# Universitatea “1 Decembrie 1918” Alba Iulia

Telegram Data Fetching Script in Python

Munteanu-Bardan Raul Octav

Chapters

1. Introduction
2. Installation and setup
3. Elastic Search
4. Code and functionality
5. Limitations
6. Introduction

This Python script is designed to facilitate the fetching of data from Telegram, gathering messages from different text groups and channels. Each channel and group is stored in the form of the link that represents the group in a text file from which the script reads and accesses each channel and group. The data gathered from Telegram is: the name of the channel or group, the text message, the date on which the message was posted, the reactions (emojis and number of each emoji) and the views. All messages are gathered within a period of days specified by the user and stored in an Elastic Search system.

1. Installation and setup

The environment on which the script was built is PyCharm, which made it easy for the downloading and importing of the libraries. The installation inside of PyCharm requires the Telethon library for Python that allows for API calls on Telegram and the library for Elastic Search. Downloading the library and then importing it was the first step of the installment.

On the other hand, the Elastic Search system requires more than just a download. Elastic Search is a system that enables the storage of data in the same manner a database would. The installation of the Elastic Search was made through Docker, a software designed for containing applications so they can be run on any environment. After downloading the image containing the Elastic Search with the latest version and running a few commands in CMD, a network was created so that the Elastic Search could be installed. Additionally, a Kibana system, which is a web-based UI, part of the Elastic Search stack, that graphically displays all the data in the Elastic Search and facilitates searching for data based on filters.

1. Elastic Search

The port for Elastic Search is 9200 and for the Kibana it is 5601. Both of these can be accessed through any browser by going to the “https://localhost:port”, depending on the system of choice. Elastic Search runs on indices and clusters, which means that data can be clumped together and separated on different locations based on the needs of the user and the category of data. All data stored inside Elastic Search is in JSON format.

1. Code and functionality

There are three methods that describe the Python script:

* greet() – which prompts the user with a description and waits for his input. This method allows the user to choose the period from which the data should be gathered
* get\_group\_messages() – this method gathers all the messages and the afferent data.
* index\_messages\_to\_elasticsearch(messages) – here all the data is dumped inside the Elastic Search system.

The first step of gathering the data is to first establish a connection to the Telegram API so that we can fetch the data. The script requires a valid Telegram account so that it can make all the necessary calls. Upon creating an account and going to the API page of Telegram, each user is prompted with a generated API ID and a hash, which are required for the methods that establish the connection. Validation of the acount is required, prior to accessing the API.

The next step is parsing the file containing all the groups that are fed to the script for data fetching. For each group the messages from the specified period are being gathered and indexed inside a Python dictionary. The dictionary has much in common with a JSON, and the following data is being gathered for each message:

* The name of the channel it belongs to
* The text message itself
* The timestamp of the post
* The number of views for the post
* All the reactions, which consist of one or multiple emojis and their afferent count

After the dictionary was built it is effectively sent to the last method which selects the desired index which will store the data. If the name of the index does not exist, a new one will be created and the data will be sent to it. The library for Elastic Search has a compatibility with the Python dictionaries, for normally, the data sent to the system should be in JSON format. The library allows for dictionaries to be considered valid JSON.

1. Limitations

The only limitations when it comes to the script is the inability to determine which user reacted with which emoji on each message. Aside from this, no other limitations can be specified when it comes to the gathering of data.

The Elastic Search system has the limitation of being on a local machine, meaning that any connection from the outside cannot be done without configuring a remote connection. Of course, this can be done by configuring DDNS and port forwarding, but for the current state and needs of the project, everything can be done locally.