

Introducing the FROG tool for gathering Telegram data

Mobile Media & Communication
2024, Vol. 12(2) 449–453
© The Author(s) 2024



Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/20501579241244973
journals.sagepub.com/home/mmc



Florian Primig 

Institute for Media and Communication Studies, Freie Universität Berlin,
Germany

Fabian Fröschl 

Software Development, Höskolan Kristianstad, Sweden

Telegram data has become more important for social scientists as the messenger service has risen in popularity and other platforms have restricted application programming interface (API) access. Users can create public and private chat rooms, called channels or groups, with a limit of 200,000 members for groups and an unlimited number of members in channels (Telegram, 2023). Telegram has proven relatively resilient against deplatforming and regulation efforts and promises relative high levels of security to its users, which makes it attractive for mobilization efforts by activists (Ebel, 2019) and extremist groups (Krona, 2020; Mahl et al., 2021). Activism and extremism are some of the main foci of research utilizing Telegram data (Curley et al., 2022; Urman et al., 2021).

Data loss is a problem when obtaining Telegram data due to, for instance, banning and voluntary deletion of content in chats (Buehling, 2023). Further, the messenger app allowing users only to download an archive version of a single channel or chat directly in the app as either JSON or html format, which leaves collection of large datasets tedious for researchers with limited coding capacity—an issue as Telegram data is clearly relevant for qualitative researchers (Dowling, 2023) and mixed-methods integrating big and small data become more important (Dehghan et al., 2020). FROG (<https://github.com/Froschi1860/F-R-O-G-Telegram-scraping-tool>) aims at mitigating these issues by offering a user-interface-based solution for Telegram data gathering that is also capable of scraping multiple channels simultaneously with multiple sets of API credentials to mitigate rate limits. Users are able to select timeframes or force a full channel collection. Potential use cases include qualitative and quantitative research as well as higher education teaching.

Current features of FROG

FROG allows users to authorize *multiple API credentials*¹ which then can scrape data simultaneously to speed up the otherwise slow process of data retrieval.² After

registration and authorization of at least one set of API credentials, the user can start the scraper after adding single channel references or a list of such as a text file. Likewise, further channels can be appended to a running job. Users can set a *date or message limit* in the scraper configuration and, for instance, only retrieve the *X* latest messages or all messages from date A to date B. By default, the scraper retrieves all messages. We advise users only to scrape all messages of a channel if it is explicitly necessary for their research purpose. To date, FROG can only extract text messages, message metadata, and channel metadata. The *name-ID-mapper* lets users retrieve a channel's ID from one of its valid references or the correct channel name from a previously saved ID,³ which is useful for monitoring and multi-level snowball sampling (as one channel might be referenced by multiple names and links over time, yet the ID remains).

In the *data view*, users can access and view⁴ their saved datasets and *export* them as either .csv or .json files individually or all at once. Channel metadata can only be exported as .json as of now. Exporting channel messages or channel metadata writes the respective files in the output folder in the directory of FROG. It is crucial to save one's datasets elsewhere for later analyses, as newer exports of the same datasets overwrite the old ones in the output folder. Name-ID mappings can be exported in the same way. Timestamps for messages are converted into CET + 02:00, "NA" is used as a placeholder for empty CSV fields, and flattened messages in CSV exports receive a "_" character as delimiter between parent and child sections of the types. The *error log* feature consistently documents all errors that occur while scraping and app-internally. Error logs can be exported as .csv files. A simple flowchart guiding the user through the steps of using FROG can be found in Figure 1 below.

Potential use cases of FROG

As Telegram is mostly used as a mobile app, a multitude of potential uses in mobile communication research is conceivable. FROG can be used, for instance, to gather data in experimental yet ecologically valid chatgroup settings. Except for individual exposure times, datasets acquired with FROG allow for extensive analyses of chat activity and reach of content. Also in less controlled environments, large datasets can be collected easily. A study using an early version of FROG gathered data for social network analysis and textual analysis of a large network of far-right and coronavirus protestors (Primig, 2024). Since the data gathering requires no coding skills, FROG makes it easier for researchers working within qualitative paradigms to integrate substantial datasets. The multiple credentials option speeds up data retrieval, minimizing data loss. In classrooms, FROG can be used to enable students to participate in computational and mixed-methods courses within the limited timeframe of a term or semester. The software allows students and student assistants to easily gather data independently. Finally, we believe FROG can be useful for monitoring projects that are often subjected to limited funding. FROG can aid them to maintain extensive channel lists and update the database regularly. For future developments, a function that automatically starts the scraping process for a predefined list of channels in pre-set intervals could be implemented. We invite the community to participate in further development of FROG.

Quick-start guide FROG

1. When you open FROG you are greeted with a welcome message directing you to the manual and API registration for Telegram.
After you've registered your API access, go to the credentials tab.
2. In the credentials tab, enter your API credentials and hit "add new credentials".
Now your saved credentials are listed below and need to be validated (you'll get a validation code in Telegram).
3. In the scraper tab settings you can specify a date range, maximum of messages, or nothing (which will then by default mean full scraping of the channels you provide).
You can provide channel references as CSV or txt file lists (one reference per row), or add them individually (also while a scraper is running by "add to queue").
While your scraper is active, a tiny frog jumps across your screen. To check whether the scraper is done, "get active scrapers".
4. In the data view tab, you can now export your data all together or by channel and view individual channel message data.
You're done, congrats!
Remember to save your exports somewhere safe!

