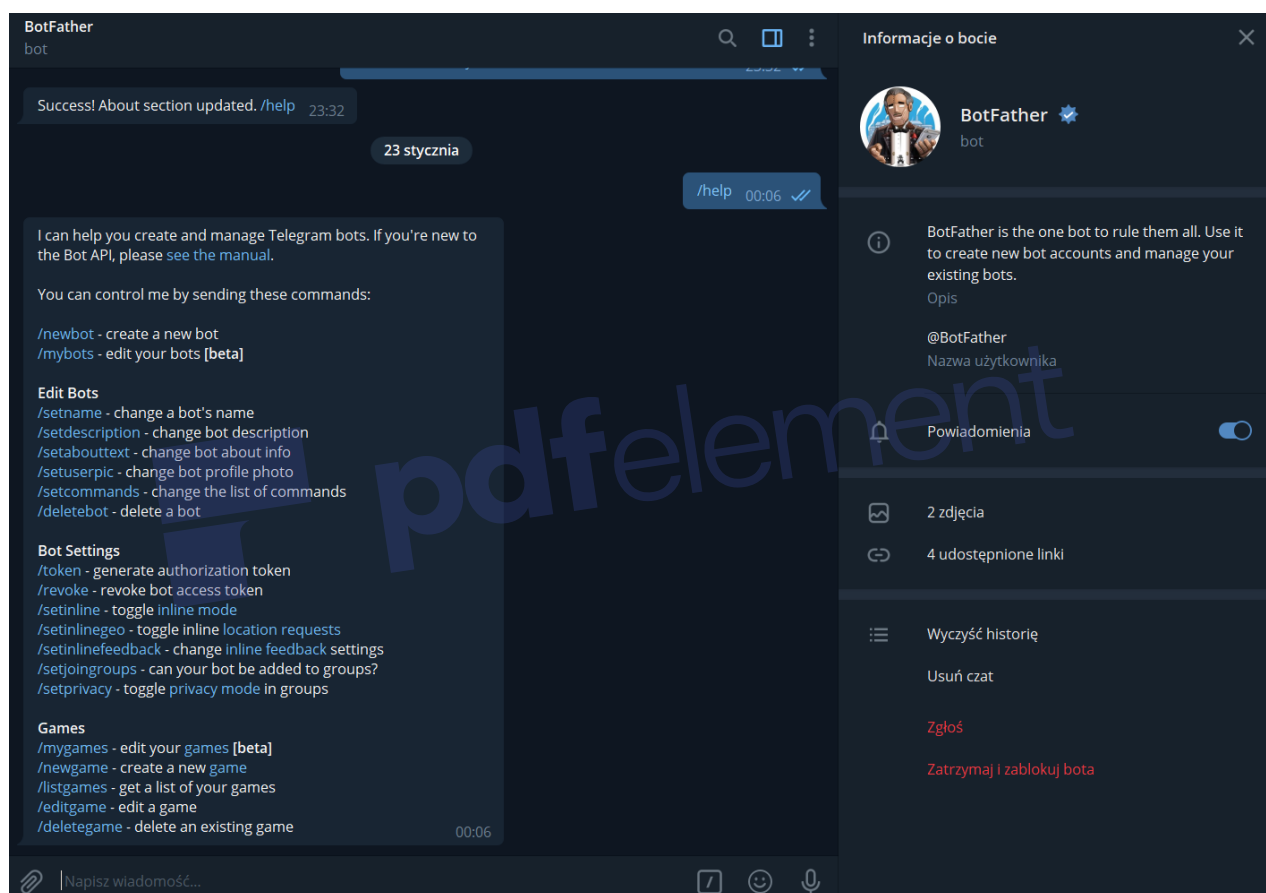
.....	0
Wstęp .....	2
Realizacja .....	3
Pobieranie i Podłączenie .....	5
Blok Nr1 .....	7
Blok Nr2 .....	8
Blok Nr3 .....	9
Feedbacks and Conclusions.....	12



