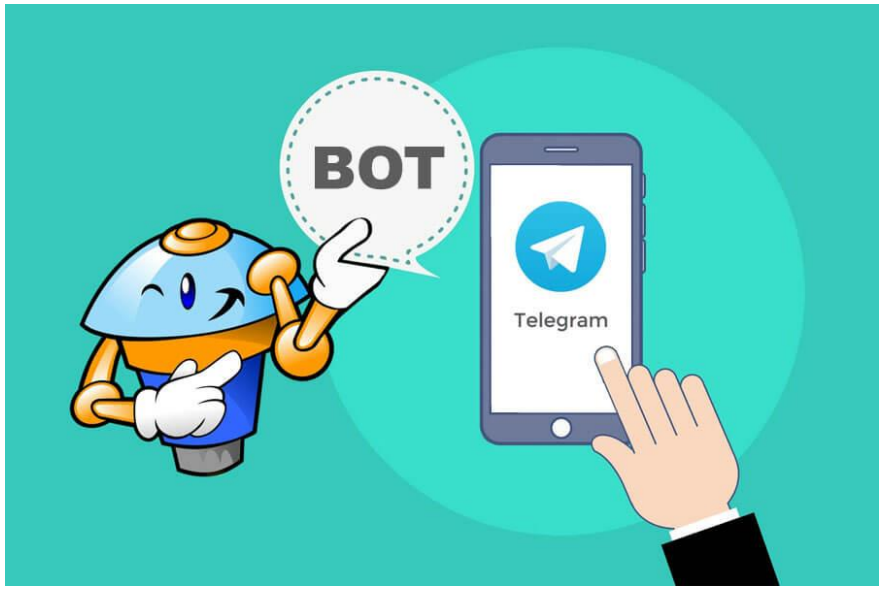
Telegram Bot



Agenda

- Introduction
- Authorization
- Communication
- The pyTelegramBotAPI package
- The message class
- Message handler

Introduction

Prerequisites: It is presumed that created a bot with @BotFather and got the API authorization token (Find the contact BotFather and type ”/newbot”. Then BotFather will lead you to the end.)
More details can be found in the official Introduction to Telegram bots.

This section is heavily-based-on and partially-copy-pasted-from the official Telegram bot API. Have a look there to better understand the content of this documentation.

Authorization

Once a bot is created in the telegram, the token must be taken and used later .
Copy it from the message you got from BotFather.

For example:

```
token = '1594806170:AAEyMsEgCJjE2BenGzO_7Viuh44w81VMrQ'
```

For install the telegram package is:

```
pip install pyTelegramBotAPI
```

Communication

This is a good opportunity to refresh yourself with HTTP concepts through the chapter about web scraping.

The communication with the bot is through HTTP requests.
Each request must follow a specific structure:
https://api.telegram.org/bot< token >/METHOD_NAME?prm1=val1&prm2=val2...

can communicate with our bot by ourselves using an HTTP package (e.g. requests) and the general Telegram API. For demonstrate this idea using the basic bot method - sendMessage.

```
url = 'https://api.telegram.org/bot{token}/
                                    sendMessage?chat_id={chat_id}&text={text}'
resp = requests.get(url)
```

Communication

For get your chatid in telegram

1. Create URL for telegram with method getUpdates

```
url = 'https://api.telegram.org/bot{token}/getUpdates'
```

2. Send Message in your bot
3. Check the chat_id in URL that created.

The requests module

The response from the server contains the data as a JSON file as described in the documentation.

```
resp = requests.get(url)
print(resp.text)
Anwser:
{"ok":true,"result":{"message_id":35,"from":{"id":1594806170,
"is_bot":true,"first_name":"avibot-15","username":"aviyashar_bot"},
"chat":{"id":865138792,"first_name":"Avi","last_name":"Yashar",
"type":"private"},"date":1610816025,"text":"This is my message"}}
```

Example 1-2

The pyTelegramBotAPI package

pyTelegramBotAPI is a Python wrapper for the Telegram API.

Our bot is implemented by the TeleBot class, which encapsulates all API calls in a single class. It provides functions such as send_xyz (send_message, send_document etc.) and several ways to listen for incoming messages.

Again, used authorization token to start the communication, but this time it through a new TeleBot object.

```
bot = telebot.TeleBot(token)
```

Using this object and use the API more easily, e.g.:

```
msg = bot.send_message(chat_id=865138792, text='This is my message')
```

Example 3

Telegram Bot

Exresices:

Send your bot a message saying "Wake up! It's {the current timestamp}".

The Message class

The bot essentially communicates through incoming and outgoing messages, and the message class encapsulates all the message information, including some relevant metadata, e.g. the content and the chat id.

- Print the type of message :

```
print(msg.content_type)
Answer: text
```

- Print the text that received

```
print(msg.text)
Answer: Wake up! It's 07:17:51
```

- Print the chat id

```
print(msg.chat.id)
Answer: 312541264
```

Message handlers

Programming Telegram bots in Python is heavily based on working with decorators.
If you are not familiar with this concept, then you should probably read our chapter about decorators.

A message handler is a function that is decorated with the message_handler decorator of a TeleBot instance. Message handlers consist of one or multiple filters. Each filter must return True for a certain message in order for a message handler to become eligible to handle that message.

```
@bot.message_handler(filters)
def function_name(message):
    bot.reply_to(message, "This is a message handler")
```

filters applied to the message have several options, including content_types, regex, commands & func. More details, along with some simple samples can be found in the docs.

Message handlers

To start the bot we use the polling() method.

```
bot.polling()
```

Replay the message with the same message

```
bot = telebot.TeleBot(test1_token)

@bot.message_handler(func=lambda message: True)
def echo_all(message):
    bot.reply_to(message, message.text)

bot.polling()
```

Example 4-7

Telegram Bot

Exresices:

Create your application with telegram bot examples:

1. Get the price of your stock in Realtime
2. what is the Weather
3. Do something else ...