

Пользовательская Инструкция - v02

Пользовательская инструкция

1. Назначение бота

Бот предназначен для переноса сообщений и структуры чатов из Telegram в eXpress.

С помощью бота пользователь может:

- подключить учетную запись Telegram;
 - получить список доступных Telegram-чатов;
 - выполнить перенос отдельного чата;
 - запустить массовый перенос нескольких чатов;
 - повторно выполнить перенос или обработать ранее неуспешно перенесенные элементы;
 - добавить пользователей в уже созданный чат eXpress с использованием Excel-файла;
 - импортировать чат из Telegram export archive без использования Telethon.
-

2. Основные термины

`source_chat_id` — идентификатор исходного Telegram-чата. Может быть получен с помощью команды `/chats`.

Целевой чат (`target chat`) — чат в eXpress, который создается ботом либо используется для выполнения переноса.

Частичный перенос (`partial migration`) — сценарий переноса, при котором часть участников не добавляется автоматически из-за отсутствия возможности их сопоставления.

3. Подготовка к началу работы

Для использования режима переноса через Telethon требуется не только доступ к боту, но и отдельная учетная запись Telegram, от имени которой сервис будет выполнять чтение чатов и истории сообщений.

3.1. Подготовка учетной записи Telegram

Рекомендуется использовать отдельную рабочую учетную запись Telegram, а не личный аккаунт.

Требования к учетной записи:

- учетная запись должна иметь доступ к чатам Telegram, которые планируется переносить;
- если требуется перенос исторических сообщений, учетная запись должна иметь доступ к соответствующей истории в Telegram;
- при включенной двухфакторной аутентификации необходимо заранее подготовить пароль для команды `/connect` ;
- для переноса групповых чатов и каналов учетная запись должна обладать достаточными правами для чтения состава участников и истории сообщений.

3.2. Создание Telegram application для Telethon

Для работы Telethon необходимо получить параметры `api_id` и `api_hash` .

Порядок действий:

1. Открыть сайт <https://my.telegram.org> .
2. Авторизоваться под номером телефона той учетной записи Telegram, которая будет подключаться к боту.



Delete Account or Manage Apps

Log in here to **manage your apps** using Telegram API or **delete your account**. Enter your number and we will send you a confirmation code via Telegram (not SMS).

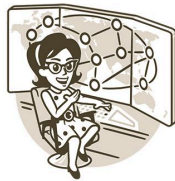
Your Phone Number

Please enter your number in [international format](#)

Next

3. Перейти в раздел [API development tools](#) .

[Home](#) [FAQ](#) [Apps](#) [API](#) [Protocol](#)



Your Telegram Core

- [API development tools](#)
- [Delete account](#)
- [Log out](#)

4. Создать новое приложение.

App configuration

| | | |
|---------------|--|---|
| App api_id: | <input type="text" value="20225351"/> | 🔒 |
| App api_hash: | <input type="text" value="86 [REDACTED]"/> | 🔒 |
| App title: | <input type="text" value="MyFirstApp17728"/> | |
| Short name: | <input type="text" value="myfapp17728"/> | |

alphanumeric, 5-32 characters

При создании приложения, как правило, требуется заполнить следующие поля:

- **App title** — произвольное понятное название, например `ExpressMigrationBot` ;
- **Short name** — краткое название латиницей без пробелов, например `expressmigratebot` .

После сохранения будут отображены параметры:

- **App api_id**
- **App api_hash**

Указанные значения необходимо использовать при настройке переменных окружения.

Примечание: данные параметры запрашиваются однократно. В дальнейшем остальные пользователи будут использовать только номер телефона в рамках самого бота.

3.3. Передача параметров в инфраструктуру

Для настройки сервиса необходимо передать вашему DevOps-специалисту следующие параметры:

- `EXTG_TELEGRAM__API_ID`

- EXTG_TELEGRAM__API_HASH

Без указанных параметров бот не сможет установить Telethon-сессию.

Важно учитывать следующее:

- `api_hash` является конфиденциальным значением и не должен передаваться через незащищенные каналы связи;
- `api_id` и `api_hash` привязаны к Telegram application и должны храниться с тем же уровнем защиты, что и учетные данные;
- при изменении приложения или Telegram-учетной записи необходимо повторно проверить корректность подключения Telethon.

3.4. Проверка готовности перед первым запуском `/connect`

Перед началом работы необходимо убедиться, что:

- `api_id` и `api_hash` уже добавлены в окружение сервиса;
- требуемая Telegram-учетная запись может пройти авторизацию в Telegram с телефона или desktop-клиента;
- имеется доступ к SMS либо коду подтверждения в Telegram;
- при включенной 2FA известен пароль;
- учетная запись уже добавлена в необходимые исходные чаты и имеет доступ к истории, подлежащей переносу.