## 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 eksplorację języka ( Python ) na którym te wszystkie rzeczy są napisane (egzamin y 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 wiedzę 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”.

```
C:\Users\fljor>pip install pyTelegramBotAPI
Requirement already satisfied: pyTelegramBotAPI in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (3.7.6)
Requirement already satisfied: requests in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from pyTelegramBotAPI) (2.25.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests->pyTelegramBotAPI) (1.26.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests->pyTelegramBotAPI) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests->pyTelegramBotAPI) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests->pyTelegramBotAPI) (2.10)
You are using pip version 18.1, however version 20.3.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

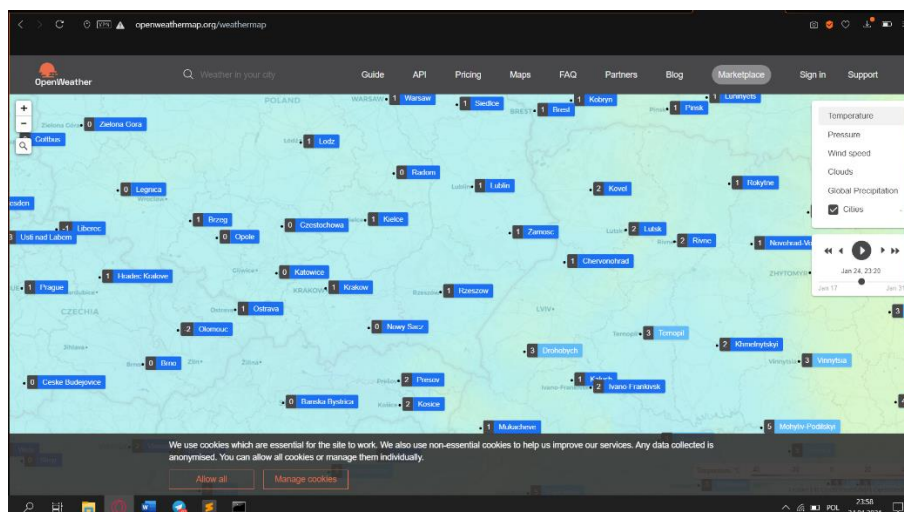
Rysunek 2.Instalacja pyTelegramBotAPI

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/>

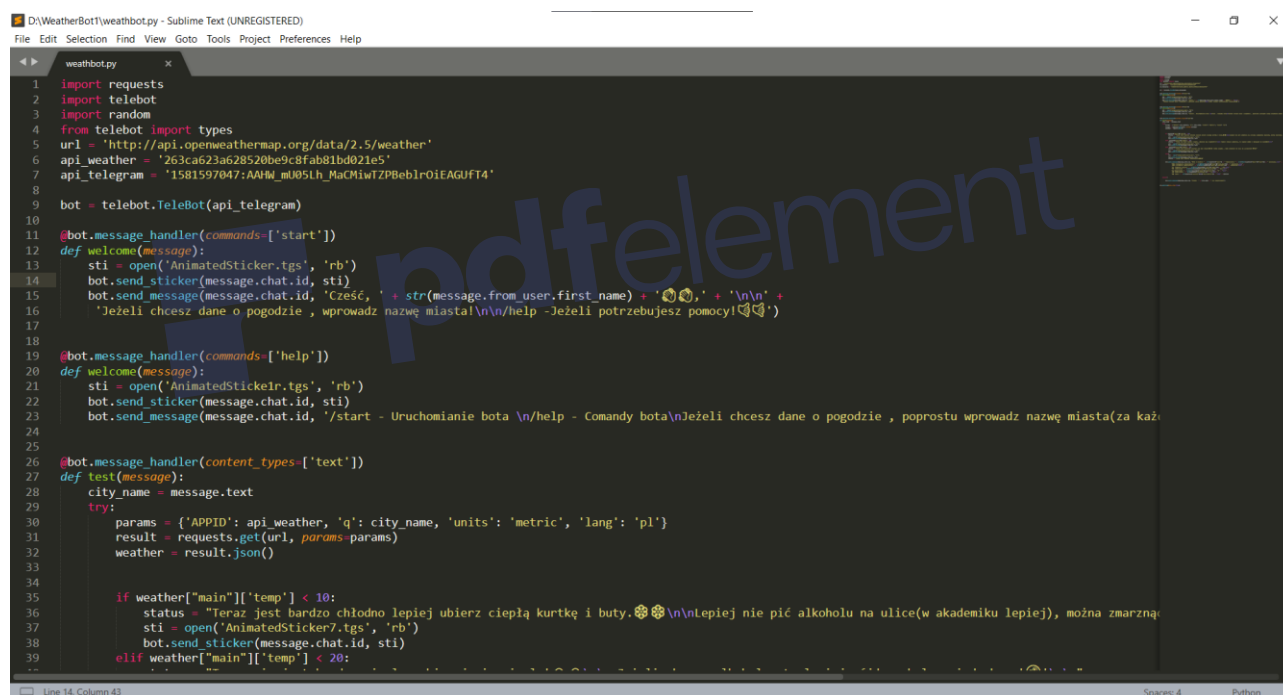
```
C:\Users\fljor>pip install pyowm
Requirement already satisfied: pyowm in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (2.10.0)
Requirement already satisfied: geojson<3,>=2.3.0 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from pyowm) (2.5.0)
Requirement already satisfied: requests<3,>=2.20.0 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from pyowm) (2.25.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests<3,>=2.20.0->pyowm) (2020.12.5)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests<3,>=2.20.0->pyowm) (1.26.2)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests<3,>=2.20.0->pyowm) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\fljor\appdata\local\programs\python\python36-32\lib\site-packages (from requests<3,>=2.20.0->pyowm) (2.10)
You are using pip version 18.1, however version 20.3.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Rysunek 3.Instalacja biblioteki pyowm



