

 **tmat** сегодня в 18:08

Всё, о чём должен знать разработчик Телеграм-ботов

Мессенджеры, API



Вы вряд ли найдете в интернете что-то про разработку ботов, кроме документаций к библиотекам, историй "как я создал такого-то бота" и туториалов вроде "как создать бота, который будет говорить hello world". При этом многие неочевидные моменты просто нигде не описаны.

Как вообще устроены боты? Как они взаимодействуют с пользователями? Что с их помощью можно реализовать, а что нельзя?

Подробный гайд о том, как работать с ботами — под катом.

▼ [Содержание](#)

Начало работы

- Telegram API vs Telegram Bot API
- На чём пишут Телеграм-ботов
- Создание бота
- Юзернеймы
- Оформление бота

Сообщения и чаты

- Запуск бота пользователем
- Сообщения
- Куда может писать бот
- Супергруппы
- id пользователей и чатов
- id сообщений
- Видимость сообщений в группах
- Команды
- Разметка сообщений
- Кнопки

Взаимодействие с ботом

- Ссылки на бота
- Инлайн-режим
- Создание наборов стикеров
- Платежи через ботов
- HTML-игры в ботах
- Telegram Login Widget

Разработка ботов

- Какие апдейты можно получать
- Лимиты
- Рассылка по пользователям
- Смена владельца бота
- Локальный сервер Bot API
- Юзерботы

Заключение

Начало работы

Telegram API vs Telegram Bot API

Рассказываю по порядку.

Телеграм использует собственный протокол шифрования [MTProto](#). MTProto API (он же Telegram API) — это API, через который ваше приложение Телеграм связывается с сервером. Telegram API полностью открыт, так что любой разработчик может написать свой клиент мессенджера.

Для написания ботов был создан Telegram Bot API — надстройка над Telegram API. Перевод с [официального сайта](#):

Чтобы использовать Bot API, вам не нужно ничего знать о том, как работает протокол шифрования MTProto — наш вспомогательный сервер будет сам обрабатывать все шифрование и связь с Telegram API. Вы соединяетесь с сервером через простой HTTPS-интерфейс, который предоставляет простую версию Telegram API.

Среди упрощений Bot API: работа через вебхуки, упрощенная разметка сообщений и прочее.

Почему-то мало кто знает о том, что боты могут работать напрямую через Telegram API. Более того, таким образом можно даже обойти некоторые ограничения, которые даёт Bot API.

[Об авторизации ботов через Telegram API в официальной документации](#)

Вся информация ниже будет по умолчанию относиться и к Bot API, и к Telegram API. О различиях я буду упоминать. От некоторых ограничений Bot API можно избавиться с помощью локального сервера, об этом в конце статьи.

На чём пишут Телеграм-ботов

Бот должен уметь отправлять запросы Телеграм-серверу и получать от него апдейты (updates, обновления).

▼ [Как получать апдейты в Bot API](#)

Получать апдейты можно одним из двух способов:

- Поллинг — просто регулярно отправлять запрос к серверу Телеграма для получения обновлений,
- Вебхук — сделать так, чтобы Телеграм сам отправлял запросы по нужному URL.

Конечно, удобнее использовать библиотеки, чем делать http-запросы "руками".

Если вы попробуете [загуглить](#), как написать Телеграм-бота на Python, вам предложат воспользоваться библиотеками python-telegram-bot и telebot. Но не стоит.

Ну, если вы только хотите познакомиться с разработкой ботов и написать своего hello-world-бота, то можете, конечно использовать и их. Но эти библиотеки могут далеко не всё. Среди разработчиков ботов лучшей библиотекой для ботов на Python считается [aiogram](#). Она асинхронная, использует декораторы и содержит удобные инструменты для разработки. Ещё был хороший [Rocketgram](#), но он давно не обновлялся.

Также ботов часто пишут на JavaScript, для этого обычно используется [Telegraf](#). Библиотеки есть и для многих других языков, но используют их реже.

Если же вы хотите использовать Telegram API, то можете воспользоваться Python'овскими [Telethon](#) и [Pyrogram](#).

▼ [Пример кода бота](#)

Если вы хотите получить какое-то представление о том, как вообще выглядит код бота, вот вам пример использования на [aiogram](#) с [его странички на GitHub](#):

```
import asyncio
from aiogram import Bot, Dispatcher, types

async def start_handler(event: types.Message):
    await event.answer(
        f"Hello, {event.from_user.get_mention(as_html=True)} ?!",
        parse_mode=types.ParseMode.HTML,
    )

async def main():
    bot = Bot(token=BOT_TOKEN)
    try:
        disp = Dispatcher(bot=bot)
        disp.register_message_handler(start_handler, commands={"start", "restart"})
        await disp.start_polling()
    finally:
        await bot.close()

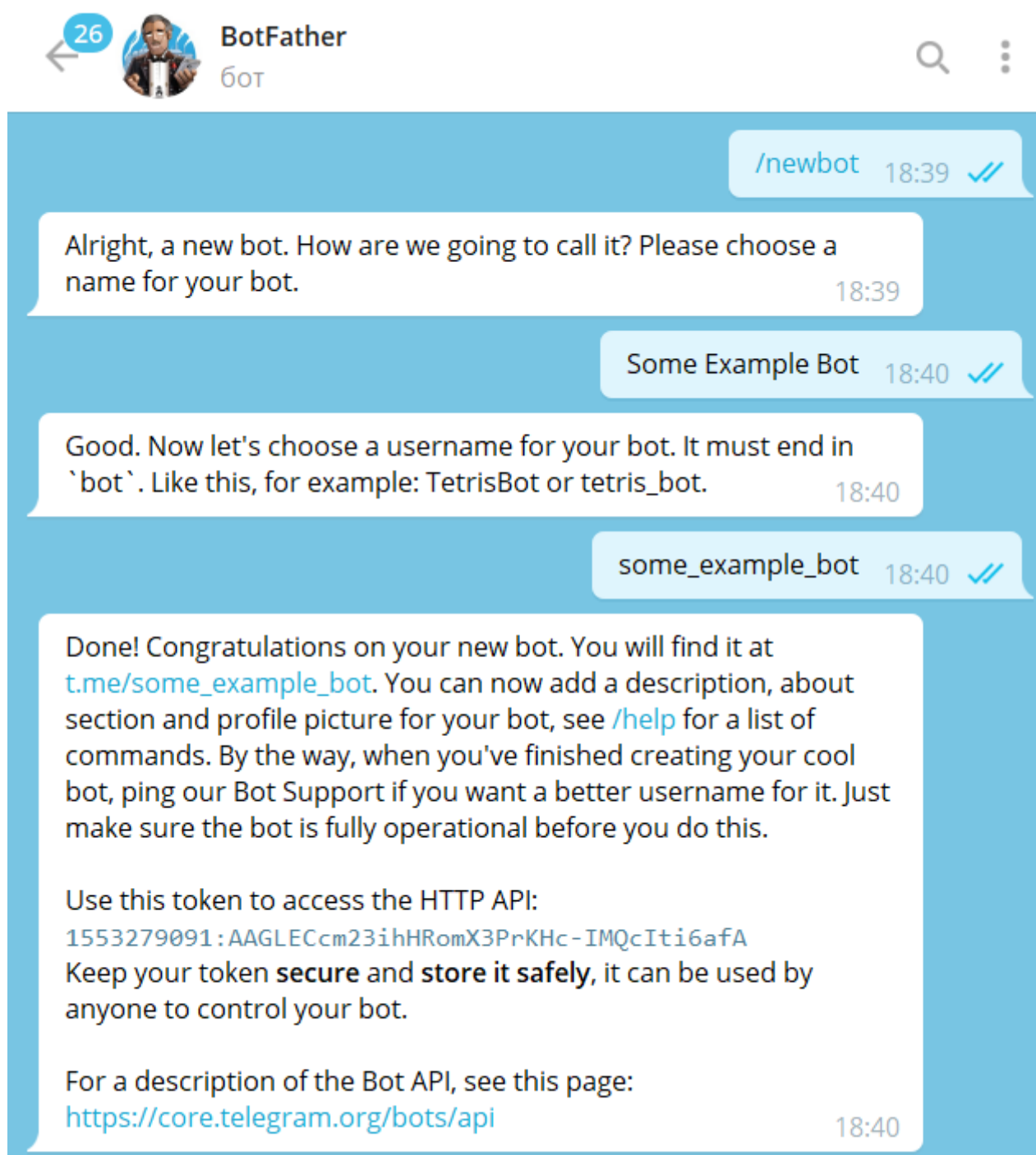
asyncio.run(main())
```

Этот бот будет отвечать на команды `/start` и `/restart`.

Создание бота

Единственная информация о Телеграм-ботах, которой в интернете полным-полно: как создать бота. Это делается через специального бота [BotFather](#). Когда вы создадите бота, BotFather даст вам его токен. Токен выглядит примерно так:

`110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw`. Именно с помощью токена вы сможете управлять ботом.



Один пользователь может создать до 20 ботов.

В BotFather удобно управлять ботами своими командой `/mybots`.

Юзернеймы

При создании бота нужно выбрать юзернейм. После этого поменять его будет очень сложно.

▼ Как поменять юзернейм бота

Если у вас есть бот, и вы хотите дать ему более короткий юзернейм (который может быть занят неработающим ботом), то вы, теоретически, можете это сделать через [@BotSupport](#).

Для этого бот должен (внезапно) быть рабочим, а также поддерживать английский язык.

Принято использовать такой формат обращения в поддержку:

1. @old_username,
2. @new_username,
3. Что бот делает.

Если вы везунчик 9999 |v| — вам ответят.

