**Документация консольного клиента tg\_cli**

Версия пакета: 0.1.0  
Дата: 18 апреля 2025 г.

**1. Назначение**

tg\_cli — консольное приложение на Python (Windows cmd), позволяющее просматривать список диалогов Telegram, быстрo переходить к чатам, читать новые сообщения и отвечать, не открывая GUI‑клиент.

**2. Установка**

pip install telethon==1.39.0 telethon-tgcrypto rich aioconsole ruamel.yaml

# или склонируйте репозиторий и установите в editable‑режиме

pip install -e .

**3. Подготовка конфигурации**

При первом запуске создаётся файл %USERPROFILE%\.tg\_cli.yaml со значениями по умолчанию:

page\_size: 20 # строк в таблице диалогов

theme: default # зарезервировано под будущие темы

only\_unread: false # фильтр "только непрочитанные" по умолчанию

api\_id: <введите>

api\_hash: <введите>

Добавьте свои api\_id и api\_hash (их можно получить на https://my.telegram.org).

**4. Запуск**

tgcli

Первый старт:

1. Введите номер телефона.
2. Введите код проверки из Telegram.
3. В каталоге %USERPROFILE%\.tg\_cli будет сохранён файл session.session и строковая сессия session.txt.

**5. Навигация и горячие клавиши**

**5.1 Состояние Диалоги**

|  |  |
| --- | --- |
| **Клавиша** | **Действие** |
| n / p | Следующая / предыдущая страница |
| u | Переключить фильтр «только непрочитанные» |
| числа + Enter | Открыть чат под соответствующим номером строки |
| q | Выход из программы |

*Буфер многозначного ввода* отображается в футере таблицы; очищается при навигации.

**5.2 Состояние Чат**

|  |  |
| --- | --- |
| Клавиша | Действие |
| ↑ / ↓ | Перемещение по сообщениям (выделение стрелкой) |
| u | Загрузить ещё 100 старых сообщений |
| r | Ответить на выделенное сообщение (появляется prompt reply>) |
| b | Назад к списку диалогов (сообщения помечаются прочитанными) |

*Все новые ответы отправляются с привязкой* ***reply\_to****, поэтому в GUI Telegram будет видна цитата.*

**6. Архитектура проекта**

tg\_cli/

├── app.py # цикл UI, маршрутизация клавиш

├── state.py # dataclass‑ы курсоров и глобальное состояние

├── config.py # YAML‑конфиг + дефолты

├── api/client.py # обёртка Telethon + cryptg

├── views/

│ ├── dialogs.py # рендер таблицы диалогов (Rich)

│ └── chat.py # рендер окна чата, подсветка строки

├── input/keys.py # асинхронное чтение клавиш (Windows cmd)

├── cli.py # entry‑point «tgcli»

└── plugins/ # место для будущих расширений

**6.1 Основные классы**

* **TGCLIApplication** — точка входа, управляет Live‑обновлением экрана.
* **TelegramClientWrapper** — тонкий фасад над Telethon, скрывающий session‑логику.
* **AppState / DialogCursor / ChatCursor** — структуры для хранения положения пользователя.

**7. Работа с медиаконтентом**

На данном этапе медиа отображаются как текст‑заглушки (<медиа>).  
Запланировано: скачивание по клавише d и открытие через start/xdg-open.

**8. Расширение функциональности**

Плагин‑API (папка plugins/) позволяет регистрировать новые команды:

from tg\_cli.plugins import register\_command

async def archive\_current\_chat(app\_state, client):

...

register\_command('a', 'chat', archive\_current\_chat)

**9. Известные ограничения**

1. **ANSI‑цвета** требуют Windows 10 1607+.
2. Высокие DPI‑терминалы могут сдвигать псевдографику.
3. Многостраничные чаты (10 000+ сообщений) могут занимать заметно больше оперативной памяти.

**10. Планы на версию 0.2**

* Поиск (/), скачивание медиа (d, v), цветовые темы.
* Интеграция с GitHub Actions: линтер ruff, pytest с мок‑Telegram‑сервером.
* Режим «export chat» — сохранение истории в Markdown/HTML.

**Автор / Поддержка**

Разработчик CLI — @your‑nick.  
Для багрепортов и предложений: issues в репозитории либо команда h внутри программы (будет открывать справку с ссылкой).