



Telegram Scripts

AUTOMATICH EXPORTEREN VAN GEGEVENS

NAGELS VIKTOR (FGP ANT)

Contents

Inleiding.....	2
Overzicht Scripts	Error! Bookmark not defined.
Installatie.....	2
Scripts.....	3
Exporteren van live user_ids + schermnaam	Error! Bookmark not defined.
Exporteren berichten van 1 individu.....	3
Exporteren media in chats	4
Exporteren van alle user_ids in chat (max 10.000).....	4

Inleiding

Dit document bevat extra uitleg over vier Telegram-scripts die eenvoudige en gemakkelijke exports van gegevens uit Telegram mogelijk maken.

Het eerste script is ontworpen om de `user_ids` en gebruikersnamen van gebruikers binnen een specifieke chat te exporteren. Dit is handig als je een overzicht wilt hebben van de actieve gebruikers in een chat, bijvoorbeeld om gerichte berichten naar hen te sturen.

Het tweede script richt zich op het exporteren van alle verzonden berichten van een individuele gebruiker. Dit is handig om belangrijke gesprekken met specifieke personen te bewaren of om bijvoorbeeld te onderzoeken wat er in een bepaalde chat is besproken.

Het derde script exporteert alle media die is gedeeld in een specifieke chat. Dit is handig als je alleen op zoek bent naar foto's of video's die binnen een chatgesprek zijn gedeeld.

Tot slot exporteert het vierde script de `user_ids` van de leden binnen een specifieke groep. Houd er rekening mee dat dit script beperkt is tot het exporteren van maximaal 10.000 gebruikers. Dit kan handig zijn als je een overzicht wilt hebben van de gebruikers in een groep en hun gebruikersnamen wilt weten.

Installatie

Eerst en vooral is het belangrijk om na te gaan of dat python al geïnstalleerd is. Dit kan door volgend commando uit te voeren in de "Command Prompt".

```
Python -v
```

Wanneer u de python versie terugkrijgt, betekent dit dat python al geïnstalleerd is. Dan mag u de volgende stap over het installeren van python overschakelen.

Installatie python

1. Ga naar de officiële Python-website: <https://www.python.org/downloads/>
2. Klik op de knop 'Downloaden' onder de nieuwste versie van Python. Kies de juiste versie voor jouw besturingssysteem (bijvoorbeeld Windows, macOS of Linux).
3. Voer het installatieprogramma uit en volg de instructies op het scherm. Tijdens de installatie kun je ervoor kiezen om Python toe te voegen aan het PATH-systeem van je computer, wat het gemakkelijker maakt om Python-scripts vanaf elke locatie op je computer uit te voeren.
4. Zodra de installatie is voltooid, open je de opdrachtprompt (Windows) of de terminal (macOS en Linux) en typ je 'python' om te controleren of Python correct is geïnstalleerd. Als Python correct is geïnstalleerd, zou je de Python-interpreter moeten zien opstarten met het versienummer.

Een virtual environment is een afgescheiden omgeving waarin je Python packages kunt installeren zonder de packages op het systeemniveau te installeren. Dit maakt het gemakkelijker om te werken met verschillende projecten die verschillende versies van dezelfde package vereisen. Hier zijn de stappen om te werken met een virtual environment in Python:

1. Installeer het "venv" module door de volgende opdracht in je terminalvenster uit te voeren:

python -m venv myenv

Hierbij vervang je "myenv" door de naam die je wilt geven aan je virtual environment.

2. Activeer de virtual environment door de volgende opdracht in je terminalvenster uit te voeren:

myenv/bin/activate

3. Je kunt nu packages installeren in de virtual environment met behulp van pip. Als je klaar bent met het werken in de virtual environment, kun je deze deactiveren door de opdracht "deactivate" in de terminal uit te voeren.

Om te werken met de packages die in de virtual environment zijn geïnstalleerd, moet je de requirements.txt file uitlezen. Deze file bevat een lijst met alle packages die in de virtual environment zijn geïnstalleerd. Om de juiste packages automatisch te installeren, voer je de volgende opdracht uit in de terminalvenster van de virtual environment:

pip install -r requirements.txt

Dit zal alle packages installeren die in de requirements.txt file staan. Als er nieuwe packages zijn toegevoegd sinds de laatste keer dat de file werd bijgewerkt, zullen deze packages ook worden geïnstalleerd.

