**Telegram .onion Link Extractor**

This Python script connects to the Telegram API and extracts .onion links from specified Telegram channels, saving them to a JSON file. This tool was created for the CodeGrills internship assignment.

**Features**

* Uses the Telethon library to connect to Telegram API
* Extracts .onion links from Telegram messages using regex
* Saves links in a structured JSON format
* Tracks the last processed message to avoid duplicate processing
* Handles rate limits and errors gracefully
* Uses async/await for efficient processing

**Requirements**

* Python 3.7+
* Telethon library
* Telegram API credentials

**Setup Instructions**

1. Create a Telegram account if you don't have one
2. Get your API credentials:
   * Go to <https://my.telegram.org/>
   * Log in with your phone number
   * Go to 'API development tools'
   * Create a new application to get your API ID and hash
3. Install required packages:

pip install telethon

1. Set environment variables with your API credentials:
2. export TELEGRAM\_API\_ID='your\_api\_id\_here'

export TELEGRAM\_API\_HASH='your\_api\_hash\_here'

**Usage**

Simply run the script:

python telegram\_onion\_extractor.py

By default, it will:

* Connect to the 'toronionlinks' channel
* Extract .onion links from the most recent messages
* Save results to 'onion\_links.json'
* Store the last processed message ID in 'last\_message\_id.txt'

**Configuration**

You can modify these variables in the script:

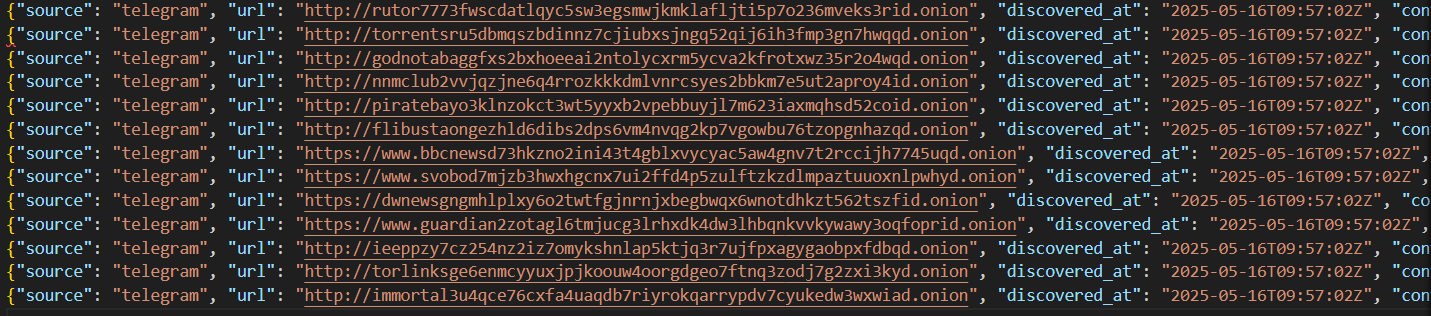
* CHANNEL\_USERNAME: The channel to monitor (default: 'toronionlinks')
* OUTPUT\_FILE: Where to save the extracted links (default: 'onion\_links.json')
* LAST\_MESSAGE\_ID\_FILE: Where to store the last processed message ID (default: 'last\_message\_id.txt')

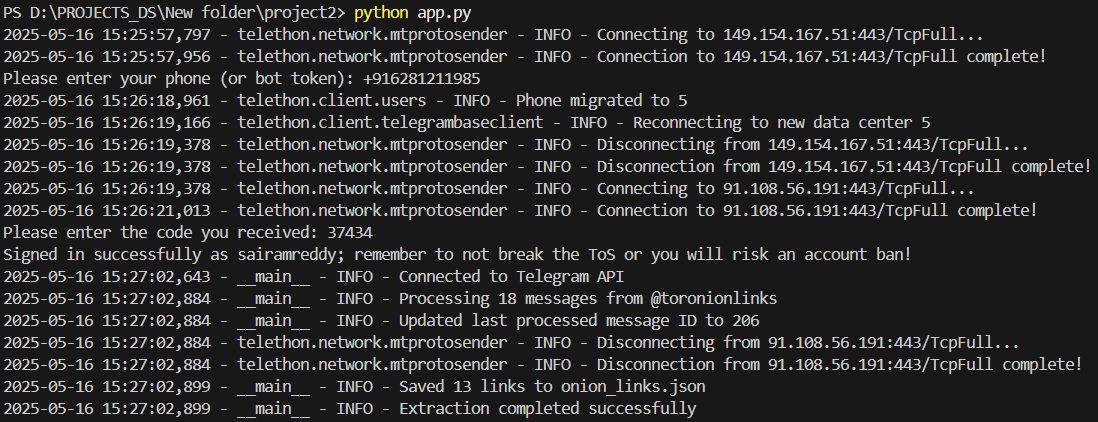
**Sample Output**

When the script runs successfully, it produces a file called onion\_links.json with entries like:

json

{"source": "telegram", "url": "http://example123abc.onion", "discovered\_at": "2025-05-16T15:42:10Z", "context": "Found in Telegram channel @toronionlinks", "status": "pending"}

{"source": "telegram", "url": "http://darksite456def.onion", "discovered\_at": "2025-05-16T15:42:10Z", "context": "Found in Telegram channel @toronionlinks", "status": "pending"}

Each line represents one .onion link discovered in the Telegram channel.  


**Error Handling**

The script includes comprehensive error handling for:

* API connection issues
* Rate limits
* File I/O errors
* General exceptions

All errors are logged with appropriate messages.