

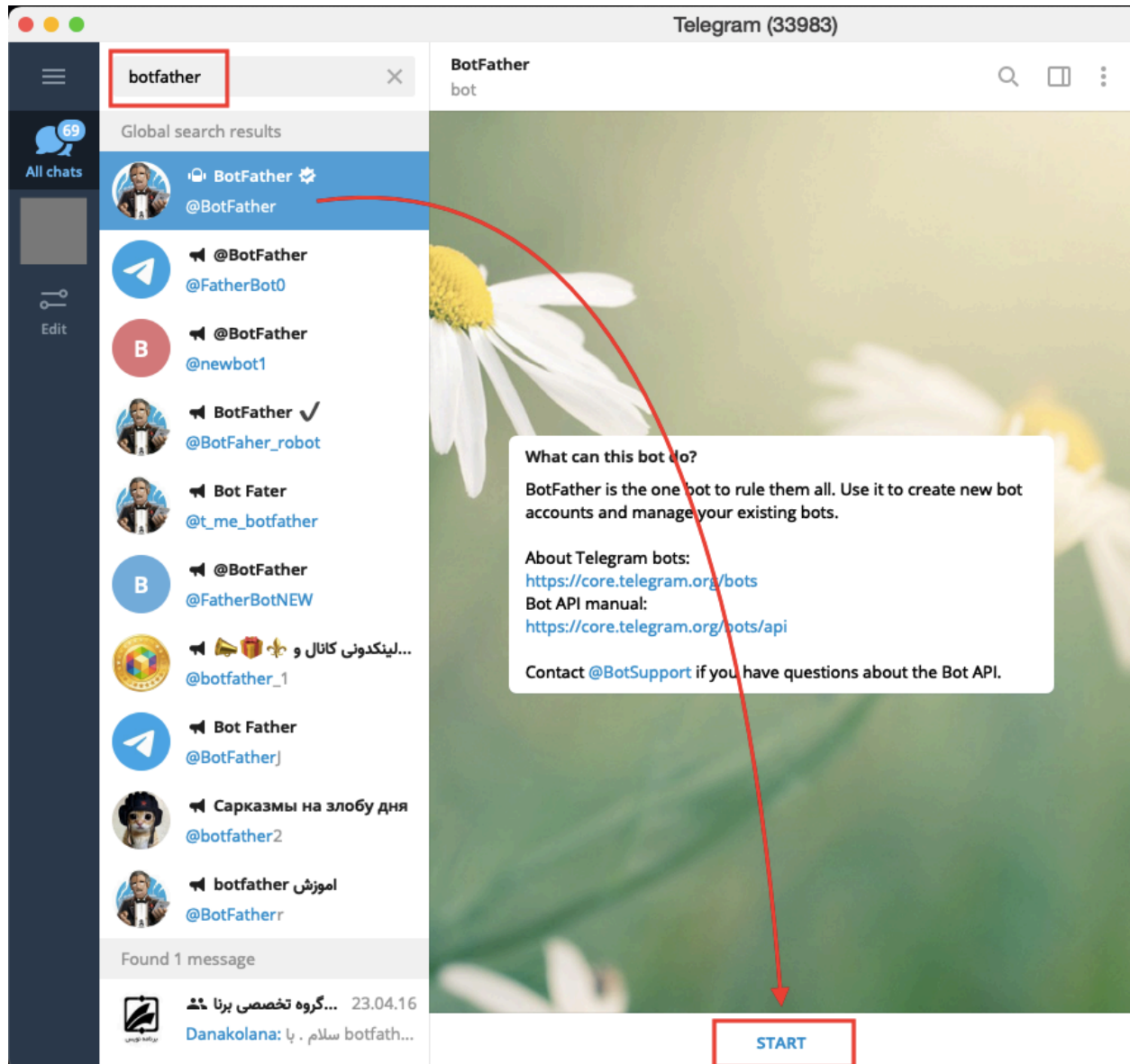
Telegram Bot API Setup

estimated reading 2 minutes

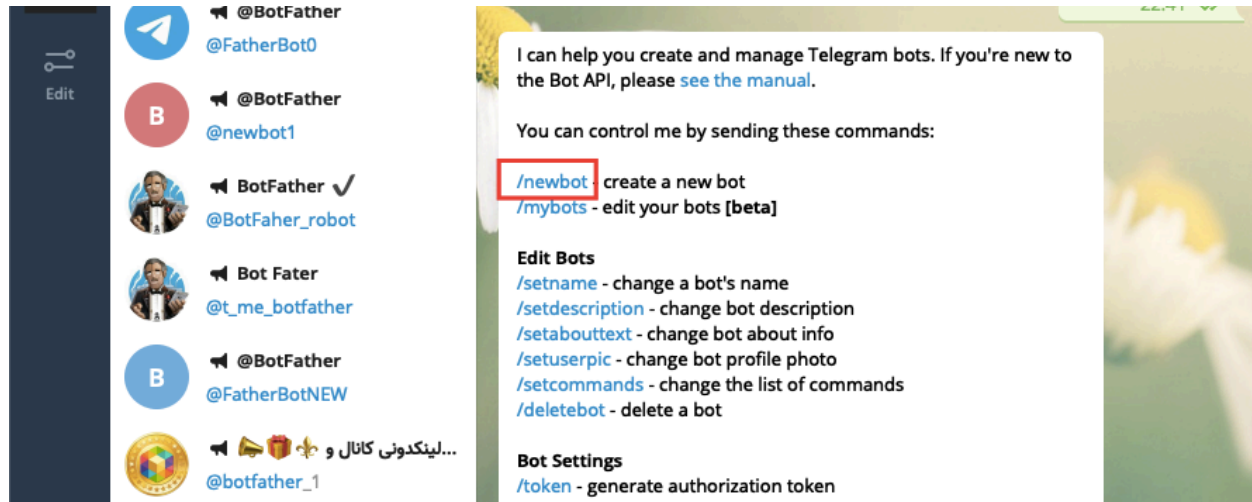
In order to use Telegram Bot API you need to create a telegram bot and receive a Token Key

1- in your Telegram App, search for **BotFather**.

