

[Home](#) [API](#) [Protocol](#) [Schema](#)

Recent changes	Authorizing your bot	Making requests	Using a Local Bot API Server	Getting updates
--------------------------------	--------------------------------------	---------------------------------	--	---------------------------------

August 15, 2025	Making requests when getting updates	Do I need a Local Bot API Server	Update
---------------------------------	--	--	------------------------

July 3, 2025	getUpdates
------------------------------	----------------------------

April 11, 2025	setWebhook
--------------------------------	----------------------------

	deleteWebhook
--	-------------------------------

	getWebhookInfo
--	--------------------------------

	WebhookInfo
--	-----------------------------

Available types	Available methods	Updating messages	Stickers
---------------------------------	-----------------------------------	-----------------------------------	--------------------------

User	getMe	editMessageText	Sticker
----------------------	-----------------------	---------------------------------	-------------------------

Chat	logOut	editMessageCaption	StickerSet
----------------------	------------------------	------------------------------------	----------------------------

ChatFullInfo	close	editMessageMedia	MaskPosition
------------------------------	-----------------------	----------------------------------	------------------------------

Message	sendMessage	editMessageLiveLocation	InputSticker
-------------------------	-----------------------------	---	------------------------------

MessageId	Formatting options	stopMessageLiveLocation	sendSticker
---------------------------	------------------------------------	---	-----------------------------

InaccessibleMessage	Paid Broadcasts	editMessageChecklist	getStickerSet
-------------------------------------	---------------------------------	--------------------------------------	-------------------------------

MaybelnaccessibleMessage	forwardMessage	editMessageReplyMarkup	getCustomEmojiStickers
--	--------------------------------	--	--

MessageEntity	forwardMessages	stopPoll	uploadStickerFile
-------------------------------	---------------------------------	--------------------------	-----------------------------------

TextQuote	copyMessage	approveSuggestedPost	createNewStickerSet
---------------------------	-----------------------------	--------------------------------------	-------------------------------------

ExternalReplyInfo	copyMessages	declineSuggestedPost	addStickerToSet
-----------------------------------	------------------------------	--------------------------------------	---------------------------------

ReplyParameters	sendPhoto	deleteMessage	setStickerPositionInSet
---------------------------------	---------------------------	-------------------------------	---

MessageOrigin	sendAudio	deleteMessages	deleteStickerFromSet
-------------------------------	---------------------------	--------------------------------	--------------------------------------

MessageOriginUser	sendDocument	replaceStickerInSet
-----------------------------------	------------------------------	-------------------------------------

MessageOriginHiddenUser	sendVideo	setStickerEmojiList
---	---------------------------	-------------------------------------

MessageOriginChat	sendAnimation	setStickerKeywords
-----------------------------------	-------------------------------	------------------------------------

MessageOriginChannel	sendVoice	setStickerMaskPosition
--------------------------------------	---------------------------	--

PhotoSize	sendVideoNote	setStickerSetTitle
---------------------------	-------------------------------	------------------------------------

Animation	sendPaidMedia	setStickerSetThumbnail
---------------------------	-------------------------------	--

Audio	sendMediaGroup	setCustomEmojiStickerSetThumbnail
-----------------------	--------------------------------	---

Document	sendLocation	deleteStickerSet
--------------------------	------------------------------	----------------------------------

Story	sendVenue	Inline mode	Payments
-----------------------	---------------------------	-----------------------------	--------------------------

Video	sendContact	InlineQuery	sendInvoice
-----------------------	-----------------------------	-----------------------------	-----------------------------

VideoNote	sendPoll	answerInlineQuery	createInvoiceLink
---------------------------	--------------------------	-----------------------------------	-----------------------------------

Voice	sendChecklist	InlineQueryResultsButton	answerShippingQuery
-----------------------	-------------------------------	--	-------------------------------------

PaidMediaInfo	sendDice	InlineQueryResult	answerPreCheckoutQuery
-------------------------------	--------------------------	-----------------------------------	--

PaidMedia	sendChatAction	InlineQueryResultArticle	getMyStarBalance
---------------------------	--------------------------------	--	----------------------------------

PaidMediaPreview	setMessageReaction	InlineQueryResultPhoto	getStarTransactions
----------------------------------	------------------------------------	--	-------------------------------------

PaidMediaPhoto	getUserProfilePhotos	InlineQueryResultGif	refundStarPayment
--------------------------------	--------------------------------------	--------------------------------------	-----------------------------------

PaidMediaVideo	setUserEmojiStatus	InlineQueryResultMpeg4Gif	editUserStarSubscription
--------------------------------	------------------------------------	---	--

Contact	getFile	InlineQueryResultVideo	LabeledPrice
-------------------------	-------------------------	--	------------------------------

Dice	banChatMember	InlineQueryResultAudio	Invoice
----------------------	-------------------------------	--	-------------------------

PollOption	unbanChatMember	InlineQueryResultVoice	ShippingAddress
InputPollOption	restrictChatMember	InlineQueryResultDocument	OrderInfo
PollAnswer	promoteChatMember	InlineQueryResultLocation	ShippingOption
Poll	setChatAdministratorCustomTitle	InlineQueryResultVenue	SuccessfulPayment
ChecklistTask	banChatSenderChat	InlineQueryResultContact	RefundedPayment
Checklist	unbanChatSenderChat	InlineQueryResultGame	ShippingQuery
InputChecklistTask	setChatPermissions	InlineQueryResultCachedPhoto	PreCheckoutQuery
InputChecklist	exportChatInviteLink	InlineQueryResultCachedGif	PaidMediaPurchased
ChecklistTasksDone	createChatInviteLink	InlineQueryResultCachedMpeg4Gif	RevenueWithdrawalState
ChecklistTasksAdded	editChatInviteLink	InlineQueryResultCachedSticker	RevenueWithdrawalStatePending
Location	createChatSubscriptionInviteLink	InlineQueryResultCachedDocument	RevenueWithdrawalStateSucceeded
Venue	editChatSubscriptionInviteLink	InlineQueryResultCachedVideo	RevenueWithdrawalStateFailed
WebAppData	revokeChatInviteLink	InlineQueryResultCachedVoice	AffiliateInfo
ProximityAlertTriggered	approveChatJoinRequest	InlineQueryResultCachedAudio	TransactionPartner
MessageAutoDeleteTimerChanged	declineChatJoinRequest	InputMessageContent	TransactionPartnerUser
ChatBoostAdded	setChatPhoto	InputTextMessageContent	TransactionPartnerChat
BackgroundFill	deleteChatPhoto	InputLocationMessageContent	TransactionPartnerAffiliateProgram
BackgroundFillSolid	setChatTitle	InputVenueMessageContent	TransactionPartnerFragment
BackgroundFillGradient	setChatDescription	InputContactMessageContent	TransactionPartnerTelegramAds
BackgroundFillFreeformGradient	pinChatMessage	InputInvoiceMessageContent	TransactionPartnerTelegramApi
BackgroundType	unpinChatMessage	ChosenInlineResult	TransactionPartnerOther
BackgroundTypeFill	unpinAllChatMessages	answerWebAppQuery	StarTransaction
BackgroundTypeWallpaper	leaveChat	SentWebAppMessage	StarTransactions
BackgroundTypePattern	getChat	savePreparedInlineMessage	Telegram Passport
BackgroundTypeChatTheme	getChatAdministrators	PreparedInlineMessage	PassportData
ChatBackground	getChatMemberCount		PassportFile
ForumTopicCreated	getChatMember		EncryptedPassportElement
ForumTopicClosed	setChatStickerSet		EncryptedCredentials
ForumTopicEdited	deleteChatStickerSet		setPassportDataErrors
ForumTopicReopened	getForumTopicIconStickers		PassportElementError
GeneralForumTopicHidden	createForumTopic		PassportElementErrorDataField
GeneralForumTopicUnhidden	editForumTopic		PassportElementErrorFrontSide
SharedUser	closeForumTopic		PassportElementErrorReverseSide
UsersShared	reopenForumTopic		PassportElementErrorSelfie
ChatShared	deleteForumTopic		PassportElementErrorFile
WriteAccessAllowed	unpinAllForumTopicMessages		PassportElementErrorFiles
VideoChatScheduled	editGeneralForumTopic		PassportElementErrorTranslationFile
VideoChatStarted	closeGeneralForumTopic		PassportElementErrorTranslationFiles
VideoChatEnded	reopenGeneralForumTopic		PassportElementErrorUnspecified
VideoChatParticipantsInvited	hideGeneralForumTopic	Games	
PaidMessagePriceChanged	unhideGeneralForumTopic	sendGame	
DirectMessagePriceChanged	unpinAllGeneralForumTopicMessages		

SuggestedPostApproved	answerCallbackQuery	Game
SuggestedPostApprovalFailed	getUserChatBoosts	CallbackGame
SuggestedPostDeclined	getBusinessConnection	setGameScore
SuggestedPostPaid	setMyCommands	getGameHighScores
SuggestedPostRefunded	deleteMyCommands	GameHighScore
GiveawayCreated	getMyCommands	
Giveaway	setMyName	
GiveawayWinners	getMyName	
GiveawayCompleted	setMyDescription	
LinkPreviewOptions	getMyDescription	
SuggestedPostPrice	setMyShortDescription	
SuggestedPostInfo	getMyShortDescription	
SuggestedPostParameters	setChatMenuButton	
DirectMessagesTopic	getChatMenuButton	
UserProfilePhotos	setMyDefaultAdministratorRights	
File	getMyDefaultAdministratorRights	
WebAppInfo	getAvailableGifts	
ReplyKeyboardMarkup	sendGift	
KeyboardButton	giftPremiumSubscription	
KeyboardButtonRequestUsers	verifyUser	
KeyboardButtonRequestChat	verifyChat	
KeyboardButtonPollType	removeUserVerification	
ReplyKeyboardRemove	removeChatVerification	
InlineKeyboardMarkup	readBusinessMessage	
InlineKeyboardButton	deleteBusinessMessages	
LoginUrl	setBusinessAccountName	
SwitchInlineQueryChosenChat	setBusinessAccountUsername	
CopyTextButton	setBusinessAccountBio	
CallbackQuery	setBusinessAccountProfilePhoto	
ForceReply	removeBusinessAccountProfilePhoto	
ChatPhoto	setBusinessAccountGiftSettings	
ChatInviteLink	getBusinessAccountStarBalance	
ChatAdministratorRights	transferBusinessAccountStars	
ChatMemberUpdated	getBusinessAccountGifts	
ChatMember	convertGiftToStars	
ChatMemberOwner	upgradeGift	
ChatMemberAdministrator	transferGift	
ChatMemberMember	postStory	
ChatMemberRestricted	editStory	
ChatMemberLeft	deleteStory	
ChatMemberBanned	Inline mode methods	
ChatJoinRequest		

ChatPermissions

Birthdate

BusinessIntro

BusinessLocation

BusinessOpeningHoursInterval

BusinessOpeningHours

StoryAreaPosition

LocationAddress

StoryAreaType

StoryAreaTypeLocation

StoryAreaTypeSuggestedReaction

StoryAreaTypeLink

StoryAreaTypeWeather

StoryAreaTypeUniqueGift

StoryArea

ChatLocation

ReactionType

ReactionTypeEmoji

ReactionTypeCustomEmoji

ReactionTypePaid

ReactionCount

MessageReactionUpdated

MessageReactionCountUpdated

ForumTopic

Gift

Gifts

UniqueGiftModel

UniqueGiftSymbol

UniqueGiftBackdropColors

UniqueGiftBackdrop

UniqueGift

GiftInfo

UniqueGiftInfo

OwnedGift

OwnedGiftRegular

OwnedGiftUnique

OwnedGifts

AcceptedGiftTypes

StarAmount

BotCommand

BotCommandScope

Determining list of commands

BotCommandScopeDefault
BotCommandScopeAllPrivateChats
BotCommandScopeAllGroupChats
BotCommandScopeAllChatAdministrators
BotCommandScopeChat
BotCommandScopeChatAdministrators
BotCommandScopeChatMember
BotName
BotDescription
BotShortDescription
MenuButton
MenuButtonCommands
MenuButtonWebApp
MenuButtonDefault
ChatBoostSource
ChatBoostSourcePremium
ChatBoostSourceGiftCode
ChatBoostSourceGiveaway
ChatBoost
ChatBoostUpdated
ChatBoostRemoved
UserChatBoosts
BusinessBotRights
BusinessConnection
BusinessMessagesDeleted
ResponseParameters
InputMedia
InputMediaPhoto
InputMediaVideo
InputMediaAnimation
InputMediaAudio
InputMediaDocument
InputFile
InputPaidMedia
InputPaidMediaPhoto
InputPaidMediaVideo
InputProfilePhoto
InputProfilePhotoStatic
InputProfilePhotoAnimated
InputStoryContent
InputStoryContentPhoto
InputStoryContentVideo

[Sending files](#)[Accent colors](#)[Profile accent colors](#)[Inline mode objects](#)[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](#)

August 15, 2025

[Bot API 9.2](#)

Checklists

- Added the field `checklist_task_id` to the class [ReplyParameters](#), allowing bots to reply to a specific checklist task.
- Added the field `reply_to_checklist_task_id` to the class [Message](#).

Gifts

- Added the field `publisher_chat` to the classes [Gift](#) and [UniqueGift](#) which can be used to get information about the chat that published a gift.

Direct Messages in Channels

- Added the field `is_direct_messages` to the classes [Chat](#) and [ChatFullInfo](#) which can be used to identify supergroups that are used as channel direct messages chats.
- Added the field `parent_chat` to the class [ChatFullInfo](#) which indicates the parent channel chat for a channel direct messages chat.
- Added the class [DirectMessagesTopic](#) and the field `direct_messages_topic` to the class [Message](#), describing a topic of a direct messages chat.
- Added the parameter `direct_messages_topic_id` to the methods `sendMessage`, `sendPhoto`, `sendVideo`, `sendAnimation`, `sendAudio`, `sendDocument`, `sendPaidMedia`, `sendSticker`, `sendVideoNote`, `sendVoice`, `sendLocation`, `sendVenue`, `sendContact`, `sendDice`, `sendInvoice`, `sendMediaGroup`, `copyMessage`, `copyMessages`, `forwardMessage` and `forwardMessages`. This parameter can be used to send a message to a direct messages chat topic.

Suggested Posts

- Added the class [SuggestedPostParameters](#) and the parameter `suggested_post_parameters` to the methods `sendMessage`, `sendPhoto`, `sendVideo`, `sendAnimation`, `sendAudio`, `sendDocument`, `sendPaidMedia`, `sendSticker`, `sendVideoNote`, `sendVoice`, `sendLocation`, `sendVenue`, `sendContact`, `sendDice`, `sendInvoice`, `copyMessage`, `forwardMessage`. This parameter can be used to send a suggested post to a direct messages chat topic.
- Added the method `approveSuggestedPost`, allowing bots to approve incoming suggested posts.
- Added the method `declineSuggestedPost`, allowing bots to decline incoming suggested posts.
- Added the field `can_manage_direct_messages` to the classes [ChatMemberAdministrator](#) and [ChatAdministratorRights](#).
- Added the parameter `can_manage_direct_messages` to the method `promoteChatMember`.
- Added the field `is_paid_post` to the class [Message](#), which can be used to identify paid posts. Such posts must not be deleted for 24 hours to receive the payment.
- Added the class [SuggestedPostPrice](#), describing the price of a suggested post.
- Added the class [SuggestedPostInfo](#) and the field `suggested_post_info` to the class [Message](#), describing a suggested post.
- Added the class [SuggestedPostApproved](#) and the field `suggested_post_approved` to the class [Message](#), describing a service message about the approval of a suggested post.
- Added the class [SuggestedPostApprovalFailed](#) and the field `suggested_post_approval_failed` to the class [Message](#), describing a service message about the failed approval of a suggested post.
- Added the class [SuggestedPostDeclined](#) and the field `suggested_post_declined` to the class [Message](#), describing a service message about the rejection of a suggested post.
- Added the class [SuggestedPostPaid](#) and the field `suggested_post_paid` to the class [Message](#), describing a service message about a successful payment for a suggested post.
- Added the class [SuggestedPostRefunded](#) and the field `suggested_post_refunded` to the class [Message](#), describing a service message about a payment refund for a suggested post.

July 3, 2025

[Bot API 9.1](#)

Checklists

- Added the class [ChecklistTask](#) representing a task in a checklist.
- Added the class [Checklist](#) representing a checklist.
- Added the class [InputChecklistTask](#) representing a task to add to a checklist.
- Added the class [InputChecklist](#) representing a checklist to create.
- Added the field `checklist` to the classes [Message](#) and [ExternalReplyInfo](#), describing a checklist in a message.
- Added the class [ChecklistTasksDone](#) and the field `checklist_tasks_done` to the class [Message](#), describing a service message about status changes for tasks in a checklist (i.e., marked as done/not done).
- Added the class [ChecklistTasksAdded](#) and the field `checklist_tasks_added` to the class [Message](#), describing a service message about the addition of new tasks to a checklist.
- Added the method `sendChecklist`, allowing bots to send a checklist on behalf of a business account.
- Added the method `editMessageChecklist`, allowing bots to edit a checklist on behalf of a business account.

Gifts

- Added the field `next_transfer_date` to the classes [OwnedGiftUnique](#) and [UniqueGiftInfo](#).
- Added the field `last_resale_star_count` to the class [UniqueGiftInfo](#).
- Added "resale" as the possible value of the field `origin` in the class [UniqueGiftInfo](#).

General

- Increased the maximum number of options in a poll to 12.
- Added the method `getMyStarBalance`, allowing bots to get their current balance of Telegram Stars.
- Added the class [DirectMessagePriceChanged](#) and the field `direct_message_price_changed` to the class [Message](#), describing a service message about a price change for direct messages sent to the channel chat.
- Added the method `hideKeyboard` to the class [WebApp](#).

April 11, 2025

[Bot API 9.0](#)

Business Accounts

- Added the class `BusinessBotRights` and replaced the field `can_reply` with the field `rights` of the type `BusinessBotRights` in the class `BusinessConnection`.
- Added the method `readBusinessMessage`, allowing bots to mark incoming messages as read on behalf of a business account.
- Added the method `deleteBusinessMessages`, allowing bots to delete messages on behalf of a business account.
- Added the method `setBusinessAccountName`, allowing bots to change the first and last name of a managed business account.
- Added the method `setBusinessAccountUsername`, allowing bots to change the username of a managed business account.
- Added the method `setBusinessAccountBio`, allowing bots to change the bio of a managed business account.
- Added the class `InputProfilePhoto`, describing a profile photo to be set.
- Added the methods `setBusinessAccountProfilePhoto` and `removeBusinessAccountProfilePhoto`, allowing bots to change the profile photo of a managed business account.
- Added the method `setBusinessAccountGiftSettings`, allowing bots to change the privacy settings pertaining to incoming gifts in a managed business account.
- Added the class `StarAmount` and the method `getBusinessAccountStarBalance`, allowing bots to check the current Telegram Star balance of a managed business account.
- Added the method `transferBusinessAccountStars`, allowing bots to transfer Telegram Stars from the balance of a managed business account to their own balance for withdrawal.
- Added the classes `OwnedGiftRegular`, `OwnedGiftUnique`, `OwnedGifts` and the method `getBusinessAccountGifts`, allowing bots to fetch the list of gifts owned by a managed business account.
- Added the method `convertGiftToStars`, allowing bots to convert gifts received by a managed business account to Telegram Stars.
- Added the method `upgradeGift`, allowing bots to upgrade regular gifts received by a managed business account to unique gifts.
- Added the method `transferGift`, allowing bots to transfer unique gifts owned by a managed business account.
- Added the classes `InputStoryContentPhoto` and `InputStoryContentVideo` representing the content of a story to post.
- Added the classes `StoryArea`, `StoryAreaPosition`, `LocationAddress`, `StoryAreaTypeLocation`, `StoryAreaTypeSuggestedReaction`, `StoryAreaTypeLink`, `StoryAreaTypeWeather` and `StoryAreaTypeUniqueGift`, describing clickable active areas on stories.
- Added the method `postStory`, allowing bots to post a story on behalf of a managed business account.
- Added the method `editStory`, allowing bots to edit stories they had previously posted on behalf of a managed business account.
- Added the method `deleteStory`, allowing bots to delete stories they had previously posted on behalf of a managed business account.

Mini Apps

- Added the field `DeviceStorage`, allowing Mini Apps to use persistent local storage on the user's device.
- Added the field `SecureStorage`, allowing Mini Apps to use a secure local storage on the user's device for sensitive data.

Gifts

- Added the classes `UniqueGiftModel`, `UniqueGiftSymbol`, `UniqueGiftBackdropColors`, and `UniqueGiftBackdrop` to describe the properties of a unique gift.
- Added the class `UniqueGift` describing a gift that was upgraded to a unique one.
- Added the class `AcceptedGiftTypes` describing the types of gifts that are accepted by a user or a chat.
- Replaced the field `can_send_gift` with the field `accepted_gift_types` of the type `AcceptedGiftTypes` in the class `ChatFullInfo`.
- Added the class `GiftInfo` and the field `gift` to the class `Message`, describing a service message about a regular gift that was sent or received.
- Added the class `UniqueGiftInfo` and the field `unique_gift` to the class `Message`, describing a service message about a unique gift that was sent or received.

Telegram Premium

- Added the method `giftPremiumSubscription`, allowing bots to gift a user a Telegram Premium subscription paid in Telegram Stars.
- Added the field `premium_subscription_duration` to the class `TransactionPartnerUser` for transactions involving a Telegram Premium subscription purchased by the bot.
- Added the field `transaction_type` to the class `TransactionPartnerUser`, simplifying the differentiation and processing of all transaction types.

General

- Increased the maximum price for paid media to 10000 Telegram Stars.
- Increased the maximum price for a subscription period to 10000 Telegram Stars.
- Added the class `PaidMessagePriceChanged` and the field `paid_message_price_changed` to the class `Message`, describing a service message about a price change for paid messages sent to the chat.
- Added the field `paid_star_count` to the class `Message`, containing the number of Telegram Stars that were paid to send the message.

[See earlier changes »](#)

Authorizing your bot

Each bot is given a unique authentication token when it is created. The token looks something like `123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11`, but we'll use simply `<token>` in this document instead. You can learn about obtaining tokens and generating new ones in [this document](#).

Making requests

All queries to the Telegram Bot API must be served over HTTPS and need to be presented in this form: https://api.telegram.org/bot<token>/METHOD_NAME. Like this for example:

```
https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/getMe
```

We support GET and POST HTTP methods. We support four ways of passing parameters in Bot API requests:

- `URL query string`
- `application/x-www-form-urlencoded`
- `application/json` (except for uploading files)
- `multipart/form-data` (use to upload files)

The response contains a JSON object, which always has a Boolean field 'ok' and may have an optional String field 'description' with a human-readable description of the result. If 'ok' equals `True`, the request was successful and the result of the query can be found in the 'result' field. In case of an unsuccessful request, 'ok' equals false and the error is explained in the 'description'. An Integer 'error_code' field is also returned, but its contents are subject to change in the future. Some errors may also have an optional field 'parameters' of the type `ResponseParameters`, which can help to automatically handle the error.

- All methods in the Bot API are case-insensitive.
- All queries must be made using UTF-8.

Making requests when getting updates

If you're using [webhooks](#), you can perform a request to the Bot API while sending an answer to the webhook. Use either `application/json` or `application/x-www-form-urlencoded` or `multipart/form-data` response content type for passing parameters. Specify the method to be invoked in the `method` parameter of the request. It's not possible to know that such a request was successful or get its result.

Please see our [FAQ](#) for examples.

Using a Local Bot API Server

The Bot API server source code is available at [telegram-bot-api](#). You can run it locally and send the requests to your own server instead of <https://api.telegram.org>. If you switch to a local Bot API server, your bot will be able to:

- Download files without a size limit.
- Upload files up to 2000 MB.
- Upload files using their local path and [the file URI scheme](#).
- Use an HTTP URL for the webhook.
- Use any local IP address for the webhook.
- Use any port for the webhook.
- Set `max_webhook_connections` up to 100000.

- Receive the absolute local path as a value of the `file_path` field without the need to download the file after a `getFile` request.

Do I need a Local Bot API Server

The majority of bots will be OK with the default configuration, running on our servers. But if you feel that you need one of [these features](#), you're welcome to switch to your own at any time.

Getting updates

There are two mutually exclusive ways of receiving updates for your bot – the `getUpdates` method on one hand and [webhooks](#) on the other. Incoming updates are stored on the server until the bot receives them either way, but they will not be kept longer than 24 hours.

Regardless of which option you choose, you will receive JSON-serialized [Update](#) objects as a result.

Update

This [object](#) represents an incoming update.

At most **one** of the optional parameters can be present in any given update.

Field	Type	Description
<code>update_id</code>	<code>Integer</code>	The update's unique identifier. Update identifiers start from a certain positive number and increase sequentially. This identifier becomes especially handy if you're using webhooks , since it allows you to ignore repeated updates or to restore the correct update sequence, should they get out of order. If there are no new updates for at least a week, then identifier of the next update will be chosen randomly instead of sequentially.
<code>message</code>	<code>Message</code>	<i>Optional.</i> New incoming message of any kind – text, photo, sticker, etc.
<code>edited_message</code>	<code>Message</code>	<i>Optional.</i> New version of a message that is known to the bot and was edited. This update may at times be triggered by changes to message fields that are either unavailable or not actively used by your bot.
<code>channel_post</code>	<code>Message</code>	<i>Optional.</i> New incoming channel post of any kind – text, photo, sticker, etc.
<code>edited_channel_post</code>	<code>Message</code>	<i>Optional.</i> New version of a channel post that is known to the bot and was edited. This update may at times be triggered by changes to message fields that are either unavailable or not actively used by your bot.
<code>business_connection</code>	<code>BusinessConnection</code>	<i>Optional.</i> The bot was connected to or disconnected from a business account, or a user edited an existing connection with the bot
<code>business_message</code>	<code>Message</code>	<i>Optional.</i> New message from a connected business account
<code>edited_business_message</code>	<code>Message</code>	<i>Optional.</i> New version of a message from a connected business account
<code>deleted_business_messages</code>	<code>BusinessMessagesDeleted</code>	<i>Optional.</i> Messages were deleted from a connected business account
<code>message_reaction</code>	<code>MessageReactionUpdated</code>	<i>Optional.</i> A reaction to a message was changed by a user. The bot must be an administrator in the chat and must explicitly specify <code>"message_reaction"</code> in the list of <code>allowed_updates</code> to receive these updates. The update isn't received for reactions set by bots.
<code>message_reaction_count</code>	<code>MessageReactionCountUpdated</code>	<i>Optional.</i> Reactions to a message with anonymous reactions were changed. The bot must be an administrator in the chat and must explicitly specify <code>"message_reaction_count"</code> in the list of <code>allowed_updates</code> to receive these updates. The updates are grouped and can be sent with delay up to a few minutes.
<code>inline_query</code>	<code>InlineQuery</code>	<i>Optional.</i> New incoming inline query
<code>chosen_inline_result</code>	<code>ChosenInlineResult</code>	<i>Optional.</i> The result of an <code>inline</code> query that was chosen by a user and sent to their chat partner. Please see our documentation on the feedback collecting for details on how to enable these updates for your bot.
<code>callback_query</code>	<code>CallbackQuery</code>	<i>Optional.</i> New incoming callback query
<code>shipping_query</code>	<code>ShippingQuery</code>	<i>Optional.</i> New incoming shipping query. Only for invoices with flexible price
<code>pre_checkout_query</code>	<code>PreCheckoutQuery</code>	<i>Optional.</i> New incoming pre-checkout query. Contains full information about checkout
<code>purchased_paid_media</code>	<code>PaidMediaPurchased</code>	<i>Optional.</i> A user purchased paid media with a non-empty payload sent by the bot in a non-channel chat
<code>poll</code>	<code>Poll</code>	<i>Optional.</i> New poll state. Bots receive only updates about manually stopped polls and polls, which are sent by the bot
<code>poll_answer</code>	<code>PollAnswer</code>	<i>Optional.</i> A user changed their answer in a non-anonymous poll. Bots receive new votes only in polls that were sent by the bot itself.
<code>my_chat_member</code>	<code>ChatMemberUpdated</code>	<i>Optional.</i> The bot's chat member status was updated in a chat. For private chats, this update is received only when the bot is blocked or unblocked by the user.
<code>chat_member</code>	<code>ChatMemberUpdated</code>	<i>Optional.</i> A chat member's status was updated in a chat. The bot must be an administrator in the chat and must explicitly specify <code>"chat_member"</code> in the list of <code>allowed_updates</code> to receive these updates.
<code>chat_join_request</code>	<code>ChatJoinRequest</code>	<i>Optional.</i> A request to join the chat has been sent. The bot must have the <code>can_invite_users</code> administrator right in the chat to receive these updates.
<code>chat_boost</code>	<code>ChatBoostUpdated</code>	<i>Optional.</i> A chat boost was added or changed. The bot must be an administrator in the chat to receive these updates.
<code>removed_chat_boost</code>	<code>ChatBoostRemoved</code>	<i>Optional.</i> A boost was removed from a chat. The bot must be an administrator in the chat to receive these updates.

getUpdates

Use this method to receive incoming updates using long polling ([wiki](#)). Returns an Array of [Update](#) objects.

Parameter	Type	Required	Description
<code>offset</code>	<code>Integer</code>	Optional	Identifier of the first update to be returned. Must be greater by one than the highest among the identifiers of previously received updates. By default, updates starting with the earliest unconfirmed update are returned. An update is considered confirmed as soon as <code>getUpdates</code>

Parameter	Type	Required	Description
is called with an <i>offset</i> higher than its <i>update_id</i> . The negative offset can be specified to retrieve updates starting from <i>-offset</i> update from the end of the updates queue. All previous updates will be forgotten.			
limit	Integer	Optional	Limits the number of updates to be retrieved. Values between 1–100 are accepted. Defaults to 100.
timeout	Integer	Optional	Timeout in seconds for long polling. Defaults to 0, i.e. usual short polling. Should be positive, short polling should be used for testing purposes only.
allowed_updates	Array of String	Optional	A JSON-serialized list of the update types you want your bot to receive. For example, specify <code>["message", "edited_channel_post", "callback_query"]</code> to only receive updates of these types. See Update for a complete list of available update types. Specify an empty list to receive all update types except <code>chat_member</code> , <code>message_reaction</code> , and <code>message_reaction_count</code> (default). If not specified, the previous setting will be used. Please note that this parameter doesn't affect updates created before the call to <code>getUpdates</code> , so unwanted updates may be received for a short period of time.

Notes

1. This method will not work if an outgoing webhook is set up.
2. In order to avoid getting duplicate updates, recalculate *offset* after each server response.

setWebhook

Use this method to specify a URL and receive incoming updates via an outgoing webhook. Whenever there is an update for the bot, we will send an HTTPS POST request to the specified URL, containing a JSON-serialized [Update](#). In case of an unsuccessful request (a request with response HTTP status code different from `2XY`), we will repeat the request and give up after a reasonable amount of attempts. Returns `True` on success.

If you'd like to make sure that the webhook was set by you, you can specify secret data in the parameter `secret_token`. If specified, the request will contain a header "X-Telegram-Bot-Api-Secret-Token" with the secret token as content.

Parameter	Type	Required	Description
url	String	Yes	HTTPS URL to send updates to. Use an empty string to remove webhook integration
certificate	InputFile	Optional	Upload your public key certificate so that the root certificate in use can be checked. See our self-signed guide for details.
ip_address	String	Optional	The fixed IP address which will be used to send webhook requests instead of the IP address resolved through DNS
max_connections	Integer	Optional	The maximum allowed number of simultaneous HTTPS connections to the webhook for update delivery, 1–100. Defaults to 40. Use lower values to limit the load on your bot's server, and higher values to increase your bot's throughput.
allowed_updates	Array of String	Optional	A JSON-serialized list of the update types you want your bot to receive. For example, specify <code>["message", "edited_channel_post", "callback_query"]</code> to only receive updates of these types. See Update for a complete list of available update types. Specify an empty list to receive all update types except <code>chat_member</code> , <code>message_reaction</code> , and <code>message_reaction_count</code> (default). If not specified, the previous setting will be used. Please note that this parameter doesn't affect updates created before the call to the <code>setWebhook</code> , so unwanted updates may be received for a short period of time.
drop_pending_updates	Boolean	Optional	Pass <code>True</code> to drop all pending updates
secret_token	String	Optional	A secret token to be sent in a header "X-Telegram-Bot-Api-Secret-Token" in every webhook request, 1–256 characters. Only characters <code>A-Z</code> , <code>a-z</code> , <code>0-9</code> , <code>_</code> and <code>-</code> are allowed. The header is useful to ensure that the request comes from a webhook set by you.

Notes

1. You will not be able to receive updates using `getUpdates` for as long as an outgoing webhook is set up.
2. To use a self-signed certificate, you need to upload your [public key certificate](#) using `certificate` parameter. Please upload as InputFile, sending a String will not work.
3. Ports currently supported for webhooks: **443, 80, 88, 8443**.

If you're having any trouble setting up webhooks, please check out this [amazing guide to webhooks](#).

deleteWebhook

Use this method to remove webhook integration if you decide to switch back to `getUpdates`. Returns `True` on success.

Parameter	Type	Required	Description
drop_pending_updates	Boolean	Optional	Pass <code>True</code> to drop all pending updates

getWebhookInfo

Use this method to get current webhook status. Requires no parameters. On success, returns a `WebhookInfo` object. If the bot is using `getUpdates`, will return an object with the `url` field empty.

WebhookInfo

Describes the current status of a webhook.

Field	Type	Description
url	String	Webhook URL, may be empty if webhook is not set up
has_custom_certificate	Boolean	<code>True</code> , if a custom certificate was provided for webhook certificate checks
pending_update_count	Integer	Number of updates awaiting delivery
ip_address	String	<i>Optional</i> . Currently used webhook IP address

Field	Type	Description
last_error_date	Integer	<i>Optional.</i> Unix time for the most recent error that happened when trying to deliver an update via webhook
last_error_message	String	<i>Optional.</i> Error message in human-readable format for the most recent error that happened when trying to deliver an update via webhook
last_synchronization_error_date	Integer	<i>Optional.</i> Unix time of the most recent error that happened when trying to synchronize available updates with Telegram datacenters
max_connections	Integer	<i>Optional.</i> The maximum allowed number of simultaneous HTTPS connections to the webhook for update delivery
allowed_updates	Array of String	<i>Optional.</i> A list of update types the bot is subscribed to. Defaults to all update types except <code>chat_member</code>

Available types

All types used in the Bot API responses are represented as JSON-objects.

It is safe to use 32-bit signed integers for storing all `Integer` fields unless otherwise noted.

Optional fields may be not returned when irrelevant.

User

This object represents a Telegram user or bot.

Field	Type	Description
id	Integer	Unique identifier for this user or bot. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a 64-bit integer or double-precision float type are safe for storing this identifier.
is_bot	Boolean	<i>True</i> , if this user is a bot
first_name	String	User's or bot's first name
last_name	String	<i>Optional.</i> User's or bot's last name
username	String	<i>Optional.</i> User's or bot's username
language_code	String	<i>Optional.</i> IETF language tag of the user's language
is_premium	True	<i>Optional.</i> <i>True</i> , if this user is a Telegram Premium user
added_to_attachment_menu	True	<i>Optional.</i> <i>True</i> , if this user added the bot to the attachment menu
can_join_groups	Boolean	<i>Optional.</i> <i>True</i> , if the bot can be invited to groups. Returned only in <code>getMe</code> .
can_read_all_group_messages	Boolean	<i>Optional.</i> <i>True</i> , if privacy mode is disabled for the bot. Returned only in <code>getMe</code> .
supports_inline_queries	Boolean	<i>Optional.</i> <i>True</i> , if the bot supports inline queries. Returned only in <code>getMe</code> .
can_connect_to_business	Boolean	<i>Optional.</i> <i>True</i> , if the bot can be connected to a Telegram Business account to receive its messages. Returned only in <code>getMe</code> .
has_main_web_app	Boolean	<i>Optional.</i> <i>True</i> , if the bot has a main Web App. Returned only in <code>getMe</code> .

Chat

This object represents a chat.

Field	Type	Description
id	Integer	Unique identifier for this chat. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.
type	String	Type of the chat, can be either "private", "group", "supergroup" or "channel"
title	String	<i>Optional.</i> Title, for supergroups, channels and group chats
username	String	<i>Optional.</i> Username, for private chats, supergroups and channels if available
first_name	String	<i>Optional.</i> First name of the other party in a private chat
last_name	String	<i>Optional.</i> Last name of the other party in a private chat
is_forum	True	<i>Optional.</i> <i>True</i> , if the supergroup chat is a forum (has <code>topics</code> enabled)
is_direct_messages	True	<i>Optional.</i> <i>True</i> , if the chat is the direct messages chat of a channel

ChatFullInfo

This object contains full information about a chat.

Field	Type	Description
id	Integer	Unique identifier for this chat. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.