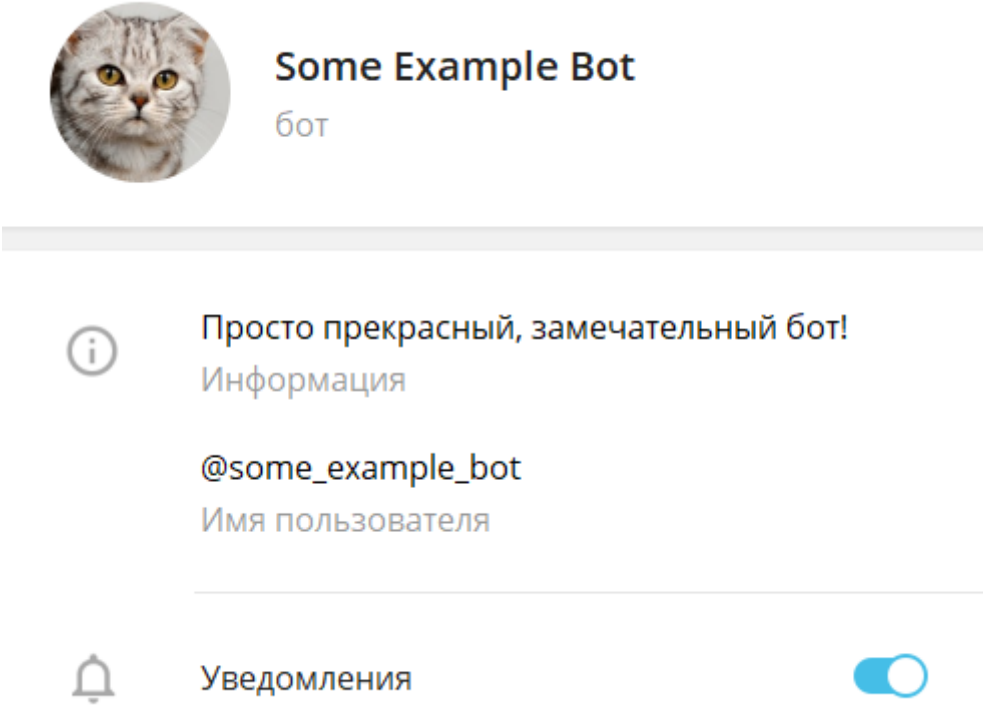
Юзернейм бота выглядит как обычный юзернейм, но он должен заканчиваться на "bot".

Вы могли видеть ботов с именами [@pic](#), [@vid](#), [@sticker](#), [@gamee](#) — это официальные боты Телеграма. Им можно нарушать все правила :)

Очень многие юзернеймы уже заняты. Свободных коротких юзернеймов осталось очень мало. И что самое грустное: почти все эти боты мертвы. Просто не отвечают на сообщения. Наверное, это просто разные любопытные люди хотят сделать бота, создают его, а потом забывают. У меня самого есть несколько лежащих ботов. Так что, думаю, лимит в 20 ботов на одного владельца вполне оправдан :)

Оформление бота

Открыв бота, пользователи могут увидеть его профиль.



Оформление бота настраивается в BotFather: меню `/mybots` → `Edit Bot`. Там можно изменить:

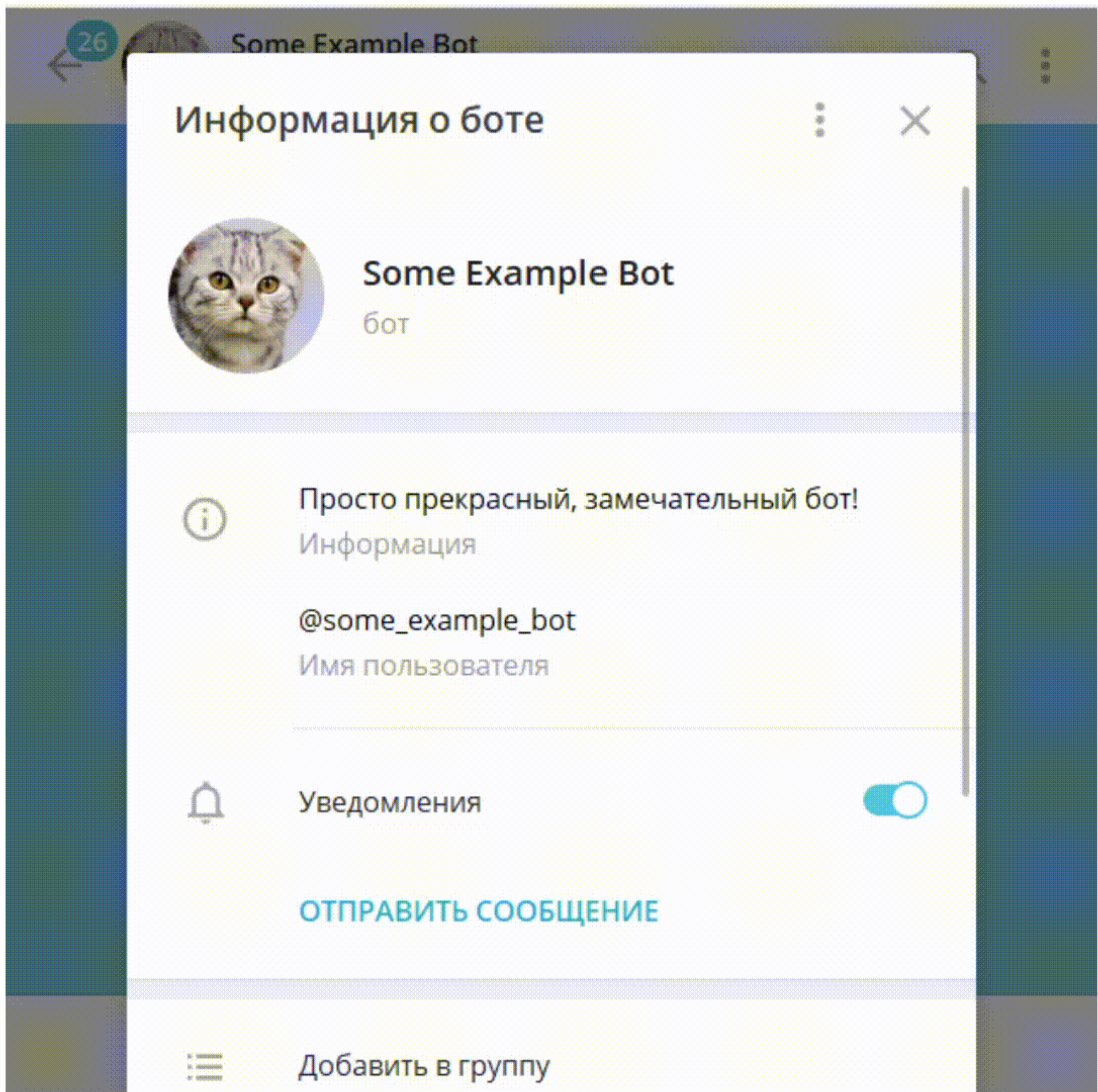
- 1. Имя бота.
- 2. Описание (Description) — это текст, который пользователи будут видеть в начале диалога с ботом под заголовком "Что может делать этот бот?"
- 3. Информация (About) — это текст, который будет виден в профиле бота.
- 4. Аватарка. Аватарки ботов, в отличие от аватарок пользователей и чатов, не могут быть анимированными. Только картинки.
- 5. Команды — тут имеются ввиду подсказки команд в боте. Подробнее о командах [ниже](#).
- 6. Inline Placeholder — об инлайн-режиме см. [ниже](#).

Стандартный совет: Потратьте свое время и заполните описание и информацию бота, чтобы пользователям было понятнее и проще его использовать. Можете оставить там свои контакты. И поставьте аватарку, чтобы бота было проще отличать от других чатов в списке.

Сообщения и чаты

Запуск бота пользователем

Когда пользователь впервые открывает бота, он видит кнопку "Запустить" или "Начать" (зависит от платформы пользователя), на английском — "Start". Нажимая на эту кнопку, он отправляет команду `/start`.



Таким образом, первое сообщение от пользователя — это всегда `/start` (либо `/start` с параметрами, об этом ниже в разделе "Диплинки").

▼ ...если пользователь использует официальный клиент

На стороне сервера это не проверяется, поэтому теоретически пользователь может отправить боту любое сообщение через Telegram API.

Сообщения

Понятно, что главная функция бота — отправлять и получать сообщения.

И то, и другое можно делать со всеми видами сообщений (фото и видео, файлы, опросы, голосовые сообщения и т. д.).

В Телеграме можно делиться файлами до 2 ГБ, но в Bot API более жесткие лимиты: боты могут скачивать файлы до 20 МБ и отправлять файлы до 50 МБ.

▼ [Работа с файлами в Bot API](#)

Если бот уже загрузил файл на сервер Телеграма, то он может использовать `file_id`, чтобы отправлять этот файл.

Загружать файл на сервер можно в том числе и по URL файла.

[Подробнее об отправке файлов в Bot API](#)

Куда может писать бот

Бот может писать в личку только тем пользователям, которые его запустили. Пользователь может заблокировать бота, и тогда бот снова не сможет ему писать.

Боты не могут писать другим ботам.

Бота можно добавить в группу (если в BotFather включена соответствующая настройка). По умолчанию он видит не все сообщения (об этом ниже, в разделе "[Видимость сообщений в группах](#)").

В группе боту можно дать права администратора, чтобы он мог выполнять действия админов.

В одной группе может быть до 20 ботов. В публичные группы (группы с юзернеймом) ботов могут добавлять только админы.



Также бота можно добавить в канал, причем только как администратора. Самый частый способ использования ботов в каналах — добавление кнопок под постами ("лайки", ссылки и прочее).

▼ [Как боты добавляют кнопки](#)

У админа канала может быть специальное право: "Редактирование чужих публикаций". С помощью него боты редактируют посты, добавляя к ним кнопки.