2- press **start**

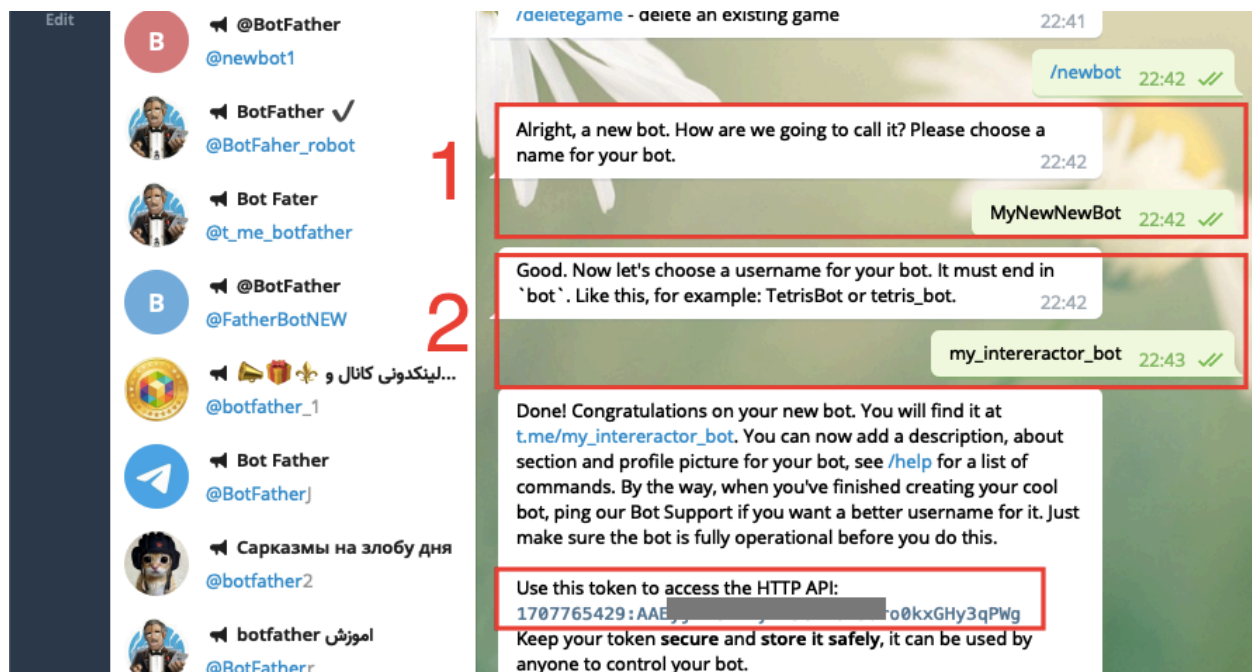


3- press `/newbot`



4- choose a name for your bot

5- choose a username for your bot (it should end with “bot” and also be unique)

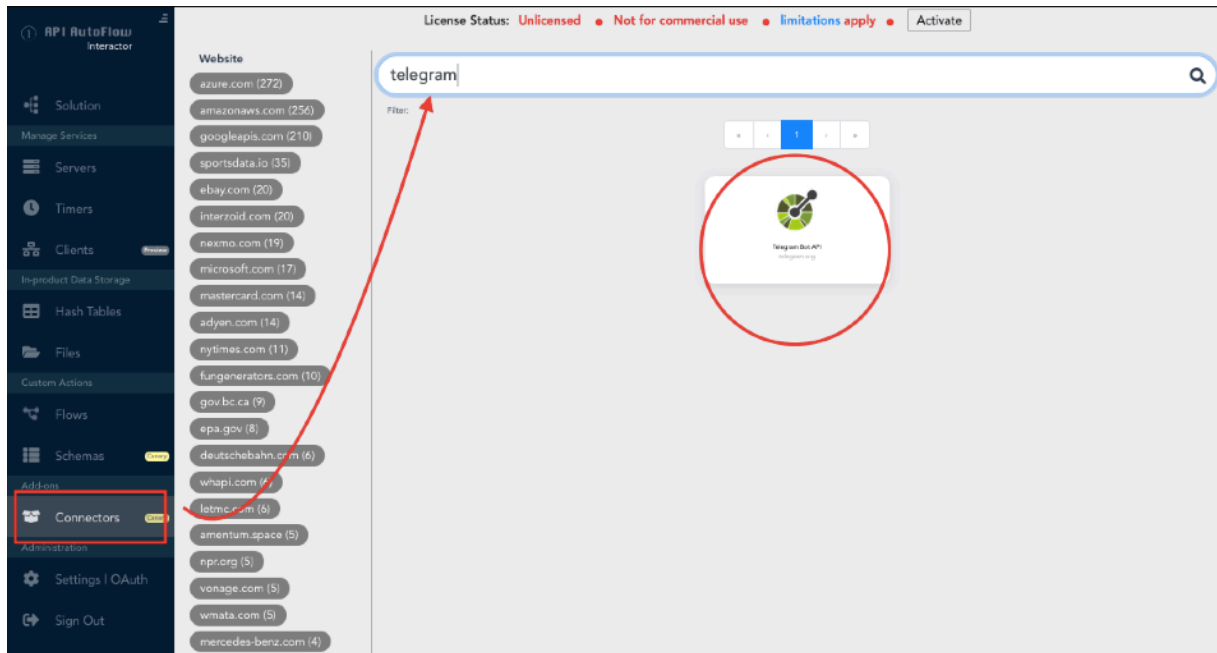


6- then you will receive a token as shown in the above picture

Next Step:

Installing Telegram Connector on API AutoFlow:

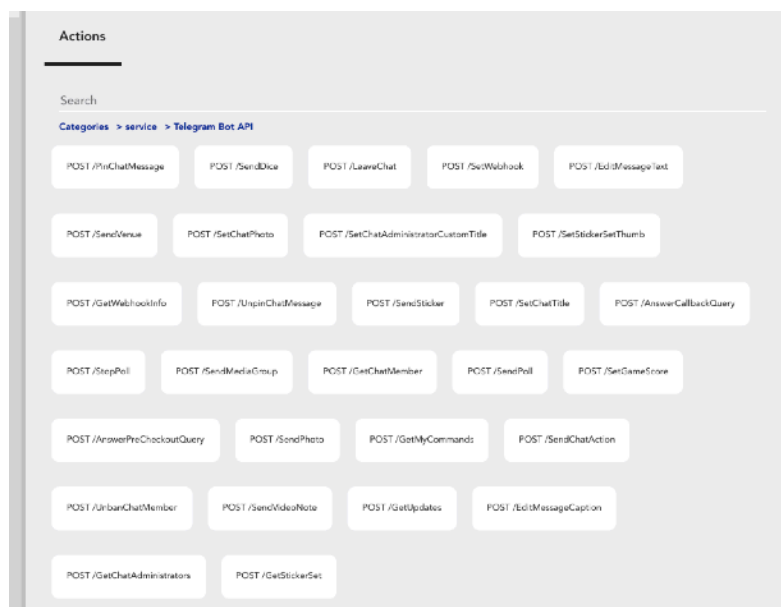
- 1- Go to Connectors
 - 2- search for “telegram”
 - 3- clic on it and just press install
- You are done



Now you can use Telegram Actions in the solution. Go to Actions, Service, Telegram Bot API

You will find all required actions here.

Before starting using these actions you need to know some basics about “Telegram Bot API”



General Information:

Reference: <https://core.telegram.org/bots/api>

in order to use API, you need to send an HTTPS Request (both GET and POST methods work). The request format is:c

https://api.telegram.org/bot<token>/<Method_Name>

- 1- **<token>** should be replaced with your BOT's token, that received in Setup Section
- 2- **<Method_Name>** should be replaced with available methods

To begin with, you can use these two following methods:

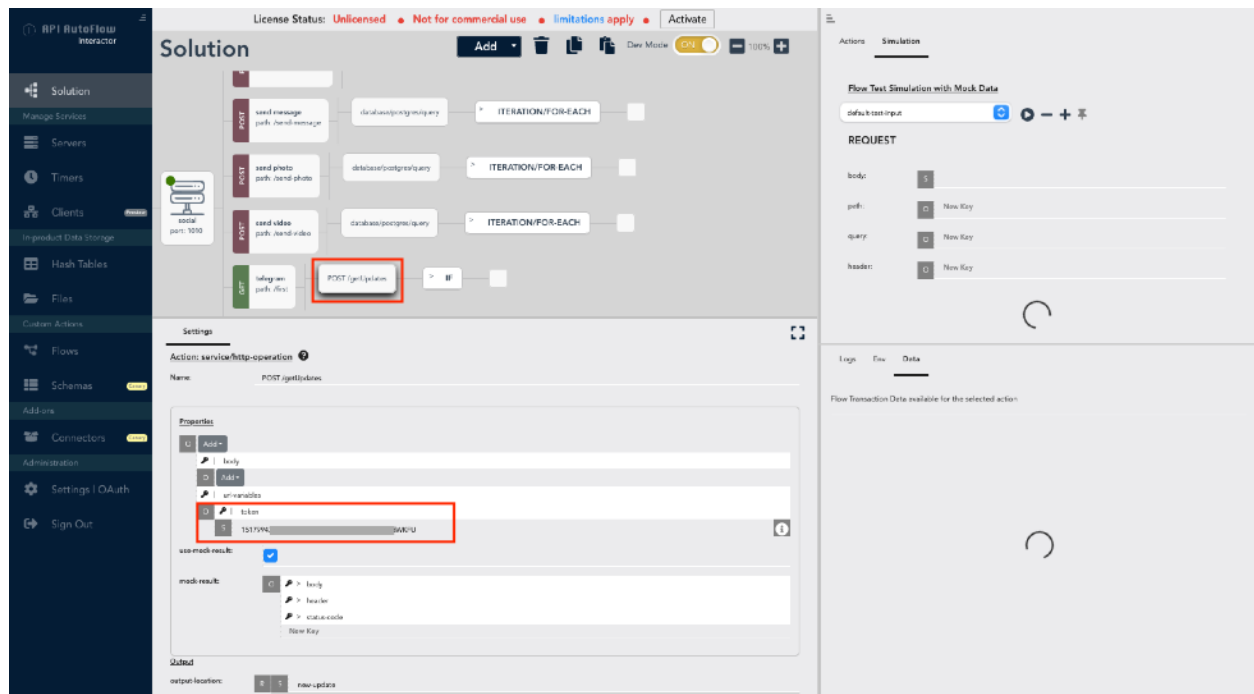
1- getMe:

this method gives information about your bot

2- getUpdates:

this method gives latest updates, including all types of messages that have been sent to your bot. This is an important method that is described in more details.

getUpdates: To use this method, you need to provide your token in the parameters.



This method returns a list of JSON format messages. Including some important information like:

- 1- chat_id: this is a unique number that is given to a chat between bot and another telegram user. In order to send a messages to a users, you will need to insert this chat_id.
- 2- username: this is username of the message sender
- 3- date: unix format date and time

An example of getUpdates method:

```
{"ok":true,"result":[{"update_id":305986041,"message":{"message_id":54,"from":{"id":132368291,"is_bot":false,"first_name":"Navid","username":"navidely","language_code":"en"},"chat":{"id":132368291,"first_name":"Navid","username":"navidely","type":"private"},"date":1617539924,"forward_from":{"id":132368291,"is_bot":false,"first_name":"Navid","username":"navidely","language_code":"en"},"forward_date":1617395178,"photo":{"file_id":"AgACAgQAAxkBAAM2YGmzVOj6PnzM7KcUsLNBkbsakslAAtq1MRsr-EFTZw1cgmiu92fKh8kmXQADAQADAgADbQADyykHAAEeBA","file_unique_id":"AQADyofJJl0AA8spBwAB","file_size":10048,"width":320,"height":240}}}]}
```

Each message has a unique “**update id**” and also some information regarding to **message**. Messages are generally in 9 categories: (1) Photo (2) Video (3) Document (4) Text (5) Sticker (6) Animation (7) Location (8) Contact (9) Poll.

For more information you can visit this reference link: <https://core.telegram.org/bots/api>
And in the getting updates section you will find more description.

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

Telegram Bots > Telegram Bot API

Telegram Bot API

The Bot API is an HTTP-based interface created for developers keen on building bots for Telegram. To learn how to create and set up a bot, please consult our [Introduction to Bots](#) and [Bot FAQ](#).

Recent changes

Subscribe to [@BotNews](#) to be the first to know about the latest updates and join the discussion in [@BotTalk](#).

March 9, 2021
Bot API 5.1

Added two new update types

- Added updates about member status changes in chats, represented by the class [ChatMemberUpdated](#) and the fields `my_chat_member` and `chat_member` in the [Update](#) class. The bot must be an administrator in the chat to receive `chat_member` updates about other chat members. By default, only `my_chat_member` updates about the bot itself are received.