Rysunek 4. WebStrona OpenWeatherMap.org

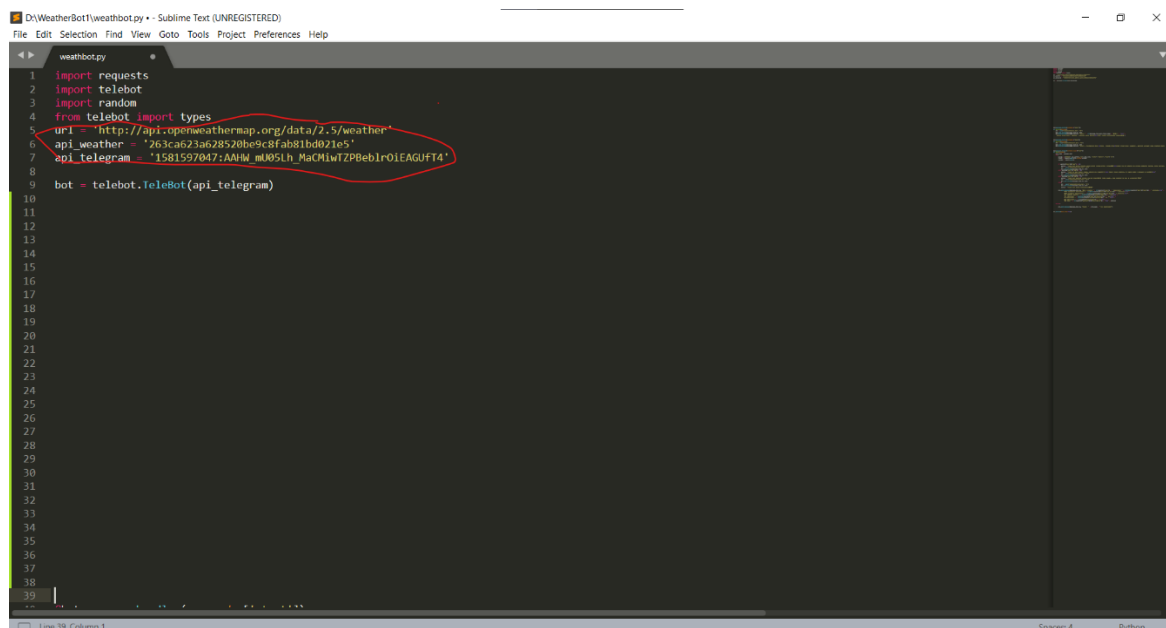
Dla wykonania kodu wykorzystałem redaktor kodu SublimeText, Pycharm dla moich celów nie był podchodzący dlatego że tam były problemy z instalacją bibliotek i po prostu chciałem na żywo zobaczyć jak wygląda taka robota bez automatycznego wyszukiwania błędów.(tak mówiąc skomplikowałem trochę sobie życie).