3.5. Если использование Telethon невозможно

Если отсутствует возможность создать Telegram application или подключить пользовательскую сессию, следует использовать альтернативный сценарий:

```
/import_archive
```

Данный режим работает на основе Telegram export archive и не требует Telethon, однако имеет ограничения:

- участники чата не переносятся;
- вложения не переносятся;
- сценарий не предоставляет полноценный live-доступ к Telegram и предназначен только для архивного импорта.

4. Быстрый старт

1. Подключить Telegram-учетную запись:

```
/connect +79990001122
```

2. Проверить, что сессия успешно подключена:

```
/account
```

3. Получить список доступных чатов:

```
/chats
```

4. Определить необходимый `source_chat_id`.

5. Запустить перенос:

```
/migrate <source_chat_id>
```

6. Отслеживать прогресс выполнения:

```
/status <source_chat_id>
```

5. Основные сценарии работы

5.1. Подключение учетной записи Telegram

Команда:

```
/connect +79990001122
```

После запуска команды:

- бот запросит код подтверждения из Telegram;
- если для учетной записи включена двухфакторная аутентификация, бот запросит пароль;
- после завершения процедуры учетная запись будет привязана к пользователю, выполнившему подключение.

Перед выполнением команды должны быть предварительно настроены параметры `EXTG_TELEGRAM__API_ID` и `EXTG_TELEGRAM__API_HASH`.

Проверка подключения:

```
/account
```

Отключение учетной записи:

```
/disconnect
```

5.2. Просмотр доступных чатов

Команда:

```
/chats
```

Дополнительные варианты использования:

```
/chats query=project  
/chats limit=50
```

В ответ бот отображает:

- `source_chat_id`;
- тип чата;
- наименование чата;
- признак наличия ранее созданной настройки;
- сведения о наличии прогресса по переносу.

5.3. Просмотр пользователей чата

Команда:

```
/chat_users <source_chat_id>
```

Команда используется для предварительной оценки:

- каких пользователей бот уже способен сопоставить с учетными записями eXpress;
- для каких пользователей потребуется ручное сопоставление;
- каких участников, возможно, потребуется добавить вручную после завершения переноса.

5.4. Настройка параметров чата перед переносом

Если команда вызывается без дополнительных параметров, бот отображает текущую конфигурацию:

```
/configure <source_chat_id>
```

Примеры настройки:

```
/configure <source_chat_id> access=link  
/configure <source_chat_id> media=off  
/configure <source_chat_id> format=quote  
/configure <source_chat_id> from=2026-01-01T00:00:00+03:00 to=2026-03-  
01T00:00:00+03:00
```

Как правило, настраиваются следующие параметры:

- `access=invite|link` — способ предоставления доступа к результату переноса;
- `media=on|off` — необходимость переноса вложений;
- `format=quote|source_id|none` — формат оформления ссылок и ответов;
- `from / to` — диапазон сообщений, подлежащих переносу.

Пример ответа бота на `/configure <source_chat_id>` :

```
migration_id=tg_to_express_bot_managed
source_chat_id=-1001234567890
source_chat_type=supergroup
source_chat_title=Release Discussion
target_strategy=create
access=direct_add
format=inline_quote
media=on
skip_in_all=no
target_title=Release Discussion
target_chat_id=-
include_from=-
include_to=-
topic_strategy=split_by_topic
telegram_chat_id=-1001234567890
```

Расшифровка полей:

- `migration_id` — идентификатор текущего migration flow.
- `source_chat_id` — идентификатор исходного Telegram-чата или логического topic-чата.
- `source_chat_type` — тип исходного чата: `private`, `group`, `supergroup`, `channel`.
- `source_chat_title` — название исходного чата в Telegram.
- `target_strategy` — способ выбора целевого чата в eXpress:
 - `create` — создать новый чат;
 - `bind` — использовать уже существующий `target_chat_id`.
- `access` — фактическая стратегия доступа в конфиге:
 - `direct_add` — бот пытается добавить участников автоматически;
 - `invite_link` — бот рассылает ссылки на вход.
- `format` — фактический режим оформления reply/source-ссылок:
 - `inline_quote`
 - `source_id`
 - `none`
- `media` — переносить ли вложения.
- `skip_in_all` — исключен ли чат из `/migrate_all`.
- `target_title` — имя целевого чата в eXpress.