Improved Invite Links

- Added the class [ChatInviteLink](#), representing an invite link to a chat.
- Added the method [createChatInviteLink](#), which can be used to create new invite links in addition to the primary invite link.
- Added the method [editChatInviteLink](#), which can be used to edit non-primary invite links created by the bot.
- Added the method [revokeChatInviteLink](#), which can be used to revoke invite links created by the bot.

Voice Chat Info

- Added the type [VoiceChatStarted](#) and the field `voice_chat_started` to the class [Message](#).
- Added the type [VoiceChatEnded](#) and the field `voice_chat_ended` to the class [Message](#).
- Added the type [VoiceChatParticipantsInvited](#) and the field `voice_chat_participants_invited` to the class [Message](#).
- Added the new administrator privilege `can_manage_voice_chats` to the class [ChatMember](#) and parameter `can_manage_voice_chats` to the method [promoteChatMember](#). For now, bots can use this privilege only for passing to other administrators.

And More

- Added the type [MessageAutoDeleteTimerChanged](#) and the field `message_auto_delete_timer_changed` to the class [Message](#).
- Added the parameter `revoke_messages` to the method [kickChatMember](#), allowing to delete all messages from a group for the user who is being removed.
- Added the new administrator privilege `can_manage_chat` to the class [ChatMember](#) and parameter `can_manage_chat` to the method [promoteChatMember](#).

Recent changes

[Authorizing your bot](#)

[Making requests](#)

[Using a Local Bot API ...](#)

[Getting updates](#)

[Available types](#)

[Available methods](#)

[Updating messages](#)

[Stickers](#)

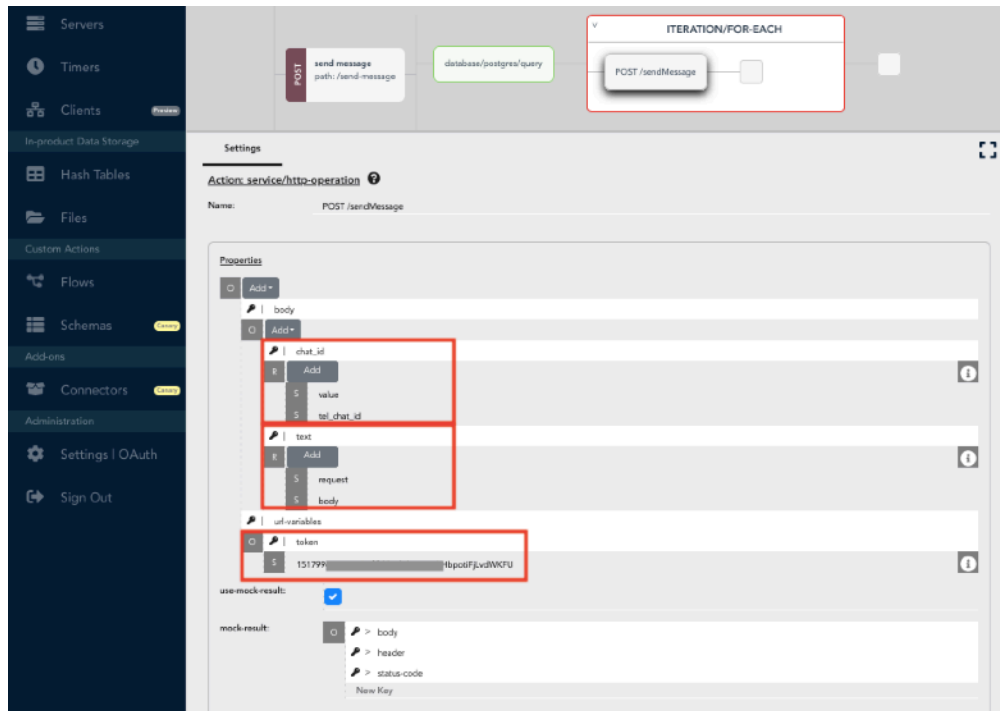
[Inline mode](#)

[Payments](#)

[Telegram Passport](#)

[Games](#)

sendMessage: to send a message you need a [chat id](#) and [text](#) parameters.