Подробнее о кнопках тоже [ниже](#).

Супергруппы

На самом деле многие группы в Телеграме являются супергруппами.

Почему так? Раньше было [четкое разделение на группы и супергруппы](#). По задумке, супергруппы — это группы для сообществ. Супергруппы могут иметь больше участников, публичные ссылки и другие плюшки.

Со временем, видимо, решили, что это неудобная концепция. Теперь обычная группа становится супергруппой, когда у группы меняются какие-нибудь настройки ([подробнее тут](#)). Вот такой костыль.

В этой статье под группами я подразумеваю и супергруппы, и обычные группы.

Супергруппу нельзя обратно превратить в группу. С точки зрения API [супергруппа устроена так же, как и канал](#). Важное отличие супергрупп от обычных групп состоит в нумерации сообщений: о нём [чуть ниже](#).

id пользователей и чатов

У каждого пользователя, бота, группы, канала в Телеграме есть собственный id. Различать чаты в коде бота следует именно по id, потому что он никогда не меняется.

В токене бота первая часть — это его id. Например, токен `110201874:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw` принадлежит боту с id `110201874`.

В Bot API перед id супергрупп и каналов пишется -100. Так, id `1356415630` превращается в `-1001356415630`. Осторожно: вы не сможете сохранить это значение в 32-битный тип числа.

id сообщений

Каждое сообщение в Телеграме имеет свой id. Это относится и к системным сообщениям (пользователь зашел в группу, изменилось название группы и т. д.)

Через Telegram API боты могут получать по запросу сообщения в любом чате по их id.

id сообщений в супергруппах и каналах уникальны для чата: первое сообщение в чате имеет номер 1, второе имеет номер 2 и так далее.

id сообщений в личных сообщениях и обычных группах работают по другому. Там, можно сказать, нумерация сквозная: id сообщения уникально для каждого отправившего его пользователя. Так, первое сообщение от пользователя во всех личках и группах имеет номер 1, второе сообщение от того же пользователя имеет номер 2 и так далее.

Видимость сообщений в группах

Обычно бот должен реагировать именно на команды. Телеграм не уведомляет бота об остальных сообщениях, и это гарантирует приватность переписки.

Но если боту нужно видеть все сообщения в группе (например, если это чат-бот или антиспам-бот), для него можно отключить Privacy mode.

Privacy mode — настройка в BotFather, которая по умолчанию включена. В таком режиме бот в группах видит только такие сообщения:

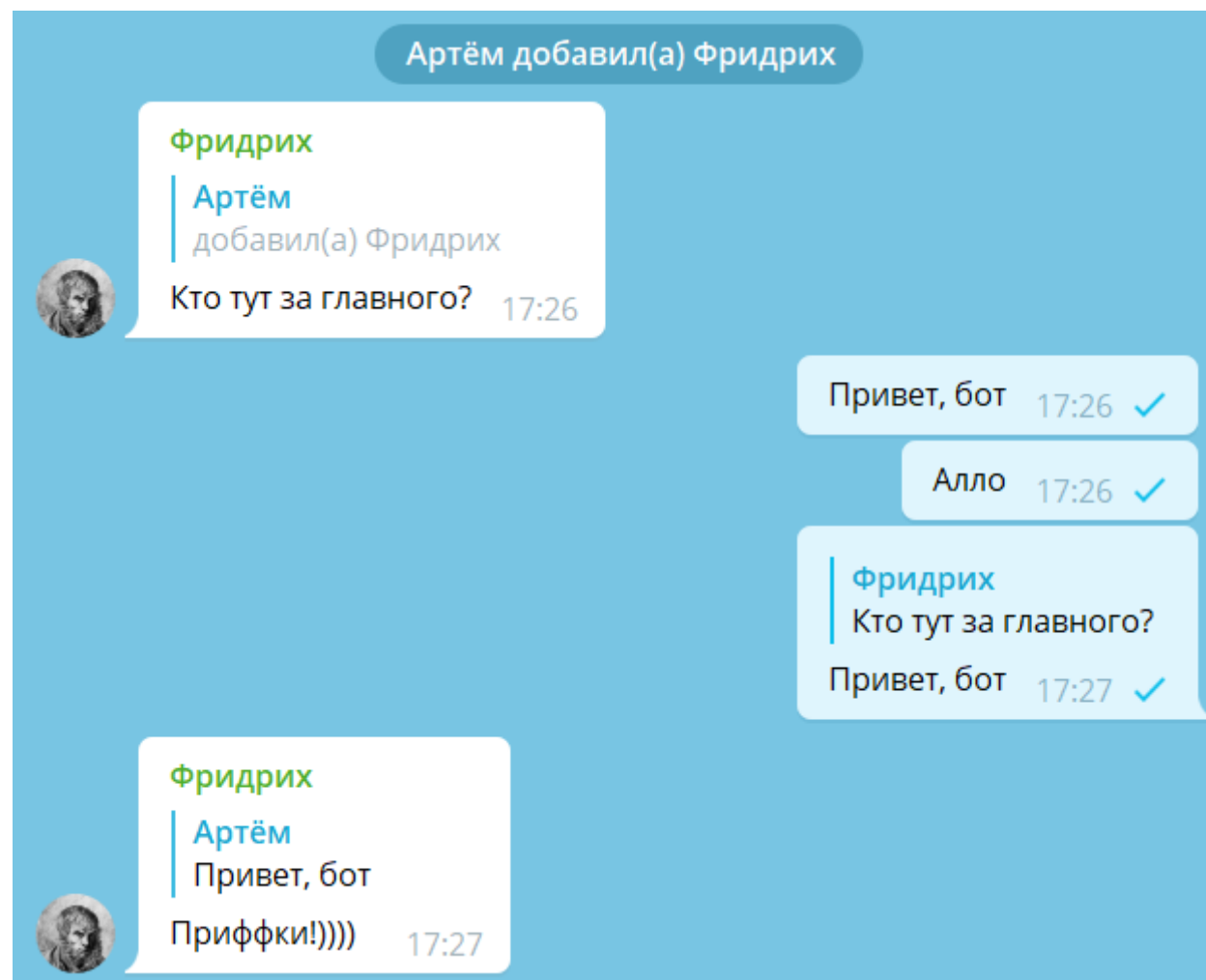
- Сообщения с упоминанием бота,
- Ответы на сообщение бота, ответы на ответы и так далее,
- Системные сообщения,

- Команды — о них в следующем пункте.

А если Privacy mode выключен, то бот видит все сообщения в группе.

Если бот — админ в группе, то он в любом случае видит все сообщения.

Бот, работающий через Bot API, в любом случае не будет видеть сообщения от других ботов.



Бот видит не все сообщения

▼ [Я включил Privacy mode, а он не работает](#)

Нужно удалить бота из группы и добавить заново.

▼ [Исправленный баг с видимостью сообщений](#)

Недавно [был найден баг](#): боты не могли видеть некоторые сообщения от юзеров, если сообщения выглядят как команды с юзернеймом пользователя на конце.

Это могло нарушить работу антиспам- и других ботов.

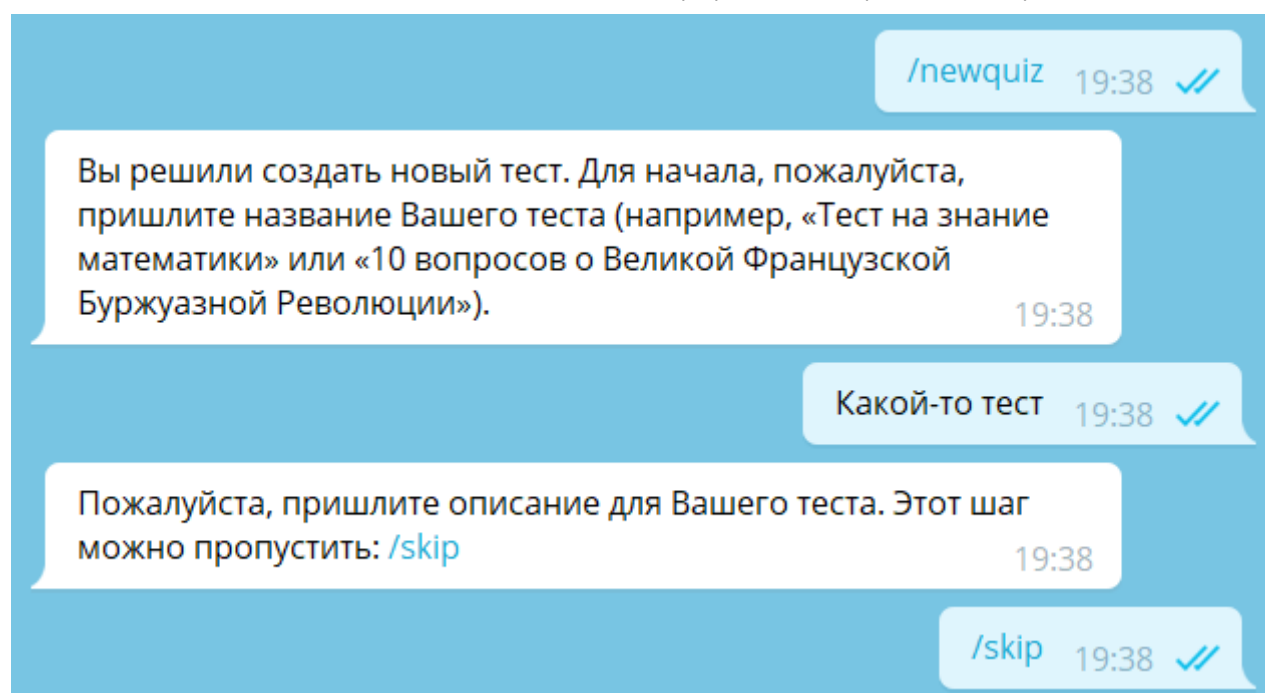
Сейчас баг уже исправлен.