Field	Type	Description
type	String	Type of the chat, can be either “private”, “group”, “supergroup” or “channel”
title	String	<i>Optional.</i> Title, for supergroups, channels and group chats
username	String	<i>Optional.</i> Username, for private chats, supergroups and channels if available
first_name	String	<i>Optional.</i> First name of the other party in a private chat
last_name	String	<i>Optional.</i> Last name of the other party in a private chat
is_forum	True	<i>Optional.</i> True, if the supergroup chat is a forum (has topics enabled)
is_direct_messages	True	<i>Optional.</i> True, if the chat is the direct messages chat of a channel
accent_color_id	Integer	Identifier of the accent color for the chat name and backgrounds of the chat photo, reply header, and link preview. See accent colors for more details.
max_reaction_count	Integer	The maximum number of reactions that can be set on a message in the chat
photo	ChatPhoto	<i>Optional.</i> Chat photo
active_usernames	Array of String	<i>Optional.</i> If non-empty, the list of all active chat usernames ; for private chats, supergroups and channels
birthdate	Birthdate	<i>Optional.</i> For private chats, the date of birth of the user
business_intro	BusinessIntro	<i>Optional.</i> For private chats with business accounts, the intro of the business
business_location	BusinessLocation	<i>Optional.</i> For private chats with business accounts, the location of the business
business_opening_hours	BusinessOpeningHours	<i>Optional.</i> For private chats with business accounts, the opening hours of the business
personal_chat	Chat	<i>Optional.</i> For private chats, the personal channel of the user
parent_chat	Chat	<i>Optional.</i> Information about the corresponding channel chat; for direct messages chats only
available_reactions	Array of ReactionType	<i>Optional.</i> List of available reactions allowed in the chat. If omitted, then all emoji reactions are allowed.
background_custom_emoji_id	String	<i>Optional.</i> Custom emoji identifier of the emoji chosen by the chat for the reply header and link preview background
profile_accent_color_id	Integer	<i>Optional.</i> Identifier of the accent color for the chat's profile background. See profile accent colors for more details.
profile_background_custom_emoji_id	String	<i>Optional.</i> Custom emoji identifier of the emoji chosen by the chat for its profile background
emoji_status_custom_emoji_id	String	<i>Optional.</i> Custom emoji identifier of the emoji status of the chat or the other party in a private chat
emoji_status_expiration_date	Integer	<i>Optional.</i> Expiration date of the emoji status of the chat or the other party in a private chat, in Unix time, if any
bio	String	<i>Optional.</i> Bio of the other party in a private chat
has_private_forwards	True	<i>Optional.</i> True, if privacy settings of the other party in the private chat allows to use <a href="tg://user?id=<user_id>">tg://user?id=<user_id> links only in chats with the user
has_restricted_voice_and_video_messages	True	<i>Optional.</i> True, if the privacy settings of the other party restrict sending voice and video note messages in the private chat
join_to_send_messages	True	<i>Optional.</i> True, if users need to join the supergroup before they can send messages
join_by_request	True	<i>Optional.</i> True, if all users directly joining the supergroup without using an invite link need to be approved by supergroup administrators
description	String	<i>Optional.</i> Description, for groups, supergroups and channel chats
invite_link	String	<i>Optional.</i> Primary invite link, for groups, supergroups and channel chats
pinned_message	Message	<i>Optional.</i> The most recent pinned message (by sending date)
permissions	ChatPermissions	<i>Optional.</i> Default chat member permissions, for groups and supergroups
accepted_gift_types	AcceptedGiftTypes	Information about types of gifts that are accepted by the chat or by the corresponding user for private chats
can_send_paid_media	True	<i>Optional.</i> True, if paid media messages can be sent or forwarded to the channel chat. The field is available only for channel chats.
slow_mode_delay	Integer	<i>Optional.</i> For supergroups, the minimum allowed delay between consecutive messages sent by each unprivileged user; in seconds
unrestrict_boost_count	Integer	<i>Optional.</i> For supergroups, the minimum number of boosts that a non-administrator user needs to add in order to ignore slow mode and chat permissions
message_auto_delete_time	Integer	<i>Optional.</i> The time after which all messages sent to the chat will be automatically deleted; in seconds
has_aggressive_anti_spam_enabled	True	<i>Optional.</i> True, if aggressive anti-spam checks are enabled in the supergroup. The field is only available to chat administrators.
has_hidden_members	True	<i>Optional.</i> True, if non-administrators can only get the list of bots and administrators in the chat

Field	Type	Description
has_protected_content	True	<i>Optional.</i> True, if messages from the chat can't be forwarded to other chats
has_visible_history	True	<i>Optional.</i> True, if new chat members will have access to old messages; available only to chat administrators
sticker_set_name	String	<i>Optional.</i> For supergroups, name of the group sticker set
can_set_sticker_set	True	<i>Optional.</i> True, if the bot can change the group sticker set
custom_emoji_sticker_set_name	String	<i>Optional.</i> For supergroups, the name of the group's custom emoji sticker set. Custom emoji from this set can be used by all users and bots in the group.
linked_chat_id	Integer	<i>Optional.</i> Unique identifier for the linked chat, i.e. the discussion group identifier for a channel and vice versa; for supergroups and channel chats. This identifier may be greater than 32 bits and some programming languages may have difficulty/silent defects in interpreting it. But it is smaller than 52 bits, so a signed 64 bit integer or double-precision float type are safe for storing this identifier.
location	ChatLocation	<i>Optional.</i> For supergroups, the location to which the supergroup is connected

Message

This object represents a message.

Field	Type	Description
message_id	Integer	Unique message identifier inside this chat. In specific instances (e.g., message containing a video sent to a big chat), the server might automatically schedule a message instead of sending it immediately. In such cases, this field will be 0 and the relevant message will be unusable until it is actually sent
message_thread_id	Integer	<i>Optional.</i> Unique identifier of a message thread to which the message belongs; for supergroups only
direct_messages_topic	DirectMessagesTopic	<i>Optional.</i> Information about the direct messages chat topic that contains the message
from	User	<i>Optional.</i> Sender of the message; may be empty for messages sent to channels. For backward compatibility, if the message was sent on behalf of a chat, the field contains a fake sender user in non-channel chats
sender_chat	Chat	<i>Optional.</i> Sender of the message when sent on behalf of a chat. For example, the supergroup itself for messages sent by its anonymous administrators or a linked channel for messages automatically forwarded to the channel's discussion group. For backward compatibility, if the message was sent on behalf of a chat, the field <i>from</i> contains a fake sender user in non-channel chats.
sender_boost_count	Integer	<i>Optional.</i> If the sender of the message boosted the chat, the number of boosts added by the user
sender_business_bot	User	<i>Optional.</i> The bot that actually sent the message on behalf of the business account. Available only for outgoing messages sent on behalf of the connected business account.
date	Integer	Date the message was sent in Unix time. It is always a positive number, representing a valid date.
business_connection_id	String	<i>Optional.</i> Unique identifier of the business connection from which the message was received. If non-empty, the message belongs to a chat of the corresponding business account that is independent from any potential bot chat which might share the same identifier.
chat	Chat	Chat the message belongs to
forward_origin	MessageOrigin	<i>Optional.</i> Information about the original message for forwarded messages
is_topic_message	True	<i>Optional.</i> True, if the message is sent to a forum topic
is_automatic_forward	True	<i>Optional.</i> True, if the message is a channel post that was automatically forwarded to the connected discussion group
reply_to_message	Message	<i>Optional.</i> For replies in the same chat and message thread, the original message. Note that the Message object in this field will not contain further <i>reply_to_message</i> fields even if it itself is a reply.
external_reply	ExternalReplyInfo	<i>Optional.</i> Information about the message that is being replied to, which may come from another chat or forum topic
quote	TextQuote	<i>Optional.</i> For replies that quote part of the original message, the quoted part of the message
reply_to_story	Story	<i>Optional.</i> For replies to a story, the original story
reply_to_checklist_task_id	Integer	<i>Optional.</i> Identifier of the specific checklist task that is being replied to
via_bot	User	<i>Optional.</i> Bot through which the message was sent
edit_date	Integer	<i>Optional.</i> Date the message was last edited in Unix time
has_protected_content	True	<i>Optional.</i> True, if the message can't be forwarded
is_from_offline	True	<i>Optional.</i> True, if the message was sent by an implicit action, for example, as an away or a greeting business message, or as a scheduled message
is_paid_post	True	<i>Optional.</i> True, if the message is a paid post. Note that such posts must not be deleted for 24 hours to receive the payment and can't be edited.
media_group_id	String	<i>Optional.</i> The unique identifier of a media message group this message belongs to

Field	Type	Description
author_signature	String	<i>Optional.</i> Signature of the post author for messages in channels, or the custom title of an anonymous group administrator
paid_star_count	Integer	<i>Optional.</i> The number of Telegram Stars that were paid by the sender of the message to send it
text	String	<i>Optional.</i> For text messages, the actual UTF-8 text of the message
entities	Array of MessageEntity	<i>Optional.</i> For text messages, special entities like usernames, URLs, bot commands, etc. that appear in the text
link_preview_options	LinkPreviewOptions	<i>Optional.</i> Options used for link preview generation for the message, if it is a text message and link preview options were changed
suggested_post_info	SuggestedPostInfo	<i>Optional.</i> Information about suggested post parameters if the message is a suggested post in a channel direct messages chat. If the message is an approved or declined suggested post, then it can't be edited.
effect_id	String	<i>Optional.</i> Unique identifier of the message effect added to the message
animation	Animation	<i>Optional.</i> Message is an animation, information about the animation. For backward compatibility, when this field is set, the <code>document</code> field will also be set
audio	Audio	<i>Optional.</i> Message is an audio file, information about the file
document	Document	<i>Optional.</i> Message is a general file, information about the file
paid_media	PaidMediaInfo	<i>Optional.</i> Message contains paid media; information about the paid media
photo	Array of PhotoSize	<i>Optional.</i> Message is a photo, available sizes of the photo
sticker	Sticker	<i>Optional.</i> Message is a sticker, information about the sticker
story	Story	<i>Optional.</i> Message is a forwarded story
video	Video	<i>Optional.</i> Message is a video, information about the video
video_note	VideoNote	<i>Optional.</i> Message is a video note , information about the video message
voice	Voice	<i>Optional.</i> Message is a voice message, information about the file
caption	String	<i>Optional.</i> Caption for the animation, audio, document, paid media, photo, video or voice
caption_entities	Array of MessageEntity	<i>Optional.</i> For messages with a caption, special entities like usernames, URLs, bot commands, etc. that appear in the caption
show_caption_above_media	True	<i>Optional.</i> <code>True</code> , if the caption must be shown above the message media
has_mediaSpoiler	True	<i>Optional.</i> <code>True</code> , if the message media is covered by a spoiler animation
checklist	Checklist	<i>Optional.</i> Message is a checklist
contact	Contact	<i>Optional.</i> Message is a shared contact, information about the contact
dice	Dice	<i>Optional.</i> Message is a dice with random value
game	Game	<i>Optional.</i> Message is a game, information about the game. More about games »
poll	Poll	<i>Optional.</i> Message is a native poll, information about the poll
venue	Venue	<i>Optional.</i> Message is a venue, information about the venue. For backward compatibility, when this field is set, the <code>location</code> field will also be set
location	Location	<i>Optional.</i> Message is a shared location, information about the location
new_chat_members	Array of User	<i>Optional.</i> New members that were added to the group or supergroup and information about them (the bot itself may be one of these members)
left_chat_member	User	<i>Optional.</i> A member was removed from the group, information about them (this member may be the bot itself)
new_chat_title	String	<i>Optional.</i> A chat title was changed to this value
new_chat_photo	Array of PhotoSize	<i>Optional.</i> A chat photo was changed to this value
delete_chat_photo	True	<i>Optional.</i> Service message: the chat photo was deleted
group_chat_created	True	<i>Optional.</i> Service message: the group has been created
supergroup_chat_created	True	<i>Optional.</i> Service message: the supergroup has been created. This field can't be received in a message coming through updates, because bot can't be a member of a supergroup when it is created. It can only be found in <code>reply_to_message</code> if someone replies to a very first message in a directly created supergroup.
channel_chat_created	True	<i>Optional.</i> Service message: the channel has been created. This field can't be received in a message coming through updates, because bot can't be a member of a channel when it is created. It can only be found in <code>reply_to_message</code> if someone replies to a very first message in a channel.
message_auto_delete_timer_changed	MessageAutoDeleteTimerChanged	<i>Optional.</i> Service message: auto-delete timer settings changed in the chat

Field	Type	Description
migrate_to_chat_id	Integer	<i>Optional.</i> The group has been migrated to a supergroup with the specified identifier. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.
migrate_from_chat_id	Integer	<i>Optional.</i> The supergroup has been migrated from a group with the specified identifier. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.
pinned_message	MaybeInaccessibleMessage	<i>Optional.</i> Specified message was pinned. Note that the Message object in this field will not contain further <code>reply_to_message</code> fields even if it itself is a reply.
invoice	Invoice	<i>Optional.</i> Message is an invoice for a payment , information about the invoice. More about payments »
successful_payment	SuccessfulPayment	<i>Optional.</i> Message is a service message about a successful payment, information about the payment. More about payments »
refunded_payment	RefundedPayment	<i>Optional.</i> Message is a service message about a refunded payment, information about the payment. More about payments »
users_shared	UsersShared	<i>Optional.</i> Service message: users were shared with the bot
chat_shared	ChatShared	<i>Optional.</i> Service message: a chat was shared with the bot
gift	GiftInfo	<i>Optional.</i> Service message: a regular gift was sent or received
unique_gift	UniqueGiftInfo	<i>Optional.</i> Service message: a unique gift was sent or received
connected_website	String	<i>Optional.</i> The domain name of the website on which the user has logged in. More about Telegram Login »
write_access_allowed	WriteAccessAllowed	<i>Optional.</i> Service message: the user allowed the bot to write messages after adding it to the attachment or side menu, launching a Web App from a link, or accepting an explicit request from a Web App sent by the method <code>requestWriteAccess</code>
passport_data	PassportData	<i>Optional.</i> Telegram Passport data
proximity_alert_triggered	ProximityAlertTriggered	<i>Optional.</i> Service message. A user in the chat triggered another user's proximity alert while sharing Live Location.
boost_added	ChatBoostAdded	<i>Optional.</i> Service message: user boosted the chat
chat_background_set	ChatBackground	<i>Optional.</i> Service message: chat background set
checklist_tasks_done	ChecklistTasksDone	<i>Optional.</i> Service message: some tasks in a checklist were marked as done or not done
checklist_tasks_added	ChecklistTasksAdded	<i>Optional.</i> Service message: tasks were added to a checklist
direct_message_price_changed	DirectMessagePriceChanged	<i>Optional.</i> Service message: the price for paid messages in the corresponding direct messages chat of a channel has changed
forum_topic_created	ForumTopicCreated	<i>Optional.</i> Service message: forum topic created
forum_topic_edited	ForumTopicEdited	<i>Optional.</i> Service message: forum topic edited
forum_topic_closed	ForumTopicClosed	<i>Optional.</i> Service message: forum topic closed
forum_topic_reopened	ForumTopicReopened	<i>Optional.</i> Service message: forum topic reopened
general_forum_topic_hidden	GeneralForumTopicHidden	<i>Optional.</i> Service message: the 'General' forum topic hidden
general_forum_topic_unhidden	GeneralForumTopicUnhidden	<i>Optional.</i> Service message: the 'General' forum topic unhidden
giveaway_created	GiveawayCreated	<i>Optional.</i> Service message: a scheduled giveaway was created
giveaway	Giveaway	<i>Optional.</i> The message is a scheduled giveaway message
giveaway_winners	GiveawayWinners	<i>Optional.</i> A giveaway with public winners was completed
giveaway_completed	GiveawayCompleted	<i>Optional.</i> Service message: a giveaway without public winners was completed
paid_message_price_changed	PaidMessagePriceChanged	<i>Optional.</i> Service message: the price for paid messages has changed in the chat
suggested_post_approved	SuggestedPostApproved	<i>Optional.</i> Service message: a suggested post was approved
suggested_post_approval_failed	SuggestedPostApprovalFailed	<i>Optional.</i> Service message: approval of a suggested post has failed
suggested_post_declined	SuggestedPostDeclined	<i>Optional.</i> Service message: a suggested post was declined
suggested_post_paid	SuggestedPostPaid	<i>Optional.</i> Service message: payment for a suggested post was received
suggested_post_refunded	SuggestedPostRefunded	<i>Optional.</i> Service message: payment for a suggested post was refunded
video_chat_scheduled	VideoChatScheduled	<i>Optional.</i> Service message: video chat scheduled
video_chat_started	VideoChatStarted	<i>Optional.</i> Service message: video chat started

Field	Type	Description
video_chat_ended	VideoChatEnded	<i>Optional.</i> Service message: video chat ended
video_chat_participants_invited	VideoChatParticipantsInvited	<i>Optional.</i> Service message: new participants invited to a video chat
web_app_data	WebAppData	<i>Optional.</i> Service message: data sent by a Web App
reply_markup	InlineKeyboardMarkup	<i>Optional.</i> Inline keyboard attached to the message. <code>login_url</code> buttons are represented as ordinary <code>url</code> buttons.

Messageld

This object represents a unique message identifier.

Field	Type	Description
message_id	Integer	Unique message identifier. In specific instances (e.g., message containing a video sent to a big chat), the server might automatically schedule a message instead of sending it immediately. In such cases, this field will be 0 and the relevant message will be unusable until it is actually sent

InaccessibleMessage

This object describes a message that was deleted or is otherwise inaccessible to the bot.

Field	Type	Description
chat	Chat	Chat the message belonged to
message_id	Integer	Unique message identifier inside the chat
date	Integer	Always 0. The field can be used to differentiate regular and inaccessible messages.

MaybeInaccessibleMessage

This object describes a message that can be inaccessible to the bot. It can be one of

- [Message](#)
- [InaccessibleMessage](#)

MessageEntity

This object represents one special entity in a text message. For example, hashtags, usernames, URLs, etc.

Field	Type	Description
type	String	Type of the entity. Currently, can be "mention" (<code>@username</code>), "hashtag" (<code>#hashtag</code> or <code>#hashtag@chatusername</code>), "cashtag" (<code>\$USD</code> or <code>\$USD@chatusername</code>), "bot_command" (<code>/start@jobs_bot</code>), "url" (https://telegram.org), "email" (<code>do-not-reply@telegram.org</code>), "phone_number" (<code>+1-212-555-0123</code>), "bold" (<code>bold text</code>), "italic" (<code>italic text</code>), "underline" (underlined text), "strikethrough" (strikethrough text), "spoiler" (spoiler message), "blockquote" (block quotation), "expandable_blockquote" (collapsed-by-default block quotation), "code" (monowidth string), "pre" (monowidth block), "text_link" (for clickable text URLs), "text_mention" (for users without usernames), "custom_emoji" (for inline custom emoji stickers)
offset	Integer	Offset in UTF-16 code units to the start of the entity
length	Integer	Length of the entity in UTF-16 code units
url	String	<i>Optional.</i> For "text_link" only, URL that will be opened after user taps on the text
user	User	<i>Optional.</i> For "text_mention" only, the mentioned user
language	String	<i>Optional.</i> For "pre" only, the programming language of the entity text
custom_emoji_id	String	<i>Optional.</i> For "custom_emoji" only, unique identifier of the custom emoji. Use <code>getCustomEmojiStickers</code> to get full information about the sticker

TextQuote

This object contains information about the quoted part of a message that is replied to by the given message.

Field	Type	Description
text	String	Text of the quoted part of a message that is replied to by the given message
entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the quote. Currently, only <code>bold</code> , <code>italic</code> , <code>underline</code> , <code>strikethrough</code> , <code>spoiler</code> , and <code>custom_emoji</code> entities are kept in quotes.
position	Integer	Approximate quote position in the original message in UTF-16 code units as specified by the sender
is_manual	True	<i>Optional.</i> <code>True</code> , if the quote was chosen manually by the message sender. Otherwise, the quote was added automatically by the server.

ExternalReplyInfo

This object contains information about a message that is being replied to, which may come from another chat or forum topic.

Field	Type	Description
origin	MessageOrigin	Origin of the message replied to by the given message
chat	Chat	<i>Optional.</i> Chat the original message belongs to. Available only if the chat is a supergroup or a channel.
message_id	Integer	<i>Optional.</i> Unique message identifier inside the original chat. Available only if the original chat is a supergroup or a channel.
link_preview_options	LinkPreviewOptions	<i>Optional.</i> Options used for link preview generation for the original message, if it is a text message

Field	Type	Description
animation	Animation	<i>Optional.</i> Message is an animation, information about the animation
audio	Audio	<i>Optional.</i> Message is an audio file, information about the file
document	Document	<i>Optional.</i> Message is a general file, information about the file
paid_media	PaidMediaInfo	<i>Optional.</i> Message contains paid media; information about the paid media
photo	Array of PhotoSize	<i>Optional.</i> Message is a photo, available sizes of the photo
sticker	Sticker	<i>Optional.</i> Message is a sticker, information about the sticker
story	Story	<i>Optional.</i> Message is a forwarded story
video	Video	<i>Optional.</i> Message is a video, information about the video
video_note	VideoNote	<i>Optional.</i> Message is a video note , information about the video message
voice	Voice	<i>Optional.</i> Message is a voice message, information about the file
has_media_spoiler	True	<i>Optional.</i> <code>True</code> , if the message media is covered by a spoiler animation
checklist	Checklist	<i>Optional.</i> Message is a checklist
contact	Contact	<i>Optional.</i> Message is a shared contact, information about the contact
dice	Dice	<i>Optional.</i> Message is a dice with random value
game	Game	<i>Optional.</i> Message is a game, information about the game. More about games »
giveaway	Giveaway	<i>Optional.</i> Message is a scheduled giveaway, information about the giveaway
giveaway_winners	GiveawayWinners	<i>Optional.</i> A giveaway with public winners was completed
invoice	Invoice	<i>Optional.</i> Message is an invoice for a payment , information about the invoice. More about payments »
location	Location	<i>Optional.</i> Message is a shared location, information about the location
poll	Poll	<i>Optional.</i> Message is a native poll, information about the poll
venue	Venue	<i>Optional.</i> Message is a venue, information about the venue

ReplyParameters

Describes reply parameters for the message that is being sent.

Field	Type	Description
message_id	Integer	Identifier of the message that will be replied to in the current chat, or in the chat <code>chat_id</code> if it is specified
chat_id	Integer or String	<i>Optional.</i> If the message to be replied to is from a different chat, unique identifier for the chat or username of the channel (in the format <code>@channelusername</code>). Not supported for messages sent on behalf of a business account and messages from channel direct messages chats.
allow_sending_without_reply	Boolean	<i>Optional.</i> Pass <code>True</code> if the message should be sent even if the specified message to be replied to is not found. Always <code>False</code> for replies in another chat or forum topic. Always <code>True</code> for messages sent on behalf of a business account.
quote	String	<i>Optional.</i> Quoted part of the message to be replied to; 0–1024 characters after entities parsing. The quote must be an exact substring of the message to be replied to, including <code>bold</code> , <code>italic</code> , <code>underline</code> , <code>strikethrough</code> , <code>spoiler</code> , and <code>custom_emoji</code> entities. The message will fail to send if the quote isn't found in the original message.
quote_parse_mode	String	<i>Optional.</i> Mode for parsing entities in the quote. See formatting options for more details.
quote_entities	Array of MessageEntity	<i>Optional.</i> A JSON-serialized list of special entities that appear in the quote. It can be specified instead of <code>quote_parse_mode</code> .
quote_position	Integer	<i>Optional.</i> Position of the quote in the original message in UTF-16 code units
checklist_task_id	Integer	<i>Optional.</i> Identifier of the specific checklist task to be replied to

MessageOrigin

This object describes the origin of a message. It can be one of

- [MessageOriginUser](#)
- [MessageOriginHiddenUser](#)
- [MessageOriginChat](#)
- [MessageOriginChannel](#)

MessageOriginUser

The message was originally sent by a known user.

Field	Type	Description
type	String	Type of the message origin, always "user"
date	Integer	Date the message was sent originally in Unix time
sender_user	User	User that sent the message originally

MessageOriginHiddenUser

The message was originally sent by an unknown user.

Field	Type	Description
type	String	Type of the message origin, always "hidden_user"
date	Integer	Date the message was sent originally in Unix time
sender_user_name	String	Name of the user that sent the message originally

MessageOriginChat

The message was originally sent on behalf of a chat to a group chat.

Field	Type	Description
type	String	Type of the message origin, always "chat"
date	Integer	Date the message was sent originally in Unix time
sender_chat	Chat	Chat that sent the message originally
author_signature	String	<i>Optional.</i> For messages originally sent by an anonymous chat administrator, original message author signature

MessageOriginChannel

The message was originally sent to a channel chat.

Field	Type	Description
type	String	Type of the message origin, always "channel"
date	Integer	Date the message was sent originally in Unix time
chat	Chat	Channel chat to which the message was originally sent
message_id	Integer	Unique message identifier inside the chat
author_signature	String	<i>Optional.</i> Signature of the original post author

PhotoSize

This object represents one size of a photo or a file / sticker thumbnail.

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
width	Integer	Photo width
height	Integer	Photo height
file_size	Integer	<i>Optional.</i> File size in bytes

Animation

This object represents an animation file (GIF or H.264/MPEG-4 AVC video without sound).

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
width	Integer	Video width as defined by the sender
height	Integer	Video height as defined by the sender
duration	Integer	Duration of the video in seconds as defined by the sender
thumbnail	PhotoSize	<i>Optional.</i> Animation thumbnail as defined by the sender
file_name	String	<i>Optional.</i> Original animation filename as defined by the sender
mime_type	String	<i>Optional.</i> MIME type of the file as defined by the sender
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.

Audio

This object represents an audio file to be treated as music by the Telegram clients.

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.

Field	Type	Description
duration	Integer	Duration of the audio in seconds as defined by the sender
performer	String	<i>Optional.</i> Performer of the audio as defined by the sender or by audio tags
title	String	<i>Optional.</i> Title of the audio as defined by the sender or by audio tags
file_name	String	<i>Optional.</i> Original filename as defined by the sender
mime_type	String	<i>Optional.</i> MIME type of the file as defined by the sender
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.
thumbnail	PhotoSize	<i>Optional.</i> Thumbnail of the album cover to which the music file belongs

Document

This object represents a general file (as opposed to [photos](#), [voice messages](#) and [audio files](#)).

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
thumbnail	PhotoSize	<i>Optional.</i> Document thumbnail as defined by the sender
file_name	String	<i>Optional.</i> Original filename as defined by the sender
mime_type	String	<i>Optional.</i> MIME type of the file as defined by the sender
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.

Story

This object represents a story.

Field	Type	Description
chat	Chat	Chat that posted the story
id	Integer	Unique identifier for the story in the chat

Video

This object represents a video file.

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
width	Integer	Video width as defined by the sender
height	Integer	Video height as defined by the sender
duration	Integer	Duration of the video in seconds as defined by the sender
thumbnail	PhotoSize	<i>Optional.</i> Video thumbnail
cover	Array of PhotoSize	<i>Optional.</i> Available sizes of the cover of the video in the message
start_timestamp	Integer	<i>Optional.</i> Timestamp in seconds from which the video will play in the message
file_name	String	<i>Optional.</i> Original filename as defined by the sender
mime_type	String	<i>Optional.</i> MIME type of the file as defined by the sender
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.

VideoNote

This object represents a [video message](#) (available in Telegram apps as of v.4.0).

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
length	Integer	Video width and height (diameter of the video message) as defined by the sender
duration	Integer	Duration of the video in seconds as defined by the sender
thumbnail	PhotoSize	<i>Optional.</i> Video thumbnail

Field	Type	Description
file_size	Integer	<i>Optional.</i> File size in bytes

Voice

This object represents a voice note.

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
duration	Integer	Duration of the audio in seconds as defined by the sender
mime_type	String	<i>Optional.</i> MIME type of the file as defined by the sender
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.

PaidMediaInfo

Describes the paid media added to a message.

Field	Type	Description
star_count	Integer	The number of Telegram Stars that must be paid to buy access to the media
paid_media	Array of PaidMedia	Information about the paid media

PaidMedia

This object describes paid media. Currently, it can be one of

- [PaidMediaPreview](#)
- [PaidMediaPhoto](#)
- [PaidMediaVideo](#)

PaidMediaPreview

The paid media isn't available before the payment.

Field	Type	Description
type	String	Type of the paid media, always "preview"
width	Integer	<i>Optional.</i> Media width as defined by the sender
height	Integer	<i>Optional.</i> Media height as defined by the sender
duration	Integer	<i>Optional.</i> Duration of the media in seconds as defined by the sender

PaidMediaPhoto

The paid media is a photo.

Field	Type	Description
type	String	Type of the paid media, always "photo"
photo	Array of PhotoSize	The photo

PaidMediaVideo

The paid media is a video.

Field	Type	Description
type	String	Type of the paid media, always "video"
video	Video	The video

Contact

This object represents a phone contact.

Field	Type	Description
phone_number	String	Contact's phone number
first_name	String	Contact's first name
last_name	String	<i>Optional.</i> Contact's last name
user_id	Integer	<i>Optional.</i> Contact's user identifier in Telegram. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.
vcard	String	<i>Optional.</i> Additional data about the contact in the form of a vCard

Dice

This object represents an animated emoji that displays a random value.

Field	Type	Description
emoji	String	Emoji on which the dice throw animation is based
value	Integer	Value of the dice, 1–6 for “🎲”, “🎯” and “🎳” base emoji, 1–5 for “🏀” and “🎳” base emoji, 1–64 for “🎰” base emoji

PollOption

This object contains information about one answer option in a poll.

Field	Type	Description
text	String	Option text, 1–100 characters
text_entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the option <code>text</code> . Currently, only custom emoji entities are allowed in poll option texts
voter_count	Integer	Number of users that voted for this option

InputPollOption

This object contains information about one answer option in a poll to be sent.

Field	Type	Description
text	String	Option text, 1–100 characters
text_parse_mode	String	<i>Optional.</i> Mode for parsing entities in the text. See formatting options for more details. Currently, only custom emoji entities are allowed
text_entities	Array of MessageEntity	<i>Optional.</i> A JSON-serialized list of special entities that appear in the poll option text. It can be specified instead of <code>text_parse_mode</code>

PollAnswer

This object represents an answer of a user in a non-anonymous poll.

Field	Type	Description
poll_id	String	Unique poll identifier
voter_chat	Chat	<i>Optional.</i> The chat that changed the answer to the poll, if the voter is anonymous
user	User	<i>Optional.</i> The user that changed the answer to the poll, if the voter isn't anonymous
option_ids	Array of Integer	0-based identifiers of chosen answer options. May be empty if the vote was retracted.

Poll

This object contains information about a poll.

Field	Type	Description
id	String	Unique poll identifier
question	String	Poll question, 1–300 characters
question_entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the <code>question</code> . Currently, only custom emoji entities are allowed in poll questions
options	Array of PollOption	List of poll options
total_voter_count	Integer	Total number of users that voted in the poll
is_closed	Boolean	<i>True</i> , if the poll is closed
is_anonymous	Boolean	<i>True</i> , if the poll is anonymous
type	String	Poll type, currently can be “regular” or “quiz”
allows_multiple_answers	Boolean	<i>True</i> , if the poll allows multiple answers
correct_option_id	Integer	<i>Optional.</i> 0-based identifier of the correct answer option. Available only for polls in the quiz mode, which are closed, or was sent (not forwarded) by the bot or to the private chat with the bot.
explanation	String	<i>Optional.</i> Text that is shown when a user chooses an incorrect answer or taps on the lamp icon in a quiz-style poll, 0–200 characters
explanation_entities	Array of MessageEntity	<i>Optional.</i> Special entities like usernames, URLs, bot commands, etc. that appear in the <code>explanation</code>
open_period	Integer	<i>Optional.</i> Amount of time in seconds the poll will be active after creation
close_date	Integer	<i>Optional.</i> Point in time (Unix timestamp) when the poll will be automatically closed

ChecklistTask

Describes a task in a checklist.

Field	Type	Description
id	Integer	Unique identifier of the task
text	String	Text of the task
text_entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the task text
completed_by_user	User	<i>Optional.</i> User that completed the task; omitted if the task wasn't completed
completion_date	Integer	<i>Optional.</i> Point in time (Unix timestamp) when the task was completed; 0 if the task wasn't completed

Checklist

Describes a checklist.

Field	Type	Description
title	String	Title of the checklist
title_entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the checklist title
tasks	Array of ChecklistTask	List of tasks in the checklist
others_can_add_tasks	True	<i>Optional.</i> True, if users other than the creator of the list can add tasks to the list
others_can_mark_tasks_as_done	True	<i>Optional.</i> True, if users other than the creator of the list can mark tasks as done or not done

InputChecklistTask

Describes a task to add to a checklist.

Field	Type	Description
id	Integer	Unique identifier of the task; must be positive and unique among all task identifiers currently present in the checklist
text	String	Text of the task; 1–100 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the text. See formatting options for more details.
text_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the text, which can be specified instead of parse_mode. Currently, only <i>bold</i> , <i>italic</i> , <i>underline</i> , <i>strikethrough</i> , <i>spoiler</i> , and <i>custom_emoji</i> entities are allowed.

InputChecklist

Describes a checklist to create.

Field	Type	Description
title	String	Title of the checklist; 1–255 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the title. See formatting options for more details.
title_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the title, which can be specified instead of parse_mode. Currently, only <i>bold</i> , <i>italic</i> , <i>underline</i> , <i>strikethrough</i> , <i>spoiler</i> , and <i>custom_emoji</i> entities are allowed.
tasks	Array of InputChecklistTask	List of 1–30 tasks in the checklist
others_can_add_tasks	Boolean	<i>Optional.</i> Pass True if other users can add tasks to the checklist
others_can_mark_tasks_as_done	Boolean	<i>Optional.</i> Pass True if other users can mark tasks as done or not done in the checklist

ChecklistTasksDone

Describes a service message about checklist tasks marked as done or not done.

Field	Type	Description
checklist_message	Message	<i>Optional.</i> Message containing the checklist whose tasks were marked as done or not done. Note that the Message object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
marked_as_done_task_ids	Array of Integer	<i>Optional.</i> Identifiers of the tasks that were marked as done
marked_as_not_done_task_ids	Array of Integer	<i>Optional.</i> Identifiers of the tasks that were marked as not done

ChecklistTasksAdded

Describes a service message about tasks added to a checklist.

Field	Type	Description
checklist_message	Message	<i>Optional.</i> Message containing the checklist to which the tasks were added. Note that the Message object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
tasks	Array of ChecklistTask	List of tasks added to the checklist

Location

This object represents a point on the map.

Field	Type	Description
latitude	Float	Latitude as defined by the sender
longitude	Float	Longitude as defined by the sender
horizontal_accuracy	Float	<i>Optional.</i> The radius of uncertainty for the location, measured in meters; 0–1500
live_period	Integer	<i>Optional.</i> Time relative to the message sending date, during which the location can be updated; in seconds. For active live locations only.
heading	Integer	<i>Optional.</i> The direction in which user is moving, in degrees; 1–360. For active live locations only.
proximity_alert_radius	Integer	<i>Optional.</i> The maximum distance for proximity alerts about approaching another chat member, in meters. For sent live locations only.

Venue

This object represents a venue.

Field	Type	Description
location	Location	Venue location. Can't be a live location
title	String	Name of the venue
address	String	Address of the venue
foursquare_id	String	<i>Optional.</i> Foursquare identifier of the venue
foursquare_type	String	<i>Optional.</i> Foursquare type of the venue. (For example, "arts_entertainment/default", "arts_entertainment/aquarium" or "food/icecream".)
google_place_id	String	<i>Optional.</i> Google Places identifier of the venue
google_place_type	String	<i>Optional.</i> Google Places type of the venue. (See supported types .)

WebAppData

Describes data sent from a [Web App](#) to the bot.

Field	Type	Description
data	String	The data. Be aware that a bad client can send arbitrary data in this field.
button_text	String	Text of the <code>web_app</code> keyboard button from which the Web App was opened. Be aware that a bad client can send arbitrary data in this field.

ProximityAlertTriggered

This object represents the content of a service message, sent whenever a user in the chat triggers a proximity alert set by another user.

Field	Type	Description
traveler	User	User that triggered the alert
watcher	User	User that set the alert
distance	Integer	The distance between the users

MessageAutoDeleteTimerChanged

This object represents a service message about a change in auto-delete timer settings.

Field	Type	Description
message_auto_delete_time	Integer	New auto-delete time for messages in the chat; in seconds

ChatBoostAdded

This object represents a service message about a user boosting a chat.

Field	Type	Description
boost_count	Integer	Number of boosts added by the user

BackgroundFill

This object describes the way a background is filled based on the selected colors. Currently, it can be one of

- `BackgroundFillSolid`
- `BackgroundFillGradient`
- `BackgroundFillFreeformGradient`

BackgroundFillSolid

The background is filled using the selected color.

Field	Type	Description
type	String	Type of the background fill, always "solid"
color	Integer	The color of the background fill in the RGB24 format

BackgroundFillGradient

The background is a gradient fill.

Field	Type	Description
type	String	Type of the background fill, always "gradient"
top_color	Integer	Top color of the gradient in the RGB24 format
bottom_color	Integer	Bottom color of the gradient in the RGB24 format
rotation_angle	Integer	Clockwise rotation angle of the background fill in degrees; 0–359

BackgroundFillFreeformGradient

The background is a freeform gradient that rotates after every message in the chat.

