```
{
 "info": {
  "_postman_id": "ec4f32ad-a516-4545-8daa-d88b7be8d2b1",
  "name": "Telegram Bot API",
  "description": "Auto-generated Postman collection by TGScraper.",
  "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
  "_exporter_id": "40011479",
  "_collection_link": "https://www.postman.com/satellite-administrator-38214876/telegram-api/collection/qejha5x/telegram-bot-api?action=share&source=collection
 },
 "item": [
  {
   "name": "getUpdates",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "offset",
       "description": "Optional. Identifier of the first update to be returned. Must be greater by one than the highest among the identifiers of previously rec
       "type": "text",
       "disabled": true
      },
      {
       "key": "limit",
       "value": "100",
       "description": "Optional. Limits the number of updates to be retrieved. Values between 1-100 are accepted. Defaults to 100.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "timeout",
       "description": "Optional. Timeout in seconds for long polling. Defaults to 0, i.e. usual short polling. Should be positive, short polling should be used
       "type": "text",
       "disabled": true
      },
      {
       "key": "allowed_updates",
       "description": "Optional. A JSON-serialized list of the update types you want your bot to receive. For example, specify [\"message\", \"edited_channel_p
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getUpdates",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "getUpdates"
     ]
    },
    "description": "Use this method to receive incoming updates using long polling (wiki). Returns an Array of Update objects."
   },
   "response": []
  },
  {
   "name": "setWebhook",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "url",
       "description": "Required. HTTPS URL to send updates to. Use an empty string to remove webhook integration",
       "type": "text"
      },
      {
       "key": "certificate",
       "description": "Optional. Upload your public key certificate so that the root certificate in use can be checked. See our self-signed guide for details."
       "type": "text",
       "disabled": true
      },
      {
       "key": "ip_address",
       "description": "Optional. The fixed IP address which will be used to send webhook requests instead of the IP address resolved through DNS",
       "type": "text",
       "disabled": true
      },
      {
       "key": "max_connections",
       "value": "40",
       "description": "Optional. The maximum allowed number of simultaneous HTTPS connections to the webhook for update delivery, 1-100. Defaults to 40. Use lo
       "type": "text",
       "disabled": true
      },
      {
       "key": "allowed_updates",
       "description": "Optional. A JSON-serialized list of the update types you want your bot to receive. For example, specify [\"message\", \"edited_channel_p
       "type": "text",
       "disabled": true
      },
      {
       "key": "drop_pending_updates",
       "description": "Optional. Pass True to drop all pending updates",
       "type": "text",
       "disabled": true
      },
       {
        "key": "secret_token",
        "description": "Optional. A secret token to be sent in a header â€œX-Telegram-Bot-Api-Secret-Tokenâ€  in every webhook request, 1-256 characters. Only c
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setWebhook",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setWebhook"
      ]
     },
     "description": "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 sen
    },
    "response": []
   },
   {
    "name": "deleteWebhook",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "drop_pending_updates",
        "description": "Optional. Pass True to drop all pending updates",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/deleteWebhook",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "deleteWebhook"
      ]
     },
     "description": "Use this method to remove webhook integration if you decide to switch back to getUpdates. Returns True on success."
    },
    "response": []
   },
   {
    "name": "getWebhookInfo",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": []
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getWebhookInfo",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "getWebhookInfo"
      ]
     },
     "description": "Use this method to get current webhook status. Requires no parameters. On success, returns a WebhookInfo object. If the bot is using getUpd
    },
    "response": []
   },
   {
    "name": "getMe",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": []
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getMe",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "getMe"
      ]
     },
     "description": "A simple method for testing your bot's authentication token. Requires no parameters. Returns basic information about the bot in form of a U
    },
    "response": []
   },
   {
    "name": "logOut",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": []
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/logOut",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "logOut"
      ]
     },
     "description": "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 local
    },
    "response": []
   },
   {
    "name": "close",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": []
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/close",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "close"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "sendMessage",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "text",
        "description": "Required. Text of the message to be sent, 1-4096 characters after entities parsing",
        "type": "text"
       },
       {
        "key": "parse_mode",
        "description": "Optional. Mode for parsing entities in the message text. See formatting options for more details.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "entities",
        "description": "Optional. A JSON-serialized list of special entities that appear in message text, which can be specified instead of parse_mode",
        "type": "text",
        "disabled": true
       },
       {
        "key": "link_preview_options",
        "description": "Optional. Link preview generation options for the message",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendMessage",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendMessage"
      ]
     },
     "description": "Use this method to send text messages. On success, the sent Message is returned."