[О Privacy mode в документации Bot API](#)

Команды

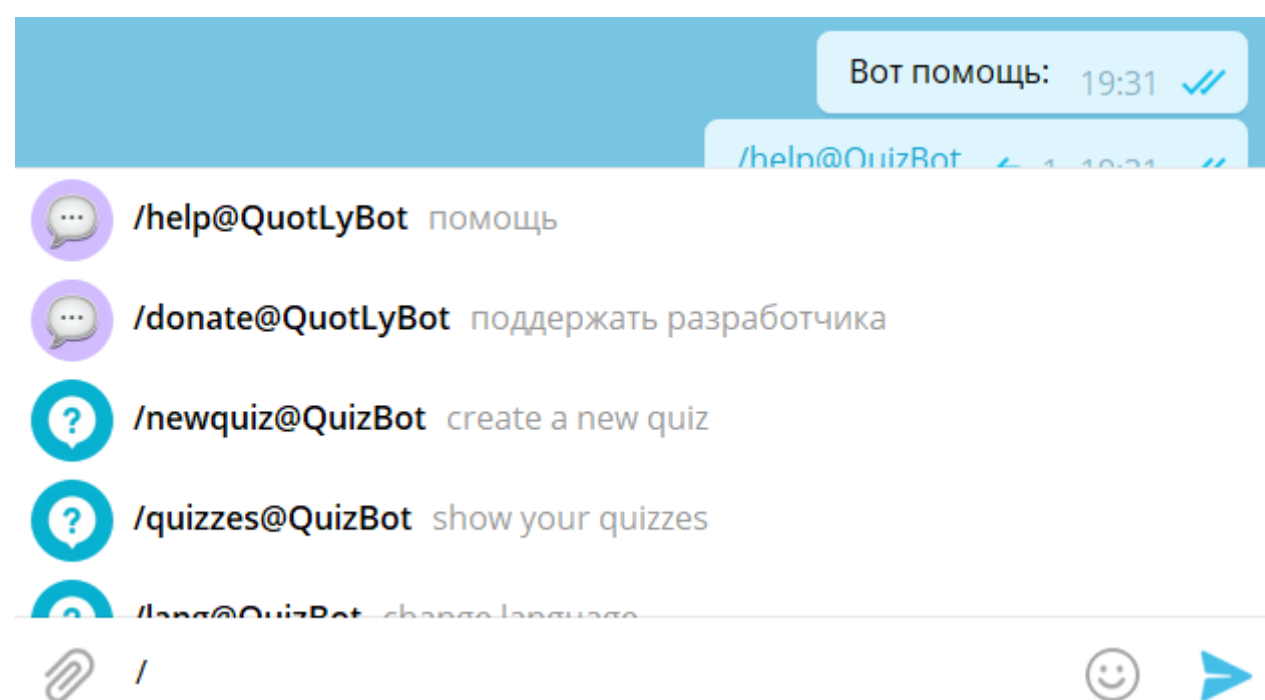
Часто используемый способ "общения" пользователей с ботом — команды. Команды начинаются на "/" и состоят из латинских букв (можно использовать цифры и нижние подчеркивания).

Команды подсвечиваются как ссылки: нажатие отправляет команду в чат.



В группах, чтобы различать команды от разных ботов, Телеграм предлагает ставить в конце команды юзернейм бота. Например: `/start@examplebot`.

В BotFather можно указать подсказки команд для бота. Он будет отображаться при вводе "/" и команд. Если есть подсказки, рядом с кнопкой "Отправить" появляется кнопка для открытия меню команд.



Если в подсказках команд есть `/help`, в профиле бота появляется кнопка "Помощь с ботом". Нажатие на кнопку отправляет эту команду.

Если в подсказках команд есть `/settings`, в профиле бота появляется кнопка "Настройки бота". Нажатие на кнопку отправляет эту команду.

Разметка сообщений

Как вы, наверное, знаете, сообщения в Телеграме могут содержать не только обычный текст, но и жирный, курсив и др. В Bot API разметку сообщений можно делать в HTML и Markdown.

▼ Разметка в Telegram API

В Telegram API для разметки надо вместе с сообщением передавать entities (MessageEntityBold, MessageEntityItalic и так далее). Хорошие библиотеки сами превращают HTML/Markdown в текст и entities.

Способы выделения текста:

- **Жирный текст**
- *Курсив*

- Подчёркнутый текст
- Зачёркнутый текст
- Моноширинный текст ("в строке" и "блоком")
- Ссылка (встроенная в текст)
- Упоминание пользователя — текст, похожий на ссылку, клик по которому открывает профиль пользователя. Если упомянуть в группе её участника, он получит уведомление.
Чтобы вставить в сообщение упоминание пользователя, в Bot API нужно встроить ссылку на `tg://user?id=123456789`.

[О разметке в документации Bot API](#)

Кнопки

Инлайн-кнопки

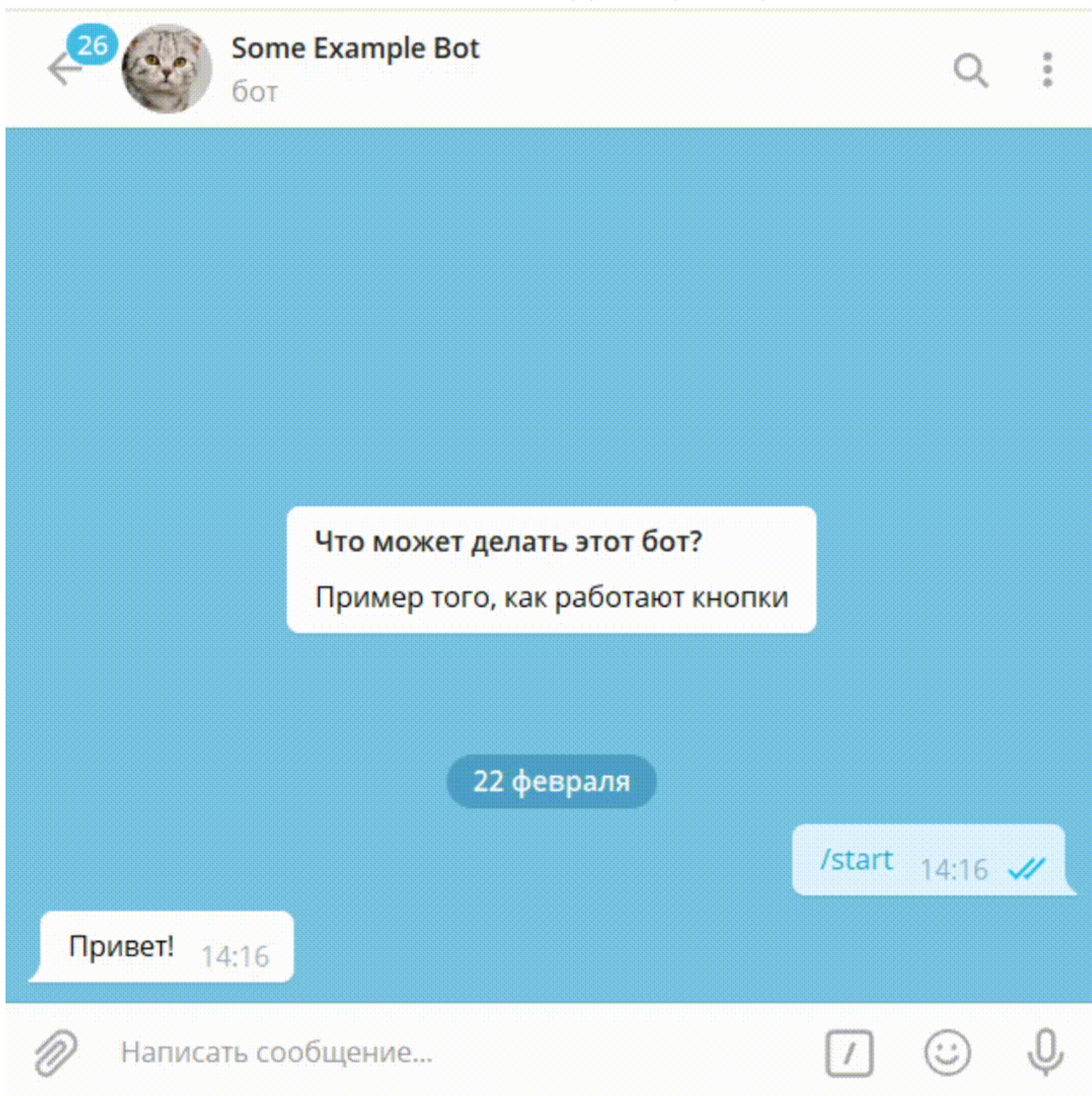
Бот может оставлять кнопки под своими сообщениями.

Кнопки под сообщениями (они же inline keyboards / inline buttons) в основном бывают трёх видов:

- URL button — кнопка с ссылкой.
- Callback button. При нажатии на такую кнопку боту придёт апдейт. С созданием кнопки можно указать параметр, который будет указан в этом апдейте (до 64 байтов). Обычно после нажатий на такие кнопки боты изменяют исходное сообщение или показывают notification или alert.
- Switch to inline button. Кнопка для переключения в инлайн-режим (об инлайн-режиме см. [ниже](#)). Кнопка может открывать инлайн в том же чате или открывать меню для выбора чата. Можно указать в кнопке запрос, который появится рядом с никнеймом бота при нажатии на кнопку.