Field	Type	Description
type	String	Type of the background fill, always "freeform_gradient"
colors	Array of Integer	A list of the 3 or 4 base colors that are used to generate the freeform gradient in the RGB24 format

BackgroundType

This object describes the type of a background. Currently, it can be one of

- [BackgroundTypeFill](#)
- [BackgroundTypeWallpaper](#)
- [BackgroundTypePattern](#)
- [BackgroundTypeChatTheme](#)

BackgroundTypeFill

The background is automatically filled based on the selected colors.

Field	Type	Description
type	String	Type of the background, always "fill"
fill	BackgroundFill	The background fill
dark_theme_dimming	Integer	Dimming of the background in dark themes, as a percentage; 0–100

BackgroundTypeWallpaper

The background is a wallpaper in the JPEG format.

Field	Type	Description
type	String	Type of the background, always "wallpaper"
document	Document	Document with the wallpaper
dark_theme_dimming	Integer	Dimming of the background in dark themes, as a percentage; 0–100
is_blurred	True	<i>Optional.</i> True, if the wallpaper is downscaled to fit in a 450x450 square and then box-blurred with radius 12
is_moving	True	<i>Optional.</i> True, if the background moves slightly when the device is tilted

BackgroundTypePattern

The background is a .PNG or .TGV (gzipped subset of SVG with MIME type "application/x-tgwallpattern") pattern to be combined with the background fill chosen by the user.

Field	Type	Description
type	String	Type of the background, always "pattern"
document	Document	Document with the pattern
fill	BackgroundFill	The background fill that is combined with the pattern
intensity	Integer	Intensity of the pattern when it is shown above the filled background; 0–100
is_inverted	True	<i>Optional.</i> True, if the background fill must be applied only to the pattern itself. All other pixels are black in this case. For dark themes only
is_moving	True	<i>Optional.</i> True, if the background moves slightly when the device is tilted

BackgroundTypeChatTheme

The background is taken directly from a built-in chat theme.

Field	Type	Description
type	String	Type of the background, always "chat_theme"
theme_name	String	Name of the chat theme, which is usually an emoji

ChatBackground

This object represents a chat background.

Field	Type	Description
type	BackgroundType	Type of the background

ForumTopicCreated

This object represents a service message about a new forum topic created in the chat.

Field	Type	Description
name	String	Name of the topic
icon_color	Integer	Color of the topic icon in RGB format
icon_custom_emoji_id	String	<i>Optional.</i> Unique identifier of the custom emoji shown as the topic icon

ForumTopicClosed

This object represents a service message about a forum topic closed in the chat. Currently holds no information.

ForumTopicEdited

This object represents a service message about an edited forum topic.

Field	Type	Description
name	String	<i>Optional.</i> New name of the topic, if it was edited
icon_custom_emoji_id	String	<i>Optional.</i> New identifier of the custom emoji shown as the topic icon, if it was edited; an empty string if the icon was removed

ForumTopicReopened

This object represents a service message about a forum topic reopened in the chat. Currently holds no information.

GeneralForumTopicHidden

This object represents a service message about General forum topic hidden in the chat. Currently holds no information.

GeneralForumTopicUnhidden

This object represents a service message about General forum topic unhidden in the chat. Currently holds no information.

SharedUser

This object contains information about a user that was shared with the bot using a [KeyboardButtonRequestUsers](#) button.

Field	Type	Description
user_id	Integer	Identifier of the shared user. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so 64-bit integers or double-precision float types are safe for storing these identifiers. The bot may not have access to the user and could be unable to use this identifier, unless the user is already known to the bot by some other means.
first_name	String	<i>Optional.</i> First name of the user, if the name was requested by the bot
last_name	String	<i>Optional.</i> Last name of the user, if the name was requested by the bot
username	String	<i>Optional.</i> Username of the user, if the username was requested by the bot
photo	Array of PhotoSize	<i>Optional.</i> Available sizes of the chat photo, if the photo was requested by the bot

UsersShared

This object contains information about the users whose identifiers were shared with the bot using a [KeyboardButtonRequestUsers](#) button.

Field	Type	Description
request_id	Integer	Identifier of the request
users	Array of SharedUser	Information about users shared with the bot.

ChatShared

This object contains information about a chat that was shared with the bot using a [KeyboardButtonRequestChat](#) button.

Field	Type	Description
request_id	Integer	Identifier of the request
chat_id	Integer	Identifier of the shared chat. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a 64-bit integer or double-precision float type are safe for storing this identifier. The bot may not have access to the chat and could be unable to use this identifier, unless the chat is already known to the bot by some other means.
title	String	<i>Optional.</i> Title of the chat, if the title was requested by the bot.
username	String	<i>Optional.</i> Username of the chat, if the username was requested by the bot and available.
photo	Array of PhotoSize	<i>Optional.</i> Available sizes of the chat photo, if the photo was requested by the bot

WriteAccessAllowed

This object represents a service message about a user allowing a bot to write messages after adding it to the attachment menu, launching a Web App from a link, or accepting an explicit request from a Web App sent by the method [requestWriteAccess](#).

Field	Type	Description
from_request	Boolean	<i>Optional.</i> <code>True</code> , if the access was granted after the user accepted an explicit request from a Web App sent by the method requestWriteAccess
web_app_name	String	<i>Optional.</i> Name of the Web App, if the access was granted when the Web App was launched from a link
from_attachment_menu	Boolean	<i>Optional.</i> <code>True</code> , if the access was granted when the bot was added to the attachment or side menu

VideoChatScheduled

This object represents a service message about a video chat scheduled in the chat.

Field	Type	Description
start_date	Integer	Point in time (Unix timestamp) when the video chat is supposed to be started by a chat administrator

VideoChatStarted

This object represents a service message about a video chat started in the chat. Currently holds no information.

VideoChatEnded

This object represents a service message about a video chat ended in the chat.

Field	Type	Description
duration	Integer	Video chat duration in seconds

VideoChatParticipantsInvited

This object represents a service message about new members invited to a video chat.

Field	Type	Description
users	Array of User	New members that were invited to the video chat

PaidMessagePriceChanged

Describes a service message about a change in the price of paid messages within a chat.

Field	Type	Description
paid_message_star_count	Integer	The new number of Telegram Stars that must be paid by non-administrator users of the supergroup chat for each sent message

DirectMessagePriceChanged

Describes a service message about a change in the price of direct messages sent to a channel chat.

Field	Type	Description
are_direct_messages_enabled	Boolean	<code>True</code> , if direct messages are enabled for the channel chat; false otherwise
direct_message_star_count	Integer	<i>Optional.</i> The new number of Telegram Stars that must be paid by users for each direct message sent to the channel. Does not apply to users who have been exempted by administrators. Defaults to 0.

SuggestedPostApproved

Describes a service message about the approval of a suggested post.

Field	Type	Description
suggested_post_message	Message	<i>Optional.</i> Message containing the suggested post. Note that the Message object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
price	SuggestedPostPrice	<i>Optional.</i> Amount paid for the post
send_date	Integer	Date when the post will be published

SuggestedPostApprovalFailed

Describes a service message about the failed approval of a suggested post. Currently, only caused by insufficient user funds at the time of approval.

Field	Type	Description
suggested_post_message	Message	<i>Optional.</i> Message containing the suggested post whose approval has failed. Note that the Message object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
price	SuggestedPostPrice	Expected price of the post

SuggestedPostDeclined

Describes a service message about the rejection of a suggested post.

Field	Type	Description
suggested_post_message	Message	<i>Optional.</i> Message containing the suggested post. Note that the Message object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
comment	String	<i>Optional.</i> Comment with which the post was declined

SuggestedPostPaid

Describes a service message about a successful payment for a suggested post.

Field	Type	Description
suggested_post_message	Message	<i>Optional.</i> Message containing the suggested post. Note that the <code>Message</code> object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
currency	String	Currency in which the payment was made. Currently, one of "XTR" for Telegram Stars or "TON" for toncoins
amount	Integer	<i>Optional.</i> The amount of the currency that was received by the channel in nanotoncoins; for payments in toncoins only
star_amount	StarAmount	<i>Optional.</i> The amount of Telegram Stars that was received by the channel; for payments in Telegram Stars only

SuggestedPostRefunded

Describes a service message about a payment refund for a suggested post.

Field	Type	Description
suggested_post_message	Message	<i>Optional.</i> Message containing the suggested post. Note that the <code>Message</code> object in this field will not contain the <code>reply_to_message</code> field even if it itself is a reply.
reason	String	Reason for the refund. Currently, one of "post_deleted" if the post was deleted within 24 hours of being posted or removed from scheduled messages without being posted, or "payment_refunded" if the payer refunded their payment.

GiveawayCreated

This object represents a service message about the creation of a scheduled giveaway.

Field	Type	Description
prize_star_count	Integer	<i>Optional.</i> The number of Telegram Stars to be split between giveaway winners; for Telegram Star giveaways only

Giveaway

This object represents a message about a scheduled giveaway.

Field	Type	Description
chats	Array of Chat	The list of chats which the user must join to participate in the giveaway
winners_selection_date	Integer	Point in time (Unix timestamp) when winners of the giveaway will be selected
winner_count	Integer	The number of users which are supposed to be selected as winners of the giveaway
only_new_members	True	<i>Optional.</i> <code>True</code> , if only users who join the chats after the giveaway started should be eligible to win
has_public_winners	True	<i>Optional.</i> <code>True</code> , if the list of giveaway winners will be visible to everyone
prize_description	String	<i>Optional.</i> Description of additional giveaway prize
country_codes	Array of String	<i>Optional.</i> A list of two-letter ISO 3166-1 alpha-2 country codes indicating the countries from which eligible users for the giveaway must come. If empty, then all users can participate in the giveaway. Users with a phone number that was bought on Fragment can always participate in giveaways.
prize_star_count	Integer	<i>Optional.</i> The number of Telegram Stars to be split between giveaway winners; for Telegram Star giveaways only
premium_subscription_month_count	Integer	<i>Optional.</i> The number of months the Telegram Premium subscription won from the giveaway will be active for; for Telegram Premium giveaways only

GiveawayWinners

This object represents a message about the completion of a giveaway with public winners.

Field	Type	Description
chat	Chat	The chat that created the giveaway
giveaway_message_id	Integer	Identifier of the message with the giveaway in the chat
winners_selection_date	Integer	Point in time (Unix timestamp) when winners of the giveaway were selected
winner_count	Integer	Total number of winners in the giveaway
winners	Array of User	List of up to 100 winners of the giveaway
additional_chat_count	Integer	<i>Optional.</i> The number of other chats the user had to join in order to be eligible for the giveaway
prize_star_count	Integer	<i>Optional.</i> The number of Telegram Stars that were split between giveaway winners; for Telegram Star giveaways only
premium_subscription_month_count	Integer	<i>Optional.</i> The number of months the Telegram Premium subscription won from the giveaway will be active for; for Telegram Premium giveaways only
unclaimed_prize_count	Integer	<i>Optional.</i> Number of undistributed prizes
only_new_members	True	<i>Optional.</i> <code>True</code> , if only users who had joined the chats after the giveaway started were eligible to win
was_refunded	True	<i>Optional.</i> <code>True</code> , if the giveaway was canceled because the payment for it was refunded

Field	Type	Description
prize_description	String	<i>Optional.</i> Description of additional giveaway prize

GiveawayCompleted

This object represents a service message about the completion of a giveaway without public winners.

Field	Type	Description
winner_count	Integer	Number of winners in the giveaway
unclaimed_prize_count	Integer	<i>Optional.</i> Number of undistributed prizes
giveaway_message	Message	<i>Optional.</i> Message with the giveaway that was completed, if it wasn't deleted
is_star_giveaway	True	<i>Optional.</i> True, if the giveaway is a Telegram Star giveaway. Otherwise, currently, the giveaway is a Telegram Premium giveaway.

LinkPreviewOptions

Describes the options used for link preview generation.

Field	Type	Description
is_disabled	Boolean	<i>Optional.</i> True, if the link preview is disabled
url	String	<i>Optional.</i> URL to use for the link preview. If empty, then the first URL found in the message text will be used
prefer_small_media	Boolean	<i>Optional.</i> True, if the media in the link preview is supposed to be shrunk; ignored if the URL isn't explicitly specified or media size change isn't supported for the preview
prefer_large_media	Boolean	<i>Optional.</i> True, if the media in the link preview is supposed to be enlarged; ignored if the URL isn't explicitly specified or media size change isn't supported for the preview
show_above_text	Boolean	<i>Optional.</i> True, if the link preview must be shown above the message text; otherwise, the link preview will be shown below the message text

SuggestedPostPrice

Describes the price of a suggested post.

Field	Type	Description
currency	String	Currency in which the post will be paid. Currently, must be one of "XTR" for Telegram Stars or "TON" for toncoins
amount	Integer	The amount of the currency that will be paid for the post in the <i>smallest units</i> of the currency, i.e. Telegram Stars or nanotoncoins. Currently, price in Telegram Stars must be between 5 and 100000, and price in nanotoncoins must be between 10000000000000 and 1000000000000000.

SuggestedPostInfo

Contains information about a suggested post.

Field	Type	Description
state	String	State of the suggested post. Currently, it can be one of "pending", "approved", "declined".
price	SuggestedPostPrice	<i>Optional.</i> Proposed price of the post. If the field is omitted, then the post is unpaid.
send_date	Integer	<i>Optional.</i> Proposed send date of the post. If the field is omitted, then the post can be published at any time within 30 days at the sole discretion of the user or administrator who approves it.

SuggestedPostParameters

Contains parameters of a post that is being suggested by the bot.

Field	Type	Description
price	SuggestedPostPrice	<i>Optional.</i> Proposed price for the post. If the field is omitted, then the post is unpaid.
send_date	Integer	<i>Optional.</i> Proposed send date of the post. If specified, then the date must be between 300 second and 2678400 seconds (30 days) in the future. If the field is omitted, then the post can be published at any time within 30 days at the sole discretion of the user who approves it.

DirectMessagesTopic

Describes a topic of a direct messages chat.

Field	Type	Description
topic_id	Integer	Unique identifier of the topic. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a 64-bit integer or double-precision float type are safe for storing this identifier.
user	User	<i>Optional.</i> Information about the user that created the topic. Currently, it is always present

UserProfilePhotos

This object represent a user's profile pictures.

Field	Type	Description
total_count	Integer	Total number of profile pictures the target user has
photos	Array of Array of PhotoSize	Requested profile pictures (in up to 4 sizes each)

File

This object represents a file ready to be downloaded. The file can be downloaded via the link https://api.telegram.org/file/bot<token>/<file_path>. It is guaranteed that the link will be valid for at least 1 hour. When the link expires, a new one can be requested by calling `getFile`.

The maximum file size to download is 20 MB

Field	Type	Description
file_id	String	Identifier for this file, which can be used to download or reuse the file
file_unique_id	String	Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
file_size	Integer	<i>Optional.</i> File size in bytes. It can be bigger than 2^{31} and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this value.
file_path	String	<i>Optional.</i> File path. Use <a href="https://api.telegram.org/file/bot<token>/<file_path>">https://api.telegram.org/file/bot<token>/<file_path> to get the file.

WebAppInfo

Describes a [Web App](#).

Field	Type	Description
url	String	An HTTPS URL of a Web App to be opened with additional data as specified in Initializing Web Apps

ReplyKeyboardMarkup

This object represents a [custom keyboard](#) with reply options (see [Introduction to bots](#) for details and examples). Not supported in channels and for messages sent on behalf of a Telegram Business account.

Field	Type	Description
keyboard	Array of Array of KeyboardButton	Array of button rows, each represented by an Array of KeyboardButton objects
is_persistent	Boolean	<i>Optional.</i> Requests clients to always show the keyboard when the regular keyboard is hidden. Defaults to <code>false</code> , in which case the custom keyboard can be hidden and opened with a keyboard icon.
resize_keyboard	Boolean	<i>Optional.</i> Requests clients to resize the keyboard vertically for optimal fit (e.g., make the keyboard smaller if there are just two rows of buttons). Defaults to <code>false</code> , in which case the custom keyboard is always of the same height as the app's standard keyboard.
one_time_keyboard	Boolean	<i>Optional.</i> Requests clients to hide the keyboard as soon as it's been used. The keyboard will still be available, but clients will automatically display the usual letter-keyboard in the chat – the user can press a special button in the input field to see the custom keyboard again. Defaults to <code>false</code> .
input_field_placeholder	String	<i>Optional.</i> The placeholder to be shown in the input field when the keyboard is active; 1–64 characters
selective	Boolean	<i>Optional.</i> Use this parameter if you want to show the keyboard to specific users only. Targets: 1) users that are @mentioned in the <code>text</code> of the Message object; 2) if the bot's message is a reply to a message in the same chat and forum topic, sender of the original message.

Example: A user requests to change the bot's language, bot replies to the request with a keyboard to select the new language. Other users in the group don't see the keyboard.

KeyboardButton

This object represents one button of the reply keyboard. At most one of the optional fields must be used to specify type of the button. For simple text buttons, `String` can be used instead of this object to specify the button text.

Field	Type	Description
text	String	Text of the button. If none of the optional fields are used, it will be sent as a message when the button is pressed
request_users	KeyboardButtonRequestUsers	<i>Optional.</i> If specified, pressing the button will open a list of suitable users. Identifiers of selected users will be sent to the bot in a "users_shared" service message. Available in private chats only.
request_chat	KeyboardButtonRequestChat	<i>Optional.</i> If specified, pressing the button will open a list of suitable chats. Tapping on a chat will send its identifier to the bot in a "chat_shared" service message. Available in private chats only.
request_contact	Boolean	<i>Optional.</i> If <code>True</code> , the user's phone number will be sent as a contact when the button is pressed. Available in private chats only.
request_location	Boolean	<i>Optional.</i> If <code>True</code> , the user's current location will be sent when the button is pressed. Available in private chats only.
request_poll	KeyboardButtonPollType	<i>Optional.</i> If specified, the user will be asked to create a poll and send it to the bot when the button is pressed. Available in private chats only.
web_app	WebAppInfo	<i>Optional.</i> If specified, the described Web App will be launched when the button is pressed. The Web App will be able to send a "web_app_data" service message. Available in private chats only.

Note: `request_users` and `request_chat` options will only work in Telegram versions released after 3 February, 2023. Older clients will display *unsupported message*.

KeyboardButtonRequestUsers

This object defines the criteria used to request suitable users. Information about the selected users will be shared with the bot when the corresponding button is pressed. [More about requesting users](#)

Field	Type	Description
request_id	Integer	Signed 32-bit identifier of the request that will be received back in the UsersShared object. Must be unique within the message

Field	Type	Description
user_is_bot	Boolean	<i>Optional.</i> Pass <code>True</code> to request bots, pass <code>False</code> to request regular users. If not specified, no additional restrictions are applied.
user_is_premium	Boolean	<i>Optional.</i> Pass <code>True</code> to request premium users, pass <code>False</code> to request non-premium users. If not specified, no additional restrictions are applied.
max_quantity	Integer	<i>Optional.</i> The maximum number of users to be selected; 1–10. Defaults to 1.
request_name	Boolean	<i>Optional.</i> Pass <code>True</code> to request the users' first and last names
request_username	Boolean	<i>Optional.</i> Pass <code>True</code> to request the users' usernames
request_photo	Boolean	<i>Optional.</i> Pass <code>True</code> to request the users' photos

KeyboardButtonRequestChat

This object defines the criteria used to request a suitable chat. Information about the selected chat will be shared with the bot when the corresponding button is pressed. The bot will be granted requested rights in the chat if appropriate. [More about requesting chats »](#).

Field	Type	Description
request_id	Integer	Signed 32-bit identifier of the request, which will be received back in the <code>ChatShared</code> object. Must be unique within the message
chat_is_channel	Boolean	Pass <code>True</code> to request a channel chat, pass <code>False</code> to request a group or a supergroup chat.
chat_is_forum	Boolean	<i>Optional.</i> Pass <code>True</code> to request a forum supergroup, pass <code>False</code> to request a non-forum chat. If not specified, no additional restrictions are applied.
chat_has_username	Boolean	<i>Optional.</i> Pass <code>True</code> to request a supergroup or a channel with a username, pass <code>False</code> to request a chat without a username. If not specified, no additional restrictions are applied.
chat_is_created	Boolean	<i>Optional.</i> Pass <code>True</code> to request a chat owned by the user. Otherwise, no additional restrictions are applied.
user_administrator_rights	ChatAdministratorRights	<i>Optional.</i> A JSON-serialized object listing the required administrator rights of the user in the chat. The rights must be a superset of <code>bot_administrator_rights</code> . If not specified, no additional restrictions are applied.
bot_administrator_rights	ChatAdministratorRights	<i>Optional.</i> A JSON-serialized object listing the required administrator rights of the bot in the chat. The rights must be a subset of <code>user_administrator_rights</code> . If not specified, no additional restrictions are applied.
bot_is_member	Boolean	<i>Optional.</i> Pass <code>True</code> to request a chat with the bot as a member. Otherwise, no additional restrictions are applied.
request_title	Boolean	<i>Optional.</i> Pass <code>True</code> to request the chat's title
request_username	Boolean	<i>Optional.</i> Pass <code>True</code> to request the chat's username
request_photo	Boolean	<i>Optional.</i> Pass <code>True</code> to request the chat's photo

KeyboardButtonPollType

This object represents type of a poll, which is allowed to be created and sent when the corresponding button is pressed.

Field	Type	Description
type	String	<i>Optional.</i> If <code>quiz</code> is passed, the user will be allowed to create only polls in the quiz mode. If <code>regular</code> is passed, only regular polls will be allowed. Otherwise, the user will be allowed to create a poll of any type.

ReplyKeyboardRemove

Upon receiving a message with this object, Telegram clients will remove the current custom keyboard and display the default letter-keyboard. By default, custom keyboards are displayed until a new keyboard is sent by a bot. An exception is made for one-time keyboards that are hidden immediately after the user presses a button (see [ReplyKeyboardMarkup](#)). Not supported in channels and for messages sent on behalf of a Telegram Business account.

Field	Type	Description
remove_keyboard	True	Requests clients to remove the custom keyboard (user will not be able to summon this keyboard; if you want to hide the keyboard from sight but keep it accessible, use <code>one_time_keyboard</code> in ReplyKeyboardMarkup)
selective	Boolean	<i>Optional.</i> Use this parameter if you want to remove the keyboard for specific users only. Targets: 1) users that are @mentioned in the <code>text</code> of the Message object; 2) if the bot's message is a reply to a message in the same chat and forum topic, sender of the original message.

Example: A user votes in a poll, bot returns confirmation message in reply to the vote and removes the keyboard for that user, while still showing the keyboard with poll options to users who haven't voted yet.

InlineKeyboardMarkup

This object represents an [inline keyboard](#) that appears right next to the message it belongs to.

Field	Type	Description
inline_keyboard	Array of Array of InlineKeyboardButton	Array of button rows, each represented by an Array of InlineKeyboardButton objects

InlineKeyboardButton

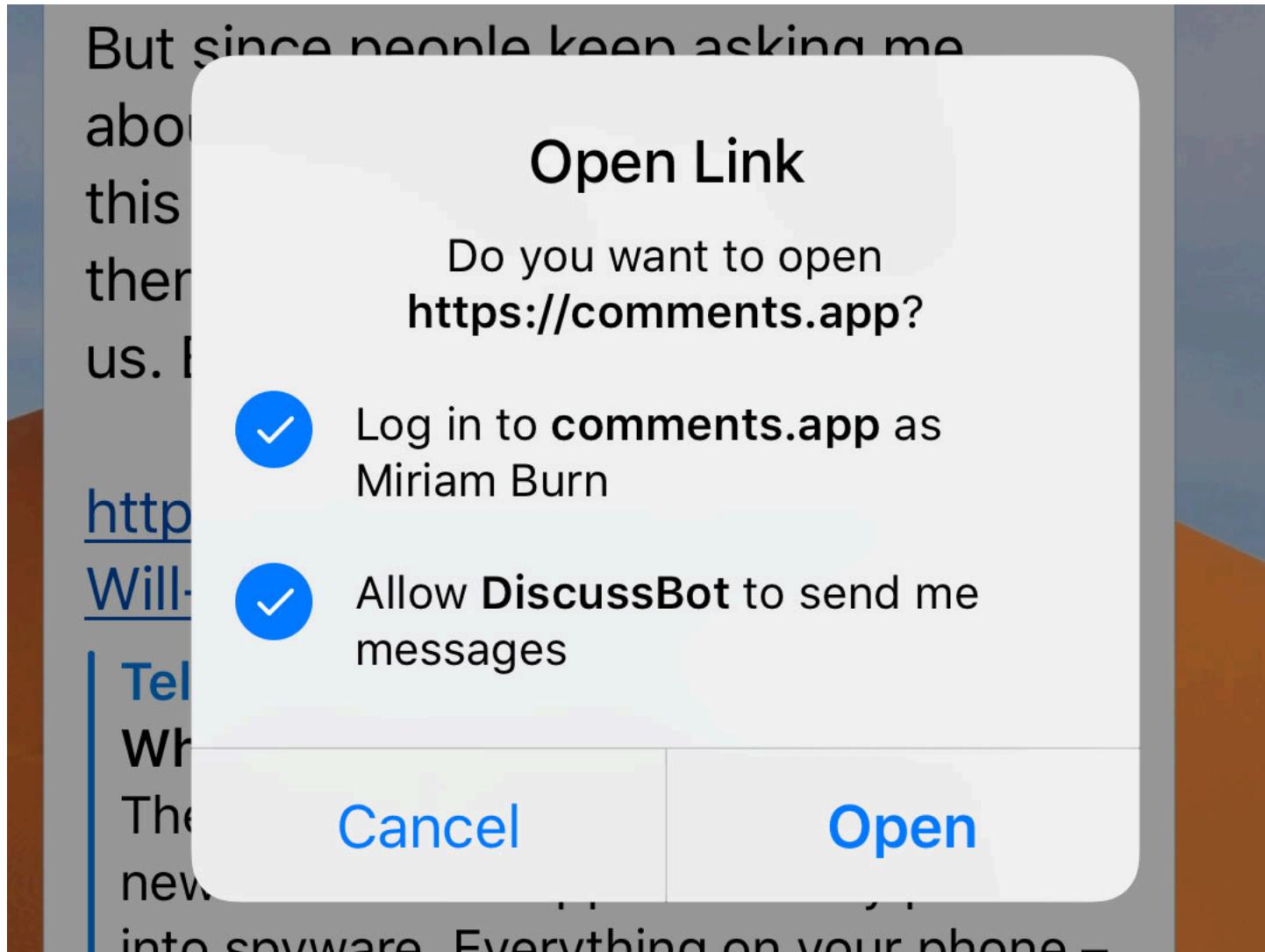
This object represents one button of an inline keyboard. Exactly one of the optional fields must be used to specify type of the button.

Field	Type	Description
text	String	Label text on the button

Field	Type	Description
url	String	<i>Optional.</i> HTTP or tg:// URL to be opened when the button is pressed. Links <code>tg://user?id=<user_id></code> can be used to mention a user by their identifier without using a username, if this is allowed by their privacy settings.
callback_data	String	<i>Optional.</i> Data to be sent in a callback query to the bot when the button is pressed, 1–64 bytes
web_app	WebAppInfo	<i>Optional.</i> Description of the Web App that will be launched when the user presses the button. The Web App will be able to send an arbitrary message on behalf of the user using the method answerWebAppQuery . Available only in private chats between a user and the bot. Not supported for messages sent on behalf of a Telegram Business account.
login_url	LoginUrl	<i>Optional.</i> An HTTPS URL used to automatically authorize the user. Can be used as a replacement for the Telegram Login Widget .
switch_inline_query	String	<i>Optional.</i> If set, pressing the button will prompt the user to select one of their chats, open that chat and insert the bot's username and the specified inline query in the input field. May be empty, in which case just the bot's username will be inserted. Not supported for messages sent in channel direct messages chats and on behalf of a Telegram Business account.
switch_inline_query_current_chat	String	<i>Optional.</i> If set, pressing the button will insert the bot's username and the specified inline query in the current chat's input field. May be empty, in which case only the bot's username will be inserted. This offers a quick way for the user to open your bot in inline mode in the same chat – good for selecting something from multiple options. Not supported in channels and for messages sent in channel direct messages chats and on behalf of a Telegram Business account.
switch_inline_query_chosen_chat	SwitchInlineQueryChosenChat	<i>Optional.</i> If set, pressing the button will prompt the user to select one of their chats of the specified type, open that chat and insert the bot's username and the specified inline query in the input field. Not supported for messages sent in channel direct messages chats and on behalf of a Telegram Business account.
copy_text	CopyTextButton	<i>Optional.</i> Description of the button that copies the specified text to the clipboard.
callback_game	CallbackGame	<i>Optional.</i> Description of the game that will be launched when the user presses the button. NOTE: This type of button must always be the first button in the first row.
pay	Boolean	<i>Optional.</i> Specify <code>True</code> , to send a Pay button. Substrings “ <code>★</code> ” and “ <code>XTR</code> ” in the button's text will be replaced with a Telegram Star icon. NOTE: This type of button must always be the first button in the first row and can only be used in invoice messages.

LoginUrl

This object represents a parameter of the inline keyboard button used to automatically authorize a user. Serves as a great replacement for the [Telegram Login Widget](#) when the user is coming from Telegram. All the user needs to do is tap/click a button and confirm that they want to log in:



Telegram apps support these buttons as of version 5.7.

Sample bot: @discussbot

Field	Type	Description
url	String	An HTTPS URL to be opened with user authorization data added to the query string when the button is pressed. If the user refuses to provide authorization data, the original URL without information about the user will be opened. The data added is the same as described in Receiving authorization data .
NOTE: You must always check the hash of the received data to verify the authentication and the integrity of the data as described in Checking authorization .		
forward_text	String	<i>Optional.</i> New text of the button in forwarded messages.
bot_username	String	<i>Optional.</i> Username of a bot, which will be used for user authorization. See Setting up a bot for more details. If not specified, the current bot's username will be assumed. The url's domain must be the same as the domain linked with the bot. See Linking your domain to the bot for more details.
request_write_access	Boolean	<i>Optional.</i> Pass <code>True</code> to request the permission for your bot to send messages to the user.

SwitchInlineQueryChosenChat

This object represents an inline button that switches the current user to inline mode in a chosen chat, with an optional default inline query.

Field	Type	Description
query	String	<i>Optional.</i> The default inline query to be inserted in the input field. If left empty, only the bot's username will be inserted
allow_user_chats	Boolean	<i>Optional.</i> <code>True</code> , if private chats with users can be chosen
allow_bot_chats	Boolean	<i>Optional.</i> <code>True</code> , if private chats with bots can be chosen
allow_group_chats	Boolean	<i>Optional.</i> <code>True</code> , if group and supergroup chats can be chosen
allow_channel_chats	Boolean	<i>Optional.</i> <code>True</code> , if channel chats can be chosen

CopyTextButton

This object represents an inline keyboard button that copies specified text to the clipboard.

Field	Type	Description
text	String	The text to be copied to the clipboard; 1–256 characters

CallbackQuery

This object represents an incoming callback query from a callback button in an [inline keyboard](#). If the button that originated the query was attached to a message sent by the bot, the field `message` will be present. If the button was attached to a message sent via the bot (in [inline mode](#)), the field `inline_message_id` will be present. Exactly one of the fields `data` or `game_short_name` will be present.

Field	Type	Description
id	String	Unique identifier for this query
from	User	Sender
message	MaybeInaccessibleMessage	<i>Optional.</i> Message sent by the bot with the callback button that originated the query
inline_message_id	String	<i>Optional.</i> Identifier of the message sent via the bot in inline mode, that originated the query.
chat_instance	String	Global identifier, uniquely corresponding to the chat to which the message with the callback button was sent. Useful for high scores in games .
data	String	<i>Optional.</i> Data associated with the callback button. Be aware that the message originated the query can contain no callback buttons with this data.
game_short_name	String	<i>Optional.</i> Short name of a Game to be returned, serves as the unique identifier for the game

NOTE: After the user presses a callback button, Telegram clients will display a progress bar until you call [answerCallbackQuery](#). It is, therefore, necessary to react by calling [answerCallbackQuery](#) even if no notification to the user is needed (e.g., without specifying any of the optional parameters).

ForceReply

Upon receiving a message with this object, Telegram clients will display a reply interface to the user (act as if the user has selected the bot's message and tapped 'Reply'). This can be extremely useful if you want to create user-friendly step-by-step interfaces without having to sacrifice [privacy mode](#). Not supported in channels and for messages sent on behalf of a Telegram Business account.

Field	Type	Description
force_reply	True	Shows reply interface to the user, as if they manually selected the bot's message and tapped 'Reply'
input_field_placeholder	String	<i>Optional.</i> The placeholder to be shown in the input field when the reply is active; 1–64 characters
selective	Boolean	<i>Optional.</i> Use this parameter if you want to force reply from specific users only. Targets: 1) users that are @mentioned in the <code>text</code> of the Message object; 2) if the bot's message is a reply to a message in the same chat and forum topic, sender of the original message.

Example: A [poll bot](#) for groups runs in privacy mode (only receives commands, replies to its messages and mentions). There could be two ways to create a new poll:

- Explain the user how to send a command with parameters (e.g. /newpoll question answer1 answer2). May be appealing for hardcore users but lacks modern day polish.
- Guide the user through a step-by-step process. 'Please send me your question', 'Cool, now let's add the first answer option', 'Great. Keep adding answer options, then send /done when you're ready'.

The last option is definitely more attractive. And if you use [ForceReply](#) in your bot's questions, it will receive the user's answers even if it only receives replies, commands and mentions – without any extra work for the user.

ChatPhoto

This object represents a chat photo.

Field	Type	Description
small_file_id	String	File identifier of small (160x160) chat photo. This file_id can be used only for photo download and only for as long as the photo is not changed.
small_file_unique_id	String	Unique file identifier of small (160x160) chat photo, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.
big_file_id	String	File identifier of big (640x640) chat photo. This file_id can be used only for photo download and only for as long as the photo is not changed.
big_file_unique_id	String	Unique file identifier of big (640x640) chat photo, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file.

ChatInviteLink

Represents an invite link for a chat.

Field	Type	Description
invite_link	String	The invite link. If the link was created by another chat administrator, then the second part of the link will be replaced with "...".
creator	User	Creator of the link
creates_join_request	Boolean	<i>True</i> , if users joining the chat via the link need to be approved by chat administrators
is_primary	Boolean	<i>True</i> , if the link is primary

Field	Type	Description
is_revoked	Boolean	<i>True</i> , if the link is revoked
name	String	<i>Optional</i> . Invite link name
expire_date	Integer	<i>Optional</i> . Point in time (Unix timestamp) when the link will expire or has been expired
member_limit	Integer	<i>Optional</i> . The maximum number of users that can be members of the chat simultaneously after joining the chat via this invite link; 1–99999
pending_join_request_count	Integer	<i>Optional</i> . Number of pending join requests created using this link
subscription_period	Integer	<i>Optional</i> . The number of seconds the subscription will be active for before the next payment
subscription_price	Integer	<i>Optional</i> . The amount of Telegram Stars a user must pay initially and after each subsequent subscription period to be a member of the chat using the link

ChatAdministratorRights

Represents the rights of an administrator in a chat.

Field	Type	Description
is_anonymous	Boolean	<i>True</i> , if the user's presence in the chat is hidden
can_manage_chat	Boolean	<i>True</i> , if the administrator can access the chat event log, get boost list, see hidden supergroup and channel members, report spam messages, ignore slow mode, and send messages to the chat without paying Telegram Stars. Implied by any other administrator privilege.
can_delete_messages	Boolean	<i>True</i> , if the administrator can delete messages of other users
can_manage_video_chats	Boolean	<i>True</i> , if the administrator can manage video chats
can_restrict_members	Boolean	<i>True</i> , if the administrator can restrict, ban or unban chat members, or access supergroup statistics
can_promote_members	Boolean	<i>True</i> , if the administrator can add new administrators with a subset of their own privileges or demote administrators that they have promoted, directly or indirectly (promoted by administrators that were appointed by the user)
can_change_info	Boolean	<i>True</i> , if the user is allowed to change the chat title, photo and other settings
can_invite_users	Boolean	<i>True</i> , if the user is allowed to invite new users to the chat
can_post_stories	Boolean	<i>True</i> , if the administrator can post stories to the chat
can_edit_stories	Boolean	<i>True</i> , if the administrator can edit stories posted by other users, post stories to the chat page, pin chat stories, and access the chat's story archive
can_delete_stories	Boolean	<i>True</i> , if the administrator can delete stories posted by other users
can_post_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can post messages in the channel, approve suggested posts, or access channel statistics; for channels only
can_edit_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can edit messages of other users and can pin messages; for channels only
can_pin_messages	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to pin messages; for groups and supergroups only
can_manage_topics	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to create, rename, close, and reopen forum topics; for supergroups only
can_manage_direct_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can manage direct messages of the channel and decline suggested posts; for channels only

ChatMemberUpdated

This object represents changes in the status of a chat member.

Field	Type	Description
chat	Chat	Chat the user belongs to
from	User	Performer of the action, which resulted in the change
date	Integer	Date the change was done in Unix time
old_chat_member	ChatMember	Previous information about the chat member
new_chat_member	ChatMember	New information about the chat member
invite_link	ChatInviteLink	<i>Optional</i> . Chat invite link, which was used by the user to join the chat; for joining by invite link events only.
via_join_request	Boolean	<i>Optional</i> . <i>True</i> , if the user joined the chat after sending a direct join request without using an invite link and being approved by an administrator
via_chat_folder_invite_link	Boolean	<i>Optional</i> . <i>True</i> , if the user joined the chat via a chat folder invite link

ChatMember

This object contains information about one member of a chat. Currently, the following 6 types of chat members are supported:

- ChatMemberOwner
- ChatMemberAdministrator
- ChatMemberMember
- ChatMemberRestricted
- ChatMemberLeft
- ChatMemberBanned

ChatMemberOwner

Represents a [chat member](#) that owns the chat and has all administrator privileges.