Rysunek 5. Printsreen 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):



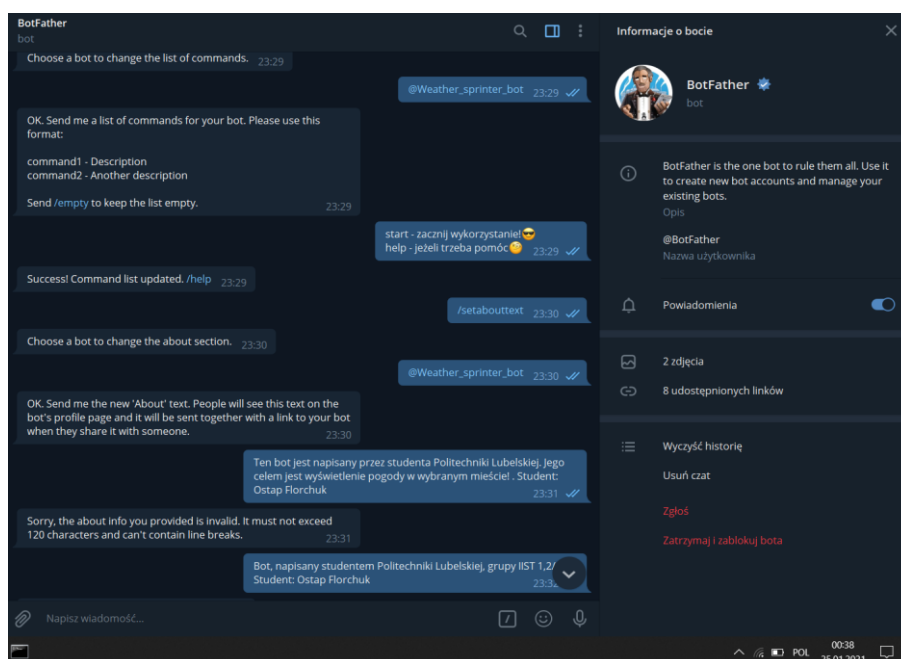
```

1 import requests
2 import telebot
3 import random
4 from telebot import types
5 url = 'http://api.openweathermap.org/data/2.5/weather'
6 api_weather = '263ca623a628520be9c8fab81bd021e5'
7 api_telegram = '1581597047:AAHW_m005Lh_MaCm1w1ZPBeb1r0IEAGUFT4'
8
9 bot = telebot.TeleBot(api_telegram)
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

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. Uzupełnianie danych/rejestracja bota

Done! Congratulations on your new bot. You will find it at [t.me/Weather\\_sprinter\\_bot](https://t.me/Weather_sprinter_bot). You can now add a description, about section and profile picture for your bot, see `/help` for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

1581597047:AAHW\_mU05Lh\_MaCMiwTZPBeb1rOiEAGUfT4

Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

17:59

Rysunek 8. Otrzymanie klucza

```

1 import requests
2 import telebot
3 import random
4 from telebot import types
5 url = "http://api.openweathermap.org/data/2.5/weather"
6 api_weather = "263ca623a628528be9c8fab81bd021e5"
7 api_telegram = "1581597047:AAHW_mU05Lh_MaCMiwTZPBeb1rOiEAGUfT4"
8
9 bot = telebot.TeleBot(api_telegram)
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