Scripts

Exporteren berichten live 1 chat

Deze code wordt gebruikt om berichten van een specifieke gebruiker in een Telegram-chat te exporteren naar een CSV-bestand. Het kan handig zijn in verschillende situaties, zoals:

1. Gegevensanalyse: Door berichten van een specifieke gebruiker in een chat te exporteren, kunnen de gegevens worden geanalyseerd en statistieken worden gegenereerd. Dit kan nuttig zijn voor het identificeren van trends, het analyseren van communicatiepatronen of het verkrijgen van inzicht in het gedrag van gebruikers.
2. Back-up van berichten: De code kan worden gebruikt om een back-up te maken van alle berichten die door een specifieke gebruiker in een chat zijn verzonden. Dit kan handig zijn als u belangrijke informatie wilt bewaren of als u een archief wilt maken van de chatgeschiedenis.
3. Rapportage en compliance: In sommige gevallen moeten berichten worden geëxporteerd voor rapportage- of nalevingsdoeleinden. Bijvoorbeeld in zakelijke omgevingen waar communicatie moet worden gearhiveerd voor juridische of regelgevende doeleinden.

Het gebruik van deze code stelt gebruikers in staat om berichten van een specifieke gebruiker in een Telegram-chat te verzamelen en op te slaan in een gestructureerd formaat (CSV), zodat ze kunnen

worden geanalyseerd, opgeslagen of gerapporteerd volgens de specifieke behoeften van de gebruiker.

Exporteren media in chats

Hier is een overzicht van wat het script doet:

1. Het importeert de vereiste modules, waaronder Telethon, dotenv, tqdm, csv en os.
2. Het laadt de API-ID, API-sleutel en telefoonnummer van de Telegram-client uit een .env-bestand.
3. Vervolgens wordt gevraagd naar de naam van de chat waaruit je media wilt exporteren (bijvoorbeeld een groep of kanaalnaam).
4. Je wordt gevraagd om je keuze te maken tussen het exporteren van foto's (1), video's (2) of bestanden (3).
5. Er zijn twee hoofdfuncties in het script:
 - a. `download_media(group, cl, name, offset_id)`: Deze functie downloadt media van een specifieke chat en slaat deze op in een opgegeven map. Het houdt rekening met de keuze van de gebruiker voor het type media dat moet worden gedownload (foto's, video's of bestanden). Het maakt ook een CSV-bestand aan (images.csv, videos.csv of files.csv) om de informatie over de gedownloade media bij te houden.
 - b. `download_all_media(chat, client, title_Download_Folder)`: Deze functie roept `download_media` herhaaldelijk aan totdat alle berichten zijn gedownload. Het maakt gebruik van een offset-ID om te bepalen welke berichten moeten worden opgehaald.
6. Het script maakt een Telegram-clientverbinding met de opgegeven API-gegevens.
7. Het vraagt de Telegram-server om een lijst met chats op te halen met behulp van de `GetDialogsRequest`-functie.
8. De `download_all_media`-functie wordt aangeroepen om alle media van de opgegeven chat te downloaden.
9. Zodra het downloaden is voltooid, wordt "DONE" afgedrukt.

Opmerking: In de code zijn er enkele opmerkingen (#) en importverklaringen die zijn uitgeschakeld door ze uit te committeren met een hekje (#). Dit geeft aan dat bepaalde functionaliteit mogelijk is uitgeschakeld of nog niet geïmplementeerd.

Exporteren van alle user_ids in chat (max 10.000)

Dit script maakt gebruik van de Telegram-clientbibliotheek genaamd Telethon om gebruikers-ID's uit een Telegram-chat te exporteren naar een CSV-bestand. Het script maakt verbinding met de Telegram API met behulp van de API-ID, API-hash en telefoonnummer die zijn opgeslagen in een .env-bestand.

Het script begint door de nodige modules en bibliotheken te importeren, zoals Telethon, tqdm, csv, os en dotenv. De dotenv-module wordt gebruikt om de omgevingsvariabelen (API-ID, API-hash, telefoonnummer) te laden uit het .env-bestand.

De variabelen `api_id`, `api_hash`, `phone_number` en `chat_name` worden geïnitieerd met de waarden uit de omgevingsvariabelen.

De functie `export_user_ids` wordt gedefinieerd om gebruikers-ID's uit een specifieke chat te exporteren. Het maakt een CSV-bestand aan om de gebruikers-ID's op te slaan. Vervolgens worden alle deelnemers aan de chat opgehaald met behulp van de `iter_participants`-methode van de client. De ID van elke deelnemer wordt vervolgens geschreven naar het CSV-bestand.

In het hoofddeel van het script wordt een Telegram-client geïnstantieerd met behulp van de opgegeven naam, API-ID en API-hash. Vervolgens wordt een oneindige lus gestart. Binnen de lus wordt de functie `export_user_ids` aangeroepen met het opgegeven `chat_name` en de client. Als er een fout optreedt, wordt deze afgedrukt en wordt de lus voortgezet. Als er geen fout optreedt, wordt de lus onderbroken.

Kortom, dit script maakt verbinding met Telegram, zoekt naar deelnemers in een opgegeven chat en exporteert hun gebruikers-ID's naar een CSV-bestand.