Field	Type	Description
status	String	The member's status in the chat, always "creator"
user	User	Information about the user
is_anonymous	Boolean	<i>True</i> , if the user's presence in the chat is hidden
custom_title	String	<i>Optional</i> . Custom title for this user

ChatMemberAdministrator

Represents a [chat member](#) that has some additional privileges.

Field	Type	Description
status	String	The member's status in the chat, always "administrator"
user	User	Information about the user
can_be_edited	Boolean	<i>True</i> , if the bot is allowed to edit administrator privileges of that user
is_anonymous	Boolean	<i>True</i> , if the user's presence in the chat is hidden
can_manage_chat	Boolean	<i>True</i> , if the administrator can access the chat event log, get boost list, see hidden supergroup and channel members, report spam messages, ignore slow mode, and send messages to the chat without paying Telegram Stars. Implied by any other administrator privilege.
can_delete_messages	Boolean	<i>True</i> , if the administrator can delete messages of other users
can_manage_video_chats	Boolean	<i>True</i> , if the administrator can manage video chats
can_restrict_members	Boolean	<i>True</i> , if the administrator can restrict, ban or unban chat members, or access supergroup statistics
can_promote_members	Boolean	<i>True</i> , if the administrator can add new administrators with a subset of their own privileges or demote administrators that they have promoted, directly or indirectly (promoted by administrators that were appointed by the user)
can_change_info	Boolean	<i>True</i> , if the user is allowed to change the chat title, photo and other settings
can_invite_users	Boolean	<i>True</i> , if the user is allowed to invite new users to the chat
can_post_stories	Boolean	<i>True</i> , if the administrator can post stories to the chat
can_edit_stories	Boolean	<i>True</i> , if the administrator can edit stories posted by other users, post stories to the chat page, pin chat stories, and access the chat's story archive
can_delete_stories	Boolean	<i>True</i> , if the administrator can delete stories posted by other users
can_post_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can post messages in the channel, approve suggested posts, or access channel statistics; for channels only
can_edit_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can edit messages of other users and can pin messages; for channels only
can_pin_messages	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to pin messages; for groups and supergroups only
can_manage_topics	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to create, rename, close, and reopen forum topics; for supergroups only
can_manage_direct_messages	Boolean	<i>Optional</i> . <i>True</i> , if the administrator can manage direct messages of the channel and decline suggested posts; for channels only
custom_title	String	<i>Optional</i> . Custom title for this user

ChatMemberMember

Represents a [chat member](#) that has no additional privileges or restrictions.

Field	Type	Description
status	String	The member's status in the chat, always "member"
user	User	Information about the user
until_date	Integer	<i>Optional</i> . Date when the user's subscription will expire; Unix time

ChatMemberRestricted

Represents a [chat member](#) that is under certain restrictions in the chat. Supergroups only.

Field	Type	Description
status	String	The member's status in the chat, always "restricted"
user	User	Information about the user
is_member	Boolean	<i>True</i> , if the user is a member of the chat at the moment of the request
can_send_messages	Boolean	<i>True</i> , if the user is allowed to send text messages, contacts, giveaways, giveaway winners, invoices, locations and venues
can_send_audios	Boolean	<i>True</i> , if the user is allowed to send audios

Field	Type	Description
can_send_documents	Boolean	<i>True</i> , if the user is allowed to send documents
can_send_photos	Boolean	<i>True</i> , if the user is allowed to send photos
can_send_videos	Boolean	<i>True</i> , if the user is allowed to send videos
can_send_video_notes	Boolean	<i>True</i> , if the user is allowed to send video notes
can_send_voice_notes	Boolean	<i>True</i> , if the user is allowed to send voice notes
can_send_polls	Boolean	<i>True</i> , if the user is allowed to send polls and checklists
can_send_other_messages	Boolean	<i>True</i> , if the user is allowed to send animations, games, stickers and use inline bots
can_add_web_page_previews	Boolean	<i>True</i> , if the user is allowed to add web page previews to their messages
can_change_info	Boolean	<i>True</i> , if the user is allowed to change the chat title, photo and other settings
can_invite_users	Boolean	<i>True</i> , if the user is allowed to invite new users to the chat
can_pin_messages	Boolean	<i>True</i> , if the user is allowed to pin messages
can_manage_topics	Boolean	<i>True</i> , if the user is allowed to create forum topics
until_date	Integer	Date when restrictions will be lifted for this user; Unix time. If 0, then the user is restricted forever

ChatMemberLeft

Represents a [chat member](#) that isn't currently a member of the chat, but may join it themselves.

Field	Type	Description
status	String	The member's status in the chat, always "left"
user	User	Information about the user

ChatMemberBanned

Represents a [chat member](#) that was banned in the chat and can't return to the chat or view chat messages.

Field	Type	Description
status	String	The member's status in the chat, always "kicked"
user	User	Information about the user
until_date	Integer	Date when restrictions will be lifted for this user; Unix time. If 0, then the user is banned forever

ChatJoinRequest

Represents a join request sent to a chat.

Field	Type	Description
chat	Chat	Chat to which the request was sent
from	User	User that sent the join request
user_chat_id	Integer	Identifier of a private chat with the user who sent the join request. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a 64-bit integer or double-precision float type are safe for storing this identifier. The bot can use this identifier for 5 minutes to send messages until the join request is processed, assuming no other administrator contacted the user.
date	Integer	Date the request was sent in Unix time
bio	String	<i>Optional</i> . Bio of the user.
invite_link	ChatInviteLink	<i>Optional</i> . Chat invite link that was used by the user to send the join request

ChatPermissions

Describes actions that a non-administrator user is allowed to take in a chat.

Field	Type	Description
can_send_messages	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send text messages, contacts, giveaways, giveaway winners, invoices, locations and venues
can_send_audios	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send audios
can_send_documents	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send documents
can_send_photos	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send photos
can_send_videos	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send videos
can_send_video_notes	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send video notes
can_send_voice_notes	Boolean	<i>Optional</i> . <i>True</i> , if the user is allowed to send voice notes

Field	Type	Description
can_send_polls	Boolean	<i>Optional.</i> True, if the user is allowed to send polls and checklists
can_send_other_messages	Boolean	<i>Optional.</i> True, if the user is allowed to send animations, games, stickers and use inline bots
can_add_web_page_previews	Boolean	<i>Optional.</i> True, if the user is allowed to add web page previews to their messages
can_change_info	Boolean	<i>Optional.</i> True, if the user is allowed to change the chat title, photo and other settings. Ignored in public supergroups
can_invite_users	Boolean	<i>Optional.</i> True, if the user is allowed to invite new users to the chat
can_pin_messages	Boolean	<i>Optional.</i> True, if the user is allowed to pin messages. Ignored in public supergroups
can_manage_topics	Boolean	<i>Optional.</i> True, if the user is allowed to create forum topics. If omitted defaults to the value of can_pin_messages

Birthdate

Describes the birthdate of a user.

Field	Type	Description
day	Integer	Day of the user's birth; 1-31
month	Integer	Month of the user's birth; 1-12
year	Integer	<i>Optional.</i> Year of the user's birth

BusinessIntro

Contains information about the start page settings of a Telegram Business account.

Field	Type	Description
title	String	<i>Optional.</i> Title text of the business intro
message	String	<i>Optional.</i> Message text of the business intro
sticker	Sticker	<i>Optional.</i> Sticker of the business intro

BusinessLocation

Contains information about the location of a Telegram Business account.

Field	Type	Description
address	String	Address of the business
location	Location	<i>Optional.</i> Location of the business

BusinessOpeningHoursInterval

Describes an interval of time during which a business is open.

Field	Type	Description
opening_minute	Integer	The minute's sequence number in a week, starting on Monday, marking the start of the time interval during which the business is open; 0 – 7 * 24 * 60
closing_minute	Integer	The minute's sequence number in a week, starting on Monday, marking the end of the time interval during which the business is open; 0 – 8 * 24 * 60

BusinessOpeningHours

Describes the opening hours of a business.

Field	Type	Description
time_zone_name	String	Unique name of the time zone for which the opening hours are defined
opening_hours	Array of BusinessOpeningHoursInterval	List of time intervals describing business opening hours

StoryAreaPosition

Describes the position of a clickable area within a story.

Field	Type	Description
x_percentage	Float	The abscissa of the area's center, as a percentage of the media width
y_percentage	Float	The ordinate of the area's center, as a percentage of the media height
width_percentage	Float	The width of the area's rectangle, as a percentage of the media width
height_percentage	Float	The height of the area's rectangle, as a percentage of the media height
rotation_angle	Float	The clockwise rotation angle of the rectangle, in degrees; 0-360
corner_radius_percentage	Float	The radius of the rectangle corner rounding, as a percentage of the media width

LocationAddress

Describes the physical address of a location.

Field	Type	Description
country_code	String	The two-letter ISO 3166-1 alpha-2 country code of the location where the location is located
state	String	<i>Optional.</i> State of the location
city	String	<i>Optional.</i> City of the location
street	String	<i>Optional.</i> Street address of the location

StoryAreaType

Describes the type of a clickable area on a story. Currently, it can be one of

- StoryAreaTypeLocation
- StoryAreaTypeSuggestedReaction
- StoryAreaTypeLink
- StoryAreaTypeWeather
- StoryAreaTypeUniqueGift

StoryAreaTypeLocation

Describes a story area pointing to a location. Currently, a story can have up to 10 location areas.

Field	Type	Description
type	String	Type of the area, always "location"
latitude	Float	Location latitude in degrees
longitude	Float	Location longitude in degrees
address	LocationAddress	<i>Optional.</i> Address of the location

StoryAreaTypeSuggestedReaction

Describes a story area pointing to a suggested reaction. Currently, a story can have up to 5 suggested reaction areas.

Field	Type	Description
type	String	Type of the area, always "suggested_reaction"
reaction_type	ReactionType	Type of the reaction
is_dark	Boolean	<i>Optional.</i> Pass <code>True</code> if the reaction area has a dark background
is_flipped	Boolean	<i>Optional.</i> Pass <code>True</code> if reaction area corner is flipped

StoryAreaTypeLink

Describes a story area pointing to an HTTP or tg:// link. Currently, a story can have up to 3 link areas.

Field	Type	Description
type	String	Type of the area, always "link"
url	String	HTTP or tg:// URL to be opened when the area is clicked

StoryAreaTypeWeather

Describes a story area containing weather information. Currently, a story can have up to 3 weather areas.

Field	Type	Description
type	String	Type of the area, always "weather"
temperature	Float	Temperature, in degree Celsius
emoji	String	Emoji representing the weather
background_color	Integer	A color of the area background in the ARGB format

StoryAreaTypeUniqueGift

Describes a story area pointing to a unique gift. Currently, a story can have at most 1 unique gift area.

Field	Type	Description
type	String	Type of the area, always "unique_gift"
name	String	Unique name of the gift

StoryArea

Describes a clickable area on a story media.

Field	Type	Description
position	StoryAreaPosition	Position of the area
type	StoryAreaType	Type of the area

ChatLocation

Represents a location to which a chat is connected.

Field	Type	Description
location	Location	The location to which the supergroup is connected. Can't be a live location.
address	String	Location address: 1-64 characters, as defined by the chat owner

ReactionType

This object describes the type of a reaction. Currently, it can be one of

- ReactionTypeEmoji
 - ReactionTypeCustomEmoji
 - ReactionTypePaid

ReactionTypeEmoji

The reaction is based on an emulsion

Field	Type	Description
type	String	Type of the reaction, always "emoji"
emoji	String	Reaction emoji. Currently, it can be one of                       

ReactionTypeCustomEmoji

The reaction is based on a custom emoji.

Field	Type	Description
type	String	Type of the reaction, always "custom_emoji"
custom_emoji_id	String	Custom emoji identifier

ReactionTypePaid

The reaction is paid.

Field	Type	Description
type	String	Type of the reaction, always "paid"

ReactionCount

Represents a reaction added to a message along with the number of times it was added.

Field	Type	Description
type	ReactionType	Type of the reaction
total_count	Integer	Number of times the reaction was added

MessageReactionUpdated

This object represents a change of a reaction on a message performed by a user.

Field	Type	Description
chat	Chat	The chat containing the message the user reacted to
message_id	Integer	Unique identifier of the message inside the chat
user	User	<i>Optional.</i> The user that changed the reaction, if the user isn't anonymous
actor_chat	Chat	<i>Optional.</i> The chat on behalf of which the reaction was changed, if the user is anonymous
date	Integer	Date of the change in Unix time
old_reaction	Array of ReactionType	Previous list of reaction types that were set by the user
new_reaction	Array of ReactionType	New list of reaction types that have been set by the user

MessageReactionCountUpdated

This object represents reaction changes on a message with anonymous reactions

Field	Type	Description
chat	Chat	The chat containing the message
message_id	Integer	Unique message identifier inside the chat
date	Integer	Date of the change in Unix time
reactions	Array of ReactionCount	List of reactions that are present on the message

ForumTopic

This object represents a forum topic.

Field	Type	Description
message_thread_id	Integer	Unique identifier of the forum topic
name	String	Name of the topic
icon_color	Integer	Color of the topic icon in RGB format
icon_custom_emoji_id	String	<i>Optional.</i> Unique identifier of the custom emoji shown as the topic icon

Gift

This object represents a gift that can be sent by the bot.

Field	Type	Description
id	String	Unique identifier of the gift
sticker	Sticker	The sticker that represents the gift
star_count	Integer	The number of Telegram Stars that must be paid to send the sticker
upgrade_star_count	Integer	<i>Optional.</i> The number of Telegram Stars that must be paid to upgrade the gift to a unique one
total_count	Integer	<i>Optional.</i> The total number of the gifts of this type that can be sent; for limited gifts only
remaining_count	Integer	<i>Optional.</i> The number of remaining gifts of this type that can be sent; for limited gifts only
publisher_chat	Chat	<i>Optional.</i> Information about the chat that published the gift

Gifts

This object represent a list of gifts.

Field	Type	Description
gifts	Array of Gift	The list of gifts

UniqueGiftModel

This object describes the model of a unique gift.

Field	Type	Description
name	String	Name of the model
sticker	Sticker	The sticker that represents the unique gift
rarity_per_mille	Integer	The number of unique gifts that receive this model for every 1000 gifts upgraded

UniqueGiftSymbol

This object describes the symbol shown on the pattern of a unique gift.

Field	Type	Description
name	String	Name of the symbol
sticker	Sticker	The sticker that represents the unique gift
rarity_per_mille	Integer	The number of unique gifts that receive this model for every 1000 gifts upgraded

UniqueGiftBackdropColors

This object describes the colors of the backdrop of a unique gift.

Field	Type	Description
center_color	Integer	The color in the center of the backdrop in RGB format
edge_color	Integer	The color on the edges of the backdrop in RGB format
symbol_color	Integer	The color to be applied to the symbol in RGB format
text_color	Integer	The color for the text on the backdrop in RGB format

UniqueGiftBackdrop

This object describes the backdrop of a unique gift.

Field	Type	Description
name	String	Name of the backdrop
colors	UniqueGiftBackdropColors	Colors of the backdrop
rarity_per_mille	Integer	The number of unique gifts that receive this backdrop for every 1000 gifts upgraded

UniqueGift

This object describes a unique gift that was upgraded from a regular gift.

Field	Type	Description
base_name	String	Human-readable name of the regular gift from which this unique gift was upgraded
name	String	Unique name of the gift. This name can be used in https://t.me/nft/... links and story areas
number	Integer	Unique number of the upgraded gift among gifts upgraded from the same regular gift
model	UniqueGiftModel	Model of the gift
symbol	UniqueGiftSymbol	Symbol of the gift
backdrop	UniqueGiftBackdrop	Backdrop of the gift
publisher_chat	Chat	<i>Optional.</i> Information about the chat that published the gift

GiftInfo

Describes a service message about a regular gift that was sent or received.

Field	Type	Description
gift	Gift	Information about the gift
owned_gift_id	String	<i>Optional.</i> Unique identifier of the received gift for the bot; only present for gifts received on behalf of business accounts
convert_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that can be claimed by the receiver by converting the gift; omitted if conversion to Telegram Stars is impossible
prepaid_upgrade_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that were prepaid by the sender for the ability to upgrade the gift
can_be_upgraded	True	<i>Optional.</i> True, if the gift can be upgraded to a unique gift
text	String	<i>Optional.</i> Text of the message that was added to the gift
entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the text
is_private	True	<i>Optional.</i> True, if the sender and gift text are shown only to the gift receiver; otherwise, everyone will be able to see them

UniqueGiftInfo

Describes a service message about a unique gift that was sent or received.

Field	Type	Description
gift	UniqueGift	Information about the gift
origin	String	Origin of the gift. Currently, either "upgrade" for gifts upgraded from regular gifts, "transfer" for gifts transferred from other users or channels, or "resale" for gifts bought from other users
last_resale_star_count	Integer	<i>Optional.</i> For gifts bought from other users, the price paid for the gift
owned_gift_id	String	<i>Optional.</i> Unique identifier of the received gift for the bot; only present for gifts received on behalf of business accounts
transfer_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that must be paid to transfer the gift; omitted if the bot cannot transfer the gift
next_transfer_date	Integer	<i>Optional.</i> Point in time (Unix timestamp) when the gift can be transferred. If it is in the past, then the gift can be transferred now

OwnedGift

This object describes a gift received and owned by a user or a chat. Currently, it can be one of

- [OwnedGiftRegular](#)
- [OwnedGiftUnique](#)

OwnedGiftRegular

Describes a regular gift owned by a user or a chat.

Field	Type	Description
type	String	Type of the gift, always "regular"
gift	Gift	Information about the regular gift
owned_gift_id	String	<i>Optional.</i> Unique identifier of the gift for the bot; for gifts received on behalf of business accounts only
sender_user	User	<i>Optional.</i> Sender of the gift if it is a known user
send_date	Integer	Date the gift was sent in Unix time
text	String	<i>Optional.</i> Text of the message that was added to the gift
entities	Array of MessageEntity	<i>Optional.</i> Special entities that appear in the text
is_private	True	<i>Optional.</i> True, if the sender and gift text are shown only to the gift receiver; otherwise, everyone will be able to see them

Field	Type	Description
is_saved	True	<i>Optional.</i> <code>True</code> , if the gift is displayed on the account's profile page; for gifts received on behalf of business accounts only
can_be_upgraded	True	<i>Optional.</i> <code>True</code> , if the gift can be upgraded to a unique gift; for gifts received on behalf of business accounts only
was_refunded	True	<i>Optional.</i> <code>True</code> , if the gift was refunded and isn't available anymore
convert_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that can be claimed by the receiver instead of the gift; omitted if the gift cannot be converted to Telegram Stars
prepaid_upgrade_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that were paid by the sender for the ability to upgrade the gift

OwnedGiftUnique

Describes a unique gift received and owned by a user or a chat.

Field	Type	Description
type	String	Type of the gift, always "unique"
gift	UniqueGift	Information about the unique gift
owned_gift_id	String	<i>Optional.</i> Unique identifier of the received gift for the bot; for gifts received on behalf of business accounts only
sender_user	User	<i>Optional.</i> Sender of the gift if it is a known user
send_date	Integer	Date the gift was sent in Unix time
is_saved	True	<i>Optional.</i> <code>True</code> , if the gift is displayed on the account's profile page; for gifts received on behalf of business accounts only
can_be_transferred	True	<i>Optional.</i> <code>True</code> , if the gift can be transferred to another owner; for gifts received on behalf of business accounts only
transfer_star_count	Integer	<i>Optional.</i> Number of Telegram Stars that must be paid to transfer the gift; omitted if the bot cannot transfer the gift
next_transfer_date	Integer	<i>Optional.</i> Point in time (Unix timestamp) when the gift can be transferred. If it is in the past, then the gift can be transferred now

OwnedGifts

Contains the list of gifts received and owned by a user or a chat.

Field	Type	Description
total_count	Integer	The total number of gifts owned by the user or the chat
gifts	Array of OwnedGift	The list of gifts
next_offset	String	<i>Optional.</i> Offset for the next request. If empty, then there are no more results

AcceptedGiftTypes

This object describes the types of gifts that can be gifted to a user or a chat.

Field	Type	Description
unlimited_gifts	Boolean	<code>True</code> , if unlimited regular gifts are accepted
limited_gifts	Boolean	<code>True</code> , if limited regular gifts are accepted
unique_gifts	Boolean	<code>True</code> , if unique gifts or gifts that can be upgraded to unique for free are accepted
premium_subscription	Boolean	<code>True</code> , if a Telegram Premium subscription is accepted

StarAmount

Describes an amount of Telegram Stars.

Field	Type	Description
amount	Integer	Integer amount of Telegram Stars, rounded to 0; can be negative
nanostar_amount	Integer	<i>Optional.</i> The number of 1/100000000 shares of Telegram Stars; from -999999999 to 999999999; can be negative if and only if <code>amount</code> is non-positive

BotCommand

This object represents a bot command.

Field	Type	Description
command	String	Text of the command; 1–32 characters. Can contain only lowercase English letters, digits and underscores.
description	String	Description of the command; 1–256 characters.

BotCommandScope

This object represents the scope to which bot commands are applied. Currently, the following 7 scopes are supported:

- BotCommandScopeDefault
- BotCommandScopeAllPrivateChats
- BotCommandScopeAllGroupChats
- BotCommandScopeAllChatAdministrators

- `BotCommandScopeChat`
- `BotCommandScopeChatAdministrators`
- `BotCommandScopeChatMember`

Determining list of commands

The following algorithm is used to determine the list of commands for a particular user viewing the bot menu. The first list of commands which is set is returned:

Commands in the chat with the bot

- `botCommandScopeChat + language_code`
- `botCommandScopeChat`
- `botCommandScopeAllPrivateChats + language_code`
- `botCommandScopeAllPrivateChats`
- `botCommandScopeDefault + language_code`
- `botCommandScopeDefault`

Commands in group and supergroup chats

- `botCommandScopeChatMember + language_code`
- `botCommandScopeChatMember`
- `botCommandScopeChatAdministrators + language_code` (administrators only)
- `botCommandScopeChatAdministrators` (administrators only)
- `botCommandScopeChat + language_code`
- `botCommandScopeChat`
- `botCommandScopeAllChatAdministrators + language_code` (administrators only)
- `botCommandScopeAllChatAdministrators` (administrators only)
- `botCommandScopeAllGroupChats + language_code`
- `botCommandScopeAllGroupChats`
- `botCommandScopeDefault + language_code`
- `botCommandScopeDefault`

BotCommandScopeDefault

Represents the default `scope` of bot commands. Default commands are used if no commands with a narrower scope are specified for the user.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>default</code>

BotCommandScopeAllPrivateChats

Represents the `scope` of bot commands, covering all private chats.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>all_private_chats</code>

BotCommandScopeAllGroupChats

Represents the `scope` of bot commands, covering all group and supergroup chats.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>all_group_chats</code>

BotCommandScopeAllChatAdministrators

Represents the `scope` of bot commands, covering all group and supergroup chat administrators.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>all_chat_administrators</code>

BotCommandScopeChat

Represents the `scope` of bot commands, covering a specific chat.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>chat</code>
<code>chat_id</code>	Integer or String	Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>). Channel direct messages chats and channel chats aren't supported.

BotCommandScopeChatAdministrators

Represents the `scope` of bot commands, covering all administrators of a specific group or supergroup chat.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>chat_administrators</code>
<code>chat_id</code>	Integer or String	Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>). Channel direct messages chats and channel chats aren't supported.

BotCommandScopeChatMember

Represents the `scope` of bot commands, covering a specific member of a group or supergroup chat.

Field	Type	Description
<code>type</code>	<code>String</code>	Scope type, must be <code>chat_member</code>

Field	Type	Description
chat_id	Integer or String	Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>). Channel direct messages chats and channel chats aren't supported.
user_id	Integer	Unique identifier of the target user

BotName

This object represents the bot's name.

Field	Type	Description
name	String	The bot's name

BotDescription

This object represents the bot's description.

Field	Type	Description
description	String	The bot's description

BotShortDescription

This object represents the bot's short description.

Field	Type	Description
short_description	String	The bot's short description

MenuButton

This object describes the bot's menu button in a private chat. It should be one of

- [MenuButtonCommands](#)
- [MenuButtonWebApp](#)
- [MenuButtonDefault](#)

If a menu button other than `MenuButtonDefault` is set for a private chat, then it is applied in the chat. Otherwise the default menu button is applied. By default, the menu button opens the list of bot commands.

MenuButtonCommands

Represents a menu button, which opens the bot's list of commands.

Field	Type	Description
type	String	Type of the button, must be <code>commands</code>

MenuButtonWebApp

Represents a menu button, which launches a [Web App](#).

Field	Type	Description
type	String	Type of the button, must be <code>web_app</code>
text	String	Text on the button
web_app	WebAppInfo	Description of the Web App that will be launched when the user presses the button. The Web App will be able to send an arbitrary message on behalf of the user using the method <code>answerWebAppQuery</code> . Alternatively, a <code>t.me</code> link to a Web App of the bot can be specified in the object instead of the Web App's URL, in which case the Web App will be opened as if the user pressed the link.

MenuButtonDefault

Describes that no specific value for the menu button was set.

Field	Type	Description
type	String	Type of the button, must be <code>default</code>

ChatBoostSource

This object describes the source of a chat boost. It can be one of

- [ChatBoostSourcePremium](#)
- [ChatBoostSourceGiftCode](#)
- [ChatBoostSourceGiveaway](#)

ChatBoostSourcePremium

The boost was obtained by subscribing to Telegram Premium or by gifting a Telegram Premium subscription to another user.

Field	Type	Description
source	String	Source of the boost, always "premium"
user	User	User that boosted the chat

ChatBoostSourceGiftCode

The boost was obtained by the creation of Telegram Premium gift codes to boost a chat. Each such code boosts the chat 4 times for the duration of the corresponding Telegram Premium subscription.

Field	Type	Description
source	String	Source of the boost, always "gift_code"
user	User	User for which the gift code was created

ChatBoostSourceGiveaway

The boost was obtained by the creation of a Telegram Premium or a Telegram Star giveaway. This boosts the chat 4 times for the duration of the corresponding Telegram Premium subscription for Telegram Premium giveaways and `prize_star_count / 500` times for one year for Telegram Star giveaways.

Field	Type	Description
source	String	Source of the boost, always "giveaway"
giveaway_message_id	Integer	Identifier of a message in the chat with the giveaway; the message could have been deleted already. May be 0 if the message isn't sent yet.
user	User	<i>Optional.</i> User that won the prize in the giveaway if any; for Telegram Premium giveaways only
prize_star_count	Integer	<i>Optional.</i> The number of Telegram Stars to be split between giveaway winners; for Telegram Star giveaways only
is_unclaimed	True	<i>Optional.</i> <code>True</code> , if the giveaway was completed, but there was no user to win the prize

ChatBoost

This object contains information about a chat boost.

Field	Type	Description
boost_id	String	Unique identifier of the boost
add_date	Integer	Point in time (Unix timestamp) when the chat was boosted
expiration_date	Integer	Point in time (Unix timestamp) when the boost will automatically expire, unless the booster's Telegram Premium subscription is prolonged
source	ChatBoostSource	Source of the added boost

ChatBoostUpdated

This object represents a boost added to a chat or changed.

Field	Type	Description
chat	Chat	Chat which was boosted
boost	ChatBoost	Information about the chat boost

ChatBoostRemoved

This object represents a boost removed from a chat.

Field	Type	Description
chat	Chat	Chat which was boosted
boost_id	String	Unique identifier of the boost
remove_date	Integer	Point in time (Unix timestamp) when the boost was removed
source	ChatBoostSource	Source of the removed boost

UserChatBoosts

This object represents a list of boosts added to a chat by a user.

Field	Type	Description
boosts	Array of ChatBoost	The list of boosts added to the chat by the user

BusinessBotRights

Represents the rights of a business bot.

Field	Type	Description
can_reply	True	<i>Optional.</i> <code>True</code> , if the bot can send and edit messages in the private chats that had incoming messages in the last 24 hours
can_read_messages	True	<i>Optional.</i> <code>True</code> , if the bot can mark incoming private messages as read
can_delete_sent_messages	True	<i>Optional.</i> <code>True</code> , if the bot can delete messages sent by the bot
can_delete_all_messages	True	<i>Optional.</i> <code>True</code> , if the bot can delete all private messages in managed chats
can_edit_name	True	<i>Optional.</i> <code>True</code> , if the bot can edit the first and last name of the business account
can_edit_bio	True	<i>Optional.</i> <code>True</code> , if the bot can edit the bio of the business account
can_edit_profile_photo	True	<i>Optional.</i> <code>True</code> , if the bot can edit the profile photo of the business account
can_edit_username	True	<i>Optional.</i> <code>True</code> , if the bot can edit the username of the business account

Field	Type	Description
can_change_gift_settings	True	<i>Optional.</i> True, if the bot can change the privacy settings pertaining to gifts for the business account
can_view_gifts_and_stars	True	<i>Optional.</i> True, if the bot can view gifts and the amount of Telegram Stars owned by the business account
can_convert_gifts_to_stars	True	<i>Optional.</i> True, if the bot can convert regular gifts owned by the business account to Telegram Stars
can_transfer_and_upgrade_gifts	True	<i>Optional.</i> True, if the bot can transfer and upgrade gifts owned by the business account
can_transfer_stars	True	<i>Optional.</i> True, if the bot can transfer Telegram Stars received by the business account to its own account, or use them to upgrade and transfer gifts
can_manage_stories	True	<i>Optional.</i> True, if the bot can post, edit and delete stories on behalf of the business account

BusinessConnection

Describes the connection of the bot with a business account.

Field	Type	Description
id	String	Unique identifier of the business connection
user	User	Business account user that created the business connection
user_chat_id	Integer	Identifier of a private chat with the user who created the business connection. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a 64-bit integer or double-precision float type are safe for storing this identifier.
date	Integer	Date the connection was established in Unix time
rights	BusinessBotRights	<i>Optional.</i> Rights of the business bot
is_enabled	Boolean	<i>True</i> , if the connection is active

BusinessMessagesDeleted

This object is received when messages are deleted from a connected business account.

Field	Type	Description
business_connection_id	String	Unique identifier of the business connection
chat	Chat	Information about a chat in the business account. The bot may not have access to the chat or the corresponding user.
message_ids	Array of Integer	The list of identifiers of deleted messages in the chat of the business account

ResponseParameters

Describes why a request was unsuccessful.

Field	Type	Description
migrate_to_chat_id	Integer	<i>Optional.</i> The group has been migrated to a supergroup with the specified identifier. This number may have more than 32 significant bits and some programming languages may have difficulty/silent defects in interpreting it. But it has at most 52 significant bits, so a signed 64-bit integer or double-precision float type are safe for storing this identifier.
retry_after	Integer	<i>Optional.</i> In case of exceeding flood control, the number of seconds left to wait before the request can be repeated

InputMedia

This object represents the content of a media message to be sent. It should be one of

- [InputMediaAnimation](#)
- [InputMediaDocument](#)
- [InputMediaAudio](#)
- [InputMediaPhoto](#)
- [InputMediaVideo](#)

InputMediaPhoto

Represents a photo to be sent.

Field	Type	Description
type	String	Type of the result, must be <i>photo</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
caption	String	<i>Optional.</i> Caption of the photo to be sent, 0–1024 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the photo caption. See formatting options for more details.
caption_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <i>parse_mode</i>
show_caption_above_media	Boolean	<i>Optional.</i> Pass <i>True</i> , if the caption must be shown above the message media
hasSpoiler	Boolean	<i>Optional.</i> Pass <i>True</i> if the photo needs to be covered with a spoiler animation

InputMediaVideo

Represents a video to be sent.

Field	Type	Description
type	String	Type of the result, must be <i>video</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
thumbnail	String	<i>Optional.</i> Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files »
cover	String	<i>Optional.</i> Cover for the video in the message. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
start_timestamp	Integer	<i>Optional.</i> Start timestamp for the video in the message
caption	String	<i>Optional.</i> Caption of the video to be sent, 0–1024 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the video caption. See formatting options for more details.
caption_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <i>parse_mode</i>
show_caption_above_media	Boolean	<i>Optional.</i> Pass <i>True</i> , if the caption must be shown above the message media
width	Integer	<i>Optional.</i> Video width
height	Integer	<i>Optional.</i> Video height
duration	Integer	<i>Optional.</i> Video duration in seconds
supports_streaming	Boolean	<i>Optional.</i> Pass <i>True</i> if the uploaded video is suitable for streaming
hasSpoiler	Boolean	<i>Optional.</i> Pass <i>True</i> if the video needs to be covered with a spoiler animation

InputMediaAnimation

Represents an animation file (GIF or H.264/MPEG-4 AVC video without sound) to be sent.

Field	Type	Description
type	String	Type of the result, must be <i>animation</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
thumbnail	String	<i>Optional.</i> Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files »
caption	String	<i>Optional.</i> Caption of the animation to be sent, 0–1024 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the animation caption. See formatting options for more details.
caption_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <i>parse_mode</i>
show_caption_above_media	Boolean	<i>Optional.</i> Pass <i>True</i> , if the caption must be shown above the message media
width	Integer	<i>Optional.</i> Animation width
height	Integer	<i>Optional.</i> Animation height
duration	Integer	<i>Optional.</i> Animation duration in seconds
hasSpoiler	Boolean	<i>Optional.</i> Pass <i>True</i> if the animation needs to be covered with a spoiler animation

InputMediaAudio

Represents an audio file to be treated as music to be sent.

Field	Type	Description
type	String	Type of the result, must be <i>audio</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
thumbnail	String	<i>Optional.</i> Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using

Field	Type	Description
		multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files »
caption	String	<i>Optional.</i> Caption of the audio to be sent, 0–1024 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the audio caption. See formatting options for more details.
caption_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <i>parse_mode</i>
duration	Integer	<i>Optional.</i> Duration of the audio in seconds
performer	String	<i>Optional.</i> Performer of the audio
title	String	<i>Optional.</i> Title of the audio

InputMediaDocument

Represents a general file to be sent.

Field	Type	Description
type	String	Type of the result, must be <i>document</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
thumbnail	String	<i>Optional.</i> Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files »
caption	String	<i>Optional.</i> Caption of the document to be sent, 0–1024 characters after entities parsing
parse_mode	String	<i>Optional.</i> Mode for parsing entities in the document caption. See formatting options for more details.
caption_entities	Array of MessageEntity	<i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <i>parse_mode</i>
disable_content_type_detection	Boolean	<i>Optional.</i> Disables automatic server-side content type detection for files uploaded using multipart/form-data. Always <i>True</i> , if the document is sent as part of an album.

InputFile

This object represents the contents of a file to be uploaded. Must be posted using multipart/form-data in the usual way that files are uploaded via the browser.

InputPaidMedia

This object describes the paid media to be sent. Currently, it can be one of

- [InputPaidMediaPhoto](#)
- [InputPaidMediaVideo](#)

InputPaidMediaPhoto

The paid media to send is a photo.

Field	Type	Description
type	String	Type of the media, must be <i>photo</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »

InputPaidMediaVideo

The paid media to send is a video.

Field	Type	Description
type	String	Type of the media, must be <i>video</i>
media	String	File to send. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
thumbnail	String	<i>Optional.</i> Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files »
cover	String	<i>Optional.</i> Cover for the video in the message. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files »
start_timestamp	Integer	<i>Optional.</i> Start timestamp for the video in the message
width	Integer	<i>Optional.</i> Video width
height	Integer	<i>Optional.</i> Video height

Field	Type	Description
duration	Integer	<i>Optional.</i> Video duration in seconds
supports_streaming	Boolean	<i>Optional.</i> Pass <code>True</code> if the uploaded video is suitable for streaming

InputProfilePhoto

This object describes a profile photo to set. Currently, it can be one of

- [InputProfilePhotoStatic](#)
- [InputProfilePhotoAnimated](#)

InputProfilePhotoStatic

A static profile photo in the .JPG format.

Field	Type	Description
type	String	Type of the profile photo, must be <code>static</code>
photo	String	The static profile photo. Profile photos can't be reused and can only be uploaded as a new file, so you can pass "attach://<file_attach_name>" if the photo was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files

InputProfilePhotoAnimated

An animated profile photo in the MPEG4 format.

Field	Type	Description
type	String	Type of the profile photo, must be <code>animated</code>
animation	String	The animated profile photo. Profile photos can't be reused and can only be uploaded as a new file, so you can pass "attach://<file_attach_name>" if the photo was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files
main_frame_timestamp	Float	<i>Optional.</i> Timestamp in seconds of the frame that will be used as the static profile photo. Defaults to 0.0.

InputStoryContent

This object describes the content of a story to post. Currently, it can be one of

- [InputStoryContentPhoto](#)
- [InputStoryContentVideo](#)

InputStoryContentPhoto

Describes a photo to post as a story.

Field	Type	Description
type	String	Type of the content, must be <code>photo</code>
photo	String	The photo to post as a story. The photo must be of the size 1080x1920 and must not exceed 10 MB. The photo can't be reused and can only be uploaded as a new file, so you can pass "attach://<file_attach_name>" if the photo was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files

InputStoryContentVideo

Describes a video to post as a story.