Rysunek 9. Wpisanie klucza w polu api\_telegram

## Blok Nr1

```

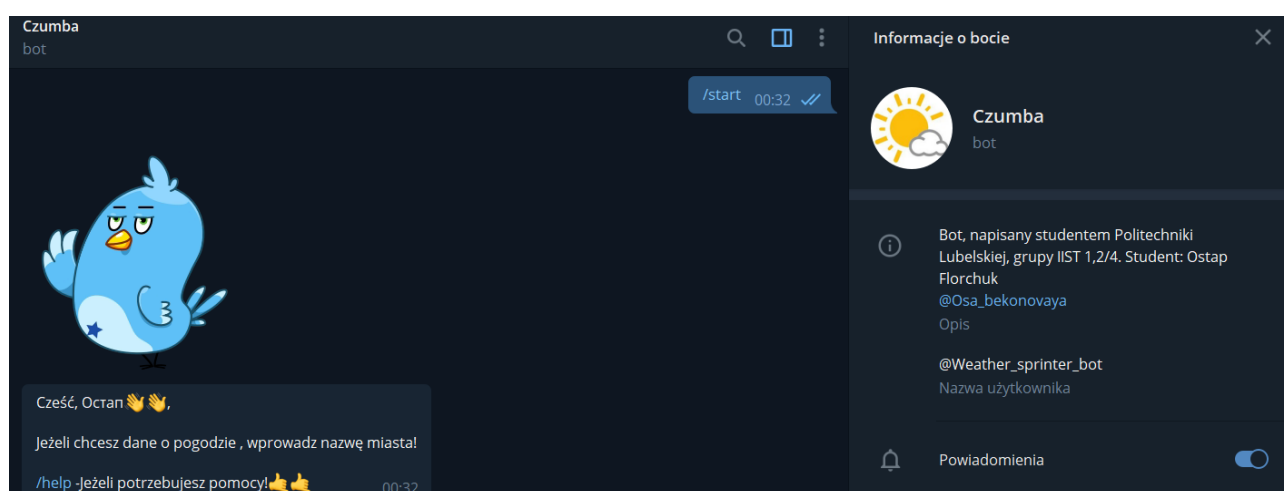
7  api_telegram = '1581597047:AAHW_mL05Lh_MaCMiwTzPBebLr0iEAGJfT4'
8
9  bot = telebot.TeleBot(api_telegram)
10
11 @bot.message_handler(commands=['start'])
12 def welcome(message):
13     sti = open('AnimatedSticker.tgs', 'rb')
14     bot.send_sticker(message.chat.id, sti)
15     bot.send_message(message.chat.id, 'Cześć, ' + str(message.from_user.first_name) + ' 🐦👋' + '\n\n' +
16                     'Jeżeli chcesz dane o pogodzie , wprowadz nazwę miasta!\n\n/help -Jeżeli potrzebujesz pomocy!👉👉')
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
--

```

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

```

13 weathbot.py
14 def weathbot(message):
15     sti = open('AnimatedSticker.tgs', 'rb')
16     bot.send_sticker(message.chat.id, sti)
17     bot.send_message(message.chat.id, 'Cześć, ' + str(message.from_user.first_name) + '!\n\n' +
18     'Jeżeli chcesz dane o pogodzie , wprowadz nazwę miasta\n\n/help - Jeżeli potrzebujesz pomocy!👉👈')
19
20 @bot.message_handler(commands=['help'])
21 def welcome(message):
22     sti = open('AnimatedSticker.tgs', 'rb')
23     bot.send_sticker(message.chat.id, sti)
24     bot.send_message(message.chat.id, '/start - Uruchomienie bota \n/help - Comandy bota\nJeżeli chcesz dane o pogodzie , poprostu wprowadz nazwę miasta(za każ
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

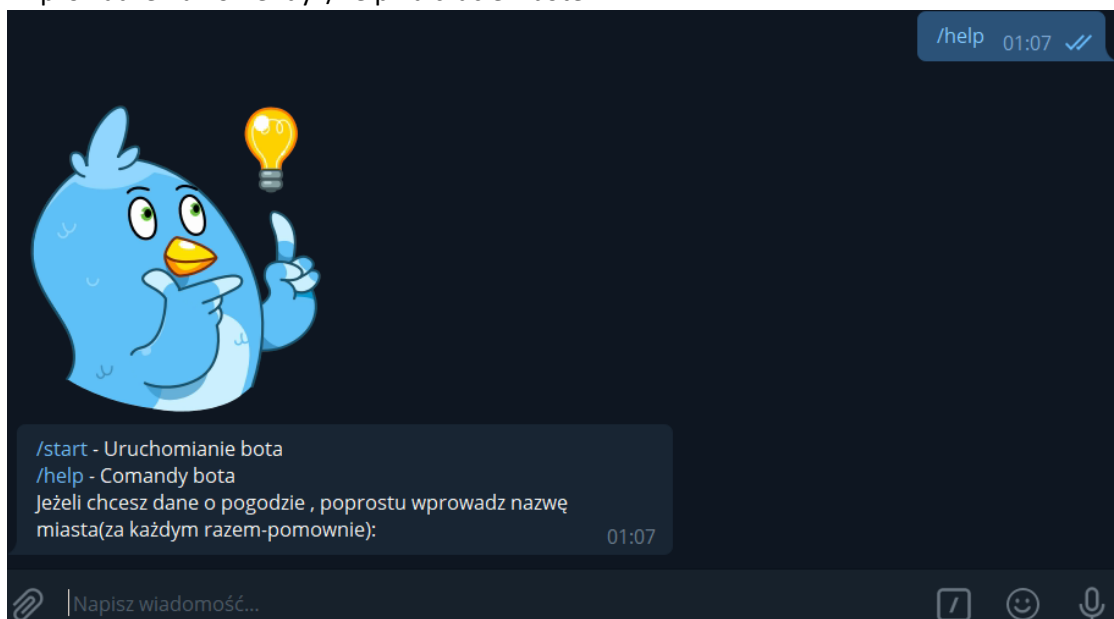
```