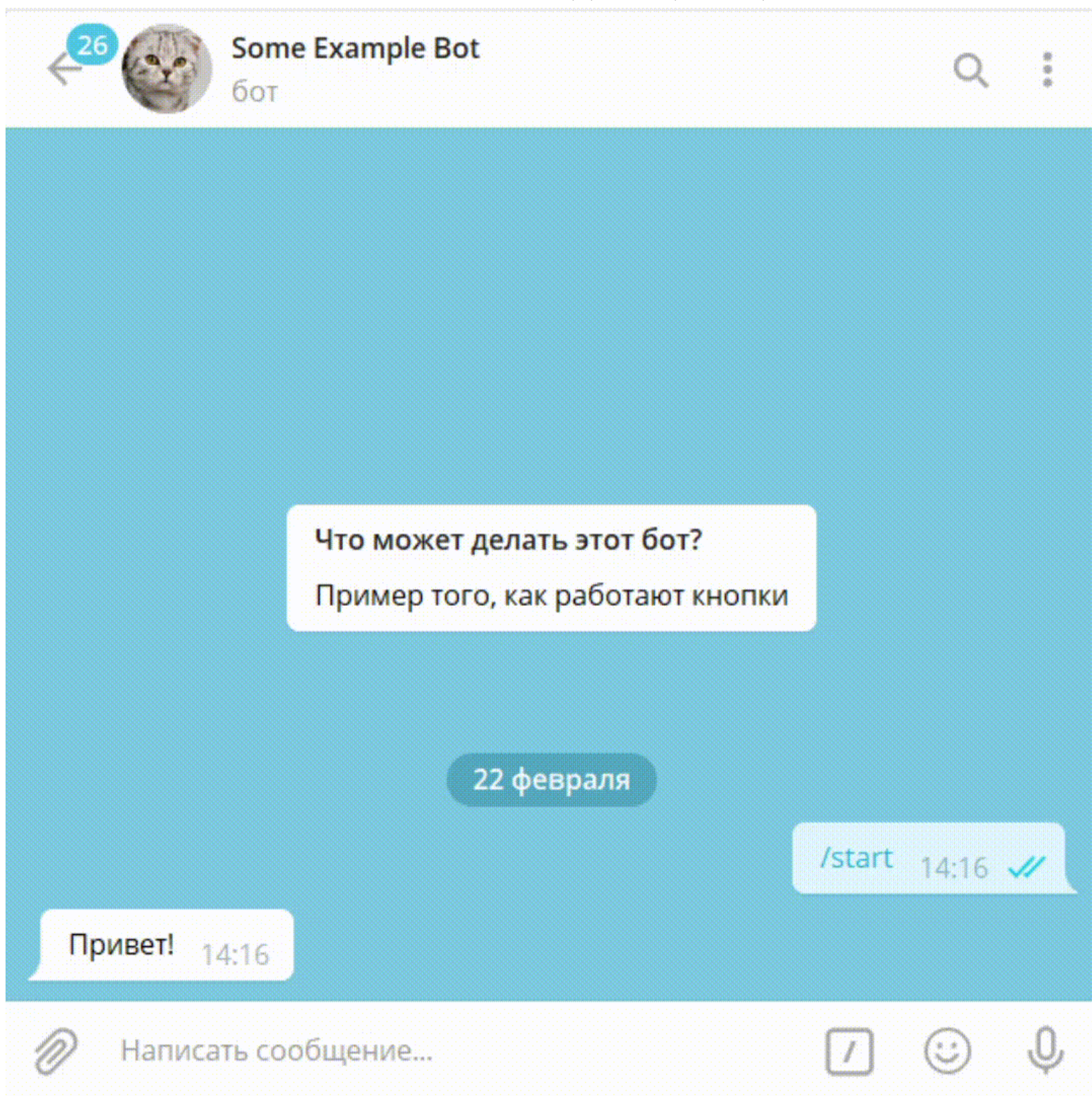
▼ [Дополнительные виды кнопок](#)

- Login URL button — специальная кнопка для авторизации пользователей на сайте. Использовалась, например, в официальном боте [@discussbot](#) (до добавления нативных комментариев в Телеграм).
- Callback game button — кнопка для открытия HTML-игры. См. пункт "[HTML-игры](#)".
- Pay button — кнопка для платежей. См. пункт "[Платежи через ботов](#)".



Клавиатурные кнопки

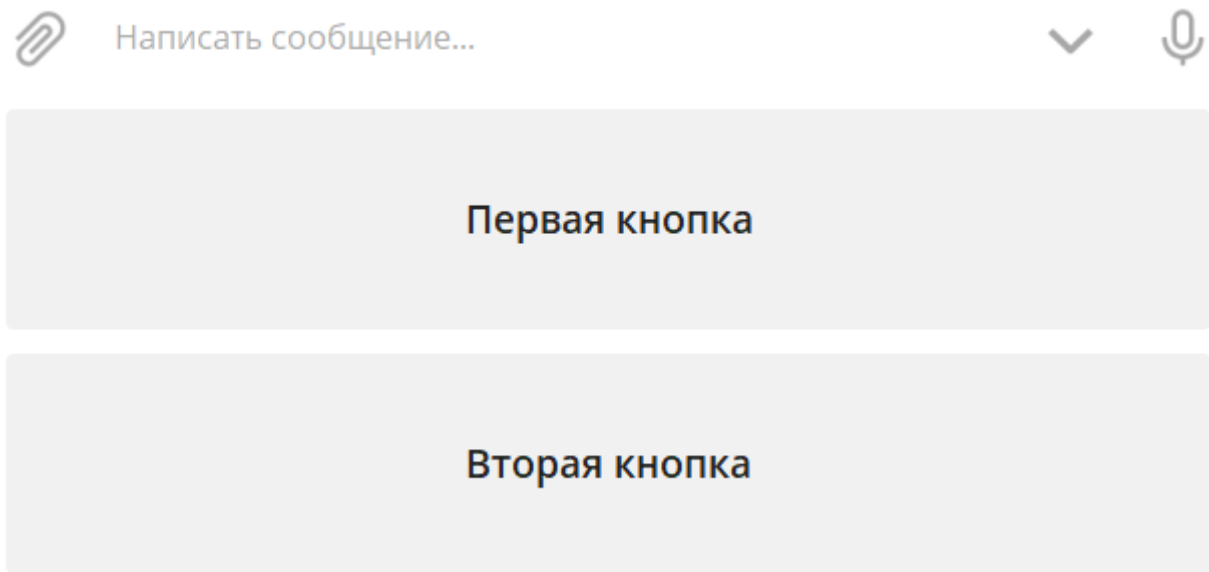
Есть другой тип кнопок: keyboard buttons. Они отображаются вместо клавиатуры как подсказки. При нажатии на такую кнопку пользователь просто отправит этот текст.



При этом в личных чатах с помощью кнопки можно:

- Запросить номер телефона пользователя,
- Запросить геолокацию пользователя,
- Открыть у пользователя меню создания опроса.

Есть опция `resize_keyboard`, которая отвечает за то, изменять ли высоту этой "клавиатуры из кнопок". По умолчанию она, почему-то, выключена, и тогда высота клавиатуры стандартная большая. Получаются кнопки как на этой картинке:



Чтобы показать клавиатурные кнопки, бот должен отправить сообщение. Можно отправить клавиатуру, которая свернётся (но не пропадёт) после нажатия на кнопку.

По умолчанию, если показать кнопки в группе, они будут видны всем пользователям. Вместо этого можно отобразить кнопки одновременно для этих пользователей:

- Для пользователей, юзернеймы которых были в тексте сообщения,
- Если это ответ на другое сообщение: для пользователя, который его отправил.

Ещё о кнопках

Оба типа кнопок могут составлять несколько рядов, в каждом из которых по несколько кнопок.

При отправке сообщения можно выбрать одно (но не больше) из следующих действий:

- Добавить к сообщению инлайн-кнопки,
- Показать клавиатурные кнопки,
- Убрать все клавиатурные кнопки,
- Force reply: автоматически заставить пользователя ответить на сообщение. Так произойдёт то же самое, что и при нажатии пользователем кнопки "Ответить". Это нужно для того, чтобы бот мог общаться с пользователями в группах, не нарушая Privacy mode.

Таким образом, нельзя показать оба типа кнопок одновременно.

Взаимодействие с ботом

Ссылки на бота

Юзернеймы ботов работают так же, как и любые другие юзернеймы в Телеграме: бота @examplebot можно открыть по ссылке t.me/examplebot.

Также существует прямая ссылка: `tg://resolve?domain=examplebot`

▼ [Подробнее о ссылках tg://](#)

Такие ссылки могут не только заменять ссылки [t.me](#), но и задавать свои действия. Например, [tg://settings](#) открывает настройки.

Список известных таких ссылок есть канале [@DeepLink](#).

Ссылка на добавление в группу

По ссылке t.me/examplebot?startgroup=true у пользователя откроется меню: выбор группы для добавления бота.