Field	Type	Description
type	String	Type of the content, must be <code>video</code>
video	String	The video to post as a story. The video must be of the size 720x1280, streamable, encoded with H.265 codec, with key frames added each second in the MPEG4 format, and must not exceed 30 MB. The video can't be reused and can only be uploaded as a new file, so you can pass "attach://<file_attach_name>" if the video was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files
duration	Float	<i>Optional.</i> Precise duration of the video in seconds; 0–60
cover_frame_timestamp	Float	<i>Optional.</i> Timestamp in seconds of the frame that will be used as the static cover for the story. Defaults to 0.0.
is_animation	Boolean	<i>Optional.</i> Pass <code>True</code> if the video has no sound

Sending files

There are three ways to send files (photos, stickers, audio, media, etc.):

1. If the file is already stored somewhere on the Telegram servers, you don't need to reupload it: each file object has a `file_id` field, simply pass this `file_id` as a parameter instead of uploading. There are **no limits** for files sent this way.
2. Provide Telegram with an HTTP URL for the file to be sent. Telegram will download and send the file. 5 MB max size for photos and 20 MB max for other types of content.
3. Post the file using multipart/form-data in the usual way that files are uploaded via the browser. 10 MB max size for photos, 50 MB for other files.

Sending by file_id

- It is not possible to change the file type when resending by `file_id`. I.e. a `video` can't be sent as a `photo`, a `photo` can't be sent as a `document`, etc.
- It is not possible to resend thumbnails.
- Resending a photo by `file_id` will send all of its `sizes`.
- `file_id` is unique for each individual bot and can't be transferred from one bot to another.
- `file_id` uniquely identifies a file, but a file can have different valid `file_ids` even for the same bot.

Sending by URL

- When sending by URL the target file must have the correct MIME type (e.g., `audio/mpeg` for `sendAudio`, etc.).
- In `sendDocument`, sending by URL will currently only work for `.PDF` and `.ZIP` files.
- To use `sendVoice`, the file must have the type `audio/ogg` and be no more than 1MB in size. 1–20MB voice notes will be sent as files.

- Other configurations may work but we can't guarantee that they will.

Accent colors

Colors with identifiers 0 (red), 1 (orange), 2 (purple/violet), 3 (green), 4 (cyan), 5 (blue), 6 (pink) can be customized by app themes. Additionally, the following colors in RGB format are currently in use.

Color identifier	Light colors	Dark colors
7	E15052 F9AE63	FF9380 992F37
8	E0802B FAC534	ECB04E C35714
9	A05FF3 F48FFF	C697FF 5E31C8
10	27A910 A7DC57	A7EB6E 167E2D
11	27ACCE 82E8D6	40D8D0 045C7F
12	3391D4 7DD3F0	52BFFF 0B5494
13	DD4371 FFBE9F	FF86A6 8E366E
14	247BED F04856 FFFFFF	3FA2FE E5424F FFFFFF
15	D67722 1EA011 FFFFFF	FF905E 32A527 FFFFFF
16	179E42 E84A3F FFFFFF	66D364 D5444F FFFFFF
17	2894AF 6FC456 FFFFFF	22BCE2 3DA240 FFFFFF
18	0C9AB3 FFAD95 FFE6B5	22BCE2 FF9778 FFDA6B
19	7757D6 F79610 FFDE8E	9791FF F2731D FFDB59
20	1585CF F2AB1D FFFFFF	3DA6EB EEA51D FFFFFF

Profile accent colors

Currently, the following colors in RGB format are in use for profile backgrounds.

Color identifier	Light colors	Dark colors
0	BA5650	9C4540
1	C27C3E	945E2C
2	956AC8	715099
3	49A355	33713B
4	3E97AD	387E87
5	5A8FBB	477194
6	B85378	944763
7	7F8B95	435261
8	C9565D D97C57	994343 AC583E
9	CF7244 CC9433	8F552F A17232
10	9662D4 B966B6	634691 9250A2
11	3D9755 89A650	296A43 5F8F44
12	3D95BA 50AD98	306C7C 3E987E
13	538BC2 4DA8BD	38618C 458BA1
14	B04F74 D1666D	884160 A65259
15	637482 7B8A97	53606E 384654

Inline mode objects

Objects and methods used in the inline mode are described in the [Inline mode section](#).

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 authentication 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
business_connection_id	String	Optional	Unique identifier of the business connection on behalf of which the message will be sent
chat_id	Integer or String	Yes	Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>)
message_thread_id	Integer	Optional	Unique identifier for the target message thread (topic) of the forum; for forum supergroups only
direct_messages_topic_id	Integer	Optional	Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat
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	A JSON-serialized list of special entities that appear in message text, which can be specified instead of <code>parse_mode</code>
link_preview_options	LinkPreviewOptions	Optional	Link preview generation options for the message
disable_notification	Boolean	Optional	Sends the message silently . Users will receive a notification with no sound.
protect_content	Boolean	Optional	Protects the contents of the sent message from forwarding and saving
allow_paid_broadcast	Boolean	Optional	Pass <i>True</i> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance
message_effect_id	String	Optional	Unique identifier of the message effect to be added to the message; for private chats only
suggested_post_parameters	SuggestedPostParameters	Optional	A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined.
reply_parameters	ReplyParameters	Optional	Description of the message to reply to
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 a reply keyboard or to force a reply from the user

Formatting options

The Bot API supports basic formatting for messages. You can use **bold**, **italic**, **underlined**, **strikethrough**, **spoiler** text, block quotations as well as inline links and pre-formatted code in your bots' messages. Telegram clients will render them accordingly. You can specify text entities directly, or use [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).

Message entities can be nested, providing following restrictions are met:

- If two entities have common characters, then one of them is fully contained inside another.
- **bold**, **italic**, **underline**, **strikethrough**, and **spoiler** entities can contain and can be part of any other entities, except **pre** and **code**.
- **blockquote** and **expandable_blockquote** entities can't be nested.
- All other entities can't contain each other.

Links `tg://user?id=<user_id>` can be used to mention a user by their identifier without using a username. Please note:

- These links will work **only** if they are used inside an inline link or in an inline keyboard button. For example, they will not work, when used in a message text.
- Unless the user is a member of the chat where they were mentioned, these mentions are only guaranteed to work if the user has contacted the bot in private in the past or has sent a callback query to the bot via an inline button and doesn't have Forwarded Messages privacy enabled for the bot.

You can find the list of programming and markup languages for which syntax highlighting is supported at [libprisma#supported-languages](#).

MarkdownV2 style

To use this mode, pass `MarkdownV2` in the `parse_mode` field. Use the following syntax in your message:

```
*bold \*text*
_italic \*text_
__underline__
~~strikethrough~~
||spoiler||

**bold _italic bold ~italic bold strikethrough ||italic bold strikethrough spoiler||~ __underline italic bold__ bold*
[inline URL](http://www.example.com/)
[inline mention of a user](tg://user?id=123456789)
![👍](tg://emoji?id=5368324170671202286)
`inline fixed-width code`
```
pre-formatted fixed-width code block
```
```python
pre-formatted fixed-width code block written in the Python programming language
```
>Block quotation started
```

```
>Block quotation continued
>Block quotation continued
>Block quotation continued
>The last line of the block quotation
**>The expandable block quotation started right after the previous block quotation
>It is separated from the previous block quotation by an empty bold entity
>Expandable block quotation continued
>Hidden by default part of the expandable block quotation started
>Expandable block quotation continued
>The last line of the expandable block quotation with the expandability mark||
```

Please note:

- Any character with code between 1 and 126 inclusively can be escaped anywhere with a preceding '\` character, in which case it is treated as an ordinary character and not a part of the markup. This implies that '\` character usually must be escaped with a preceding '\` character.
- Inside `pre` and `code` entities, all '*' and '\` characters must be escaped with a preceding '\` character.
- Inside the (...) part of the inline link and custom emoji definition, all ')' and '\` must be escaped with a preceding '\` character.
- In all other places characters '.', ',', '[', ']', '(', ')', '^', '!', '>', '#', '+', '=', '|', '!', '{', '}', '!', '!" must be escaped with the preceding character '\`.
- In case of ambiguity between *italic* and underline entities _ is always greedily treated from left to right as beginning or end of an underline entity, so instead of italic underline use italic underline**, adding an empty bold entity as a separator.
- A valid emoji must be provided as an alternative value for the custom emoji. The emoji will be shown instead of the custom emoji in places where a custom emoji cannot be displayed (e.g., system notifications) or if the message is forwarded by a non-premium user. It is recommended to use the emoji from the `emoji` field of the custom emoji sticker.
- Custom emoji entities can only be used by bots that purchased additional usernames on [Fragment](#).

HTML style

To use this mode, pass `HTML` in the `parse_mode` field. The following tags are currently supported:

```
<b>bold</b>, <strong>bold</strong>
<i>italic</i>, <em>italic</em>
<u>underline</u>, <ins>underline</ins>
<s>strikethrough</s>, <strike>strikethrough</strike>, <del>strikethrough</del>
<span class="tg-spoiler">spoiler</span>, <tg-spoiler>spoiler</tg-spoiler>
<b>bold <i>italic bold <s>italic bold strikethrough <span class="tg-spoiler">italic bold strikethrough spoiler</span></s> <u>underline italic bold</u></i> bold</b>
<a href="http://www.example.com/">inline URL</a>
<a href="tg://user?id=123456789">inline mention of a user</a>
<tg-emoji emoji-id="5368324170671202286">👉</tg-emoji>
<code>inline fixed-width code</code>
<pre>pre-formatted fixed-width code block</pre>
<pre><code class="language-python">pre-formatted fixed-width code block written in the Python programming language</code></pre>
<blockquote>Block quotation started\nBlock quotation continued\nThe last line of the block quotation</blockquote>
<blockquote expandable>Expandable block quotation started\nExpandable block quotation continued\nExpandable block quotation continued\nExpandable block quotation continued\nHidden by default part of the block quotation started\nExpandable block quotation continued\nThe last line of the block quotation</blockquote>
```

Please note:

- Only the tags mentioned above are currently supported.
- All <, > and & symbols that are not a part of a tag or an HTML entity must be replaced with the corresponding HTML entities (< with `<`, > with `>` and & with `&`).
- All numerical HTML entities are supported.
- The API currently supports only the following named HTML entities: `<`, `>`, `&` and `"`.
- Use nested `pre` and `code` tags, to define programming language for `pre` entity.
- Programming language can't be specified for standalone `code` tags.
- A valid emoji must be used as the content of the `tg-emoji` tag. The emoji will be shown instead of the custom emoji in places where a custom emoji cannot be displayed (e.g., system notifications) or if the message is forwarded by a non-premium user. It is recommended to use the emoji from the `emoji` field of the custom emoji sticker.
- Custom emoji entities can only be used by bots that purchased additional usernames on [Fragment](#).

Markdown style

This is a legacy mode, retained for backward compatibility. To use this mode, pass `Markdown` in the `parse_mode` field. Use the following syntax in your message:

```
*bold text*
_italic text_
[inline URL](http://www.example.com/)
[inline mention of a user](tg://user?id=123456789)
`inline fixed-width code`
```
pre-formatted fixed-width code block
```
```python
pre-formatted fixed-width code block written in the Python programming language
```

```

Please note:

- Entities must not be nested, use parse mode [MarkdownV2](#) instead.
- There is no way to specify "underline", "strikethrough", "spoiler", "blockquote", "expandable_blockquote" and "custom_emoji" entities, use parse mode [MarkdownV2](#) instead.
- To escape characters '_', '*', '^', '!' outside of an entity, prepend the character '\` before them.
- Escaping inside entities is not allowed, so entity must be closed first and reopened again: use snake_\case_ for italic `snake_case` and `*2***2=4*` for bold `2*2=4`.

Paid Broadcasts

By default, all bots are able to broadcast up to 30 messages per second to their users. Developers can increase this limit by enabling [Paid Broadcasts](#) in [@Botfather](#) – allowing their bot to broadcast up to 1000 messages per second.

Each message broadcasted over the free amount of 30 messages per second incurs a cost of 0.1 Stars per message, paid with Telegram Stars from the bot's balance. In order to use this feature, a bot must have at least 10,000 Stars on its balance.

Bots with increased limits are only charged for messages that are broadcasted successfully.

forwardMessage

Use this method to forward messages of any kind. Service messages and messages with protected content can't be forwarded. On success, the sent `Message` is returned.

| Parameter | Type | Required | Description |
|--------------------------------|-------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>message_thread_id</code> | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |

| Parameter | Type | Required | Description |
|---------------------------|---|----------|--|
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be forwarded; required if the message is forwarded to a direct messages chat |
| from_chat_id | Integer or String | Yes | Unique identifier for the chat where the original message was sent (or channel username in the format <code>@channelusername</code>) |
| video_start_timestamp | Integer | Optional | New start timestamp for the forwarded video in the message |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the forwarded message from forwarding and saving |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only |
| message_id | Integer | Yes | Message identifier in the chat specified in <code>from_chat_id</code> |

forwardMessages

Use this method to forward multiple messages of any kind. If some of the specified messages can't be found or forwarded, they are skipped. Service messages and messages with protected content can't be forwarded. Album grouping is kept for forwarded messages. On success, an array of [MessageId](#) of the sent messages 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>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the messages will be forwarded; required if the messages are forwarded to a direct messages chat |
| from_chat_id | Integer or String | Yes | Unique identifier for the chat where the original messages were sent (or channel username in the format <code>@channelusername</code>) |
| message_ids | Array of Integer | Yes | A JSON-serialized list of 1–100 identifiers of messages in the chat <code>from_chat_id</code> to forward. The identifiers must be specified in a strictly increasing order. |
| disable_notification | Boolean | Optional | Sends the messages <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the forwarded messages from forwarding and saving |

copyMessage

Use this method to copy messages of any kind. Service messages, paid media messages, giveaway messages, giveaway winners messages, and invoice messages can't be copied. A quiz poll can be copied only if the value of the field `correct_option_id` is known to the bot. The method is analogous to the method [forwardMessage](#), but the copied message doesn't have a link to the original message. Returns the [MessageId](#) of the sent message on success.

| 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>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| from_chat_id | Integer or String | Yes | Unique identifier for the chat where the original message was sent (or channel username in the format <code>@channelusername</code>) |
| message_id | Integer | Yes | Message identifier in the chat specified in <code>from_chat_id</code> |
| video_start_timestamp | Integer | Optional | New start timestamp for the copied video in the message |
| caption | String | Optional | New caption for media, 0–1024 characters after entities parsing. If not specified, the original caption is kept |
| parse_mode | String | Optional | Mode for parsing entities in the new caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the new caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media. Ignored if a new caption isn't specified. |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |

| Parameter | Type | Required | Description |
|------------------|--|----------|---|
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

copyMessages

Use this method to copy messages of any kind. If some of the specified messages can't be found or copied, they are skipped. Service messages, paid media messages, giveaway messages, giveaway winners messages, and invoice messages can't be copied. A quiz poll can be copied only if the value of the field `correct_option_id` is known to the bot. The method is analogous to the method [forwardMessages](#), but the copied messages don't have a link to the original message. Album grouping is kept for copied messages. On success, an array of [MessageId](#) of the sent messages 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 @channelusername) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the messages will be sent; required if the messages are sent to a direct messages chat |
| from_chat_id | Integer or String | Yes | Unique identifier for the chat where the original messages were sent (or channel username in the format @channelusername) |
| message_ids | Array of Integer | Yes | A JSON-serialized list of 1-100 identifiers of messages in the chat <code>from_chat_id</code> to copy. The identifiers must be specified in a strictly increasing order. |
| disable_notification | Boolean | Optional | Sends the messages silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent messages from forwarding and saving |
| remove_caption | Boolean | Optional | Pass <code>True</code> to copy the messages without their captions |

sendPhoto

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

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format @channelusername) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| photo | InputFile or String | Yes | Photo to send. Pass a file_id as String to send a photo that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a photo from the Internet, or upload a new photo using multipart/form-data. The photo must be at most 10 MB in size. The photo's width and height must not exceed 10000 in total. Width and height ratio must be at most 20. More information on Sending Files » |
| caption | String | Optional | Photo caption (may also be used when resending photos by <code>file_id</code>), 0-1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the photo caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media |
| hasSpoiler | Boolean | Optional | Pass <code>True</code> if the photo needs to be covered with a spoiler animation |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendAudio

Use this method to send audio files, if you want Telegram clients to display them in the music player. Your audio must be in the .MP3 or .M4A format. On success, the sent [Message](#) is returned. Bots can currently send audio files of up to 50 MB in size, this limit may be changed in the future.

For sending voice messages, use the [sendVoice](#) method instead.

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| audio | InputFile or String | Yes | Audio file to send. Pass a file_id as String to send an audio file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get an audio file from the Internet, or upload a new one using multipart/form-data. More information on Sending Files » |
| caption | String | Optional | Audio caption, 0-1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the audio caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| duration | Integer | Optional | Duration of the audio in seconds |
| performer | String | Optional | Performer |
| title | String | Optional | Track name |
| thumbnail | InputFile or String | Optional | Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files » |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendDocument

Use this method to send general files. On success, the sent [Message](#) is returned. Bots can currently send files of any type of up to 50 MB in size, this limit may be changed in the future.

| Parameter | Type | Required | Description |
|--------------------------|-------------------------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| document | InputFile or String | Yes | File to send. Pass a file_id as String to send a file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a file from the Internet, or upload a new one using multipart/form-data. More information on Sending Files » |
| thumbnail | InputFile or String | Optional | Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files » |
| caption | String | Optional | Document caption (may also be used when resending documents by <code>file_id</code>), 0-1024 characters after entities parsing |

| Parameter | Type | Required | Description |
|--------------------------------|--|----------|--|
| parse_mode | String | Optional | Mode for parsing entities in the document caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| disable_content_type_detection | Boolean | Optional | Disables automatic server-side content type detection for files uploaded using multipart/form-data |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendVideo

Use this method to send video files. Telegram clients support MPEG4 videos (other formats may be sent as [Document](#)). On success, the sent [Message](#) is returned. Bots can currently send video files of up to 50 MB in size, this limit may be changed in the future.

| Parameter | Type | Required | Description |
|--------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| video | InputFile or String | Yes | Video to send. Pass a file_id as String to send a video that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a video from the Internet, or upload a new video using multipart/form-data. More information on Sending Files » |
| duration | Integer | Optional | Duration of sent video in seconds |
| width | Integer | Optional | Video width |
| height | Integer | Optional | Video height |
| thumbnail | InputFile or String | Optional | Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files » |
| cover | InputFile or String | Optional | Cover for the video in the message. Pass a file_id to send a file that exists on the Telegram servers (recommended), pass an HTTP URL for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new one using multipart/form-data under <file_attach_name> name. More information on Sending Files » |
| start_timestamp | Integer | Optional | Start timestamp for the video in the message |
| caption | String | Optional | Video caption (may also be used when resending videos by <code>file_id</code>), 0-1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the video caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media |
| hasSpoiler | Boolean | Optional | Pass <code>True</code> if the video needs to be covered with a spoiler animation |
| supports_streaming | Boolean | Optional | Pass <code>True</code> if the uploaded video is suitable for streaming |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendAnimation

Use this method to send animation files (GIF or H.264/MPEG-4 AVC video without sound). On success, the sent [Message](#) is returned. Bots can currently send animation files of up to 50 MB in size, this limit may be changed in the future.

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| animation | InputFile or String | Yes | Animation to send. Pass a <code>file_id</code> as String to send an animation that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get an animation from the Internet, or upload a new animation using multipart/form-data. More information on Sending Files » |
| duration | Integer | Optional | Duration of sent animation in seconds |
| width | Integer | Optional | Animation width |
| height | Integer | Optional | Animation height |
| thumbnail | InputFile or String | Optional | Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files » |
| caption | String | Optional | Animation caption (may also be used when resending animation by <code>file_id</code>), 0–1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the animation caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media |
| hasSpoiler | Boolean | Optional | Pass <code>True</code> if the animation needs to be covered with a spoiler animation |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendVoice

Use this method to send audio files, if you want Telegram clients to display the file as a playable voice message. For this to work, your audio must be in an .OGG file encoded with OPUS, or in .MP3 format, or in .M4A format (other formats may be sent as [Audio](#) or [Document](#)). On success, the sent [Message](#) is returned. Bots can currently send voice messages of up to 50 MB in size, this limit may be changed in the future.

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| voice | InputFile or String | Yes | Audio file to send. Pass a file_id as String to send a file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a file from the Internet, or upload a new one using multipart/form-data. More information on Sending Files » |
| caption | String | Optional | Voice message caption, 0–1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the voice message caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| duration | Integer | Optional | Duration of the voice message in seconds |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendVideoNote

As of v.4.0, Telegram clients support rounded square MPEG4 videos of up to 1 minute long. Use this method to send video messages. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|---------------------------|-------------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| video_note | InputFile or String | Yes | Video note to send. Pass a file_id as String to send a video note that exists on the Telegram servers (recommended) or upload a new video using multipart/form-data. More information on Sending Files » . Sending video notes by a URL is currently unsupported |
| duration | Integer | Optional | Duration of sent video in seconds |
| length | Integer | Optional | Video width and height, i.e. diameter of the video message |
| thumbnail | InputFile or String | Optional | Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>. More information on Sending Files » |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |

| Parameter | Type | Required | Description |
|------------------|--|----------|---|
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendPaidMedia

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

| Parameter | Type | Required | Description |
|---------------------------|--|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>). If the chat is a channel, all Telegram Star proceeds from this media will be credited to the chat's balance. Otherwise, they will be credited to the bot's balance. |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| star_count | Integer | Yes | The number of Telegram Stars that must be paid to buy access to the media; 1–10000 |
| media | Array of InputPaidMedia | Yes | A JSON-serialized array describing the media to be sent; up to 10 items |
| payload | String | Optional | Bot-defined paid media payload, 0–128 bytes. This will not be displayed to the user, use it for your internal processes. |
| caption | String | Optional | Media caption, 0–1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the media caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendMediaGroup

Use this method to send a group of photos, videos, documents or audios as an album. Documents and audio files can be only grouped in an album with messages of the same type. On success, an array of [Message](#) objects that were sent is returned.

| Parameter | Type | Required | Description |
|--------------------------|---|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the messages will be sent; required if the messages are sent to a direct messages chat |
| media | Array of InputMediaAudio , InputMediaDocument , InputMediaPhoto and InputMediaVideo | Yes | A JSON-serialized array describing messages to be sent, must include 2–10 items |
| disable_notification | Boolean | Optional | Sends messages silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent messages from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |

| Parameter | Type | Required | Description |
|---|--|----------|--|
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| sendLocation | | | |
| Use this method to send point on the map. On success, the sent Message is returned. | | | |
| Parameter | Type | Required | Description |
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| latitude | Float | Yes | Latitude of the location |
| longitude | Float | Yes | Longitude of the location |
| horizontal_accuracy | Float | Optional | The radius of uncertainty for the location, measured in meters; 0-1500 |
| live_period | Integer | Optional | Period in seconds during which the location will be updated (see Live Locations , should be between 60 and 86400, or 0xFFFFFFFF for live locations that can be edited indefinitely) |
| heading | Integer | Optional | For live locations, a direction in which the user is moving, in degrees. Must be between 1 and 360 if specified. |
| proximity_alert_radius | Integer | Optional | For live locations, a maximum distance for proximity alerts about approaching another chat member, in meters. Must be between 1 and 100000 if specified. |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendVenueUse this method to send information about a venue. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|--------------------------|-------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| latitude | Float | Yes | Latitude of the venue |
| longitude | Float | Yes | Longitude of the venue |
| title | String | Yes | Name of the venue |
| address | String | Yes | Address of the venue |
| foursquare_id | String | Optional | Foursquare identifier of the venue |
| foursquare_type | String | Optional | Foursquare type of the venue, if known. (For example, "arts_entertainment/default", "arts_entertainment/aquarium" or "food/icecream".) |

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| google_place_id | String | Optional | Google Places identifier of the venue |
| google_place_type | String | Optional | Google Places type of the venue. (See supported types .) |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendContact

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

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| phone_number | String | Yes | Contact's phone number |
| first_name | String | Yes | Contact's first name |
| last_name | String | Optional | Contact's last name |
| vcard | String | Optional | Additional data about the contact in the form of a vCard , 0–2048 bytes |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendPoll

Use this method to send a native poll. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>). Polls can't be sent to channel direct messages chats. |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| question | String | Yes | Poll question, 1–300 characters |
| question_parse_mode | String | Optional | Mode for parsing entities in the question. See formatting options for more details. Currently, only custom emoji entities are allowed |

| Parameter | Type | Required | Description |
|-------------------------|--|----------|---|
| question_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the poll question. It can be specified instead of <code>question_parse_mode</code> |
| options | Array of InputPollOption | Yes | A JSON-serialized list of 2–12 answer options |
| is_anonymous | Boolean | Optional | <code>True</code> , if the poll needs to be anonymous, defaults to <code>True</code> |
| type | String | Optional | Poll type, “quiz” or “regular”, defaults to “regular” |
| allows_multiple_answers | Boolean | Optional | <code>True</code> , if the poll allows multiple answers, ignored for polls in quiz mode, defaults to <code>False</code> |
| correct_option_id | Integer | Optional | 0-based identifier of the correct answer option, required for polls in quiz mode |
| explanation | String | Optional | Text that is shown when a user chooses an incorrect answer or taps on the lamp icon in a quiz-style poll, 0–200 characters with at most 2 line feeds after entities parsing |
| explanation_parse_mode | String | Optional | Mode for parsing entities in the explanation. See formatting options for more details. |
| explanation_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the poll explanation. It can be specified instead of <code>explanation_parse_mode</code> |
| open_period | Integer | Optional | Amount of time in seconds the poll will be active after creation, 5–600. Can’t be used together with <code>close_date</code> . |
| close_date | Integer | Optional | Point in time (Unix timestamp) when the poll will be automatically closed. Must be at least 5 and no more than 600 seconds in the future. Can’t be used together with <code>open_period</code> . |
| is_closed | Boolean | Optional | Pass <code>True</code> if the poll needs to be immediately closed. This can be useful for poll preview. |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot’s balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendChecklist

Use this method to send a checklist on behalf of a connected business account. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer | Yes | Unique identifier for the target chat |
| checklist | InputChecklist | Yes | A JSON-serialized object for the checklist to send |
| disable_notification | Boolean | Optional | Sends the message silently. Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message |
| reply_parameters | ReplyParameters | Optional | A JSON-serialized object for description of the message to reply to |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard |

sendDice

Use this method to send an animated emoji that will display a random value. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|--------------------------|-------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| emoji | String | Optional | Emoji on which the dice throw animation is based. Currently, must be one of “🏀”, “🎯”, “🎳”, “⚽”, “🎰”, or “🎰”. Dice can have values 1–6 for “🏀”, “🎯” and “🎳”, values 1–5 for “🎳” and “⚽”, and values 1–64 for “🎰”. Defaults to “🏀” |
| disable_notification | Boolean | Optional | Sends the message <code>silently</code> . Users will receive a notification with no sound. |

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

sendChatAction

Use this method when you need to tell the user that something is happening on the bot's side. The status is set for 5 seconds or less (when a message arrives from your bot, Telegram clients clear its typing status). Returns `True` on success.

Example: The [ImageBot](#) needs some time to process a request and upload the image. Instead of sending a text message along the lines of "Retrieving image, please wait...", the bot may use [sendChatAction](#) with `action = upload_photo`. The user will see a "sending photo" status for the bot.

We only recommend using this method when a response from the bot will take a **noticeable** amount of time to arrive.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the action will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>). Channel chats and channel direct messages chats aren't supported. |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread; for supergroups only |
| action | String | Yes | Type of action to broadcast. Choose one, depending on what the user is about to receive: <code>typing</code> for text messages , <code>upload_photo</code> for photos , <code>record_video</code> or <code>upload_video</code> for videos , <code>record_voice</code> or <code>upload_voice</code> for voice notes , <code>upload_document</code> for general files , <code>choose_sticker</code> for stickers , <code>find_location</code> for location data , <code>record_video_note</code> or <code>upload_video_note</code> for video notes . |

setMessageReaction

Use this method to change the chosen reactions on a message. Service messages of some types can't be reacted to. Automatically forwarded messages from a channel to its discussion group have the same available reactions as messages in the channel. Bots can't use paid reactions. Returns `True` on success.

| 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>) |
| message_id | Integer | Yes | Identifier of the target message. If the message belongs to a media group, the reaction is set to the first non-deleted message in the group instead. |
| reaction | Array of ReactionType | Optional | A JSON-serialized list of reaction types to set on the message. Currently, as non-premium users, bots can set up to one reaction per message. A custom emoji reaction can be used if it is either already present on the message or explicitly allowed by chat administrators. Paid reactions can't be used by bots. |
| is_big | Boolean | Optional | Pass <code>True</code> to set the reaction with a big animation |

getUserProfilePhotos

Use this method to get a list of profile pictures for a user. Returns a [UserProfilePhotos](#) object.

| Parameter | Type | Required | Description |
|-----------|---------|----------|--|
| user_id | Integer | Yes | Unique identifier of the target user |
| offset | Integer | Optional | Sequential number of the first photo to be returned. By default, all photos are returned. |
| limit | Integer | Optional | Limits the number of photos to be retrieved. Values between 1-100 are accepted. Defaults to 100. |

setUserEmojiStatus

Changes the emoji status for a given user that previously allowed the bot to manage their emoji status via the Mini App method [requestEmojiStatusAccess](#). Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------------|---------|----------|--|
| user_id | Integer | Yes | Unique identifier of the target user |
| emoji_status_custom_emoji_id | String | Optional | Custom emoji identifier of the emoji status to set. Pass an empty string to remove the status. |
| emoji_status_expiration_date | Integer | Optional | Expiration date of the emoji status, if any |

getFile

Use this method to get basic information about a file and prepare it for downloading. For the moment, bots can download files of up to 20MB in size. On success, a [File](#) object is returned. The file can then be downloaded via the link https://api.telegram.org/file/bot<token>/<file_path>, where `<file_path>` is taken from the response. It is guaranteed that the

link will be valid for at least 1 hour. When the link expires, a new one can be requested by calling [getFile](#) again.

| Parameter | Type | Required | Description |
|-----------|--------|----------|--|
| file_id | String | Yes | File identifier to get information about |

Note: This function may not preserve the original file name and MIME type. You should save the file's MIME type and name (if available) when the File object is received.

banChatMember

Use this method to ban a user in a group, a supergroup or a channel. In the case of supergroups and channels, the user will not be able to return to the chat on their own using invite links, etc., unless unbanned first. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------------|-------------------|----------|---|
| chat_id | Integer or String | Yes | Unique identifier for the target group or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| until_date | Integer | Optional | Date when the user will be unbanned; Unix time. If user is banned for more than 366 days or less than 30 seconds from the current time they are considered to be banned forever. Applied for supergroups and channels only. |
| revoke_messages | Boolean | Optional | Pass <i>True</i> to delete all messages from the chat for the user that is being removed. If <i>False</i> , the user will be able to see messages in the group that were sent before the user was removed. Always <i>True</i> for supergroups and channels. |

unbanChatMember

Use this method to unban a previously banned user in a supergroup or channel. The user will **not** return to the group or channel automatically, but will be able to join via link, etc. The bot must be an administrator for this to work. By default, this method guarantees that after the call the user is not a member of the chat, but will be able to join it. So if the user is a member of the chat they will also be **removed** from the chat. If you don't want this, use the parameter `only_if_banned`. Returns *True* on success.

| Parameter | Type | Required | Description |
|----------------|-------------------|----------|---|
| chat_id | Integer or String | Yes | Unique identifier for the target group or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| only_if_banned | Boolean | Optional | Do nothing if the user is not banned |

restrictChatMember

Use this method to restrict a user in a supergroup. The bot must be an administrator in the supergroup for this to work and must have the appropriate administrator rights. Pass *True* for all permissions to lift restrictions from a user. Returns *True* on success.

| Parameter | Type | Required | Description |
|----------------------------------|-------------------|----------|---|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| permissions | ChatPermissions | Yes | A JSON-serialized object for new user permissions |
| use_independent_chat_permissions | Boolean | Optional | Pass <i>True</i> if chat permissions are set independently. Otherwise, the <code>can_send_other_messages</code> and <code>can_add_web_page_previews</code> permissions will imply the <code>can_send_messages</code> , <code>can_send_audios</code> , <code>can_send_documents</code> , <code>can_send_photos</code> , <code>can_send_videos</code> , <code>can_send_video_notes</code> , and <code>can_send_voice_notes</code> permissions; the <code>can_send_polls</code> permission will imply the <code>can_send_messages</code> permission. |
| until_date | Integer | Optional | Date when restrictions will be lifted for the user; Unix time. If user is restricted for more than 366 days or less than 30 seconds from the current time, they are considered to be restricted forever |

promoteChatMember

Use this method to promote or demote a user in a supergroup or a channel. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Pass *False* for all boolean parameters to demote a user. Returns *True* on success.

| 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>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| is_anonymous | Boolean | Optional | Pass <i>True</i> if the administrator's presence in the chat is hidden |
| can_manage_chat | Boolean | Optional | Pass <i>True</i> if the administrator can access the chat event log, get boost list, see hidden supergroup and channel members, report spam messages, ignore slow mode, and send messages to the chat without paying Telegram Stars. Implied by any other administrator privilege. |
| can_delete_messages | Boolean | Optional | Pass <i>True</i> if the administrator can delete messages of other users |
| can_manage_video_chats | Boolean | Optional | Pass <i>True</i> if the administrator can manage video chats |
| can_restrict_members | Boolean | Optional | Pass <i>True</i> if the administrator can restrict, ban or unban chat members, or access supergroup statistics |
| can_promote_members | Boolean | Optional | Pass <i>True</i> if the administrator can add new administrators with a subset of their own privileges or demote administrators that they have promoted, directly or indirectly (promoted by administrators that were appointed by him) |
| can_change_info | Boolean | Optional | Pass <i>True</i> if the administrator can change chat title, photo and other settings |
| can_invite_users | Boolean | Optional | Pass <i>True</i> if the administrator can invite new users to the chat |

| Parameter | Type | Required | Description |
|----------------------------|---------|----------|--|
| can_post_stories | Boolean | Optional | Pass <code>True</code> if the administrator can post stories to the chat |
| can_edit_stories | Boolean | Optional | Pass <code>True</code> if the administrator can edit stories posted by other users, post stories to the chat page, pin chat stories, and access the chat's story archive |
| can_delete_stories | Boolean | Optional | Pass <code>True</code> if the administrator can delete stories posted by other users |
| can_post_messages | Boolean | Optional | Pass <code>True</code> if the administrator can post messages in the channel, approve suggested posts, or access channel statistics; for channels only |
| can_edit_messages | Boolean | Optional | Pass <code>True</code> if the administrator can edit messages of other users and can pin messages; for channels only |
| can_pin_messages | Boolean | Optional | Pass <code>True</code> if the administrator can pin messages; for supergroups only |
| can_manage_topics | Boolean | Optional | Pass <code>True</code> if the user is allowed to create, rename, close, and reopen forum topics; for supergroups only |
| can_manage_direct_messages | Boolean | Optional | Pass <code>True</code> if the administrator can manage direct messages within the channel and decline suggested posts; for channels only |

setChatAdministratorCustomTitle

Use this method to set a custom title for an administrator in a supergroup promoted by the bot. Returns `True` on success.

| Parameter | Type | Required | Description |
|--------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| custom_title | String | Yes | New custom title for the administrator; 0–16 characters, emoji are not allowed |

banChatSenderChat

Use this method to ban a channel chat in a supergroup or a channel. Until the chat is [unbanned](#), the owner of the banned chat won't be able to send messages on behalf of [any of their channels](#). The bot must be an administrator in the supergroup or channel for this to work and must have the appropriate administrator rights. Returns `True` on success.

| 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>) |
| sender_chat_id | Integer | Yes | Unique identifier of the target sender chat |

unbanChatSenderChat

Use this method to unban a previously banned channel chat in a supergroup or channel. The bot must be an administrator for this to work and must have the appropriate administrator rights. Returns `True` on success.

| 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>) |
| sender_chat_id | Integer | Yes | Unique identifier of the target sender chat |

setChatPermissions

Use this method to set default chat permissions for all members. The bot must be an administrator in the group or a supergroup for this to work and must have the `can_restrict_members` administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------------------|-------------------|----------|---|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| permissions | ChatPermissions | Yes | A JSON-serialized object for new default chat permissions |
| use_independent_chat_permissions | Boolean | Optional | Pass <code>True</code> if chat permissions are set independently. Otherwise, the <code>can_send_other_messages</code> and <code>can_add_web_page_previews</code> permissions will imply the <code>can_send_messages</code> , <code>can_send_audios</code> , <code>can_send_documents</code> , <code>can_send_photos</code> , <code>can_send_videos</code> , <code>can_send_video_notes</code> , and <code>can_send_voice_notes</code> permissions; the <code>can_send_polls</code> permission will imply the <code>can_send_messages</code> permission. |

exportChatInviteLink

Use this method to generate a new primary invite link for a chat; any previously generated primary link is revoked. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns the new invite link as `String` on success.

| 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>) |

Note: Each administrator in a chat generates their own invite links. Bots can't use invite links generated by other administrators. If you want your bot to work with invite links, it will need to generate its own link using [exportChatInviteLink](#) or by calling the [getChat](#) method. If your bot needs to generate a new primary invite link replacing its previous one, use [exportChatInviteLink](#) again.