- `target_chat_id` — ID целевого чата, если используется `bind` или чат уже был создан.
- `include_from` / `include_to` — временные границы переноса.
- `topic_strategy` — стратегия работы с Telegram forum topics:
 - `single_chat` — все темы переносятся в один чат;
 - `split_by_topic` — каждая тема переносится в отдельный чат.
- `telegram_chat_id` — физический ID исходного Telegram-форума; полезен, если `source_chat_id` логический, например `-100...#topic:35892`.
- `source_thread_id` — ID top-message темы в Telegram.
- `source_thread_title` — название конкретной темы.

Важно:

- в команде используется `access=invite|link`, а в ответе сохраняется `direct_add|invite_link`;
- в команде используется `format=quote`, а в ответе сохраняется `inline_quote`.

5.5. Перенос одного чата

Команда:

```
/migrate <source_chat_id>
```

В процессе переноса возможны дополнительные шаги:

- если исходный чат является личным (`personal chat`), бот может запросить указание второго участника через `mention`, `email` или `huid`;
- если переносится группа или канал, бот может направить Excel-файл для сопоставления участников;
- если часть пользователей не сопоставлена, допускается выполнение команды `/skip` для продолжения частичного переноса;
- для остановки мастера (`wizard`) или задачи может использоваться команда `/cancel`.

Полезные параметры:

```
/migrate <source_chat_id> batch=200
/migrate <source_chat_id> progress=resume
/migrate <source_chat_id> target_title="Новый чат"
```

5.5.1. Перенос `supergroup` с топиками

Если исходный чат является `forum/supergroup` с темами, можно выбрать одну из двух стратегий:

```
/migrate -1001421126432 topics=single_chat  
/migrate -1001421126432 topics=split_by_topic
```

Варианты поведения:

- `topics=single_chat` — весь форум переносится в один целевой чат.
- `topics=split_by_topic` — каждая тема переносится в отдельный целевой чат.

Для `topics=split_by_topic` бот создает логические `source_chat_id` вида:

```
-1001421126432#topic:35892  
-1001421126432#topic:35887
```

Именно они затем используются в статусе и внутренних `job`.

Пример запуска:

```
/migrate -1001421126432 topics=split_by_topic batch=300 progress=resume
```

Типичный ответ бота:

```
Задачи приняты.  
operation=migrate_chat  
migration_id=tg_to_express_bot_managed  
accepted_jobs=2  
accepted_source_chat_ids=-1001421126432#topic:35892,-1001421126432#topic:35887  
job_keys:  
- migrate_chat:tg_to_express_bot_managed:-1001421126432#topic:35892:....  
- migrate_chat:tg_to_express_bot_managed:-1001421126432#topic:35887:....
```

Расшифровка:

- `accepted_jobs` — сколько фоновых задач было поставлено в очередь.
- `accepted_source_chat_ids` — какие `logical topic chats` реально ушли в обработку.
- `job_keys` — ключи отдельных фоновых задач по темам.

Практически это означает, что при `split_by_topic` нужно смотреть прогресс не только по `root forum`, но и по `topic child IDs`.

5.5.2. Перенос канала

Для канала используется та же команда:

```
/migrate <source_chat_id>
```

Пример:

```
/migrate -1001234567000 access=link progress=resume
```

Особенности сценария для `channel` :

- бот проверяет доступ текущей Telegram-сессии к каналу;
- если участников канала можно прочитать и среди них есть несопоставленные пользователи, бот может выдать Excel для сопоставления;
- если стратегия доступа к подписчикам явно не выбрана, бот может попросить выбрать ее в wizard;
- если автоматическое добавление подписчиков невозможно, используется сценарий со ссылкой на вход.

Для `channel` особенно важно:

- Telegram-учетная запись должна иметь достаточные права в исходном канале;
- если нужно переносить подписчиков автоматически, учетная запись обычно должна быть администратором канала;
- если переносится только история сообщений, можно использовать `access=link`, чтобы не блокировать миграцию на авто-добавлении подписчиков.

5.6. Массовый перенос всех доступных чатов

Команда:

```
/migrate_all
```

Данная команда:

- получает перечень всех доступных Telegram-чатов;
- автоматически создает недостающие конфигурации;
- запускает перенос каждого чата в виде отдельной фоновой задачи.

Особенности:

- для автоматически создаваемой конфигурации по умолчанию используется `access=link`;
- если для конкретного чата требуется дополнительное подтверждение или действие, бот останавливается на соответствующем шаге;
- массовый перенос не изменяет правило владения чатом: одним и тем же чатом управляет только его владелец.