For more information you can visit this website, <https://core.telegram.org/bots/api>, And in available methods section and sendMessage you will find all required information to use this method

sendMessage
Use this method to send text messages. On success, the sent [Message](#) is returned.

Parameter	Type	Required	Description
chat_id	Integer or String	Yes	Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>)
text	String	Yes	Text of the message to be sent, 1-4096 characters after entities parsing
parse_mode	String	Optional	Mode for parsing entities in the message text. See formatting options for more details.
entities	Array of MessageEntity	Optional	List of special entities that appear in message text, which can be specified instead of <code>parse_mode</code>
disable_web_page_preview	Boolean	Optional	Disables link previews for links in this message
disable_notification	Boolean	Optional	Sends the message silently . Users will receive a notification with no sound.
reply_to_message_id	Integer	Optional	If the message is a reply, ID of the original message
allow_sending_without_reply	Boolean	Optional	Pass <code>True</code> , if the message should be sent even if the specified replied-to message is not found
reply_markup	InlineKeyboardMarkup or ReplyKeyboardMarkup or ReplyKeyboardRemove or ForceReply	Optional	Additional interface options. A JSON-serialized object for an inline keyboard , custom reply keyboard , instructions to remove reply keyboard or to force a reply from the user.

Recent changes
[Authorizing your bot](#)
[Making requests](#)
[Using a Local Bot API ...](#)
[Getting updates](#)
[Available types](#)
[Available methods](#)
[sendMessage](#)
[Formatting options](#)
[forwardMessage](#)
[copyMessage](#)
[sendPhoto](#)
[sendAudio](#)
[sendDocument](#)
[sendVideo](#)
[sendAnimation](#)
[sendVoice](#)
[sendVideoNote](#)
[sendMediaGroup](#)
[sendLocation](#)
[editMessageLiveLocation](#)
[stopMessageLiveLocation](#)
[sendVenue](#)
[sendContact](#)
[sendPoll](#)
[sendDice](#)
[sendChatAction](#)
[getUserProfilePhotos](#)
[getFile](#)
[kickChatMember](#)
[unbanChatMember](#)
[restrictChatMember](#)
[promoteChatMember](#)
[setChatAdministratorC...](#)
[setChatPermissions](#)
[exportChatInviteLink](#)
[createChatInviteLink](#)
[editChatInviteLink](#)
[revokeChatInviteLink](#)
[setChatPhoto](#)
[deleteChatPhoto](#)
[setProfilePhoto](#)

Formatting options
The Bot API supports basic formatting for messages. You can use bold, italic, underlined and strikethrough text, as well as inline links and pre-formatted code in your bots' messages. Telegram clients will render them accordingly. You can use either markdown-style or HTML-style formatting.
Note that Telegram clients will display an alert to the user before opening an inline link ('Open this link?' together with the full URL).

SendVideo: to send a video you need a chat_id and file_id. **file_id** is a unique id for each video on the telegram's server. You can find it in the getUpdates response. You can find more information about sendVideo method and lots of other methods in the reference website, <https://core.telegram.org/bots/api>.

Available methods

All methods in the Bot API are case-insensitive. We support GET and POST HTTP methods. Use either [URL query string](#) or *application/json* or *application/x-www-form-urlencoded* or *multipart/form-data* for passing parameters in Bot API requests.