createChatInviteLink

Use this method to create an additional invite link for a chat. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. The link can be revoked using the method [revokeChatInviteLink](#). Returns the new invite link as [ChatInviteLink](#) object.

| 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>) |
| name | String | Optional | Invite link name; 0–32 characters |
| expire_date | Integer | Optional | Point in time (Unix timestamp) when the link will expire |
| member_limit | Integer | Optional | The maximum number of users that can be members of the chat simultaneously after joining the chat via this invite link; 1–99999 |
| creates_join_request | Boolean | Optional | <i>True</i> , if users joining the chat via the link need to be approved by chat administrators. If <i>True</i> , <code>member_limit</code> can't be specified |

editChatInviteLink

Use this method to edit a non-primary invite link created by the bot. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns the edited invite link as a [ChatInviteLink](#) object.

| 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>) |
| invite_link | String | Yes | The invite link to edit |
| name | String | Optional | Invite link name; 0–32 characters |
| expire_date | Integer | Optional | Point in time (Unix timestamp) when the link will expire |
| member_limit | Integer | Optional | The maximum number of users that can be members of the chat simultaneously after joining the chat via this invite link; 1–99999 |
| creates_join_request | Boolean | Optional | <i>True</i> , if users joining the chat via the link need to be approved by chat administrators. If <i>True</i> , <code>member_limit</code> can't be specified |

createChatSubscriptionInviteLink

Use this method to create a [subscription invite link](#) for a channel chat. The bot must have the `can_invite_users` administrator rights. The link can be edited using the method [editChatSubscriptionInviteLink](#) or revoked using the method [revokeChatInviteLink](#). Returns the new invite link as a [ChatInviteLink](#) object.

| Parameter | Type | Required | Description |
|---------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target channel chat or username of the target channel (in the format <code>@channelusername</code>) |
| name | String | Optional | Invite link name; 0–32 characters |
| subscription_period | Integer | Yes | The number of seconds the subscription will be active for before the next payment. Currently, it must always be 2592000 (30 days). |
| subscription_price | Integer | Yes | The amount of Telegram Stars a user must pay initially and after each subsequent subscription period to be a member of the chat; 1–10000 |

editChatSubscriptionInviteLink

Use this method to edit a subscription invite link created by the bot. The bot must have the `can_invite_users` administrator rights. Returns the edited invite link as a [ChatInviteLink](#) object.

| 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>) |
| invite_link | String | Yes | The invite link to edit |
| name | String | Optional | Invite link name; 0–32 characters |

revokeChatInviteLink

Use this method to revoke an invite link created by the bot. If the primary link is revoked, a new link is automatically generated. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns the revoked invite link as a [ChatInviteLink](#) object.

| Parameter | Type | Required | Description |
|-------------|-------------------|----------|---|
| chat_id | Integer or String | Yes | Unique identifier of the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| invite_link | String | Yes | The invite link to revoke |

approveChatJoinRequest

Use this method to approve a chat join request. The bot must be an administrator in the chat for this to work and must have the `can_invite_users` administrator right. Returns *True* on success.

| 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>) |
| user_id | Integer | Yes | Unique identifier of the target user |

declineChatJoinRequest

Use this method to decline a chat join request. The bot must be an administrator in the chat for this to work and must have the `can_invite_users` administrator right. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------|-------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>user_id</code> | Integer | Yes | Unique identifier of the target user |

setChatPhoto

Use this method to set a new profile photo for the chat. Photos can't be changed for private chats. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------|------------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>photo</code> | <code>InputFile</code> | Yes | New chat photo, uploaded using multipart/form-data |

deleteChatPhoto

Use this method to delete a chat photo. Photos can't be changed for private chats. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------|-------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |

setChatTitle

Use this method to change the title of a chat. Titles can't be changed for private chats. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------|-------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>title</code> | String | Yes | New chat title, 1–128 characters |

setChatDescription

Use this method to change the description of a group, a supergroup or a channel. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|--------------------------|-------------------|----------|--|
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>description</code> | String | Optional | New chat description, 0–255 characters |

pinChatMessage

Use this method to add a message to the list of pinned messages in a chat. In private chats and channel direct messages chats, all non-service messages can be pinned. Conversely, the bot must be an administrator with the 'can_pin_messages' right or the 'can_edit_messages' right to pin messages in groups and channels respectively. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------------------------|-------------------|----------|---|
| <code>business_connection_id</code> | String | Optional | Unique identifier of the business connection on behalf of which the message will be pinned |
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>message_id</code> | Integer | Yes | Identifier of a message to pin |
| <code>disable_notification</code> | Boolean | Optional | Pass <code>True</code> if it is not necessary to send a notification to all chat members about the new pinned message. Notifications are always disabled in channels and private chats. |

unpinChatMessage

Use this method to remove a message from the list of pinned messages in a chat. In private chats and channel direct messages chats, all messages can be unpinned. Conversely, the bot must be an administrator with the 'can_pin_messages' right or the 'can_edit_messages' right to unpin messages in groups and channels respectively. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------------------------|-------------------|----------|--|
| <code>business_connection_id</code> | String | Optional | Unique identifier of the business connection on behalf of which the message will be unpinned |
| <code>chat_id</code> | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| <code>message_id</code> | Integer | Optional | Identifier of the message to unpin. Required if <code>business_connection_id</code> is specified. If not specified, the most recent pinned message (by sending date) will be unpinned. |

unpinAllChatMessages

Use this method to clear the list of pinned messages in a chat. In private chats and channel direct messages chats, no additional rights are required to unpin all pinned messages. Conversely, the bot must be an administrator with the 'can_pin_messages' right or the 'can_edit_messages' right to unpin all pinned messages in groups and channels respectively. Returns `True` on success.

| 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>) |
| leaveChat | | | |
| Use this method for your bot to leave a group, supergroup or channel. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup or channel (in the format <code>@channelusername</code>). Channel direct messages chats aren't supported; leave the corresponding channel instead. |
| getChat | | | |
| Use this method to get up-to-date information about the chat. Returns a <code>ChatFullInfo</code> object on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| getChatAdministrators | | | |
| Use this method to get a list of administrators in a chat, which aren't bots. Returns an Array of <code>ChatMember</code> objects. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| getChatMemberCount | | | |
| Use this method to get the number of members in a chat. Returns <code>Int</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| getChatMember | | | |
| Use this method to get information about a member of a chat. The method is only guaranteed to work for other users if the bot is an administrator in the chat. Returns a <code>ChatMember</code> object on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup or channel (in the format <code>@channelusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| setChatStickerSet | | | |
| Use this method to set a new group sticker set for a supergroup. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Use the field <code>can_set_sticker_set</code> optionally returned in <code>getChat</code> requests to check if the bot can use this method. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| sticker_set_name | String | Yes | Name of the sticker set to be set as the group sticker set |
| deleteChatStickerSet | | | |
| Use this method to delete a group sticker set from a supergroup. The bot must be an administrator in the chat for this to work and must have the appropriate administrator rights. Use the field <code>can_set_sticker_set</code> optionally returned in <code>getChat</code> requests to check if the bot can use this method. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| getForumTopicIconStickers | | | |
| Use this method to get custom emoji stickers, which can be used as a forum topic icon by any user. Requires no parameters. Returns an Array of <code>Sticker</code> objects. | | | |
| createForumTopic | | | |
| Use this method to create a topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the <code>can_manage_topics</code> administrator rights. Returns information about the created topic as a <code>ForumTopic</code> object. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| name | String | Yes | Topic name, 1–128 characters |
| icon_color | Integer | Optional | Color of the topic icon in RGB format. Currently, must be one of 7322096 (0x6FB9F0), 16766590 (0xFFD67E), 13338331 (0xCB86DB), 9367192 (0x8EEE98), 16749490 (0xFF93B2), or 16478047 (0xFB6F5F) |
| icon_custom_emoji_id | String | Optional | Unique identifier of the custom emoji shown as the topic icon. Use <code>getForumTopicIconStickers</code> to get all allowed custom emoji identifiers. |
| editForumTopic | | | |

Use this method to edit name and icon of a topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights, unless it is the creator of the topic. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| message_thread_id | Integer | Yes | Unique identifier for the target message thread of the forum topic |
| name | String | Optional | New topic name, 0–128 characters. If not specified or empty, the current name of the topic will be kept |
| icon_custom_emoji_id | String | Optional | New unique identifier of the custom emoji shown as the topic icon. Use getForumTopicIconStickers to get all allowed custom emoji identifiers. Pass an empty string to remove the icon. If not specified, the current icon will be kept |

closeForumTopic

Use this method to close an open topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights, unless it is the creator of the topic. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| message_thread_id | Integer | Yes | Unique identifier for the target message thread of the forum topic |

reopenForumTopic

Use this method to reopen a closed topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights, unless it is the creator of the topic. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| message_thread_id | Integer | Yes | Unique identifier for the target message thread of the forum topic |

deleteForumTopic

Use this method to delete a forum topic along with all its messages in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_delete_messages` administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| message_thread_id | Integer | Yes | Unique identifier for the target message thread of the forum topic |

unpinAllForumTopicMessages

Use this method to clear the list of pinned messages in a forum topic. The bot must be an administrator in the chat for this to work and must have the `can_pin_messages` administrator right in the supergroup. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| message_thread_id | Integer | Yes | Unique identifier for the target message thread of the forum topic |

editGeneralForumTopic

Use this method to edit the name of the 'General' topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|-----------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| name | String | Yes | New topic name, 1–128 characters |

closeGeneralForumTopic

Use this method to close an open 'General' topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights. Returns `True` on success.

| Parameter | Type | Required | Description |
|-----------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |

reopenGeneralForumTopic

Use this method to reopen a closed 'General' topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights. The topic will be automatically unhidden if it was hidden. Returns `True` on success.

| Parameter | Type | Required | Description |
|-----------|-------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |

hideGeneralForumTopic

Use this method to hide the 'General' topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the `can_manage_topics` administrator rights. The topic will be automatically closed if it was open. Returns `True` on success.

| Parameter | Type | Required | Description |
|---|-------------------------------------|----------|--|
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| unhideGeneralForumTopic | | | |
| Use this method to unhide the 'General' topic in a forum supergroup chat. The bot must be an administrator in the chat for this to work and must have the <code>can_manage_topics</code> administrator rights. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| unpinAllGeneralForumTopicMessages | | | |
| Use this method to clear the list of pinned messages in a General forum topic. The bot must be an administrator in the chat for this to work and must have the <code>can_pin_messages</code> administrator right in the supergroup. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target supergroup (in the format <code>@supergroupusername</code>) |
| answerCallbackQuery | | | |
| Use this method to send answers to callback queries sent from inline keyboards . The answer will be displayed to the user as a notification at the top of the chat screen or as an alert. On success, <code>True</code> is returned. | | | |
| Alternatively, the user can be redirected to the specified Game URL. For this option to work, you must first create a game for your bot via @BotFather and accept the terms. Otherwise, you may use links like <code>t.me/your_bot?start=XXXX</code> that open your bot with a parameter. | | | |
| Parameter | Type | Required | Description |
| callback_query_id | String | Yes | Unique identifier for the query to be answered |
| text | String | Optional | Text of the notification. If not specified, nothing will be shown to the user, 0-200 characters |
| show_alert | Boolean | Optional | If <code>True</code> , an alert will be shown by the client instead of a notification at the top of the chat screen. Defaults to <code>false</code> . |
| url | String | Optional | URL that will be opened by the user's client. If you have created a Game and accepted the conditions via @BotFather , specify the URL that opens your game – note that this will only work if the query comes from a callback_game button. |
| Otherwise, you may use links like <code>t.me/your_bot?start=XXXX</code> that open your bot with a parameter. | | | |
| cache_time | Integer | Optional | The maximum amount of time in seconds that the result of the callback query may be cached client-side. Telegram apps will support caching starting in version 3.14. Defaults to 0. |
| getUserChatBoots | | | |
| Use this method to get the list of boosts added to a chat by a user. Requires administrator rights in the chat. Returns a UserChatBoots object. | | | |
| Parameter | Type | Required | Description |
| chat_id | Integer or String | Yes | Unique identifier for the chat or username of the channel (in the format <code>@channelusername</code>) |
| user_id | Integer | Yes | Unique identifier of the target user |
| getBusinessConnection | | | |
| Use this method to get information about the connection of the bot with a business account. Returns a BusinessConnection object on success. | | | |
| Parameter | Type | Required | Description |
| business_connection_id | String | Yes | Unique identifier of the business connection |
| setMyCommands | | | |
| Use this method to change the list of the bot's commands. See this manual for more details about bot commands. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| commands | Array of BotCommand | Yes | A JSON-serialized list of bot commands to be set as the list of the bot's commands. At most 100 commands can be specified. |
| scope | BotCommandScope | Optional | A JSON-serialized object, describing scope of users for which the commands are relevant. Defaults to BotCommandScopeDefault . |
| language_code | String | Optional | A two-letter ISO 639-1 language code. If empty, commands will be applied to all users from the given scope, for whose language there are no dedicated commands |
| deleteMyCommands | | | |
| Use this method to delete the list of the bot's commands for the given scope and user language. After deletion, higher level commands will be shown to affected users. Returns <code>True</code> on success. | | | |
| Parameter | Type | Required | Description |
| scope | BotCommandScope | Optional | A JSON-serialized object, describing scope of users for which the commands are relevant. Defaults to BotCommandScopeDefault . |

| Parameter | Type | Required | Description |
|---------------|--------|----------|--|
| language_code | String | Optional | A two-letter ISO 639-1 language code. If empty, commands will be applied to all users from the given scope, for whose language there are no dedicated commands |

getMyCommands

Use this method to get the current list of the bot's commands for the given scope and user language. Returns an Array of `BotCommand` objects. If commands aren't set, an empty list is returned.

| Parameter | Type | Required | Description |
|---------------|------------------------------|----------|--|
| scope | <code>BotCommandScope</code> | Optional | A JSON-serialized object, describing scope of users. Defaults to <code>BotCommandScopeDefault</code> . |
| language_code | String | Optional | A two-letter ISO 639-1 language code or an empty string |

setMyName

Use this method to change the bot's name. Returns `True` on success.

| Parameter | Type | Required | Description |
|---------------|--------|----------|--|
| name | String | Optional | New bot name; 0-64 characters. Pass an empty string to remove the dedicated name for the given language. |
| language_code | String | Optional | A two-letter ISO 639-1 language code. If empty, the name will be shown to all users for whose language there is no dedicated name. |

getMyName

Use this method to get the current bot name for the given user language. Returns `BotName` on success.

| Parameter | Type | Required | Description |
|---------------|--------|----------|---|
| language_code | String | Optional | A two-letter ISO 639-1 language code or an empty string |

setMyDescription

Use this method to change the bot's description, which is shown in the chat with the bot if the chat is empty. Returns `True` on success.

| Parameter | Type | Required | Description |
|---------------|--------|----------|--|
| description | String | Optional | New bot description; 0-512 characters. Pass an empty string to remove the dedicated description for the given language. |
| language_code | String | Optional | A two-letter ISO 639-1 language code. If empty, the description will be applied to all users for whose language there is no dedicated description. |

getMyDescription

Use this method to get the current bot description for the given user language. Returns `BotDescription` on success.

| Parameter | Type | Required | Description |
|---------------|--------|----------|---|
| language_code | String | Optional | A two-letter ISO 639-1 language code or an empty string |

setMyShortDescription

Use this method to change the bot's short description, which is shown on the bot's profile page and is sent together with the link when users share the bot. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------------|--------|----------|--|
| short_description | String | Optional | New short description for the bot; 0-120 characters. Pass an empty string to remove the dedicated short description for the given language. |
| language_code | String | Optional | A two-letter ISO 639-1 language code. If empty, the short description will be applied to all users for whose language there is no dedicated short description. |

getMyShortDescription

Use this method to get the current bot short description for the given user language. Returns `BotShortDescription` on success.

| Parameter | Type | Required | Description |
|---------------|--------|----------|---|
| language_code | String | Optional | A two-letter ISO 639-1 language code or an empty string |

setChatMenuButton

Use this method to change the bot's menu button in a private chat, or the default menu button. Returns `True` on success.

| Parameter | Type | Required | Description |
|-------------|-------------------------|----------|--|
| chat_id | Integer | Optional | Unique identifier for the target private chat. If not specified, default bot's menu button will be changed |
| menu_button | <code>MenuButton</code> | Optional | A JSON-serialized object for the bot's new menu button. Defaults to <code>MenuButtonDefault</code> |

getChatMenuButton

Use this method to get the current value of the bot's menu button in a private chat, or the default menu button. Returns `MenuButton` on success.

| Parameter | Type | Required | Description |
|-----------|---------|----------|---|
| chat_id | Integer | Optional | Unique identifier for the target private chat. If not specified, default bot's menu button will be returned |

setMyDefaultAdministratorRights

Use this method to change the default administrator rights requested by the bot when it's added as an administrator to groups or channels. These rights will be suggested to users, but they are free to modify the list before adding the bot. Returns `True` on success.

| Parameter | Type | Required | Description |
|---------------------------|--------------------------------------|----------|--|
| <code>rights</code> | <code>ChatAdministratorRights</code> | Optional | A JSON-serialized object describing new default administrator rights. If not specified, the default administrator rights will be cleared. |
| <code>for_channels</code> | <code>Boolean</code> | Optional | Pass <code>True</code> to change the default administrator rights of the bot in channels. Otherwise, the default administrator rights of the bot for groups and supergroups will be changed. |

getMyDefaultAdministratorRights

Use this method to get the current default administrator rights of the bot. Returns `ChatAdministratorRights` on success.

| Parameter | Type | Required | Description |
|---------------------------|----------------------|----------|--|
| <code>for_channels</code> | <code>Boolean</code> | Optional | Pass <code>True</code> to get default administrator rights of the bot in channels. Otherwise, default administrator rights of the bot for groups and supergroups will be returned. |

getAvailableGifts

Returns the list of gifts that can be sent by the bot to users and channel chats. Requires no parameters. Returns a `Gifts` object.

sendGift

Sends a gift to the given user or channel chat. The gift can't be converted to Telegram Stars by the receiver. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------------|-------------------------------------|----------|---|
| <code>user_id</code> | <code>Integer</code> | Optional | Required if <code>chat_id</code> is not specified. Unique identifier of the target user who will receive the gift. |
| <code>chat_id</code> | <code>Integer or String</code> | Optional | Required if <code>user_id</code> is not specified. Unique identifier for the chat or username of the channel (in the format <code>@channelusername</code>) that will receive the gift. |
| <code>gift_id</code> | <code>String</code> | Yes | Identifier of the gift |
| <code>pay_for_upgrade</code> | <code>Boolean</code> | Optional | Pass <code>True</code> to pay for the gift upgrade from the bot's balance, thereby making the upgrade free for the receiver |
| <code>text</code> | <code>String</code> | Optional | Text that will be shown along with the gift; 0–128 characters |
| <code>text_parse_mode</code> | <code>String</code> | Optional | Mode for parsing entities in the text. See formatting options for more details. Entities other than "bold", "italic", "underline", "strikethrough", "spoiler", and "custom_emoji" are ignored. |
| <code>text_entities</code> | Array of <code>MessageEntity</code> | Optional | A JSON-serialized list of special entities that appear in the gift text. It can be specified instead of <code>text_parse_mode</code> . Entities other than "bold", "italic", "underline", "strikethrough", "spoiler", and "custom_emoji" are ignored. |

giftPremiumSubscription

Gifts a Telegram Premium subscription to the given user. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------------|-------------------------------------|----------|---|
| <code>user_id</code> | <code>Integer</code> | Yes | Unique identifier of the target user who will receive a Telegram Premium subscription |
| <code>month_count</code> | <code>Integer</code> | Yes | Number of months the Telegram Premium subscription will be active for the user; must be one of 3, 6, or 12 |
| <code>star_count</code> | <code>Integer</code> | Yes | Number of Telegram Stars to pay for the Telegram Premium subscription; must be 1000 for 3 months, 1500 for 6 months, and 2500 for 12 months |
| <code>text</code> | <code>String</code> | Optional | Text that will be shown along with the service message about the subscription; 0–128 characters |
| <code>text_parse_mode</code> | <code>String</code> | Optional | Mode for parsing entities in the text. See formatting options for more details. Entities other than "bold", "italic", "underline", "strikethrough", "spoiler", and "custom_emoji" are ignored. |
| <code>text_entities</code> | Array of <code>MessageEntity</code> | Optional | A JSON-serialized list of special entities that appear in the gift text. It can be specified instead of <code>text_parse_mode</code> . Entities other than "bold", "italic", "underline", "strikethrough", "spoiler", and "custom_emoji" are ignored. |

verifyUser

Verifies a user [on behalf of the organization](#) which is represented by the bot. Returns `True` on success.

| Parameter | Type | Required | Description |
|---------------------------------|----------------------|----------|---|
| <code>user_id</code> | <code>Integer</code> | Yes | Unique identifier of the target user |
| <code>custom_description</code> | <code>String</code> | Optional | Custom description for the verification; 0–70 characters. Must be empty if the organization isn't allowed to provide a custom verification description. |

verifyChat

Verifies a chat [on behalf of the organization](#) which is represented by the bot. Returns `True` on success.

| Parameter | Type | Required | Description |
|---------------------------------|--------------------------------|----------|--|
| <code>chat_id</code> | <code>Integer or String</code> | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>). Channel direct messages chats can't be verified. |
| <code>custom_description</code> | <code>String</code> | Optional | Custom description for the verification; 0–70 characters. Must be empty if the organization isn't allowed to provide a custom verification description. |

removeUserVerification

Removes verification from a user who is currently verified on behalf of the organization represented by the bot. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------|---------|----------|--------------------------------------|
| user_id | Integer | Yes | Unique identifier of the target user |

removeChatVerification

Removes verification from a chat that is currently verified on behalf of the organization represented by the bot. Returns *True* on success.

| 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>) |

readBusinessMessage

Marks incoming message as read on behalf of a business account. Requires the `can_read_messages` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection on behalf of which to read the message |
| chat_id | Integer | Yes | Unique identifier of the chat in which the message was received. The chat must have been active in the last 24 hours. |
| message_id | Integer | Yes | Unique identifier of the message to mark as read |

deleteBusinessMessages

Delete messages on behalf of a business account. Requires the `can_delete_sent_messages` business bot right to delete messages sent by the bot itself, or the `can_delete_all_messages` business bot right to delete any message. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|------------------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection on behalf of which to delete the messages |
| message_ids | Array of Integer | Yes | A JSON-serialized list of 1–100 identifiers of messages to delete. All messages must be from the same chat. See deleteMessage for limitations on which messages can be deleted |

setBusinessAccountName

Changes the first and last name of a managed business account. Requires the `can_change_name` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| first_name | String | Yes | The new value of the first name for the business account; 1–64 characters |
| last_name | String | Optional | The new value of the last name for the business account; 0–64 characters |

setBusinessAccountUsername

Changes the username of a managed business account. Requires the `can_change_username` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| username | String | Optional | The new value of the username for the business account; 0–32 characters |

setBusinessAccountBio

Changes the bio of a managed business account. Requires the `can_change_bio` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| bio | String | Optional | The new value of the bio for the business account; 0–140 characters |

setBusinessAccountProfilePhoto

Changes the profile photo of a managed business account. Requires the `can_edit_profile_photo` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| photo | InputProfilePhoto | Yes | The new profile photo to set |
| is_public | Boolean | Optional | Pass <i>True</i> to set the public photo, which will be visible even if the main photo is hidden by the business account's privacy settings. An account can have only one public photo. |

removeBusinessAccountProfilePhoto

Removes the current profile photo of a managed business account. Requires the `can_edit_profile_photo` business bot right. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |

| Parameter | Type | Required | Description |
|-----------|---------|----------|---|
| is_public | Boolean | Optional | Pass <code>True</code> to remove the public photo, which is visible even if the main photo is hidden by the business account's privacy settings. After the main photo is removed, the previous profile photo (if present) becomes the main photo. |

setBusinessAccountGiftSettings

Changes the privacy settings pertaining to incoming gifts in a managed business account. Requires the `can_change_gift_settings` business bot right. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| show_gift_button | Boolean | Yes | Pass <code>True</code> , if a button for sending a gift to the user or by the business account must always be shown in the input field |
| accepted_gift_types | AcceptedGiftTypes | Yes | Types of gifts accepted by the business account |

getBusinessAccountStarBalance

Returns the amount of Telegram Stars owned by a managed business account. Requires the `can_view_gifts_and_stars` business bot right. Returns `StarAmount` on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |

transferBusinessAccountStars

Transfers Telegram Stars from the business account balance to the bot's balance. Requires the `can_transfer_stars` business bot right. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| star_count | Integer | Yes | Number of Telegram Stars to transfer; 1–10000 |

getBusinessAccountGifts

Returns the gifts received and owned by a managed business account. Requires the `can_view_gifts_and_stars` business bot right. Returns `OwnedGifts` on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| exclude_unsaved | Boolean | Optional | Pass <code>True</code> to exclude gifts that aren't saved to the account's profile page |
| exclude_saved | Boolean | Optional | Pass <code>True</code> to exclude gifts that are saved to the account's profile page |
| exclude_unlimited | Boolean | Optional | Pass <code>True</code> to exclude gifts that can be purchased an unlimited number of times |
| exclude_limited | Boolean | Optional | Pass <code>True</code> to exclude gifts that can be purchased a limited number of times |
| exclude_unique | Boolean | Optional | Pass <code>True</code> to exclude unique gifts |
| sort_by_price | Boolean | Optional | Pass <code>True</code> to sort results by gift price instead of send date. Sorting is applied before pagination. |
| offset | String | Optional | Offset of the first entry to return as received from the previous request; use empty string to get the first chunk of results |
| limit | Integer | Optional | The maximum number of gifts to be returned; 1–100. Defaults to 100 |

convertGiftToStars

Converts a given regular gift to Telegram Stars. Requires the `can_convert_gifts_to_stars` business bot right. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|--------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| owned_gift_id | String | Yes | Unique identifier of the regular gift that should be converted to Telegram Stars |

upgradeGift

Upgrades a given regular gift to a unique gift. Requires the `can_transfer_and_upgrade_gifts` business bot right. Additionally requires the `can_transfer_stars` business bot right if the upgrade is paid. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|---|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| owned_gift_id | String | Yes | Unique identifier of the regular gift that should be upgraded to a unique one |
| keep_original_details | Boolean | Optional | Pass <code>True</code> to keep the original gift text, sender and receiver in the upgraded gift |
| star_count | Integer | Optional | The amount of Telegram Stars that will be paid for the upgrade from the business account balance. If <code>gift.paid_prepaid_upgrade_star_count > 0</code> , then pass 0, otherwise, the <code>can_transfer_stars</code> business bot right is required and <code>gift.upgrade_star_count</code> must be passed. |

transferGift

Transfers an owned unique gift to another user. Requires the `can_transfer_and_upgrade_gifts` business bot right. Requires `can_transfer_stars` business bot right if the transfer is paid. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| owned_gift_id | String | Yes | Unique identifier of the regular gift that should be transferred |
| new_owner_chat_id | Integer | Yes | Unique identifier of the chat which will own the gift. The chat must be active in the last 24 hours. |
| star_count | Integer | Optional | The amount of Telegram Stars that will be paid for the transfer from the business account balance. If positive, then the <code>can_transfer_stars</code> business bot right is required. |

postStory

Posts a story on behalf of a managed business account. Requires the `can_manage_stories` business bot right. Returns [Story](#) on success.

| Parameter | Type | Required | Description |
|------------------------|--|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| content | InputStoryContent | Yes | Content of the story |
| active_period | Integer | Yes | Period after which the story is moved to the archive, in seconds; must be one of <code>6 * 3600</code> , <code>12 * 3600</code> , <code>86400</code> , or <code>2 * 86400</code> |
| caption | String | Optional | Caption of the story, 0–2048 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the story caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| areas | Array of StoryArea | Optional | A JSON-serialized list of clickable areas to be shown on the story |
| post_to_chat_page | Boolean | Optional | Pass <code>True</code> to keep the story accessible after it expires |
| protect_content | Boolean | Optional | Pass <code>True</code> if the content of the story must be protected from forwarding and screenshotting |

editStory

Edits a story previously posted by the bot on behalf of a managed business account. Requires the `can_manage_stories` business bot right. Returns [Story](#) on success.

| Parameter | Type | Required | Description |
|------------------------|--|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| story_id | Integer | Yes | Unique identifier of the story to edit |
| content | InputStoryContent | Yes | Content of the story |
| caption | String | Optional | Caption of the story, 0–2048 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the story caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| areas | Array of StoryArea | Optional | A JSON-serialized list of clickable areas to be shown on the story |

deleteStory

Deletes a story previously posted by the bot on behalf of a managed business account. Requires the `can_manage_stories` business bot right. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------------------|---------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection |
| story_id | Integer | Yes | Unique identifier of the story to delete |

Inline mode methods

Methods and objects used in the inline mode are described in the [Inline mode](#) section.

Updating messages

The following methods allow you to change an existing message in the message history instead of sending a new one with a result of an action. This is most useful for messages with [inline keyboards](#) using callback queries, but can also help reduce clutter in conversations with regular chat bots.

Please note, that it is currently only possible to edit messages without `reply_markup` or with [inline keyboards](#).

editMessageText

Use this method to edit text and [game](#) messages. On success, if the edited message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned. Note that business messages that were not sent by the bot and do not contain an inline keyboard can only be edited within **48 hours** from the time they were sent.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message to edit |

| Parameter | Type | Required | Description |
|----------------------|--|----------|---|
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| text | String | Yes | New text of the message, 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 | A JSON-serialized list of special entities that appear in message text, which can be specified instead of <code>parse_mode</code> |
| link_preview_options | LinkPreviewOptions | Optional | Link preview generation options for the message |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard . |

editMessageCaption

Use this method to edit captions of messages. On success, if the edited message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned. Note that business messages that were not sent by the bot and do not contain an inline keyboard can only be edited within **48 hours** from the time they were sent.

| Parameter | Type | Required | Description |
|--------------------------|--|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message to edit |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| caption | String | Optional | New caption of the message, 0–1024 characters after entities parsing |
| parse_mode | String | Optional | Mode for parsing entities in the message caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | Optional | A JSON-serialized list of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | Optional | Pass <code>True</code> , if the caption must be shown above the message media. Supported only for animation, photo and video messages. |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard . |

editMessageMedia

Use this method to edit animation, audio, document, photo, or video messages, or to add media to text messages. If a message is part of a message album, then it can be edited only to an audio for audio albums, only to a document for document albums and to a photo or a video otherwise. When an inline message is edited, a new file can't be uploaded; use a previously uploaded file via its `file_id` or specify a URL. On success, if the edited message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned. Note that business messages that were not sent by the bot and do not contain an inline keyboard can only be edited within **48 hours** from the time they were sent.

| Parameter | Type | Required | Description |
|------------------------|--------------------------------------|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message to edit |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| media | InputMedia | Yes | A JSON-serialized object for a new media content of the message |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for a new inline keyboard . |

editMessageLiveLocation

Use this method to edit live location messages. A location can be edited until its `live_period` expires or editing is explicitly disabled by a call to [stopMessageLiveLocation](#). On success, if the edited message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned.

| Parameter | Type | Required | Description |
|------------------------|-------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message to edit |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| latitude | Float | Yes | Latitude of new location |
| longitude | Float | Yes | Longitude of new location |
| live_period | Integer | Optional | New period in seconds during which the location can be updated, starting from the message send date. If <code>0xFFFFFFFF</code> is specified, then the location can be updated forever. Otherwise, the new value must not exceed the current <code>live_period</code> by more than a day, and the live location expiration date must remain within the next 90 days. If not specified, then <code>live_period</code> remains unchanged |
| horizontal_accuracy | Float | Optional | The radius of uncertainty for the location, measured in meters; 0–1500 |

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|--|
| heading | Integer | Optional | Direction in which the user is moving, in degrees. Must be between 1 and 360 if specified. |
| proximity_alert_radius | Integer | Optional | The maximum distance for proximity alerts about approaching another chat member, in meters. Must be between 1 and 100000 if specified. |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for a new inline keyboard . |

stopMessageLiveLocation

Use this method to stop updating a live location message before `live_period` expires. On success, if the message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned.

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message with live location to stop |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for a new inline keyboard . |

editMessageChecklist

Use this method to edit a checklist on behalf of a connected business account. On success, the edited [Message](#) is returned.

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|--|
| business_connection_id | String | Yes | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer | Yes | Unique identifier for the target chat |
| message_id | Integer | Yes | Unique identifier for the target message |
| checklist | InputChecklist | Yes | A JSON-serialized object for the new checklist |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for the new inline keyboard for the message |

editMessageReplyMarkup

Use this method to edit only the reply markup of messages. On success, if the edited message is not an inline message, the edited [Message](#) is returned, otherwise `True` is returned. Note that business messages that were not sent by the bot and do not contain an inline keyboard can only be edited within **48 hours** from the time they were sent.

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|---|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the message to edit |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard . |

stopPoll

Use this method to stop a poll which was sent by the bot. On success, the stopped [Poll](#) is returned.

| Parameter | Type | Required | Description |
|------------------------|----------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message to be edited was sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format <code>@channelusername</code>) |
| message_id | Integer | Yes | Identifier of the original message with the poll |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for a new message inline keyboard . |

approveSuggestedPost

Use this method to approve a suggested post in a direct messages chat. The bot must have the 'can_post_messages' administrator right in the corresponding channel chat. Returns `True` on success.

| Parameter | Type | Required | Description |
|------------|---------|----------|--|
| chat_id | Integer | Yes | Unique identifier for the target direct messages chat |
| message_id | Integer | Yes | Identifier of a suggested post message to approve |
| send_date | Integer | Optional | Point in time (Unix timestamp) when the post is expected to be published; omit if the date has already been specified when the suggested post was created. If specified, then the date must be not more than 2678400 seconds (30 days) in the future |

declineSuggestedPost

Use this method to decline a suggested post in a direct messages chat. The bot must have the 'can_manage_direct_messages' administrator right in the corresponding channel chat. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------|---------|----------|---|
| chat_id | Integer | Yes | Unique identifier for the target direct messages chat |
| message_id | Integer | Yes | Identifier of a suggested post message to decline |
| comment | String | Optional | Comment for the creator of the suggested post; 0–128 characters |

deleteMessage

Use this method to delete a message, including service messages, with the following limitations:

- A message can only be deleted if it was sent less than 48 hours ago.
- Service messages about a supergroup, channel, or forum topic creation can't be deleted.
- A dice message in a private chat can only be deleted if it was sent more than 24 hours ago.
- Bots can delete outgoing messages in private chats, groups, and supergroups.
- Bots can delete incoming messages in private chats.
- Bots granted *can_post_messages* permissions can delete outgoing messages in channels.
- If the bot is an administrator of a group, it can delete any message there.
- If the bot has *can_delete_messages* administrator right in a supergroup or a channel, it can delete any message there.
- If the bot has *can_manage_direct_messages* administrator right in a channel, it can delete any message in the corresponding direct messages chat.

Returns *True* on success.

| 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>) |
| message_id | Integer | Yes | Identifier of the message to delete |

deleteMessages

Use this method to delete multiple messages simultaneously. If some of the specified messages can't be found, they are skipped. Returns *True* on success.

| 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>) |
| message_ids | Array of Integer | Yes | A JSON-serialized list of 1–100 identifiers of messages to delete. See deleteMessage for limitations on which messages can be deleted |

Stickers

The following methods and objects allow your bot to handle stickers and sticker sets.