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 pomoc.

Tutaj są wykorzystane metody:

- `sti = open('AnimatedSticker.tgs', 'rb')`- wysła wiadomość z miłym ptakiem w rozszerzeniu .tgs.
- `bot.send_message(message.chat.id,...)`- wysła 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 😊

```

@bot.message_handler(content_types=['text'])
def test(message):
    city_name = message.text
    try:
        params = {'APPID': api_weather, 'q': city_name, 'units': 'metric', 'lang': 'pl'}
        result = requests.get(url, params=params)
        weather = result.json()

        if weather["main"]["temp"] < 10:
            status = "Teraz jest bardzo chłodno lepiej ubierz ciepłą kurtkę i buty.❄️❄️\n\nLepiej nie pić alkoholu na ulice, można zmarznąć, poczucie ciepła j"
            sti = open('AnimatedSticker7.tgs', 'rb')
            bot.send_sticker(message.chat.id, sti)
        elif weather["main"]["temp"] < 20:
            status = "Teraz nie jest bardzo ciepło, ubieraj się ciepło!☁️☁️\n\nJeżeli chcesz alkoholu, to lepiej pójdz z kolegami do baru!🍷\n\n"
            sti = open('AnimatedSticker6.tgs', 'rb')
            bot.send_sticker(message.chat.id, sti)
        elif weather["main"]["temp"] > 22:
            status = "Teraz jest pięknie! ubieraj się jak chcesz!☀️☀️ Dobra pogoda , żeby pojechać do lasu na wycieczke! 😊☀️"
            sti = open('AnimatedSticker2r.tgs', 'rb')
            bot.send_sticker(message.chat.id, sti)
        else:
            sti = open('AnimatedSticker5.tgs', 'rb')
            bot.send_sticker(message.chat.id, sti)
            status = "Teraz jest piękna temperatura!😊😊"

        bot.send_message(message.chat.id, "🌤️ W mieście " + str(weather["name"]) + " temperatura " + str(float(weather["main"]["temp"])) + " celsjusza.\n\n"
            "🌡️ Maksymalna temperatura: " + str(float(weather["main"]["temp_max"])) + " celsjusza.\n\n" +
            "🌡️ Minimalna temperatura: " + str(float(weather["main"]["temp_min"])) + " celsjusza.\n\n" +
            "💨 Prędkość wiatru: " + str(float(weather["wind"]["speed"])) + " m/s.\n\n" +
            "📉 Ciśnienie: " + str(float(weather["main"]["pressure"])) + " bar.\n\n" +
            "💧 Wilgotność: " + str(int(weather["main"]["humidity"])) + "%" + "\n\n" +
            "🌫️ Widoczność: " + str(weather["visibility"]) + " m.\n\n" +
            "📄 Opis: " + str(weather["weather"][0]["description"]) + "\n\n" + status)

    except:
        bot.send_message(message.chat.id, "Miasto " + city_name + " nie znaleziono")

```

Rysunek 14.Blok Nr3

Tutaj są wiele komand i metod pobranych z biblioteki pyowm:

