# Practical Usage Guide: Telegram .onion Link Extractor

This guide walks you through setting up and running the Telegram .onion Link Extractor script.

## 1. Prerequisites Setup

### A. Obtain Telegram API Credentials

1. Visit my.telegram.org and log in with your Telegram account.
2. Click on "API development tools".
3. Fill in the required fields to create a new application:
   - App title: Onion Link Extractor
   - Short name: onion_extractor
   - Platform: Desktop
   - Description: Tool to extract .onion links from Telegram channels
4. Click "Create Application".
5. Note down your **API ID** (a number) and **API Hash** (a string).

### B. Install Required Software

1. Make sure Python 3.6+ is installed:

   ```bash
   python --version
   ```

2. Install the Telethon library:

   ```bash
   pip install telethon
   ```

## 2. Configure the Script

Open the `telegram_onion_extractor.py` file and update the following variables:

```python
# Telegram API credentials - replace with your own from my.telegram.org
API_ID = 12345  # Replace with your API ID
API_HASH = "abcdef123456789abcdef123456789ab"  # Replace with your API hash
CHANNEL_USERNAME = "toronionlinks"  # The target channel username
```

Notes:

- `API_ID` should be a number (no quotes)
- `API_HASH` should be in quotes
- `CHANNEL_USERNAME` should be the username without the @ symbol

## 3. Running the Script

Execute the script from your terminal:

```bash
python telegram_onion_extractor.py
```

### First Run

On the first run, you'll be prompted to enter your phone number to authenticate with Telegram:

1. Enter your full phone number (with country code, e.g., +12025551234)
2. You'll receive a verification code via Telegram
3. Enter the verification code when prompted

This authentication is only required once. The script creates a session file that stores your authentication for future runs.

### Expected Output

If everything works correctly, you should see output similar to:

```
Connected to Telegram!
Found channel: Onion Links Collection
Extracted 12 .onion links from toronionlinks
Results saved to onion_links.json
Disconnected from Telegram
```

## 4. Examining the Results

The script creates two files:

1. **onion_links.json** - Contains the extracted .onion links in JSON format:

```json
{"source": "telegram", "url": "http://abcdef123456.onion", "discovered_at": "2025-05-15T10:
{"source": "telegram", "url": "http://example7890xyz.onion/forum", "discovered_at": "2025-0
```

2. **last_message_id.txt** - Contains the ID of the last processed message.

## 5. Subsequent Runs

When you run the script again, it will:

1. Read the last processed message ID from `last_message_id.txt`
2. Skip messages that have already been processed
3. Only extract links from new messages
4. Append new findings to `onion_links.json`

## 6. Troubleshooting

### Connection Issues

If you see errors like "Connection failed" or "Network error":

- Check your internet connection
- Verify that your API credentials are correct
- Make sure your IP address is not blocked by Telegram

### Channel Access Issues

If you see "Channel not found" or "Access denied":

- Verify the channel username is correct
- Make sure the channel is public (private channels require membership)
- Try another public channel as a test

### Rate Limiting

If you see "FloodWaitError":

- The script is making too many requests too quickly
- Wait for the specified time before trying again
- Consider adding delays between requests

## 7. Security Notes

- Keep your API credentials private
- The session file contains sensitive authentication data
- Be cautious with .onion links as they may point to illegal content
- This tool is for educational purposes only

## 8. Maintaining the Script

To keep the script running smoothly:

1. Periodically check for updates to the Telethon library:

   ```bash
   pip install --upgrade telethon
   ```

2. Clean up old session files if you're no longer using them:

   ```bash
   rm onion_link_session*
   ```

3. Consider implementing a log rotation system for `onion_links.json` if it grows too large.

By following this guide, you should be able to successfully run the Telegram .onion Link Extractor script and extract links from Telegram channels.