Sticker

This object represents a sticker.

| Field | Type | Description |
|-------------------|--------------|---|
| file_id | String | Identifier for this file, which can be used to download or reuse the file |
| file_unique_id | String | Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file. |
| type | String | Type of the sticker, currently one of "regular", "mask", "custom_emoji". The type of the sticker is independent from its format, which is determined by the fields <i>is_animated</i> and <i>is_video</i> . |
| width | Integer | Sticker width |
| height | Integer | Sticker height |
| is_animated | Boolean | <i>True</i> , if the sticker is animated |
| is_video | Boolean | <i>True</i> , if the sticker is a video sticker |
| thumbnail | PhotoSize | <i>Optional</i> . Sticker thumbnail in the .WEBP or .JPG format |
| emoji | String | <i>Optional</i> . Emoji associated with the sticker |
| set_name | String | <i>Optional</i> . Name of the sticker set to which the sticker belongs |
| premium_animation | File | <i>Optional</i> . For premium regular stickers, premium animation for the sticker |
| mask_position | MaskPosition | <i>Optional</i> . For mask stickers, the position where the mask should be placed |
| custom_emoji_id | String | <i>Optional</i> . For custom emoji stickers, unique identifier of the custom emoji |
| needs_repainting | True | <i>Optional</i> . <i>True</i> , if the sticker must be repainted to a text color in messages, the color of the Telegram Premium badge in emoji status, white color on chat photos, or another appropriate color in other places |
| file_size | Integer | <i>Optional</i> . File size in bytes |

StickerSet

This object represents a sticker set.

| Field | Type | Description |
|-------|--------|-------------------|
| name | String | Sticker set name |
| title | String | Sticker set title |

| Field | Type | Description |
|--------------|------------------|---|
| sticker_type | String | Type of stickers in the set, currently one of "regular", "mask", "custom_emoji" |
| stickers | Array of Sticker | List of all set stickers |
| thumbnail | PhotoSize | <i>Optional.</i> Sticker set thumbnail in the .WEBP, .TGS, or .WEBM format |

MaskPosition

This object describes the position on faces where a mask should be placed by default.

| Field | Type | Description |
|---------|--------|---|
| point | String | The part of the face relative to which the mask should be placed. One of "forehead", "eyes", "mouth", or "chin". |
| x_shift | Float | Shift by X-axis measured in widths of the mask scaled to the face size, from left to right. For example, choosing -1.0 will place mask just to the left of the default mask position. |
| y_shift | Float | Shift by Y-axis measured in heights of the mask scaled to the face size, from top to bottom. For example, 1.0 will place the mask just below the default mask position. |
| scale | Float | Mask scaling coefficient. For example, 2.0 means double size. |

InputSticker

This object describes a sticker to be added to a sticker set.

| Field | Type | Description |
|---------------|-----------------|---|
| sticker | String | The added sticker. Pass a <i>file_id</i> as a String to send a file that already exists on the Telegram servers, pass an HTTP URL as a String for Telegram to get a file from the Internet, or pass "attach://<file_attach_name>" to upload a new file using multipart/form-data under <file_attach_name> name. Animated and video stickers can't be uploaded via HTTP URL. More information on Sending Files » |
| format | String | Format of the added sticker, must be one of "static" for a .WEBP or .PNG image, "animated" for a .TGS animation, "video" for a .WEBM video |
| emoji_list | Array of String | List of 1–20 emoji associated with the sticker |
| mask_position | MaskPosition | <i>Optional.</i> Position where the mask should be placed on faces. For "mask" stickers only. |
| keywords | Array of String | <i>Optional.</i> List of 0–20 search keywords for the sticker with total length of up to 64 characters. For "regular" and "custom_emoji" stickers only. |

sendSticker

Use this method to send static .WEBP, animated .TGS, or video .WEBM stickers. On success, the sent [Message](#) is returned.

| Parameter | Type | Required | Description |
|---------------------------|--|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer or String | Yes | Unique identifier for the target chat or username of the target channel (in the format @channelusername) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| sticker | InputFile or String | Yes | Sticker to send. Pass a <i>file_id</i> as String to send a file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a .WEBP sticker from the Internet, or upload a new .WEBP, .TGS, or .WEBM sticker using multipart/form-data. More information on Sending Files » . Video and animated stickers can't be sent via an HTTP URL. |
| emoji | String | Optional | Emoji associated with the sticker; only for just uploaded stickers |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <i>True</i> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| 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 a reply keyboard or to force a reply from the user |

getStickerSet

Use this method to get a sticker set. On success, a [StickerSet](#) object is returned.

| Parameter | Type | Required | Description |
|---|---------------------------------------|----------|---|
| name | String | Yes | Name of the sticker set |
| getCustomEmojiStickers | | | |
| Use this method to get information about custom emoji stickers by their identifiers. Returns an Array of Sticker objects. | | | |
| Parameter | Type | Required | Description |
| custom_emoji_ids | Array of String | Yes | A JSON-serialized list of custom emoji identifiers. At most 200 custom emoji identifiers can be specified. |
| uploadStickerFile | | | |
| Use this method to upload a file with a sticker for later use in the createNewStickerSet , addStickerToSet , or replaceStickerInSet methods (the file can be used multiple times). Returns the uploaded File on success. | | | |
| Parameter | Type | Required | Description |
| user_id | Integer | Yes | User identifier of sticker file owner |
| sticker | InputFile | Yes | A file with the sticker in .WEBP, .PNG, .TGS, or .WEBM format. See https://core.telegram.org/stickers for technical requirements. More information on Sending Files » |
| sticker_format | String | Yes | Format of the sticker, must be one of "static", "animated", "video" |
| createNewStickerSet | | | |
| Use this method to create a new sticker set owned by a user. The bot will be able to edit the sticker set thus created. Returns True on success. | | | |
| Parameter | Type | Required | Description |
| user_id | Integer | Yes | User identifier of created sticker set owner |
| name | String | Yes | Short name of sticker set, to be used in t.me/addstickers/ URLs (e.g., <i>animals</i>). Can contain only English letters, digits and underscores. Must begin with a letter, can't contain consecutive underscores and must end in " _by_<bot_username> ". <bot_username> is case insensitive. 1–64 characters. |
| title | String | Yes | Sticker set title, 1–64 characters |
| stickers | Array of InputSticker | Yes | A JSON-serialized list of 1–50 initial stickers to be added to the sticker set |
| sticker_type | String | Optional | Type of stickers in the set, pass "regular", "mask", or "custom_emoji". By default, a regular sticker set is created. |
| needs_repainting | Boolean | Optional | Pass True if stickers in the sticker set must be repainted to the color of text when used in messages, the accent color if used as emoji status, white on chat photos, or another appropriate color based on context; for custom emoji sticker sets only |
| addStickerToSet | | | |
| Use this method to add a new sticker to a set created by the bot. Emoji sticker sets can have up to 200 stickers. Other sticker sets can have up to 120 stickers. Returns True on success. | | | |
| Parameter | Type | Required | Description |
| user_id | Integer | Yes | User identifier of sticker set owner |
| name | String | Yes | Sticker set name |
| sticker | InputSticker | Yes | A JSON-serialized object with information about the added sticker. If exactly the same sticker had already been added to the set, then the set isn't changed. |
| setStickerPositionInSet | | | |
| Use this method to move a sticker in a set created by the bot to a specific position. Returns True on success. | | | |
| Parameter | Type | Required | Description |
| sticker | String | Yes | File identifier of the sticker |
| position | Integer | Yes | New sticker position in the set, zero-based |
| deleteStickerFromSet | | | |
| Use this method to delete a sticker from a set created by the bot. Returns True on success. | | | |
| Parameter | Type | Required | Description |
| sticker | String | Yes | File identifier of the sticker |
| replaceStickerInSet | | | |
| Use this method to replace an existing sticker in a sticker set with a new one. The method is equivalent to calling deleteStickerFromSet , then addStickerToSet , then setStickerPositionInSet . Returns True on success. | | | |
| Parameter | Type | Required | Description |
| user_id | Integer | Yes | User identifier of the sticker set owner |
| name | String | Yes | Sticker set name |
| old_sticker | String | Yes | File identifier of the replaced sticker |

| Parameter | Type | Required | Description |
|-----------|--------------|----------|---|
| sticker | InputSticker | Yes | A JSON-serialized object with information about the added sticker. If exactly the same sticker had already been added to the set, then the set remains unchanged. |

setStickerEmojiList

Use this method to change the list of emoji assigned to a regular or custom emoji sticker. The sticker must belong to a sticker set created by the bot. Returns *True* on success.

| Parameter | Type | Required | Description |
|------------|-----------------|----------|--|
| sticker | String | Yes | File identifier of the sticker |
| emoji_list | Array of String | Yes | A JSON-serialized list of 1–20 emoji associated with the sticker |

setStickerKeywords

Use this method to change search keywords assigned to a regular or custom emoji sticker. The sticker must belong to a sticker set created by the bot. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------|-----------------|----------|---|
| sticker | String | Yes | File identifier of the sticker |
| keywords | Array of String | Optional | A JSON-serialized list of 0–20 search keywords for the sticker with total length of up to 64 characters |

setStickerMaskPosition

Use this method to change the *mask position* of a mask sticker. The sticker must belong to a sticker set that was created by the bot. Returns *True* on success.

| Parameter | Type | Required | Description |
|---------------|--------------|----------|--|
| sticker | String | Yes | File identifier of the sticker |
| mask_position | MaskPosition | Optional | A JSON-serialized object with the position where the mask should be placed on faces. Omit the parameter to remove the mask position. |

setStickerSetTitle

Use this method to set the title of a created sticker set. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------|--------|----------|------------------------------------|
| name | String | Yes | Sticker set name |
| title | String | Yes | Sticker set title, 1–64 characters |

setStickerSetThumbnail

Use this method to set the thumbnail of a regular or mask sticker set. The format of the thumbnail file must match the format of the stickers in the set. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------|---------------------|----------|--|
| name | String | Yes | Sticker set name |
| user_id | Integer | Yes | User identifier of the sticker set owner |
| thumbnail | InputFile or String | Optional | A .WEBP or .PNG image with the thumbnail, must be up to 128 kilobytes in size and have a width and height of exactly 100px, or a .TGS animation with a thumbnail up to 32 kilobytes in size (see https://core.telegram.org/stickers#animation-requirements for animated sticker technical requirements), or a .WEBM video with the thumbnail up to 32 kilobytes in size; see https://core.telegram.org/stickers#video-requirements for video sticker technical requirements. Pass a <i>file_id</i> as a String to send a file that already exists on the Telegram servers, pass an HTTP URL as a String for Telegram to get a file from the Internet, or upload a new one using multipart/form-data. More information on Sending Files . Animated and video sticker set thumbnails can't be uploaded via HTTP URL. If omitted, then the thumbnail is dropped and the first sticker is used as the thumbnail. |
| format | String | Yes | Format of the thumbnail, must be one of "static" for a .WEBP or .PNG image, "animated" for a .TGS animation, or "video" for a .WEBM video |

setCustomEmojiStickerSetThumbnail

Use this method to set the thumbnail of a custom emoji sticker set. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------------|--------|----------|---|
| name | String | Yes | Sticker set name |
| custom_emoji_id | String | Optional | Custom emoji identifier of a sticker from the sticker set; pass an empty string to drop the thumbnail and use the first sticker as the thumbnail. |

deleteStickerSet

Use this method to delete a sticker set that was created by the bot. Returns *True* on success.

| Parameter | Type | Required | Description |
|-----------|--------|----------|------------------|
| name | String | Yes | Sticker set name |

Inline mode

The following methods and objects allow your bot to work in **inline mode**.

Please see our [Introduction to Inline bots](#) for more details.

To enable this option, send the `/setinline` command to `@BotFather` and provide the placeholder text that the user will see in the input field after typing your bot's name.

InlineQuery

This object represents an incoming inline query. When the user sends an empty query, your bot could return some default or trending results.

| Field | Type | Description |
|-----------|----------|---|
| id | String | Unique identifier for this query |
| from | User | Sender |
| query | String | Text of the query (up to 256 characters) |
| offset | String | Offset of the results to be returned, can be controlled by the bot |
| chat_type | String | <i>Optional.</i> Type of the chat from which the inline query was sent. Can be either "sender" for a private chat with the inline query sender, "private", "group", "supergroup", or "channel". The chat type should be always known for requests sent from official clients and most third-party clients, unless the request was sent from a secret chat |
| location | Location | <i>Optional.</i> Sender location, only for bots that request user location |

answerInlineQuery

Use this method to send answers to an inline query. On success, `True` is returned.

No more than 50 results per query are allowed.

| Parameter | Type | Required | Description |
|-----------------|--|----------|--|
| inline_query_id | String | Yes | Unique identifier for the answered query |
| results | Array of InlineQueryResult | Yes | A JSON-serialized array of results for the inline query |
| cache_time | Integer | Optional | The maximum amount of time in seconds that the result of the inline query may be cached on the server. Defaults to 300. |
| is_personal | Boolean | Optional | Pass <code>True</code> if results may be cached on the server side only for the user that sent the query. By default, results may be returned to any user who sends the same query. |
| next_offset | String | Optional | Pass the offset that a client should send in the next query with the same text to receive more results. Pass an empty string if there are no more results or if you don't support pagination. Offset length can't exceed 64 bytes. |
| button | InlineQueryResultsButton | Optional | A JSON-serialized object describing a button to be shown above inline query results |

InlineQueryResultsButton

This object represents a button to be shown above inline query results. You **must** use exactly one of the optional fields.

| Field | Type | Description |
|-----------------|----------------------------|--|
| text | String | Label text on the button |
| web_app | WebAppInfo | <i>Optional.</i> Description of the Web App that will be launched when the user presses the button. The Web App will be able to switch back to the inline mode using the method <code>switchInlineQuery</code> inside the Web App. |
| start_parameter | String | <i>Optional.</i> Deep-linking parameter for the /start message sent to the bot when a user presses the button. 1–64 characters, only <code>A-Z</code> , <code>a-z</code> , <code>0-9</code> , <code>_</code> and <code>-</code> are allowed. |

Example: An inline bot that sends YouTube videos can ask the user to connect the bot to their YouTube account to adapt search results accordingly. To do this, it displays a 'Connect your YouTube account' button above the results, or even before showing any. The user presses the button, switches to a private chat with the bot and, in doing so, passes a start parameter that instructs the bot to return an OAuth link. Once done, the bot can offer a `switch_inline` button so that the user can easily return to the chat where they wanted to use the bot's inline capabilities.

InlineQueryResult

This object represents one result of an inline query. Telegram clients currently support results of the following 20 types:

- [InlineQueryResultCachedAudio](#)
- [InlineQueryResultCachedDocument](#)
- [InlineQueryResultCachedGif](#)
- [InlineQueryResultCachedMpeg4Gif](#)
- [InlineQueryResultCachedPhoto](#)
- [InlineQueryResultCachedSticker](#)
- [InlineQueryResultCachedVideo](#)
- [InlineQueryResultCachedVoice](#)
- [InlineQueryResultArticle](#)
- [InlineQueryResultAudio](#)
- [InlineQueryResultContact](#)
- [InlineQueryResultGame](#)
- [InlineQueryResultDocument](#)
- [InlineQueryResultGif](#)
- [InlineQueryResultLocation](#)
- [InlineQueryResultMpeg4Gif](#)
- [InlineQueryResultPhoto](#)
- [InlineQueryResultVenue](#)
- [InlineQueryResultVideo](#)
- [InlineQueryResultVoice](#)

Note: All URLs passed in inline query results will be available to end users and therefore must be assumed to be `public`.

InlineQueryResultArticle

Represents a link to an article or web page.

| Field | Type | Description |
|-------|--------|--|
| type | String | Type of the result, must be <code>article</code> |

| Field | Type | Description |
|-----------------------|----------------------|--|
| id | String | Unique identifier for this result, 1–64 Bytes |
| title | String | Title of the result |
| input_message_content | InputMessageContent | Content of the message to be sent |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| url | String | <i>Optional.</i> URL of the result |
| description | String | <i>Optional.</i> Short description of the result |
| thumbnail_url | String | <i>Optional.</i> Url of the thumbnail for the result |
| thumbnail_width | Integer | <i>Optional.</i> Thumbnail width |
| thumbnail_height | Integer | <i>Optional.</i> Thumbnail height |

InlineQueryResultPhoto

Represents a link to a photo. By default, this photo will be sent by the user with optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the photo.

| Field | Type | Description |
|--------------------------|------------------------|---|
| type | String | Type of the result, must be <code>photo</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| photo_url | String | A valid URL of the photo. Photo must be in JPEG format. Photo size must not exceed 5MB |
| thumbnail_url | String | URL of the thumbnail for the photo |
| photo_width | Integer | <i>Optional.</i> Width of the photo |
| photo_height | Integer | <i>Optional.</i> Height of the photo |
| title | String | <i>Optional.</i> Title for the result |
| description | String | <i>Optional.</i> Short description of the result |
| caption | String | <i>Optional.</i> Caption of the photo to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the photo caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the photo |

InlineQueryResultGif

Represents a link to an animated GIF file. By default, this animated GIF file will be sent by the user with optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the animation.

| Field | Type | Description |
|--------------------------|------------------------|---|
| type | String | Type of the result, must be <code>gif</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| gif_url | String | A valid URL for the GIF file |
| gif_width | Integer | <i>Optional.</i> Width of the GIF |
| gif_height | Integer | <i>Optional.</i> Height of the GIF |
| gif_duration | Integer | <i>Optional.</i> Duration of the GIF in seconds |
| thumbnail_url | String | URL of the static (JPEG or GIF) or animated (MPEG4) thumbnail for the result |
| thumbnail_mime_type | String | <i>Optional.</i> MIME type of the thumbnail, must be one of “image/jpeg”, “image/gif”, or “video/mp4”. Defaults to “image/jpeg” |
| title | String | <i>Optional.</i> Title for the result |
| caption | String | <i>Optional.</i> Caption of the GIF file to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the GIF animation |

InlineQueryResultMpeg4Gif

Represents a link to a video animation (H.264/MPEG-4 AVC video without sound). By default, this animated MPEG-4 file will be sent by the user with optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the animation.

| Field | Type | Description |
|--------------------------|--|---|
| type | String | Type of the result, must be <code>mpeg4_gif</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| mpeg4_url | String | A valid URL for the MPEG4 file |
| mpeg4_width | Integer | <i>Optional.</i> Video width |
| mpeg4_height | Integer | <i>Optional.</i> Video height |
| mpeg4_duration | Integer | <i>Optional.</i> Video duration in seconds |
| thumbnail_url | String | URL of the static (JPEG or GIF) or animated (MPEG4) thumbnail for the result |
| thumbnail_mime_type | String | <i>Optional.</i> MIME type of the thumbnail, must be one of "image/jpeg", "image/gif", or "video/mp4". Defaults to "image/jpeg" |
| title | String | <i>Optional.</i> Title for the result |
| caption | String | <i>Optional.</i> Caption of the MPEG-4 file to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the video animation |

InlineQueryResultVideo

Represents a link to a page containing an embedded video player or a video file. By default, this video file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the video.

If an `InlineQueryResultVideo` message contains an embedded video (e.g., YouTube), you **must** replace its content using `input_message_content`.

| Field | Type | Description |
|--------------------------|--|---|
| type | String | Type of the result, must be <code>video</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| video_url | String | A valid URL for the embedded video player or video file |
| mime_type | String | MIME type of the content of the video URL, "text/html" or "video/mp4" |
| thumbnail_url | String | URL of the thumbnail (JPEG only) for the video |
| title | String | Title for the result |
| caption | String | <i>Optional.</i> Caption of the video to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the video caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| video_width | Integer | <i>Optional.</i> Video width |
| video_height | Integer | <i>Optional.</i> Video height |
| video_duration | Integer | <i>Optional.</i> Video duration in seconds |
| description | String | <i>Optional.</i> Short description of the result |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the video. This field is required if <code>InlineQueryResultVideo</code> is used to send an HTML-page as a result (e.g., a YouTube video). |

InlineQueryResultAudio

Represents a link to an MP3 audio file. By default, this audio file will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the audio.

| Field | Type | Description |
|-------|--------|--|
| type | String | Type of the result, must be <code>audio</code> |

| Field | Type | Description |
|-----------------------|--|---|
| id | String | Unique identifier for this result, 1–64 bytes |
| audio_url | String | A valid URL for the audio file |
| title | String | Title |
| caption | String | <i>Optional.</i> Caption, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the audio caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| performer | String | <i>Optional.</i> Performer |
| audio_duration | Integer | <i>Optional.</i> Audio duration in seconds |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the audio |

InlineQueryResultVoice

Represents a link to a voice recording in an .OGG container encoded with OPUS. By default, this voice recording will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the the voice message.

| Field | Type | Description |
|-----------------------|--|---|
| type | String | Type of the result, must be <code>voice</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| voice_url | String | A valid URL for the voice recording |
| title | String | Recording title |
| caption | String | <i>Optional.</i> Caption, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the voice message caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| voice_duration | Integer | <i>Optional.</i> Recording duration in seconds |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the voice recording |

InlineQueryResultDocument

Represents a link to a file. By default, this file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the file. Currently, only **.PDF** and **.ZIP** files can be sent using this method.

| Field | Type | Description |
|-----------------------|--|---|
| type | String | Type of the result, must be <code>document</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| title | String | Title for the result |
| caption | String | <i>Optional.</i> Caption of the document to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the document caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| document_url | String | A valid URL for the file |
| mime_type | String | MIME type of the content of the file, either “application/pdf” or “application/zip” |
| description | String | <i>Optional.</i> Short description of the result |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the file |
| thumbnail_url | String | <i>Optional.</i> URL of the thumbnail (JPEG only) for the file |
| thumbnail_width | Integer | <i>Optional.</i> Thumbnail width |
| thumbnail_height | Integer | <i>Optional.</i> Thumbnail height |

InlineQueryResultLocation

Represents a location on a map. By default, the location will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the location.

| Field | Type | Description |
|------------------------|----------------------|--|
| type | String | Type of the result, must be <i>location</i> |
| id | String | Unique identifier for this result, 1–64 Bytes |
| latitude | Float | Location latitude in degrees |
| longitude | Float | Location longitude in degrees |
| title | String | Location title |
| horizontal_accuracy | Float | <i>Optional.</i> The radius of uncertainty for the location, measured in meters; 0–1500 |
| live_period | Integer | <i>Optional.</i> Period in seconds during which the location can be updated, should be between 60 and 86400, or 0xFFFFFFFF for live locations that can be edited indefinitely. |
| heading | Integer | <i>Optional.</i> For live locations, a direction in which the user is moving, in degrees. Must be between 1 and 360 if specified. |
| proximity_alert_radius | Integer | <i>Optional.</i> For live locations, a maximum distance for proximity alerts about approaching another chat member, in meters. Must be between 1 and 100000 if specified. |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the location |
| thumbnail_url | String | <i>Optional.</i> Url of the thumbnail for the result |
| thumbnail_width | Integer | <i>Optional.</i> Thumbnail width |
| thumbnail_height | Integer | <i>Optional.</i> Thumbnail height |

InlineQueryResultVenue

Represents a venue. By default, the venue will be sent by the user. Alternatively, you can use *input_message_content* to send a message with the specified content instead of the venue.

| Field | Type | Description |
|-----------------------|----------------------|---|
| type | String | Type of the result, must be <i>venue</i> |
| id | String | Unique identifier for this result, 1–64 Bytes |
| latitude | Float | Latitude of the venue location in degrees |
| longitude | Float | Longitude of the venue location in degrees |
| title | String | Title of the venue |
| address | String | Address of the venue |
| foursquare_id | String | <i>Optional.</i> Foursquare identifier of the venue if known |
| foursquare_type | String | <i>Optional.</i> Foursquare type of the venue, if known. (For example, "arts_entertainment/default", "arts_entertainment/aquarium" or "food/icecream".) |
| google_place_id | String | <i>Optional.</i> Google Places identifier of the venue |
| google_place_type | String | <i>Optional.</i> Google Places type of the venue. (See supported types .) |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the venue |
| thumbnail_url | String | <i>Optional.</i> Url of the thumbnail for the result |
| thumbnail_width | Integer | <i>Optional.</i> Thumbnail width |
| thumbnail_height | Integer | <i>Optional.</i> Thumbnail height |

InlineQueryResultContact

Represents a contact with a phone number. By default, this contact will be sent by the user. Alternatively, you can use *input_message_content* to send a message with the specified content instead of the contact.

| Field | Type | Description |
|-----------------------|----------------------|---|
| type | String | Type of the result, must be <i>contact</i> |
| id | String | Unique identifier for this result, 1–64 Bytes |
| phone_number | String | Contact's phone number |
| first_name | String | Contact's first name |
| last_name | String | <i>Optional.</i> Contact's last name |
| vcard | String | <i>Optional.</i> Additional data about the contact in the form of a vCard, 0–2048 bytes |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the contact |

| Field | Type | Description |
|------------------|---------|--|
| thumbnail_url | String | <i>Optional.</i> Url of the thumbnail for the result |
| thumbnail_width | Integer | <i>Optional.</i> Thumbnail width |
| thumbnail_height | Integer | <i>Optional.</i> Thumbnail height |

InlineQueryResultGame

Represents a [Game](#).

| Field | Type | Description |
|-----------------|--------------------------------------|--|
| type | String | Type of the result, must be <code>game</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| game_short_name | String | Short name of the game |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |

InlineQueryResultCachedPhoto

Represents a link to a photo stored on the Telegram servers. By default, this photo will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the photo.

| Field | Type | Description |
|--------------------------|--|---|
| type | String | Type of the result, must be <code>photo</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| photo_file_id | String | A valid file identifier of the photo |
| title | String | <i>Optional.</i> Title for the result |
| description | String | <i>Optional.</i> Short description of the result |
| caption | String | <i>Optional.</i> Caption of the photo to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the photo caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the photo |

InlineQueryResultCachedGif

Represents a link to an animated GIF file stored on the Telegram servers. By default, this animated GIF file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with specified content instead of the animation.

| Field | Type | Description |
|--------------------------|--|---|
| type | String | Type of the result, must be <code>gif</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| gif_file_id | String | A valid file identifier for the GIF file |
| title | String | <i>Optional.</i> Title for the result |
| caption | String | <i>Optional.</i> Caption of the GIF file to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the GIF animation |

InlineQueryResultCachedMpeg4Gif

Represents a link to a video animation (H.264/MPEG-4 AVC video without sound) stored on the Telegram servers. By default, this animated MPEG-4 file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the animation.

| Field | Type | Description |
|---------------|--------|--|
| type | String | Type of the result, must be <code>mpeg4_gif</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| mpeg4_file_id | String | A valid file identifier for the MPEG4 file |

| Field | Type | Description |
|--------------------------|--|---|
| title | String | <i>Optional.</i> Title for the result |
| caption | String | <i>Optional.</i> Caption of the MPEG-4 file to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the video animation |

InlineQueryResultCachedSticker

Represents a link to a sticker stored on the Telegram servers. By default, this sticker will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the sticker.

| Field | Type | Description |
|-----------------------|--------------------------------------|---|
| type | String | Type of the result, must be <code>sticker</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| sticker_file_id | String | A valid file identifier of the sticker |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the sticker |

InlineQueryResultCachedDocument

Represents a link to a file stored on the Telegram servers. By default, this file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the file.

| Field | Type | Description |
|-----------------------|--|---|
| type | String | Type of the result, must be <code>document</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| title | String | Title for the result |
| document_file_id | String | A valid file identifier for the file |
| description | String | <i>Optional.</i> Short description of the result |
| caption | String | <i>Optional.</i> Caption of the document to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the document caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the file |

InlineQueryResultCachedVideo

Represents a link to a video file stored on the Telegram servers. By default, this video file will be sent by the user with an optional caption. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the video.

| Field | Type | Description |
|--------------------------|--|---|
| type | String | Type of the result, must be <code>video</code> |
| id | String | Unique identifier for this result, 1–64 bytes |
| video_file_id | String | A valid file identifier for the video file |
| title | String | Title for the result |
| description | String | <i>Optional.</i> Short description of the result |
| caption | String | <i>Optional.</i> Caption of the video to be sent, 0–1024 characters after entities parsing |
| parse_mode | String | <i>Optional.</i> Mode for parsing entities in the video caption. See formatting options for more details. |
| caption_entities | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| show_caption_above_media | Boolean | <i>Optional.</i> Pass <code>True</code> , if the caption must be shown above the message media |
| reply_markup | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| input_message_content | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the video |

InlineQueryResultCachedVoice

Represents a link to a voice message stored on the Telegram servers. By default, this voice message will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the voice message.

| Field | Type | Description |
|------------------------------------|--|---|
| <code>type</code> | String | Type of the result, must be <code>voice</code> |
| <code>id</code> | String | Unique identifier for this result, 1–64 bytes |
| <code>voice_file_id</code> | String | A valid file identifier for the voice message |
| <code>title</code> | String | Voice message title |
| <code>caption</code> | String | <i>Optional.</i> Caption, 0–1024 characters after entities parsing |
| <code>parse_mode</code> | String | <i>Optional.</i> Mode for parsing entities in the voice message caption. See formatting options for more details. |
| <code>caption_entities</code> | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| <code>reply_markup</code> | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| <code>input_message_content</code> | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the voice message |

InlineQueryResultCachedAudio

Represents a link to an MP3 audio file stored on the Telegram servers. By default, this audio file will be sent by the user. Alternatively, you can use `input_message_content` to send a message with the specified content instead of the audio.

| Field | Type | Description |
|------------------------------------|--|---|
| <code>type</code> | String | Type of the result, must be <code>audio</code> |
| <code>id</code> | String | Unique identifier for this result, 1–64 bytes |
| <code>audio_file_id</code> | String | A valid file identifier for the audio file |
| <code>caption</code> | String | <i>Optional.</i> Caption, 0–1024 characters after entities parsing |
| <code>parse_mode</code> | String | <i>Optional.</i> Mode for parsing entities in the audio caption. See formatting options for more details. |
| <code>caption_entities</code> | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in the caption, which can be specified instead of <code>parse_mode</code> |
| <code>reply_markup</code> | InlineKeyboardMarkup | <i>Optional.</i> Inline keyboard attached to the message |
| <code>input_message_content</code> | InputMessageContent | <i>Optional.</i> Content of the message to be sent instead of the audio |

InputMessageContent

This object represents the content of a message to be sent as a result of an inline query. Telegram clients currently support the following 5 types:

- [InputTextMessageContent](#)
- [InputLocationMessageContent](#)
- [InputVenueMessageContent](#)
- [InputContactMessageContent](#)
- [InputInvoiceMessageContent](#)

InputTextMessageContent

Represents the [content](#) of a text message to be sent as the result of an inline query.

| Field | Type | Description |
|-----------------------------------|--|--|
| <code>message_text</code> | String | Text of the message to be sent, 1–4096 characters |
| <code>parse_mode</code> | String | <i>Optional.</i> Mode for parsing entities in the message text. See formatting options for more details. |
| <code>entities</code> | Array of MessageEntity | <i>Optional.</i> List of special entities that appear in message text, which can be specified instead of <code>parse_mode</code> |
| <code>link_preview_options</code> | LinkPreviewOptions | <i>Optional.</i> Link preview generation options for the message |

InputLocationMessageContent

Represents the [content](#) of a location message to be sent as the result of an inline query.

| Field | Type | Description |
|-------------------------------------|---------|--|
| <code>latitude</code> | Float | Latitude of the location in degrees |
| <code>longitude</code> | Float | Longitude of the location in degrees |
| <code>horizontal_accuracy</code> | Float | <i>Optional.</i> The radius of uncertainty for the location, measured in meters; 0–1500 |
| <code>live_period</code> | Integer | <i>Optional.</i> Period in seconds during which the location can be updated, should be between 60 and 86400, or 0xFFFFFFFF for live locations that can be edited indefinitely. |
| <code>heading</code> | Integer | <i>Optional.</i> For live locations, a direction in which the user is moving, in degrees. Must be between 1 and 360 if specified. |
| <code>proximity_alert_radius</code> | Integer | <i>Optional.</i> For live locations, a maximum distance for proximity alerts about approaching another chat member, in meters. Must be between 1 and 100000 if specified. |

InputVenueMessageContent

Represents the [content](#) of a venue message to be sent as the result of an inline query.

| Field | Type | Description |
|-------------------|--------|---|
| latitude | Float | Latitude of the venue in degrees |
| longitude | Float | Longitude of the venue in degrees |
| title | String | Name of the venue |
| address | String | Address of the venue |
| foursquare_id | String | <i>Optional.</i> Foursquare identifier of the venue, if known |
| foursquare_type | String | <i>Optional.</i> Foursquare type of the venue, if known. (For example, "arts_entertainment/default", "arts_entertainment/aquarium" or "food/icecream".) |
| google_place_id | String | <i>Optional.</i> Google Places identifier of the venue |
| google_place_type | String | <i>Optional.</i> Google Places type of the venue. (See supported types .) |

InputContactMessageContent

Represents the [content](#) of a contact message to be sent as the result of an inline query.

| Field | Type | Description |
|--------------|--------|---|
| phone_number | String | Contact's phone number |
| first_name | String | Contact's first name |
| last_name | String | <i>Optional.</i> Contact's last name |
| vcard | String | <i>Optional.</i> Additional data about the contact in the form of a vCard, 0–2048 bytes |

InputInvoiceMessageContent

Represents the [content](#) of an invoice message to be sent as the result of an inline query.

| Field | Type | Description |
|-------------------------------|---------------------------------------|--|
| title | String | Product name, 1–32 characters |
| description | String | Product description, 1–255 characters |
| payload | String | Bot-defined invoice payload, 1–128 bytes. This will not be displayed to the user, use it for your internal processes. |
| provider_token | String | <i>Optional.</i> Payment provider token, obtained via @BotFather . Pass an empty string for payments in Telegram Stars . |
| currency | String | Three-letter ISO 4217 currency code, see more on currencies . Pass "XTR" for payments in Telegram Stars . |
| prices | Array of LabeledPrice | Price breakdown, a JSON-serialized list of components (e.g. product price, tax, discount, delivery cost, delivery tax, bonus, etc.). Must contain exactly one item for payments in Telegram Stars . |
| max_tip_amount | Integer | <i>Optional.</i> The maximum accepted amount for tips in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a maximum tip of US\$ 1.45 pass <code>max_tip_amount = 145</code> . See the <code>exp</code> parameter in currencies.json , it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). Defaults to 0. Not supported for payments in Telegram Stars . |
| suggested_tip_amounts | Array of Integer | <i>Optional.</i> A JSON-serialized array of suggested amounts of tip in the <i>smallest units</i> of the currency (integer, not float/double). At most 4 suggested tip amounts can be specified. The suggested tip amounts must be positive, passed in a strictly increased order and must not exceed <code>max_tip_amount</code> . |
| provider_data | String | <i>Optional.</i> A JSON-serialized object for data about the invoice, which will be shared with the payment provider. A detailed description of the required fields should be provided by the payment provider. |
| photo_url | String | <i>Optional.</i> URL of the product photo for the invoice. Can be a photo of the goods or a marketing image for a service. |
| photo_size | Integer | <i>Optional.</i> Photo size in bytes |
| photo_width | Integer | <i>Optional.</i> Photo width |
| photo_height | Integer | <i>Optional.</i> Photo height |
| need_name | Boolean | <i>Optional.</i> Pass <code>True</code> if you require the user's full name to complete the order. Ignored for payments in Telegram Stars . |
| need_phone_number | Boolean | <i>Optional.</i> Pass <code>True</code> if you require the user's phone number to complete the order. Ignored for payments in Telegram Stars . |
| need_email | Boolean | <i>Optional.</i> Pass <code>True</code> if you require the user's email address to complete the order. Ignored for payments in Telegram Stars . |
| need_shipping_address | Boolean | <i>Optional.</i> Pass <code>True</code> if you require the user's shipping address to complete the order. Ignored for payments in Telegram Stars . |
| send_phone_number_to_provider | Boolean | <i>Optional.</i> Pass <code>True</code> if the user's phone number should be sent to the provider. Ignored for payments in Telegram Stars . |
| send_email_to_provider | Boolean | <i>Optional.</i> Pass <code>True</code> if the user's email address should be sent to the provider. Ignored for payments in Telegram Stars . |
| is_flexible | Boolean | <i>Optional.</i> Pass <code>True</code> if the final price depends on the shipping method. Ignored for payments in Telegram Stars . |

ChosenInlineResult

Represents a [result](#) of an inline query that was chosen by the user and sent to their chat partner.

| Field | Type | Description |
|-------------------|----------|---|
| result_id | String | The unique identifier for the result that was chosen |
| from | User | The user that chose the result |
| location | Location | <i>Optional.</i> Sender location, only for bots that require user location |
| inline_message_id | String | <i>Optional.</i> Identifier of the sent inline message. Available only if there is an inline keyboard attached to the message. Will be also received in callback queries and can be used to edit the message. |
| query | String | The query that was used to obtain the result |

Note: It is necessary to enable [inline feedback](#) via [@BotFather](#) in order to receive these objects in updates.

answerWebAppQuery

Use this method to set the result of an interaction with a [Web App](#) and send a corresponding message on behalf of the user to the chat from which the query originated. On success, a [SentWebAppMessage](#) object is returned.

| Parameter | Type | Required | Description |
|------------------|-------------------|----------|--|
| web_app_query_id | String | Yes | Unique identifier for the query to be answered |
| result | InlineQueryResult | Yes | A JSON-serialized object describing the message to be sent |

SentWebAppMessage

Describes an inline message sent by a [Web App](#) on behalf of a user.

| Field | Type | Description |
|-------------------|--------|--|
| inline_message_id | String | <i>Optional.</i> Identifier of the sent inline message. Available only if there is an inline keyboard attached to the message. |

savePreparedInlineMessage

Stores a message that can be sent by a user of a Mini App. Returns a [PreparedInlineMessage](#) object.

| Parameter | Type | Required | Description |
|---------------------|-------------------|----------|---|
| user_id | Integer | Yes | Unique identifier of the target user that can use the prepared message |
| result | InlineQueryResult | Yes | A JSON-serialized object describing the message to be sent |
| allow_user_chats | Boolean | Optional | Pass <code>True</code> if the message can be sent to private chats with users |
| allow_bot_chats | Boolean | Optional | Pass <code>True</code> if the message can be sent to private chats with bots |
| allow_group_chats | Boolean | Optional | Pass <code>True</code> if the message can be sent to group and supergroup chats |
| allow_channel_chats | Boolean | Optional | Pass <code>True</code> if the message can be sent to channel chats |

PreparedInlineMessage

Describes an inline message to be sent by a user of a Mini App.

| Field | Type | Description |
|-----------------|---------|--|
| id | String | Unique identifier of the prepared message |
| expiration_date | Integer | Expiration date of the prepared message, in Unix time. Expired prepared messages can no longer be used |

Payments

Your bot can accept payments from Telegram users. Please see the [introduction to payments](#) for more details on the process and how to set up payments for your bot.