On successful call, a JSON-object containing the result will be returned.

getMe

A simple method for testing your bot's auth token. Requires no parameters. Returns basic information about the bot in form of a [User](#) object.

logOut

Use this method to log out from the cloud Bot API server before launching the bot locally. You **must** log out the bot before running it locally, otherwise there is no guarantee that the bot will receive updates. After a successful call, you can immediately log in on a local server, but will not be able to log in back to the cloud Bot API server for 10 minutes. Returns *True* on success. Requires no parameters.

close

Use this method to close the bot instance before moving it from one local server to another. You need to delete the webhook before calling this method to ensure that the bot isn't launched again after server restart. The method will return error 429 in the first 10 minutes after the bot is launched. Returns *True* on success. Requires no parameters.

sendMessage

Use this method to send text messages. On success, the sent [Message](#) is returned.

Parameter	Type	Required	Description
chat_id	Integer or String	Yes	Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>)
text	String	Yes	Text of the message to be sent, 1–4096 characters after entities parsing
parse_mode	String	Optional	Mode for parsing entities in the message text. See formatting options for more details.
entities	Array of MessageEntity	Optional	List of special entities that appear in message text, which can be specified instead of <i>parse_mode</i>
disable_web_page_preview	Boolean	Optional	Disables link previews for links in this

Available types

User
Chat
Message
MessageId
MessageEntity
PhotoSize
Animation
Audio
Document
Video
VideoNote
Voice
Contact
Dice
PollOption
PollAnswer
Poll
Location
Venue
ProximityAlertTriggered
MessageAutoDeleteTim...
VoiceChatStarted
VoiceChatEnded
VoiceChatParticipantsIn...
UserProfilePhotos
File
ReplyKeyboardMarkup
KeyboardButton
KeyboardButtonPollType
ReplyKeyboardRemove
InlineKeyboardMarkup
InlineKeyboardButton
LoginUrl
CallbackQuery
ForceReply
ChatPhoto
ChatInviteLink
ChatMember
ChatMemberUpdated
ChatPermissions
ChatLocation
BotCommand
ResponseParameters
InputMedia
InputMediaPhoto
InputMediaVideo

List of Actions (API Requests): (Sorted)

POST /getGameHighScores
POST /getChatMembersCount
POST /getMe
POST /getWebhookInfo
POST /getChat
POST /getUpdates
POST /getStickerSet
POST /getChatAdministrators
POST /getUserProfilePhotos
POST /getFile
POST /getMyCommands
POST /getChatMember
POST /sendAnimation
POST /sendContact
POST /sendAudio
POST /sendDocument
POST /sendLocation
POST /sendDice
POST /sendChatAction
POST /sendMediaGroup
POST /sendSticker
POST /sendGame
POST /sendMessage
POST /sendVenue
POST /sendVideoNote
POST /sendPhoto
POST /sendVideo
POST /sendInvoice
POST /sendPoll
POST /sendVoice
POST /setChatAdministratorCustomTitle
POST /promoteChatMember
POST /stopMessageLiveLocation
POST /pinChatMessage
POST /createNewStickerSet
POST /editMessageReplyMarkup
POST /deleteWebhook
POST /setChatStickerSet
POST /unpinAllChatMessages
POST /answerPreCheckoutQuery
POST /answerInlineQuery
POST /unpinChatMessage
POST /uploadStickerFile

POST /unbanChatMember
POST /addStickerToSet
POST /setStickerPositionInSet
POST /close
POST /deleteChatPhoto
POST /editMessageText
POST /editMessageCaption
POST /setGameScore
POST /setStickerSetThumb
POST /exportChatInviteLink
POST /leaveChat
POST /copyMessage
POST /editMessageLiveLocation
POST /setChatPhoto
POST /restrictChatMember
POST /forwardMessage
POST /answerShippingQuery
POST /kickChatMember
POST /setMyCommands
POST /setChatTitle
POST /logOut
POST /setChatDescription
POST /setPassportDataErrors
POST /deleteChatStickerSet
POST /setWebhook
POST /deleteMessage
POST /setChatPermissions
POST /answerCallbackQuery
POST /deleteStickerFromSet
POST /editMessageMedia
POST /stopPoll