5.7. Добавление пользователей в уже мигрированный чат

Команда:

```
/add_users
```

или для конкретного чата:

```
/add_users <source_chat_id>
```

Порядок работы:

1. бот отображает список уже мигрированных чатов либо сразу формирует Excel-файл;
2. пользователь заполняет файл;
3. заполненный файл загружается обратно в тот же диалог;
4. бот добавляет пользователей в существующий целевой чат.

Допустимые колонки для заполнения:

- target_huid
- corporate_email
- ad_login
- other_id

При наличии нескольких заполненных полей используется следующий приоритет:

1. target_huid
2. corporate_email
3. ad_login
4. other_id

Важно:

- допускается добавление новых пользователей, которых не было в исходном Telegram-чате;
- в таком случае Telegram-поля можно оставить пустыми, указав только корпоративный идентификатор.

Пример ответа после загрузки заполненного файла:

```
Пользователи обработаны.  
source_chat_id=-5186712067  
target_chat_id=0e8f4c05-4db0-5914-b315-01211cda8d01  
target_chat_title=TestExpressChat  
requested_access_strategy=direct_add  
effective_result=partial_direct_add
```

```
note=direct_add could not add all users; invite_link fallback was used
processed_rows=4
imported_identity_mappings=4
mapping_skipped_rows=0
resolved_targets=4
direct_added=3
invited=1
skipped_rows=0
failed_to_add=1
failed_user=corporate_email=user@example.ru
```

Расшифровка полей ответа:

- `source_chat_id` — исходный Telegram-чат, к которому относится `workbook`.
- `target_chat_id` — целевой чат eXpress, куда добавлялись пользователи.
- `target_chat_title` — название целевого чата.
- `requested_access_strategy` — стратегия, которую система пыталась применить:
 - `direct_add`
 - `invite_link`
- `effective_result` — фактический итог выполнения:
 - `direct_add` — все пользователи добавлены напрямую;
 - `partial_direct_add` — часть добавлена напрямую, часть обработана `fallback`-сценарием;
 - `invite_link` — вместо прямого добавления отправлены ссылки;
 - возможны и другие `service-specific` значения, если `flow` деградировал.
- `note` — пояснение, если сработал `fallback`.
- `processed_rows` — количество строк `workbook`, реально обработанных сервисом.
- `imported_identity_mappings` — сколько строк использовано для сохранения новых Telegram→eXpress соответствий.
- `mapping_skipped_rows` — сколько строк не дало нового `identity mapping`, хотя сам пользователь был обработан.
- `resolved_targets` — сколько целевых пользователей eXpress было успешно найдено.
- `direct_added` — сколько пользователей реально добавлено в чат напрямую.
- `invited` — сколько пользователей получило `invite link` вместо `direct add`.
- `skipped_rows` — сколько строк было пропущено как пустые, `self`-строки или нерелевантные.
- `failed_to_add` — сколько пользователей не удалось ни добавить, ни обработать `fallback`.
- `failed_user` — человекочитаемый идентификатор проблемной строки `workbook`.

5.8. Импорт Telegram export archive без Telethon

Команда:

```
/import_archive
```

Порядок действий:

1. экспортировать чат из Telegram;
2. направить боту архив в формате `.json`, `.zip` или `.rar`.

В результате бот:

- создаст целевой чат;
- назначит инициатора администратором;
- перенесет только текстовые сообщения.

Ограничения режима:

- участники не добавляются;
- вложения не переносятся;
- режим предназначен для отдельных случаев, когда использование Telethon невозможно или нежелательно.

5.9. Повторный перенос и обработка ошибок

Повторный перенос всего чата:

```
/remigrate <source_chat_id>
```

Повторная обработка только неуспешных элементов:

```
/retry_failed  
/retry_failed <source_chat_id>  
/retry_failed <source_chat_id> limit=100 batch=50
```

Команды рекомендуется использовать в случаях, когда:

- перенос ранее уже запускался;
- часть сообщений или вложений не была успешно перенесена;
- требуется продолжить обработку без полного запуска с нуля.

5.10. Проверка прогресса выполнения

Статус по отдельному чату:

```
/status <source_chat_id>
```

Общий статус:

```
/status
```

Сводная статистика:

```
/stats
```

Данные команды позволяют:

- определить наличие активных фоновых задач;
- увидеть количество уже импортированных сообщений;
- проверить наличие элементов в статусах `failed`, `ambiguous`, `processing`.

5.11. Остановка выполнения

Отмена всех активных задач пользователя:

```
/cancel
```

Отмена задачи по конкретному чату:

```
/cancel <source_chat_id>
```

Команда выполняет следующие действия:

- останавливает активный мастер (`wizard`);
- снимает задачи в состоянии `queued`;
- для задач в состоянии `running` отправляет запрос на остановку.