The screenshot shows the FROG web interface with four numbered steps highlighted by green arrows:

- 1. Home:** The user is greeted with a welcome message and a "Home" button. A "Welcome to the F-R-O-G Telegram scraping tool" message is displayed.
- 2. Credentials:** The user is prompted to "Add new credentials" with a form for "Your phone number", "Your Telegram ID", "Your Telegram channel", and "Add to queue (CSV)".
- 3. Scraper configuration:** The user is prompted to "Scraper configuration" with fields for "Access token of bot account", "Bot channel ID", "Bot token for sending messages", "Bot channel ID", "Bot token for receiving messages", "Bot token for receiving messages", "Bot token for receiving messages", "Current scraping configuration", and "Scraper Telegram channels".
- 4. Metadata export:** The user is prompted to "Metadata export" with fields for "Phone number of channel ID or reference to channel metadata", "Bot token ID", "Export channel reference", "Get messages by ID", and "Get messages by ID".

Additional tip:

- The Name-ID Mapper helps you to keep track of channel references.
- You might find it helpful for step 3, adding channels to your scraper list.

The screenshot shows the "Name-ID Mapper" tool in the FROG interface. It has a "Find channel by reference" section with fields for "Name" and "Reference". It also has a "Find channel name by ID" section with fields for "ID" and "Name".

Link to the FROG software: <https://github.com/Froschl1860/F-R-O-G-Telegram-scraping-tool/tree/main>



Figure 1. A simple quick-start guide for using FROG.


Declaration of conflicting interests


The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Florian Primig  <https://orcid.org/0000-0002-2034-4367>

Fabian Fröschl  <https://orcid.org/0009-0008-8500-8736>

Notes

1. To register an app for API access, follow the steps outlined here: https://core.telegram.org/api/obtaining_api_id
2. When trying to scrape too many channels in too short a timeframe, Flood Wait Errors can occur which Telegram uses to avoid an overload of their API. FROG tries to avoid such errors but they may occur, which leads to a phone number being unusable for scraping for a period of time. The credentials overview displays the end of the waiting time for a given phone number in case a Flood Wait Error occurred. FROG automatically recognizes when this waiting time is over and makes the phone number available again for scraping. Further, on startup, FROG checks the authorization status for all registered credentials. Thus, usually an authorization must be done only once per phone number.
3. Be aware that a channel name can only be retrieved from an ID if the channel was found via a reference before. This function can only be used while no scraping job is running and requires at least one authorized set of credentials.
4. The viewing function is meant as a quick and convenient way to get a first glimpse of the data. It is not recommended to open large channels in the viewer as this might freeze the program for a while.

References

- Buehling, K. (2023). Message deletion on Telegram: Affected data types and implications for computational analysis. *Communication Methods and Measures*, 18(1), 92–114. <https://doi.org/10.1080/19312458.2023.2183188> .
- Curley, C., Siapera, E., & Carthy, J. (2022). COVID-19 protesters and the far right on Telegram: Co-conspirators or accidental bedfellows? *Social Media + Society*, 8(4), 20563051221129187. <https://doi.org/10.1177/20563051221129187>
- Dehghan, E., Bruns, A., Mitchell, P., & Moon, B. (2020). Discourse-analytical studies on social media platforms: A data-driven mixed-methods approach. In Lind, R. A. (Ed.), *Digital Formations: Vol. 3. Producing theory in a digital world: The intersection of audiences and production in contemporary theory* (pp. 159–177). Peter Lang.
- Dowling, M.-E. (2023). News to me: Far-right news sharing on social media. *Information, Communication & Society*, 27(1), 39–55. <https://doi.org/10.1080/1369118X.2023.2166796> .
- Ebel, F. (2019, September 15). Outlawed app Telegram emerges as key tool for Russian protesters. *The Denver Post*. <https://www.denverpost.com/2019/09/15/russia-protest-telegram-app/>

- Krona, M. (2020). Mediating Islamic State collaborative media practices and interconnected digital strategies of Islamic State (IS) and Pro-IS supporter networks on Telegram. *International Journal of Communication*, 14(2020), 23. <https://ijoc.org/index.php/ijoc/article/view/9861>
- Mahl, D., Zeng, J., & Schäfer, M. S. (2021). From “Nasa lies” to “reptilian eyes”: Mapping communication about 10 conspiracy theories, their communities, and main propagators on Twitter. *Social Media + Society*, 7(2), 205630512110174. <https://doi.org/10.1177/20563051211017482>
- Primig, F. (2024). Thinking different as an act of resistance: Reconceptualizing the German protests in the COVID-19 pandemic as an emergent counter-knowledge order. *Discourse & Society*. <https://doi.org/10.1177/09579265241231593>
- Telegram (2023, February 23). *Telegram FAQ*. <https://telegram.org/faq?setln=en>
- Urman, A., Ho, J.C.-T., & Katz, S. (2021). Analyzing protest mobilization on Telegram: The case of 2019 anti-extradition bill movement in Hong Kong. *PLoS ONE*, 16(10), e0256675. <https://doi.org/10.1371/journal.pone.0256675>

Author biographies

Florian Primig is a research associate at the Institute for Media and Communication Studies at Freie Universität Berlin in the Division Digitalization and Participation and associate researcher in the research group 'Digital News Dynamics' at the Weizenbaum Institute for the Networked Society. His research interests include media trust, digital and social media as well as mis- and disinformation.

Fabian Fröschl is a student of Software Development at Högskolan Kristianstad.