Прямая ссылка: `tg://resolve?domain=examplebot&startgroup=true`

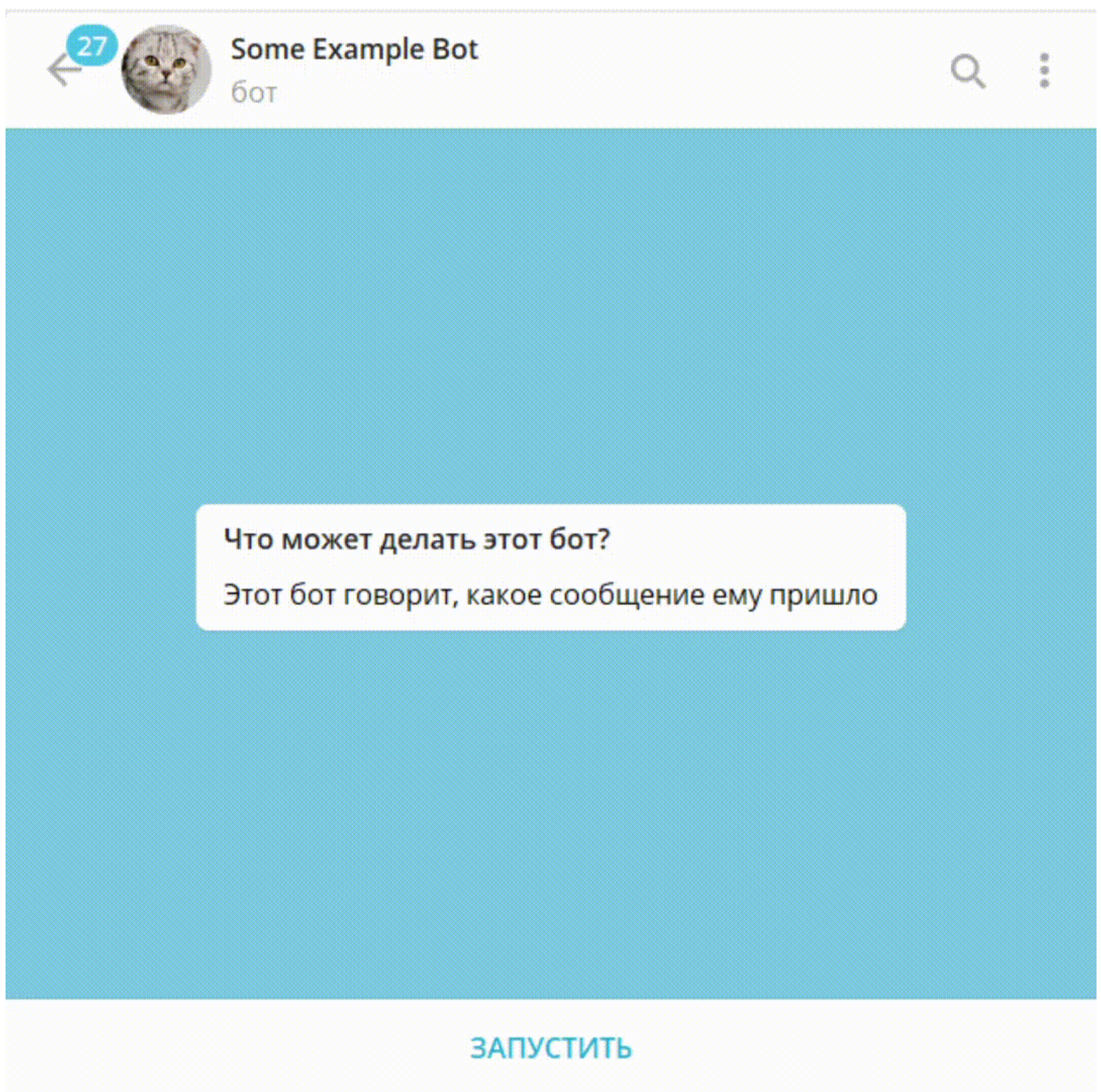
Диплинки

По ссылке `t.me/examplebot?start=<ваш текст>` пользователь может запустить бота с каким-то стартовым параметром (`<ваш текст>`).

Как это выглядит:

1. При переходе по ссылке бот открывается как обычно.
2. Отображается кнопка "Запустить", даже если пользователь уже запускал бота.
3. Пользователь нажимает на кнопку и видит сообщение `/start` (всё как обычно).
4. Боту вместо этого приходит сообщение `/start <ваш текст>`

Так бот может отреагировать на запуск не как на обычный `/start`, а другим способом.



Часто диплинки используются для реферальных программ (в качестве параметра можно передавать id пользователя, который поделился ссылкой). Есть и другие применения.

Прямая ссылка: `tg://resolve?domain=examplebot&start=<ваш текст>`

[О диплинках в документации Bot API](#)


Инлайн-режим

Инлайн-режим (inline mode) — это специальный режим работы бота, с помощью которого пользователь может использовать бота во всех чатах.

Выглядит это так: пользователь вводит юзернейм бота в поле для ввода сообщения. После юзернейма можно ещё записать запрос (текст до 256 символов).

Появляется менюшка с результатами. Выбирая результат, пользователь отправляет сообщение.

Как выглядит инлайн-режим



Инлайн-режим можно включить в BotFather, там же можно выбрать плейсхолдер вместо стандартного "Search..."

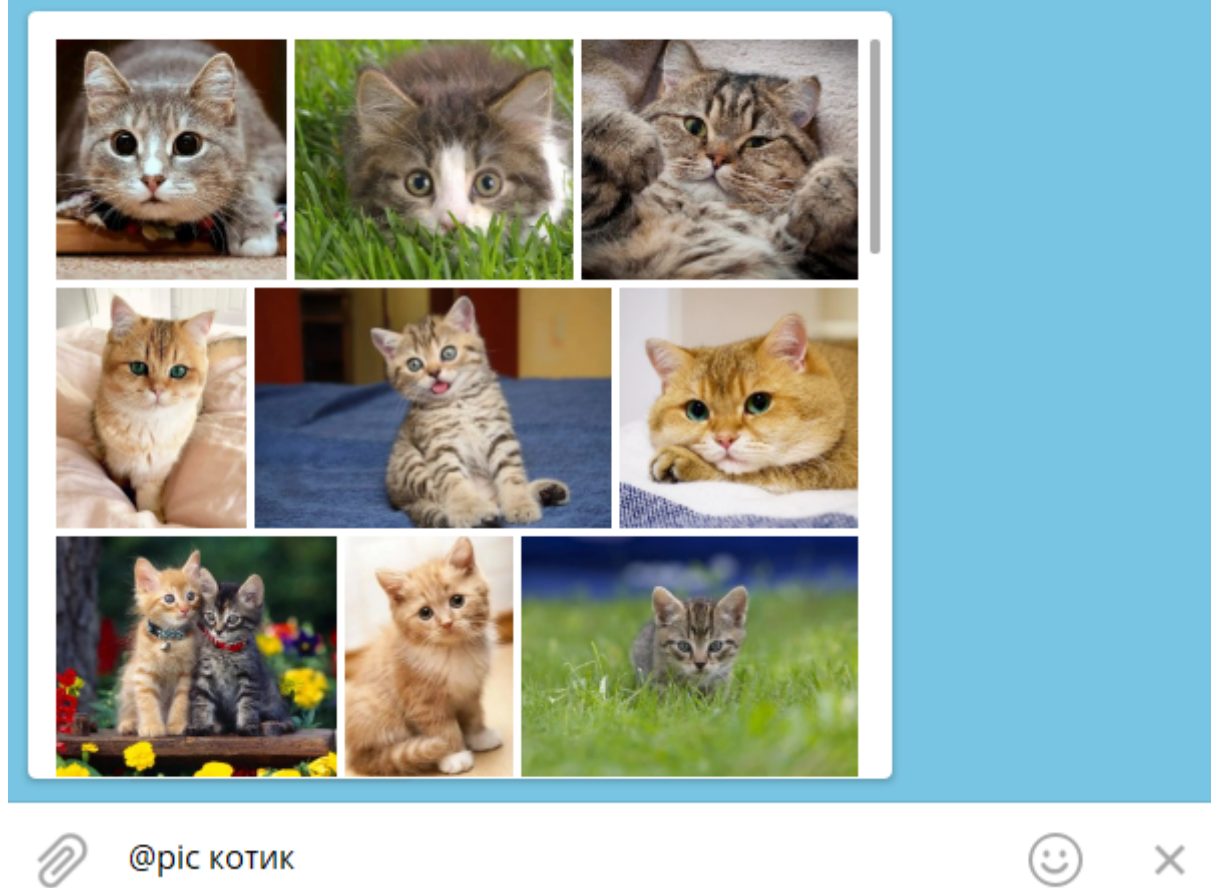
В группе можно запретить использовать инлайн всем или некоторым участникам. В официальных приложениях Телеграм это ограничение объединено с ограничением на отправку стикеров и GIF.

[Страничка об инлайн-режиме на сайте Telegram](#)