sendInvoice

Use this method to send invoices. 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>) |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| direct_messages_topic_id | Integer | Optional | Identifier of the direct messages topic to which the message will be sent; required if the message is sent to a direct messages chat |
| title | String | Yes | Product name, 1–32 characters |
| description | String | Yes | Product description, 1–255 characters |
| payload | String | Yes | Bot-defined invoice payload, 1–128 bytes. This will not be displayed to the user, use it for your internal processes. |
| provider_token | String | Optional | Payment provider token, obtained via @BotFather . Pass an empty string for payments in Telegram Stars . |
| currency | String | Yes | Three-letter ISO 4217 currency code, see more on currencies . Pass "XTR" for payments in Telegram Stars . |
| prices | Array of LabeledPrice | Yes | Price breakdown, a JSON-serialized list of components (e.g. product price, tax, discount, delivery cost, delivery tax, bonus, etc.). Must contain exactly one item for payments in Telegram Stars . |

| Parameter | Type | Required | Description |
|-------------------------------|-------------------------|----------|--|
| max_tip_amount | Integer | Optional | The maximum accepted amount for tips in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a maximum tip of US\$ 1.45 pass <code>max_tip_amount = 145</code> . See the <code>exp</code> parameter in currencies.json , it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). Defaults to 0. Not supported for payments in Telegram Stars . |
| suggested_tip_amounts | Array of Integer | Optional | A JSON-serialized array of suggested amounts of tips in the <i>smallest units</i> of the currency (integer, not float/double). At most 4 suggested tip amounts can be specified. The suggested tip amounts must be positive, passed in a strictly increased order and must not exceed <code>max_tip_amount</code> . |
| start_parameter | String | Optional | Unique deep-linking parameter. If left empty, forwarded copies of the sent message will have a <i>Pay</i> button, allowing multiple users to pay directly from the forwarded message, using the same invoice. If non-empty, forwarded copies of the sent message will have a <i>URL</i> button with a deep link to the bot (instead of a <i>Pay</i> button), with the value used as the start parameter |
| provider_data | String | Optional | JSON-serialized data about the invoice, which will be shared with the payment provider. A detailed description of required fields should be provided by the payment provider. |
| photo_url | String | Optional | URL of the product photo for the invoice. Can be a photo of the goods or a marketing image for a service. People like it better when they see what they are paying for. |
| photo_size | Integer | Optional | Photo size in bytes |
| photo_width | Integer | Optional | Photo width |
| photo_height | Integer | Optional | Photo height |
| need_name | Boolean | Optional | Pass <code>True</code> if you require the user's full name to complete the order. Ignored for payments in Telegram Stars . |
| need_phone_number | Boolean | Optional | Pass <code>True</code> if you require the user's phone number to complete the order. Ignored for payments in Telegram Stars . |
| need_email | Boolean | Optional | Pass <code>True</code> if you require the user's email address to complete the order. Ignored for payments in Telegram Stars . |
| need_shipping_address | Boolean | Optional | Pass <code>True</code> if you require the user's shipping address to complete the order. Ignored for payments in Telegram Stars . |
| send_phone_number_to_provider | Boolean | Optional | Pass <code>True</code> if the user's phone number should be sent to the provider. Ignored for payments in Telegram Stars . |
| send_email_to_provider | Boolean | Optional | Pass <code>True</code> if the user's email address should be sent to the provider. Ignored for payments in Telegram Stars . |
| is_flexible | Boolean | Optional | Pass <code>True</code> if the final price depends on the shipping method. Ignored for payments in Telegram Stars . |
| disable_notification | Boolean | Optional | Sends the message <i>silently</i> . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| suggested_post_parameters | SuggestedPostParameters | Optional | A JSON-serialized object containing the parameters of the suggested post to send; for direct messages chats only. If the message is sent as a reply to another suggested post, then that suggested post is automatically declined. |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard . If empty, one 'Pay <code>total_price</code> ' button will be shown. If not empty, the first button must be a Pay button. |

createInvoiceLink

Use this method to create a link for an invoice. Returns the created invoice link as `String` on success.

| Parameter | Type | Required | Description |
|------------------------|-----------------------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the link will be created. For payments in Telegram Stars only. |
| title | String | Yes | Product name, 1–32 characters |
| description | String | Yes | Product description, 1–255 characters |
| payload | String | Yes | Bot-defined invoice payload, 1–128 bytes. This will not be displayed to the user, use it for your internal processes. |
| provider_token | String | Optional | Payment provider token, obtained via @BotFather . Pass an empty string for payments in Telegram Stars . |
| currency | String | Yes | Three-letter ISO 4217 currency code, see more on currencies . Pass "XTR" for payments in Telegram Stars . |
| prices | Array of LabeledPrice | Yes | Price breakdown, a JSON-serialized list of components (e.g. product price, tax, discount, delivery cost, delivery tax, bonus, etc.). Must contain exactly one item for payments in Telegram Stars . |
| subscription_period | Integer | Optional | The number of seconds the subscription will be active for before the next payment. The currency must be set to "XTR" (Telegram Stars) if the parameter is used. Currently, it must always be 2592000 (30 days) if specified. Any |

| Parameter | Type | Required | Description |
|-------------------------------|------------------|----------|---|
| | | | number of subscriptions can be active for a given bot at the same time, including multiple concurrent subscriptions from the same user. Subscription price must no exceed 10000 Telegram Stars. |
| max_tip_amount | Integer | Optional | The maximum accepted amount for tips in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a maximum tip of <code>US\$ 1.45</code> pass <code>max_tip_amount = 145</code> . See the <code>exp</code> parameter in <code>currencies.json</code> , it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). Defaults to 0. Not supported for payments in Telegram Stars . |
| suggested_tip_amounts | Array of Integer | Optional | A JSON-serialized array of suggested amounts of tips in the <i>smallest units</i> of the currency (integer, not float/double). At most 4 suggested tip amounts can be specified. The suggested tip amounts must be positive, passed in a strictly increased order and must not exceed <code>max_tip_amount</code> . |
| provider_data | String | Optional | JSON-serialized data about the invoice, which will be shared with the payment provider. A detailed description of required fields should be provided by the payment provider. |
| photo_url | String | Optional | URL of the product photo for the invoice. Can be a photo of the goods or a marketing image for a service. |
| photo_size | Integer | Optional | Photo size in bytes |
| photo_width | Integer | Optional | Photo width |
| photo_height | Integer | Optional | Photo height |
| need_name | Boolean | Optional | Pass <code>True</code> if you require the user's full name to complete the order. Ignored for payments in Telegram Stars . |
| need_phone_number | Boolean | Optional | Pass <code>True</code> if you require the user's phone number to complete the order. Ignored for payments in Telegram Stars . |
| need_email | Boolean | Optional | Pass <code>True</code> if you require the user's email address to complete the order. Ignored for payments in Telegram Stars . |
| need_shipping_address | Boolean | Optional | Pass <code>True</code> if you require the user's shipping address to complete the order. Ignored for payments in Telegram Stars . |
| send_phone_number_to_provider | Boolean | Optional | Pass <code>True</code> if the user's phone number should be sent to the provider. Ignored for payments in Telegram Stars . |
| send_email_to_provider | Boolean | Optional | Pass <code>True</code> if the user's email address should be sent to the provider. Ignored for payments in Telegram Stars . |
| is_flexible | Boolean | Optional | Pass <code>True</code> if the final price depends on the shipping method. Ignored for payments in Telegram Stars . |

answerShippingQuery

If you sent an invoice requesting a shipping address and the parameter `is_flexible` was specified, the Bot API will send an `Update` with a `shipping_query` field to the bot. Use this method to reply to shipping queries. On success, `True` is returned.

| Parameter | Type | Required | Description |
|-------------------|---|----------|--|
| shipping_query_id | String | Yes | Unique identifier for the query to be answered |
| ok | Boolean | Yes | Pass <code>True</code> if delivery to the specified address is possible and <code>False</code> if there are any problems (for example, if delivery to the specified address is not possible) |
| shipping_options | Array of ShippingOption | Optional | Required if <code>ok</code> is <code>True</code> . A JSON-serialized array of available shipping options. |
| error_message | String | Optional | Required if <code>ok</code> is <code>False</code> . Error message in human readable form that explains why it is impossible to complete the order (e.g. "Sorry, delivery to your desired address is unavailable"). Telegram will display this message to the user. |

answerPreCheckoutQuery

Once the user has confirmed their payment and shipping details, the Bot API sends the final confirmation in the form of an `Update` with the field `pre_checkout_query`. Use this method to respond to such pre-checkout queries. On success, `True` is returned. **Note:** The Bot API must receive an answer within 10 seconds after the pre-checkout query was sent.

| Parameter | Type | Required | Description |
|-----------------------|---------|----------|---|
| pre_checkout_query_id | String | Yes | Unique identifier for the query to be answered |
| ok | Boolean | Yes | Specify <code>True</code> if everything is alright (goods are available, etc.) and the bot is ready to proceed with the order. Use <code>False</code> if there are any problems. |
| error_message | String | Optional | Required if <code>ok</code> is <code>False</code> . Error message in human readable form that explains the reason for failure to proceed with the checkout (e.g. "Sorry, somebody just bought the last of our amazing black T-shirts while you were busy filling out your payment details. Please choose a different color or garment!"). Telegram will display this message to the user. |

getMyStarBalance

A method to get the current Telegram Stars balance of the bot. Requires no parameters. On success, returns a `StarAmount` object.

getStarTransactions

Returns the bot's Telegram Star transactions in chronological order. On success, returns a `StarTransactions` object.

| Parameter | Type | Required | Description |
|-----------|---------|----------|---|
| offset | Integer | Optional | Number of transactions to skip in the response |
| limit | Integer | Optional | The maximum number of transactions to be retrieved. Values between 1–100 are accepted. Defaults to 100. |

refundStarPayment

Refunds a successful payment in Telegram Stars. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------------|---------|----------|---|
| user_id | Integer | Yes | Identifier of the user whose payment will be refunded |
| telegram_payment_charge_id | String | Yes | Telegram payment identifier |

editUserStarSubscription

Allows the bot to cancel or re-enable extension of a subscription paid in Telegram Stars. Returns `True` on success.

| Parameter | Type | Required | Description |
|----------------------------|---------|----------|---|
| user_id | Integer | Yes | Identifier of the user whose subscription will be edited |
| telegram_payment_charge_id | String | Yes | Telegram payment identifier for the subscription |
| is_canceled | Boolean | Yes | Pass <code>True</code> to cancel extension of the user subscription; the subscription must be active up to the end of the current subscription period. Pass <code>False</code> to allow the user to re-enable a subscription that was previously canceled by the bot. |

LabeledPrice

This object represents a portion of the price for goods or services.

| Field | Type | Description |
|--------|---------|---|
| label | String | Portion label |
| amount | Integer | Price of the product in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a price of <code>US\$ 1.45</code> pass <code>amount = 145</code> . See the <code>exp</code> parameter in <code>currencies.json</code> , it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). |

Invoice

This object contains basic information about an invoice.

| Field | Type | Description |
|-----------------|---------|--|
| title | String | Product name |
| description | String | Product description |
| start_parameter | String | Unique bot deep-linking parameter that can be used to generate this invoice |
| currency | String | Three-letter ISO 4217 currency code, or "XTR" for payments in Telegram Stars |
| total_amount | Integer | Total price in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a price of <code>US\$ 1.45</code> pass <code>amount = 145</code> . See the <code>exp</code> parameter in <code>currencies.json</code> , it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). |

ShippingAddress

This object represents a shipping address.

| Field | Type | Description |
|--------------|--------|--|
| country_code | String | Two-letter ISO 3166-1 alpha-2 country code |
| state | String | State, if applicable |
| city | String | City |
| street_line1 | String | First line for the address |
| street_line2 | String | Second line for the address |
| post_code | String | Address post code |

OrderInfo

This object represents information about an order.

| Field | Type | Description |
|------------------|-----------------|--|
| name | String | <i>Optional.</i> User name |
| phone_number | String | <i>Optional.</i> User's phone number |
| email | String | <i>Optional.</i> User email |
| shipping_address | ShippingAddress | <i>Optional.</i> User shipping address |

ShippingOption

This object represents one shipping option.

| Field | Type | Description |
|--------|-----------------------|----------------------------|
| id | String | Shipping option identifier |
| title | String | Option title |
| prices | Array of LabeledPrice | List of price portions |

SuccessfulPayment

This object contains basic information about a successful payment. Note that if the buyer initiates a chargeback with the relevant payment provider following this transaction, the funds may be debited from your balance. This is outside of Telegram's control.

| Field | Type | Description |
|------------------------------|-----------|---|
| currency | String | Three-letter ISO 4217 currency code, or "XTR" for payments in Telegram Stars |
| total_amount | Integer | Total price in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a price of US\$ 1.45 pass amount = 145 . See the exp parameter in currencies.json, it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). |
| invoice_payload | String | Bot-specified invoice payload |
| subscription_expiration_date | Integer | <i>Optional.</i> Expiration date of the subscription, in Unix time; for recurring payments only |
| is_recurring | True | <i>Optional.</i> True, if the payment is a recurring payment for a subscription |
| is_first_recurring | True | <i>Optional.</i> True, if the payment is the first payment for a subscription |
| shipping_option_id | String | <i>Optional.</i> Identifier of the shipping option chosen by the user |
| order_info | OrderInfo | <i>Optional.</i> Order information provided by the user |
| telegram_payment_charge_id | String | Telegram payment identifier |
| provider_payment_charge_id | String | Provider payment identifier |

RefundedPayment

This object contains basic information about a refunded payment.

| Field | Type | Description |
|----------------------------|---------|---|
| currency | String | Three-letter ISO 4217 currency code, or "XTR" for payments in Telegram Stars. Currently, always "XTR" |
| total_amount | Integer | Total refunded price in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a price of US\$ 1.45 , total_amount = 145 . See the exp parameter in currencies.json, it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). |
| invoice_payload | String | Bot-specified invoice payload |
| telegram_payment_charge_id | String | Telegram payment identifier |
| provider_payment_charge_id | String | <i>Optional.</i> Provider payment identifier |

ShippingQuery

This object contains information about an incoming shipping query.

| Field | Type | Description |
|------------------|-----------------|---------------------------------|
| id | String | Unique query identifier |
| from | User | User who sent the query |
| invoice_payload | String | Bot-specified invoice payload |
| shipping_address | ShippingAddress | User specified shipping address |

PreCheckoutQuery

This object contains information about an incoming pre-checkout query.

| Field | Type | Description |
|--------------------|-----------|---|
| id | String | Unique query identifier |
| from | User | User who sent the query |
| currency | String | Three-letter ISO 4217 currency code, or "XTR" for payments in Telegram Stars |
| total_amount | Integer | Total price in the <i>smallest units</i> of the currency (integer, not float/double). For example, for a price of US\$ 1.45 pass amount = 145 . See the exp parameter in currencies.json, it shows the number of digits past the decimal point for each currency (2 for the majority of currencies). |
| invoice_payload | String | Bot-specified invoice payload |
| shipping_option_id | String | <i>Optional.</i> Identifier of the shipping option chosen by the user |
| order_info | OrderInfo | <i>Optional.</i> Order information provided by the user |

PaidMediaPurchased

This object contains information about a paid media purchase.

| Field | Type | Description |
|--------------------|--------|----------------------------------|
| from | User | User who purchased the media |
| paid_media_payload | String | Bot-specified paid media payload |

RevenueWithdrawalState

This object describes the state of a revenue withdrawal operation. Currently, it can be one of

- [RevenueWithdrawalStatePending](#)
- [RevenueWithdrawalStateSucceeded](#)
- [RevenueWithdrawalStateFailed](#)

RevenueWithdrawalStatePending

The withdrawal is in progress.

| Field | Type | Description |
|-------|--------|-------------------------------------|
| type | String | Type of the state, always "pending" |

RevenueWithdrawalStateSucceeded

The withdrawal succeeded.

| Field | Type | Description |
|-------|---------|--|
| type | String | Type of the state, always "succeeded" |
| date | Integer | Date the withdrawal was completed in Unix time |
| url | String | An HTTPS URL that can be used to see transaction details |

RevenueWithdrawalStateFailed

The withdrawal failed and the transaction was refunded.

| Field | Type | Description |
|-------|--------|------------------------------------|
| type | String | Type of the state, always "failed" |

AffiliateInfo

Contains information about the affiliate that received a commission via this transaction.

| Field | Type | Description |
|----------------------|---------|---|
| affiliate_user | User | <i>Optional.</i> The bot or the user that received an affiliate commission if it was received by a bot or a user |
| affiliate_chat | Chat | <i>Optional.</i> The chat that received an affiliate commission if it was received by a chat |
| commission_per_mille | Integer | The number of Telegram Stars received by the affiliate for each 1000 Telegram Stars received by the bot from referred users |
| amount | Integer | Integer amount of Telegram Stars received by the affiliate from the transaction, rounded to 0; can be negative for refunds |
| nanostar_amount | Integer | <i>Optional.</i> The number of 1/1000000000 shares of Telegram Stars received by the affiliate; from -999999999 to 999999999; can be negative for refunds |

TransactionPartner

This object describes the source of a transaction, or its recipient for outgoing transactions. Currently, it can be one of

- [TransactionPartnerUser](#)
- [TransactionPartnerChat](#)
- [TransactionPartnerAffiliateProgram](#)
- [TransactionPartnerFragment](#)
- [TransactionPartnerTelegramAds](#)
- [TransactionPartnerTelegramApi](#)
- [TransactionPartnerOther](#)

TransactionPartnerUser

Describes a transaction with a user.

| Field | Type | Description |
|-------------------------------|--------------------|--|
| type | String | Type of the transaction partner, always "user" |
| transaction_type | String | Type of the transaction, currently one of "invoice_payment" for payments via invoices, "paid_media_payment" for payments for paid media, "gift_purchase" for gifts sent by the bot, "premium_purchase" for Telegram Premium subscriptions gifted by the bot, "business_account_transfer" for direct transfers from managed business accounts |
| user | User | Information about the user |
| affiliate | AffiliateInfo | <i>Optional.</i> Information about the affiliate that received a commission via this transaction. Can be available only for "invoice_payment" and "paid_media_payment" transactions. |
| invoice_payload | String | <i>Optional.</i> Bot-specified invoice payload. Can be available only for "invoice_payment" transactions. |
| subscription_period | Integer | <i>Optional.</i> The duration of the paid subscription. Can be available only for "invoice_payment" transactions. |
| paid_media | Array of PaidMedia | <i>Optional.</i> Information about the paid media bought by the user; for "paid_media_payment" transactions only |
| paid_media_payload | String | <i>Optional.</i> Bot-specified paid media payload. Can be available only for "paid_media_payment" transactions. |
| gift | Gift | <i>Optional.</i> The gift sent to the user by the bot; for "gift_purchase" transactions only |
| premium_subscription_duration | Integer | <i>Optional.</i> Number of months the gifted Telegram Premium subscription will be active for; for "premium_purchase" transactions only |

TransactionPartnerChat

Describes a transaction with a chat.

| Field | Type | Description |
|-------|--------|---|
| type | String | Type of the transaction partner, always "chat" |
| chat | Chat | Information about the chat |
| gift | Gift | <i>Optional.</i> The gift sent to the chat by the bot |

TransactionPartnerAffiliateProgram

Describes the affiliate program that issued the affiliate commission received via this transaction.

| Field | Type | Description |
|----------------------|---------|---|
| type | String | Type of the transaction partner, always "affiliate_program" |
| sponsor_user | User | <i>Optional.</i> Information about the bot that sponsored the affiliate program |
| commission_per_mille | Integer | The number of Telegram Stars received by the bot for each 1000 Telegram Stars received by the affiliate program sponsor from referred users |

TransactionPartnerFragment

Describes a withdrawal transaction with Fragment.

| Field | Type | Description |
|------------------|------------------------|--|
| type | String | Type of the transaction partner, always "fragment" |
| withdrawal_state | RevenueWithdrawalState | <i>Optional.</i> State of the transaction if the transaction is outgoing |

TransactionPartnerTelegramAds

Describes a withdrawal transaction to the Telegram Ads platform.

| Field | Type | Description |
|-------|--------|--|
| type | String | Type of the transaction partner, always "telegram_ads" |

TransactionPartnerTelegramApi

Describes a transaction with payment for paid broadcasting.

| Field | Type | Description |
|---------------|---------|--|
| type | String | Type of the transaction partner, always "telegram_api" |
| request_count | Integer | The number of successful requests that exceeded regular limits and were therefore billed |

TransactionPartnerOther

Describes a transaction with an unknown source or recipient.

| Field | Type | Description |
|-------|--------|---|
| type | String | Type of the transaction partner, always "other" |

StarTransaction

Describes a Telegram Star transaction. Note that if the buyer initiates a chargeback with the payment provider from whom they acquired Stars (e.g., Apple, Google) following this transaction, the refunded Stars will be deducted from the bot's balance. This is outside of Telegram's control.

| Field | Type | Description |
|-----------------|--------------------|--|
| id | String | Unique identifier of the transaction. Coincides with the identifier of the original transaction for refund transactions. Coincides with <code>SuccessfulPayment.telegram_payment_charge_id</code> for successful incoming payments from users. |
| amount | Integer | Integer amount of Telegram Stars transferred by the transaction |
| nanostar_amount | Integer | <i>Optional.</i> The number of 1/1000000000 shares of Telegram Stars transferred by the transaction; from 0 to 999999999 |
| date | Integer | Date the transaction was created in Unix time |
| source | TransactionPartner | <i>Optional.</i> Source of an incoming transaction (e.g., a user purchasing goods or services, Fragment refunding a failed withdrawal). Only for incoming transactions |
| receiver | TransactionPartner | <i>Optional.</i> Receiver of an outgoing transaction (e.g., a user for a purchase refund, Fragment for a withdrawal). Only for outgoing transactions |

StarTransactions

Contains a list of Telegram Star transactions.

| Field | Type | Description |
|--------------|--------------------------|--------------------------|
| transactions | Array of StarTransaction | The list of transactions |

Telegram Passport

Telegram Passport is a unified authorization method for services that require personal identification. Users can upload their documents once, then instantly share their data with services that require real-world ID (finance, ICOs, etc.). Please see the [manual](#) for details.

PassportData

Describes Telegram Passport data shared with the bot by the user.

| Field | Type | Description |
|-------------|---|--|
| data | Array of EncryptedPassportElement | Array with information about documents and other Telegram Passport elements that was shared with the bot |
| credentials | EncryptedCredentials | Encrypted credentials required to decrypt the data |

PassportFile

This object represents a file uploaded to Telegram Passport. Currently all Telegram Passport files are in JPEG format when decrypted and don't exceed 10MB.

| Field | Type | Description |
|----------------|---------|--|
| file_id | String | Identifier for this file, which can be used to download or reuse the file |
| file_unique_id | String | Unique identifier for this file, which is supposed to be the same over time and for different bots. Can't be used to download or reuse the file. |
| file_size | Integer | File size in bytes |
| file_date | Integer | Unix time when the file was uploaded |

EncryptedPassportElement

Describes documents or other Telegram Passport elements shared with the bot by the user.

| Field | Type | Description |
|--------------|---------------------------------------|--|
| type | String | Element type. One of "personal_details", "passport", "driver_license", "identity_card", "internal_passport", "address", "utility_bill", "bank_statement", "rental_agreement", "passport_registration", "temporary_registration", "phone_number", "email". |
| data | String | <i>Optional.</i> Base64-encoded encrypted Telegram Passport element data provided by the user; available only for "personal_details", "passport", "driver_license", "identity_card", "internal_passport" and "address" types. Can be decrypted and verified using the accompanying EncryptedCredentials . |
| phone_number | String | <i>Optional.</i> User's verified phone number; available only for "phone_number" type |
| email | String | <i>Optional.</i> User's verified email address; available only for "email" type |
| files | Array of PassportFile | <i>Optional.</i> Array of encrypted files with documents provided by the user; available only for "utility_bill", "bank_statement", "rental_agreement", "passport_registration" and "temporary_registration" types. Files can be decrypted and verified using the accompanying EncryptedCredentials . |
| front_side | PassportFile | <i>Optional.</i> Encrypted file with the front side of the document, provided by the user; available only for "passport", "driver_license", "identity_card" and "internal_passport". The file can be decrypted and verified using the accompanying EncryptedCredentials . |
| reverse_side | PassportFile | <i>Optional.</i> Encrypted file with the reverse side of the document, provided by the user; available only for "driver_license" and "identity_card". The file can be decrypted and verified using the accompanying EncryptedCredentials . |
| selfie | PassportFile | <i>Optional.</i> Encrypted file with the selfie of the user holding a document, provided by the user; available if requested for "passport", "driver_license", "identity_card" and "internal_passport". The file can be decrypted and verified using the accompanying EncryptedCredentials . |
| translation | Array of PassportFile | <i>Optional.</i> Array of encrypted files with translated versions of documents provided by the user; available if requested for "passport", "driver_license", "identity_card", "internal_passport", "utility_bill", "bank_statement", "rental_agreement", "passport_registration" and "temporary_registration" types. Files can be decrypted and verified using the accompanying EncryptedCredentials . |
| hash | String | Base64-encoded element hash for using in PassportElementErrorUnspecified |

EncryptedCredentials

Describes data required for decrypting and authenticating [EncryptedPassportElement](#). See the [Telegram Passport Documentation](#) for a complete description of the data decryption and authentication processes.

| Field | Type | Description |
|--------|--------|---|
| data | String | Base64-encoded encrypted JSON-serialized data with unique user's payload, data hashes and secrets required for EncryptedPassportElement decryption and authentication |
| hash | String | Base64-encoded data hash for data authentication |
| secret | String | Base64-encoded secret, encrypted with the bot's public RSA key, required for data decryption |

setPassportDataErrors

Informs a user that some of the Telegram Passport elements they provided contains errors. The user will not be able to re-submit their Passport to you until the errors are fixed (the contents of the field for which you returned the error must change). Returns *True* on success.

Use this if the data submitted by the user doesn't satisfy the standards your service requires for any reason. For example, if a birthday date seems invalid, a submitted document is blurry, a scan shows evidence of tampering, etc. Supply some details in the error message to make sure the user knows how to correct the issues.

| Parameter | Type | Required | Description |
|-----------|---|----------|---|
| user_id | Integer | Yes | User identifier |
| errors | Array of PassportElementError | Yes | A JSON-serialized array describing the errors |

PassportElementError

This object represents an error in the Telegram Passport element which was submitted that should be resolved by the user. It should be one of:

- `PassportElementErrorDataField`
- `PassportElementErrorFrontSide`
- `PassportElementErrorReverseSide`
- `PassportElementErrorSelfie`
- `PassportElementErrorFile`
- `PassportElementErrorFiles`
- `PassportElementErrorTranslationFile`
- `PassportElementErrorTranslationFiles`
- `PassportElementErrorUnspecified`

PassportElementErrorDataField

Represents an issue in one of the data fields that was provided by the user. The error is considered resolved when the field's value changes.

| Field | Type | Description |
|-------------------------|---------------------|---|
| <code>source</code> | <code>String</code> | Error source, must be <code>data</code> |
| <code>type</code> | <code>String</code> | The section of the user's Telegram Passport which has the error, one of "personal_details", "passport", "driver_license", "identity_card", "internal_passport", "address" |
| <code>field_name</code> | <code>String</code> | Name of the data field which has the error |
| <code>data_hash</code> | <code>String</code> | Base64-encoded data hash |
| <code>message</code> | <code>String</code> | Error message |

PassportElementErrorFrontSide

Represents an issue with the front side of a document. The error is considered resolved when the file with the front side of the document changes.

| Field | Type | Description |
|------------------------|---------------------|--|
| <code>source</code> | <code>String</code> | Error source, must be <code>front_side</code> |
| <code>type</code> | <code>String</code> | The section of the user's Telegram Passport which has the issue, one of "passport", "driver_license", "identity_card", "internal_passport" |
| <code>file_hash</code> | <code>String</code> | Base64-encoded hash of the file with the front side of the document |
| <code>message</code> | <code>String</code> | Error message |

PassportElementErrorReverseSide

Represents an issue with the reverse side of a document. The error is considered resolved when the file with reverse side of the document changes.

| Field | Type | Description |
|------------------------|---------------------|---|
| <code>source</code> | <code>String</code> | Error source, must be <code>reverse_side</code> |
| <code>type</code> | <code>String</code> | The section of the user's Telegram Passport which has the issue, one of "driver_license", "identity_card" |
| <code>file_hash</code> | <code>String</code> | Base64-encoded hash of the file with the reverse side of the document |
| <code>message</code> | <code>String</code> | Error message |

PassportElementErrorSelfie

Represents an issue with the selfie with a document. The error is considered resolved when the file with the selfie changes.

| Field | Type | Description |
|------------------------|---------------------|--|
| <code>source</code> | <code>String</code> | Error source, must be <code>selfie</code> |
| <code>type</code> | <code>String</code> | The section of the user's Telegram Passport which has the issue, one of "passport", "driver_license", "identity_card", "internal_passport" |
| <code>file_hash</code> | <code>String</code> | Base64-encoded hash of the file with the selfie |
| <code>message</code> | <code>String</code> | Error message |

PassportElementErrorFile

Represents an issue with a document scan. The error is considered resolved when the file with the document scan changes.

| Field | Type | Description |
|------------------------|---------------------|---|
| <code>source</code> | <code>String</code> | Error source, must be <code>file</code> |
| <code>type</code> | <code>String</code> | The section of the user's Telegram Passport which has the issue, one of "utility_bill", "bank_statement", "rental_agreement", "passport_registration", "temporary_registration" |
| <code>file_hash</code> | <code>String</code> | Base64-encoded file hash |
| <code>message</code> | <code>String</code> | Error message |

PassportElementErrorFiles

Represents an issue with a list of scans. The error is considered resolved when the list of files containing the scans changes.

| Field | Type | Description |
|---------------------|---------------------|--|
| <code>source</code> | <code>String</code> | Error source, must be <code>files</code> |

| Field | Type | Description |
|-------------|-----------------|---|
| type | String | The section of the user's Telegram Passport which has the issue, one of "utility_bill", "bank_statement", "rental_agreement", "passport_registration", "temporary_registration" |
| file_hashes | Array of String | List of base64-encoded file hashes |
| message | String | Error message |

PassportElementErrorTranslationFile

Represents an issue with one of the files that constitute the translation of a document. The error is considered resolved when the file changes.

| Field | Type | Description |
|-----------|--------|---|
| source | String | Error source, must be <i>translation_file</i> |
| type | String | Type of element of the user's Telegram Passport which has the issue, one of "passport", "driver_license", "identity_card", "internal_passport", "utility_bill", "bank_statement", "rental_agreement", "passport_registration", "temporary_registration" |
| file_hash | String | Base64-encoded file hash |
| message | String | Error message |

PassportElementErrorTranslationFiles

Represents an issue with the translated version of a document. The error is considered resolved when a file with the document translation change.

| Field | Type | Description |
|-------------|-----------------|---|
| source | String | Error source, must be <i>translation_files</i> |
| type | String | Type of element of the user's Telegram Passport which has the issue, one of "passport", "driver_license", "identity_card", "internal_passport", "utility_bill", "bank_statement", "rental_agreement", "passport_registration", "temporary_registration" |
| file_hashes | Array of String | List of base64-encoded file hashes |
| message | String | Error message |

PassportElementErrorUnspecified

Represents an issue in an unspecified place. The error is considered resolved when new data is added.

| Field | Type | Description |
|--------------|--------|---|
| source | String | Error source, must be <i>unspecified</i> |
| type | String | Type of element of the user's Telegram Passport which has the issue |
| element_hash | String | Base64-encoded element hash |
| message | String | Error message |

Games

Your bot can offer users **HTML5 games** to play solo or to compete against each other in groups and one-on-one chats. Create games via **@BotFather** using the `/newgame` command. Please note that this kind of power requires responsibility: you will need to accept the terms for each game that your bots will be offering.

- Games are a new type of content on Telegram, represented by the [Game](#) and [InlineQueryResultGame](#) objects.
- Once you've created a game via [BotFather](#), you can send games to chats as regular messages using the [sendGame](#) method, or use [inline mode](#) with [InlineQueryResultGame](#).
- If you send the game message without any buttons, it will automatically have a 'Play *GameName*' button. When this button is pressed, your bot gets a [CallbackQuery](#) with the *game_short_name* of the requested game. You provide the correct URL for this particular user and the app opens the game in the in-app browser.
- You can manually add multiple buttons to your game message. Please note that the first button in the first row **must always** launch the game, using the field *callback_game* in [InlineKeyboardButton](#). You can add extra buttons according to taste: e.g., for a description of the rules, or to open the game's official community.
- To make your game more attractive, you can upload a GIF animation that demonstrates the game to the users via [BotFather](#) (see [Lumberjack](#) for example).
- A game message will also display high scores for the current chat. Use [setGameScore](#) to post high scores to the chat with the game, add the *disable_edit_message* parameter to disable automatic update of the message with the current scoreboard.
- Use [getGameHighScores](#) to get data for in-game high score tables.
- You can also add an extra [sharing button](#) for users to share their best score to different chats.
- For examples of what can be done using this new stuff, check the [@gamebot](#) and [@gamee](#) bots.

sendGame

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

| Parameter | Type | Required | Description |
|------------------------|---------|----------|--|
| business_connection_id | String | Optional | Unique identifier of the business connection on behalf of which the message will be sent |
| chat_id | Integer | Yes | Unique identifier for the target chat. Games can't be sent to channel direct messages chats and channel chats. |
| message_thread_id | Integer | Optional | Unique identifier for the target message thread (topic) of the forum; for forum supergroups only |
| game_short_name | String | Yes | Short name of the game, serves as the unique identifier for the game. Set up your games via @BotFather . |
| disable_notification | Boolean | Optional | Sends the message silently . Users will receive a notification with no sound. |
| protect_content | Boolean | Optional | Protects the contents of the sent message from forwarding and saving |

| Parameter | Type | Required | Description |
|----------------------|----------------------|----------|---|
| allow_paid_broadcast | Boolean | Optional | Pass <code>True</code> to allow up to 1000 messages per second, ignoring broadcasting limits for a fee of 0.1 Telegram Stars per message. The relevant Stars will be withdrawn from the bot's balance |
| message_effect_id | String | Optional | Unique identifier of the message effect to be added to the message; for private chats only |
| reply_parameters | ReplyParameters | Optional | Description of the message to reply to |
| reply_markup | InlineKeyboardMarkup | Optional | A JSON-serialized object for an inline keyboard . If empty, one 'Play game_title' button will be shown. If not empty, the first button must launch the game. |

Game

This object represents a game. Use [BotFather](#) to create and edit games, their short names will act as unique identifiers.

| Field | Type | Description |
|---------------|------------------------|---|
| title | String | Title of the game |
| description | String | Description of the game |
| photo | Array of PhotoSize | Photo that will be displayed in the game message in chats. |
| text | String | <i>Optional.</i> Brief description of the game or high scores included in the game message. Can be automatically edited to include current high scores for the game when the bot calls <code>setGameScore</code> , or manually edited using <code>editMessageText</code> . 0–4096 characters. |
| text_entities | Array of MessageEntity | <i>Optional.</i> Special entities that appear in <code>text</code> , such as usernames, URLs, bot commands, etc. |
| animation | Animation | <i>Optional.</i> Animation that will be displayed in the game message in chats. Upload via BotFather |

CallbackGame

A placeholder, currently holds no information. Use [BotFather](#) to set up your game.

setGameScore

Use this method to set the score of the specified user in a game message. On success, if the message is not an inline message, the [Message](#) is returned, otherwise `True` is returned. Returns an error, if the new score is not greater than the user's current score in the chat and `force` is `False`.

| Parameter | Type | Required | Description |
|----------------------|---------|----------|--|
| user_id | Integer | Yes | User identifier |
| score | Integer | Yes | New score, must be non-negative |
| force | Boolean | Optional | Pass <code>True</code> if the high score is allowed to decrease. This can be useful when fixing mistakes or banning cheaters |
| disable_edit_message | Boolean | Optional | Pass <code>True</code> if the game message should not be automatically edited to include the current scoreboard |
| chat_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the sent message |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |

getGameHighScores

Use this method to get data for high score tables. Will return the score of the specified user and several of their neighbors in a game. Returns an Array of [GameHighScore](#) objects.

This method will currently return scores for the target user, plus two of their closest neighbors on each side. Will also return the top three users if the user and their neighbors are not among them. Please note that this behavior is subject to change.

| Parameter | Type | Required | Description |
|-------------------|---------|----------|--|
| user_id | Integer | Yes | Target user id |
| chat_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Unique identifier for the target chat |
| message_id | Integer | Optional | Required if <code>inline_message_id</code> is not specified. Identifier of the sent message |
| inline_message_id | String | Optional | Required if <code>chat_id</code> and <code>message_id</code> are not specified. Identifier of the inline message |

GameHighScore

This object represents one row of the high scores table for a game.

| Field | Type | Description |
|----------|---------|---|
| position | Integer | Position in high score table for the game |
| user | User | User |
| score | Integer | Score |

And that's about all we've got for now.

If you've got any questions, please check out our [Bot FAQ](#) »

Telegram

Telegram is a cloud-based mobile and desktop messaging app with a focus on security and speed.

About

- [FAQ](#)
- [Privacy](#)
- [Press](#)

Mobile Apps

- [iPhone/iPad](#)
- [Android](#)
- [Mobile Web](#)

Desktop Apps

- [PC/Mac/Linux](#)
- [macOS](#)
- [Web-browser](#)

Platform

- [API](#)
- [Translations](#)
- [Instant View](#)

About

Blog

Press

Moderation

[Go up](#)