После выполнения отмены рекомендуется через несколько секунд проверить текущее состояние:

```
/status
```

6. Справочник команд

| Команда | Назначение | Пример |
|--------------------|-------------------------------|--------------------|
| <code>/help</code> | Вывод списка доступных команд | <code>/help</code> |

| Команда | Назначение | Пример |
|---|--|---|
| <code>/connect [PHONE]</code> | Подключение учетной записи Telegram | <code>/connect +79990001122</code> |
| <code>/account</code> | Проверка Telegram-сессии | <code>/account</code> |
| <code>/disconnect</code> | Отключение Telegram-сессии | <code>/disconnect</code> |
| <code>/chats [query=TEXT] [limit=N]</code> | Получение списка доступных Telegram-чатов | <code>/chats query=finance limit=30</code> |
| <code>/chat_users <source_chat_id></code> | Просмотр пользователей конкретного Telegram-чата | <code>/chat_users -1001234567890</code> |
| <code>/configure <source_chat_id> ...</code> | Просмотр или изменение параметров переноса | <code>/configure -1001234567890 media=off access=link</code> |
| <code>/migrate <source_chat_id></code> | Запуск переноса одного чата | <code>/migrate -1001234567890</code> |
| <code>/migrate_all</code> | Запуск массового переноса всех доступных чатов | <code>/migrate_all</code> |
| <code>/remigrate <source_chat_id></code> | Повторный перенос чата | <code>/remigrate -1001234567890</code> |
| <code>/retry_failed [source_chat_id] [limit=N] [batch=N]</code> | Повторная обработка только неуспешных элементов | <code>/retry_failed -1001234567890 limit=50 batch=25</code> |
| <code>/add_users [source_chat_id]</code> | Добавление пользователей в уже созданный чат через Excel | <code>/add_users -1001234567890</code> |
| <code>/import_archive</code> | Импорт Telegram export archive | <code>/import_archive</code> |
| <code>/map_identity ...</code> | Сохранение ручного соответствия между пользователем Telegram и eXpress | <code>/map_identity tg_id=123456 email=user@company.ru</code> |
| <code>/status [source_chat_id]</code> | Просмотр статуса переноса | <code>/status -1001234567890</code> |
| <code>/stats</code> | Просмотр общей статистики | <code>/stats</code> |
| <code>/cancel [source_chat_id]</code> | Остановка всех активных задач | <code>/cancel -1001234567890</code> |

| Команда | Назначение | Пример |
|---------|---|--------|
| | пользователя либо одной конкретной задачи | |

7. Ручное сопоставление пользователя

Если бот не смог автоматически сопоставить пользователя Telegram с пользователем eXpress, это можно сделать вручную.

Пример:

```
/map_identity tg_id=123456789 email=user@company.ru
```

или:

```
/map_identity username=nickname email=user@company.ru
```

Данную команду рекомендуется использовать:

- до запуска миграции;
- после выполнения команды `/chat_users` ;
- в случаях, когда в мастере (wizard) отображается информация о нераспознанном пользователе.

8. Действия в типовых ситуациях

8.1. Бот направил Excel-файл

Необходимо:

- заполнить файл;
- загрузить его обратно в тот же диалог;
- при необходимости продолжить перенос без нераспознанных пользователей с помощью команды `/skip` , если бот предлагает такой вариант.

8.2. Требуется перенос чата без Telethon

Следует использовать команду:

```
/import_archive
```

После этого необходимо загрузить экспорт из Telegram в формате `.json`, `.zip` или `.rar`.

8.3. Требуется только добавить пользователей в уже существующий чат

Следует использовать команду:

```
/add_users
```

Для этой цели не требуется запускать новый перенос через `/migrate`.

8.4. Перенос выполняется слишком долго

Рекомендуется проверить текущее состояние с помощью команд:

```
/status  
/status <source_chat_id>  
/stats
```

При необходимости можно:

- повторно обработать только ошибки через `/retry_failed`;
- остановить перенос конкретного чата через `/cancel <source_chat_id>`;
- остановить все активные задачи пользователя через `/cancel`.

9. Ограничения и практические замечания

- Один и тот же исходный чат может быть закреплен только за первым оператором, начавшим с ним работу.
- Пользователь видит и может управлять только своими миграциями и собственной Telegram-сессией.
- В режиме `archive-flow` участники и вложения не переносятся.
- В отдельных сценариях бот намеренно переводит процесс в режим пошагового мастера (`wizard`) для подтверждения доступа, участников или способа сопоставления.