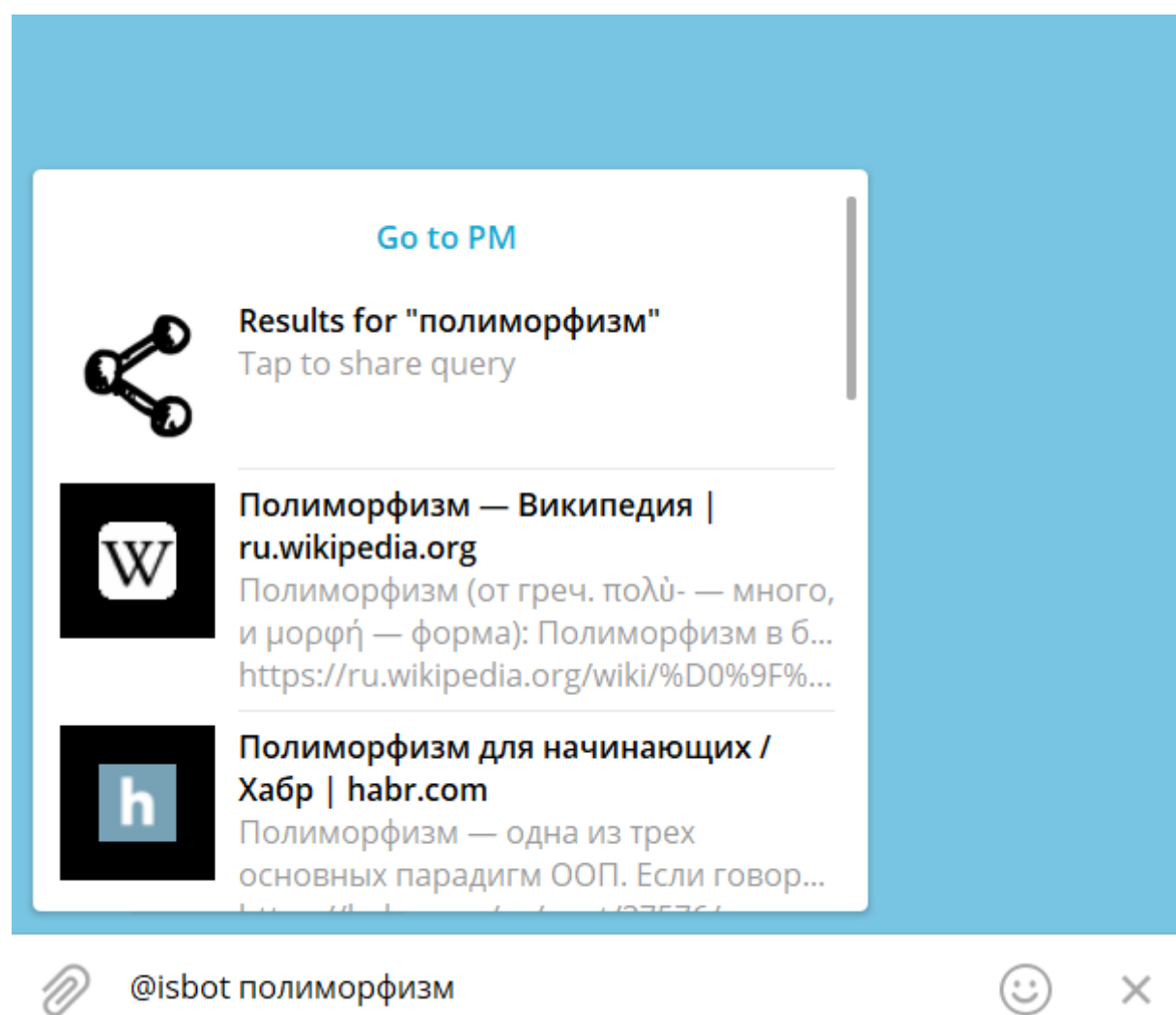
Результаты инлайн-режима

Результаты можно отображать двумя способами:

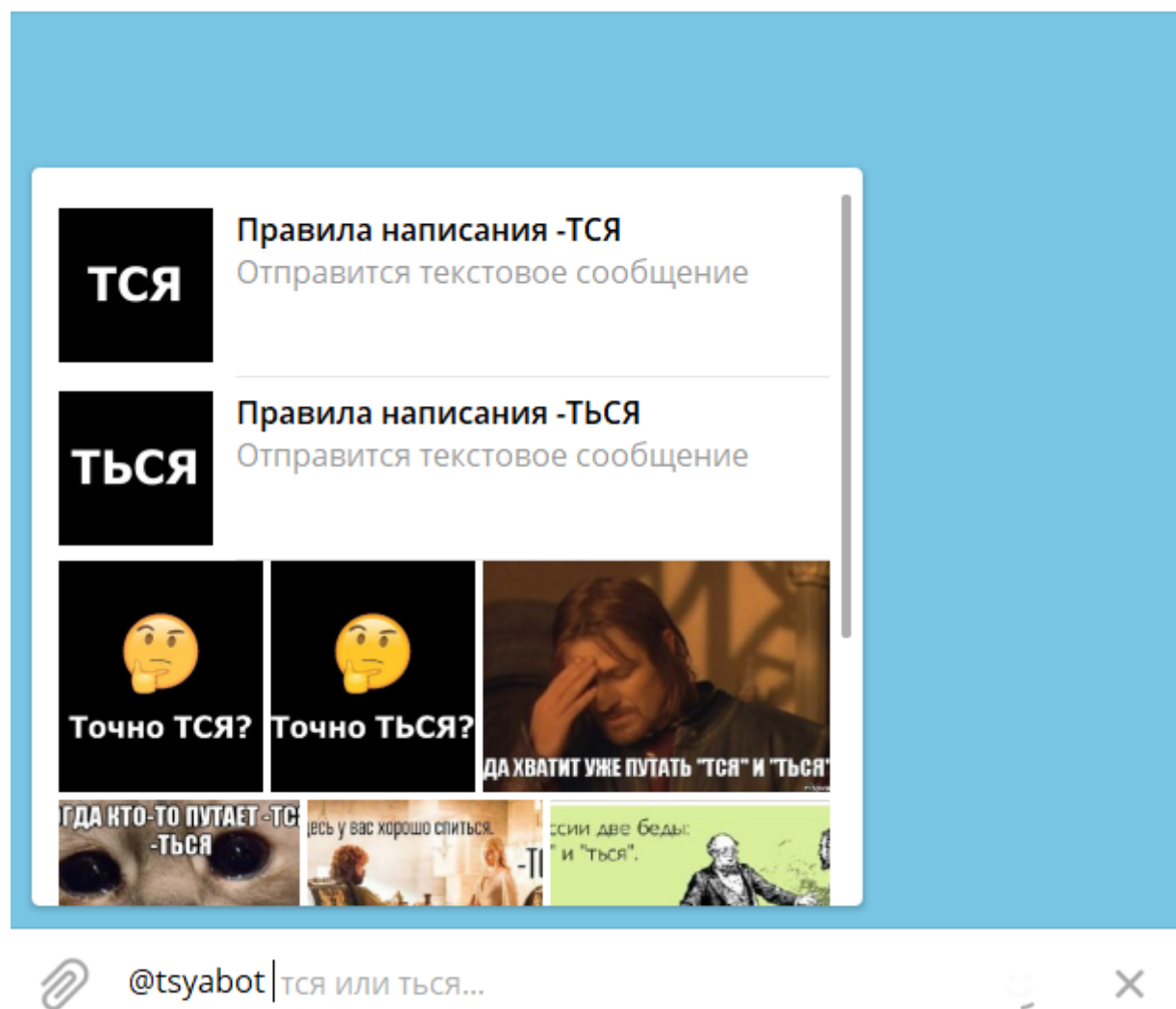
- Сеткой. Удобно для выдачи картинок.



- Вертикальным списком. Удобно для выдачи текста.



Можно совмещать два типа, но корректно отображается это только на Telegram Desktop.



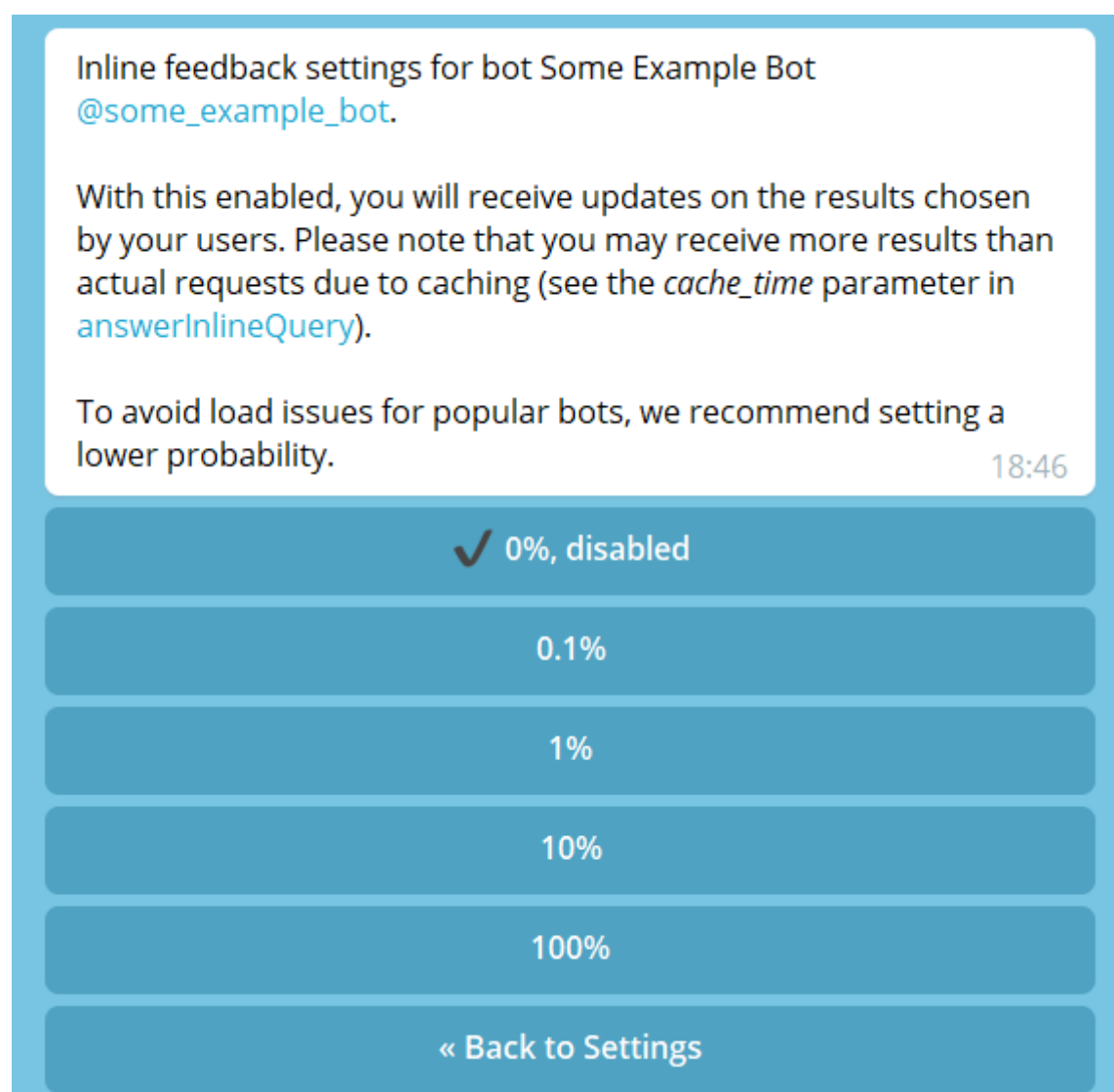
Приватность и геопозиция в инлайне

Когда пользователь вызывает инлайн-режим, бот не может получить никакую информацию о контексте, кроме id пользователя. Таким образом, бот не может узнать ни чат, в котором вызвали инлайн, ни сообщение, на которое пользователь отвечает, ни даже имя самого пользователя.