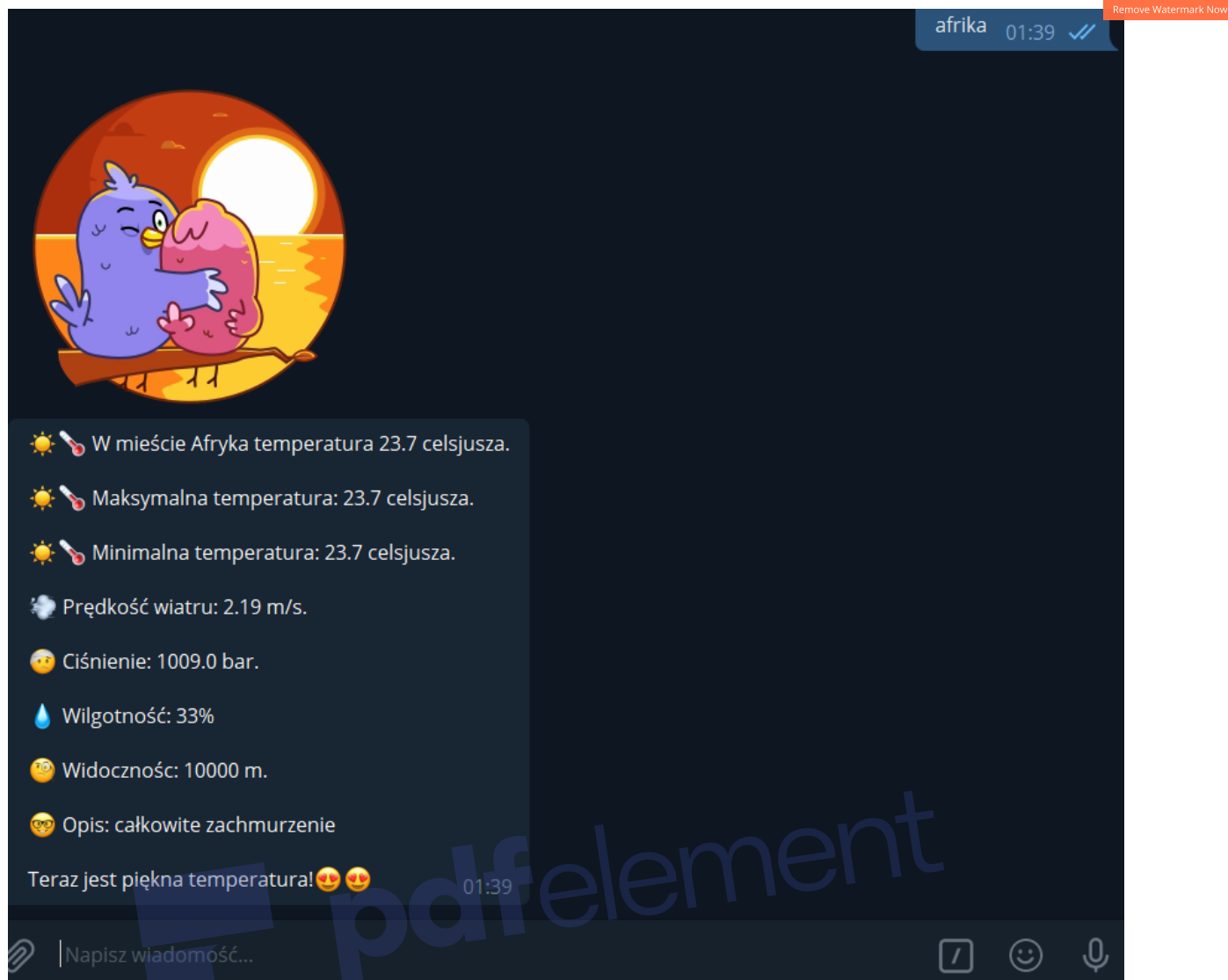
- Try/except standardowe bloki języka Python
- params = {'APPID': api\_weather, 'q': city\_name, 'units': 'metric', 'lang': 'pl'} - pobiera pogodę na moment konwersacji z botem w zadanej lokalizacji(mieście)
- result = requests.get(url, params=params)
- weather = result.json() – jeżeli dobrze zrozumiałem 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 trochę 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 dawne 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 funkcjonalności.

**\*komentarz: bot działa tylko wtedy, kiedy mój komputer jest włączony i działa jako serwer. Dla go uruchomienia 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.

[Remove Watermark Now](#)