Но зато если включить в BotFather настройку "Inline Location Data", то бот сможет видеть геопозицию пользователей, когда они используют инлайн (на мобильных устройствах). Перед этим у пользователей показывается предупреждение.

Inline feedback

Inline feedback — это апдейты о выбранных инлайн-результатах. Включаются через BotFather.



Предполагается использование inline feedback для сбора статистики, но не всегда он используется так. Inline feedback позволяет "подгружать" не все результаты сразу, а только выбранный. Например, если бот используется для поиска музыки, то он может загружать не все песни сразу, а только одну.

Важный момент: если вы получили апдейт об отправке инлайн-сообщения, то вы можете его редактировать, только если к нему прикреплены инлайн-кнопки. (Если кнопок нет, то в апдейте не указывается id инлайн-сообщения, по которому происходит редактирование).

Создание наборов стикеров

Боты (и только боты!) могут создавать наборы стикеров. При этом каждый набор стикеров должен принадлежать какому-то пользователю. Посмотреть свои наборы стикеров пользователь может с помощью бота [@Stickers](#).

Платежи через ботов

Телеграм предоставляет ботам возможность принимать платежи от пользователей. Это делается через провайдеров ЮMoney, Сбербанк, Stripe и ещё 7.

Эта возможность используется редко, потому что для использования провайдеров нужно юридическое лицо.

Платежи, к тому же, не работают на iOS из-за ограничений Apple.

[Страница Bot Payments API](#)

HTML-игры в ботах

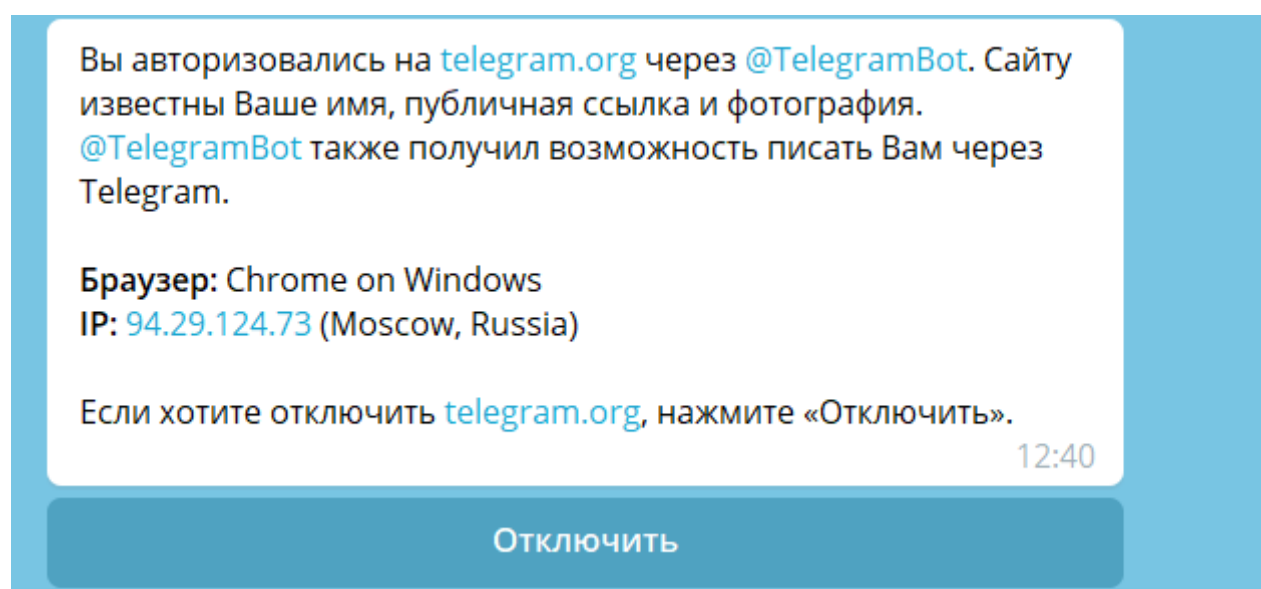
Боты могут позволять пользователям играть в HTML5-игры в чатах. Бот может отправлять сообщения-игры или создавать их через инлайн-режим. Как это работает, можно посмотреть на примере официального [@gamebot](#).

[Страница Bot Gaming Platform](#)

Telegram Login Widget

Вы можете добавить на свой сайт авторизацию через Телеграм. Процесс авторизации будет проходить так:

1. Пользователь должен будет ввести свой номер телефона.
2. Бот Telegram попросит подтвердить вход.
3. Пользователь авторизуется и нажимает на "Принять" на сайте.



Telegram Login Widget не связан с Login URL button (см. [раздел про кнопки](#) выше), а является его альтернативой.

[О Telegram Login Widget на сайте Телеграм](#)

Разработка ботов

Какие апдейты можно получать

Бот не может получить старые сообщения из чата. Бот не может получить список всех своих пользователей. Все, что может получать бот — это информацию об обновлениях. В этом заключается главная сложность разработки ботов.

Вы можете получать информацию о новых сообщениях в боте и других событиях, но только один раз. Вам придётся самим хранить список чатов, старых сообщений (если это зачем-то нужно) и так далее. Если вы случайно сотрёте/потеряете эту информацию, вы её больше никак не получите.

В Telegram API бот может чуточку больше: он может получать сообщения по id, получать список участников группы и прочее.

▼ Получение апдейтов: Bot API vs Telegram API

Если вы получили апдейт в Bot API, то второй раз вы его уже не получите.

В Telegram API это не так: пользователь может пользоваться мессенджером через несколько клиентов, каждый из которых должен получать апдейты.

Если запустить для одного бота несколько программ на Telegram API, то каждая из них будет получать все апдейты.

Ограничения Bot API не позволяют сделать то же самое.

Лимиты

Конечно, на запросы к серверу существуют лимиты. В [Bots FAQ на сайте Telegram](#) названы следующие:

- Не больше одного сообщения в секунду в один чат,
- Не больше 30 сообщений в секунду вообще,
- Не больше 20 сообщений в минуту в одну группу.

Эти лимиты не строгие, а примерные. Лимиты могут быть увеличены для больших ботов через поддержку.

Другие известные ограничения в Telegram собраны на limits.tginfo.me — см. раздел про ботов.

Рассылка по пользователям

Ниже в [Bots FAQ](#) сказано, что Bot API не позволяет рассылать сообщения всем юзерам одновременно и что в будущем, может быть, они что-то для этого сделают. И написано это уже несколько лет.

Они советуют растянуть рассылку на длительное время (8-12 часов) и замечают, что API не позволит отправлять сообщения более чем ~30 пользователям в секунду.

Смена владельца бота

Осенью 2020 года появилась возможность передавать ботов другому человеку. Это можно сделать в настройках бота в BotFather. Для этого на вашем аккаунте должна быть включена двухфакторная авторизация — не менее, чем за 7 дней до передачи. Передать бота можно только пользователю, который что-либо ему писал.

Локальный сервер Bot API

Также осенью 2020 года исходники Bot API [выложили на GitHub](#). Теперь вы можете поднять собственный сервер Bot API. На GitHub перечислены следующие преимущества:

- Скачивание файлов с сервера без ограничения (ограничение на отправку файлов пользователями в Телеграме — 2 ГБ),
- Загрузка файлов на сервер до 2000 МБ,
- Загрузка файлов на сервер с помощью локального пути и URI файла,
- Использование HTTP URL для вебхука,

- Использование любого локального IP-адреса для вебхука,
- Использование любого порта для вебхука,
- Возможность увеличить максимальное число соединений до 100000,
- Получение локального пути файла вместо загрузки файла с сервера.

Юзерботы

В начале статьи я рассказывал о том, что такое Telegram API и Telegram Bot API.

Telegram API используется не только для ботов — тогда в чём проблема управлять аккаунтами пользователей, как ботами? Люди это делают. Кто-то автоматически [ставит текущее время себе на аватарку](#), кто-то скриптом реагирует на свои сообщения как на команды, кто-то сохраняет сообщения из публичных групп и каналов. Всё это называют юзерботами.

Юзерботов следует использовать аккуратно: за большую подозрительную активность аккаунт могут ограничить или забанить.

Заключение

Я постарался собрать в одном месте и структурировать информацию о всех возможностях Телеграм-ботов. Большое спасибо [vanutp](#), [NToneE](#) и [Grinrill](#) за помощь с фактами. Если мы что-то забыли — пишите, исправлю.

Я специально не разделял большую статью на несколько постов, чтобы можно было быстро найти нужную информацию. К тому же, в начале статьи есть её содержание. Так что можете сохранить её к себе и использовать как справочник :)

Вообще интерфейс бота (то есть интерфейс чата) имеет много ограничений. Но плохо ли это? Действительно удобнее использовать инструмент, когда это часть привычной среды. Я часто прямо в переписке нахожу нужную картинку или информацию с помощью инлайн-ботов. Как заядлый пользователь Телеграма, я люблю использовать ботов. И создаю ботов. И вы создавайте.

Теги: [Telegram](#), [боты](#), [телеграм](#)

Хабы: [Мессенджеры](#), [API](#)