    },
    "response": []
   },
   {
    "name": "forwardMessage",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "from_chat_id",
        "description": "Required. Unique identifier for the chat where the original message was sent (or channel username in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the forwarded message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "message_id",
        "description": "Required. Message identifier in the chat specified in from_chat_id",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/forwardMessage",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "forwardMessage"
      ]
     },
     "description": "Use this method to forward messages of any kind. Service messages and messages with protected content can't be forwarded. On success, the s
    },
    "response": []
   },
   {
    "name": "forwardMessages",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
         "type": "text"
        },
        {
         "key": "message_thread_id",
         "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
         "type": "text",
         "disabled": true
        },
        {
         "key": "from_chat_id",
         "description": "Required. Unique identifier for the chat where the original messages were sent (or channel username in the format @channelusername)",
         "type": "text"
        },
        {
         "key": "message_ids",
         "description": "Required. Identifiers of 1-100 messages in the chat from_chat_id to forward. The identifiers must be specified in a strictly increasing
         "type": "text"
        },
        {
         "key": "disable_notification",
         "description": "Optional. Sends the messages silently. Users will receive a notification with no sound.",
         "type": "text",
         "disabled": true
        },
        {
         "key": "protect_content",
         "description": "Optional. Protects the contents of the forwarded messages from forwarding and saving",
         "type": "text",
         "disabled": true
        }
       ]
      },
      "url": {
       "raw": "https://api.telegram.org/bot{{token}}/forwardMessages",
       "protocol": "https",
       "host": [
        "api",
        "telegram",
        "org"
       ],
       "path": [
        "bot{{token}}",
        "forwardMessages"
       ]
      },
      "description": "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.
     },
     "response": []
    },
    {
     "name": "copyMessage",
     "request": {
      "method": "POST",
      "header": [],
      "body": {
       "mode": "formdata",
       "formdata": [
        {
         "key": "chat_id",
         "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
         "type": "text"
        },
        {
         "key": "message_thread_id",
         "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
         "type": "text",
         "disabled": true
        },
        {
         "key": "from_chat_id",
         "description": "Required. Unique identifier for the chat where the original message was sent (or channel username in the format @channelusername)",
         "type": "text"
        },
        {
         "key": "message_id",
         "description": "Required. Message identifier in the chat specified in from_chat_id",
         "type": "text"
        },
        {
         "key": "caption",
         "description": "Optional. New caption for media, 0-1024 characters after entities parsing. If not specified, the original caption is kept",
         "type": "text",
         "disabled": true
        },
        {
         "key": "parse_mode",
         "description": "Optional. Mode for parsing entities in the new caption. See formatting options for more details.",
         "type": "text",
         "disabled": true
        },
        {
         "key": "caption_entities",
         "description": "Optional. A JSON-serialized list of special entities that appear in the new caption, which can be specified instead of parse_mode",
         "type": "text",
         "disabled": true
        },
        {
         "key": "disable_notification",
         "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
         "type": "text",
         "disabled": true
        },
        {
         "key": "protect_content",
         "description": "Optional. Protects the contents of the sent message from forwarding and saving",
         "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/copyMessage",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "copyMessage"
     ]
    },
    "description": "Use this method to copy messages of any kind. Service messages, giveaway messages, giveaway winners messages, and invoice messages can't be
   },
   "response": []
  },
  {
   "name": "copyMessages",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "from_chat_id",
       "description": "Required. Unique identifier for the chat where the original messages were sent (or channel username in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_ids",
       "description": "Required. Identifiers of 1-100 messages in the chat from_chat_id to copy. The identifiers must be specified in a strictly increasing ord
       "type": "text"
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the messages silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent messages from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "remove_caption",
       "description": "Optional. Pass True to copy the messages without their captions",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/copyMessages",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "copyMessages"
     ]
    },
    "description": "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 message
   },
   "response": []
  },
  {
   "name": "sendPhoto",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "photo",
        "description": "Required. 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
        "type": "text"
       },
       {
        "key": "caption",
        "description": "Optional. Photo caption (may also be used when resending photos by file_id), 0-1024 characters after entities parsing",
        "type": "text",
        "disabled": true
       },
       {
        "key": "parse_mode",
        "description": "Optional. Mode for parsing entities in the photo caption. See formatting options for more details.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "caption_entities",
        "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
        "type": "text",
        "disabled": true
       },
       {
        "key": "has_spoiler",
        "description": "Optional. Pass True if the photo needs to be covered with a spoiler animation",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendPhoto",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendPhoto"
      ]
     },
     "description": "Use this method to send photos. On success, the sent Message is returned."
    },
    "response": []
   },
   {
    "name": "sendAudio",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "audio",
        "description": "Required. Audio file to send. Pass a file_id as String to send an audio file that exists on the Telegram servers (recommended), pass an 
       "type": "text"
      },
      {
       "key": "caption",
       "description": "Optional. Audio caption, 0-1024 characters after entities parsing",
       "type": "text",
       "disabled": true
      },
      {
       "key": "parse_mode",
       "description": "Optional. Mode for parsing entities in the audio caption. See formatting options for more details.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "caption_entities",
       "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
       "type": "text",
       "disabled": true
      },
      {
       "key": "duration",
       "description": "Optional. Duration of the audio in seconds",
       "type": "text",
       "disabled": true
      },
      {
       "key": "performer",
       "description": "Optional. Performer",
       "type": "text",
       "disabled": true
      },
      {
       "key": "title",
       "description": "Optional. Track name",
       "type": "text",
       "disabled": true
      },
      {
       "key": "thumbnail",
       "description": "Optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendAudio",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendAudio"
     ]
    },
    "description": "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 .M
   },
   "response": []
  },
  {
   "name": "sendDocument",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
       {
        "key": "document",
        "description": "Required. 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
        "type": "text"
       },
       {
        "key": "thumbnail",
        "description": "Optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should
        "type": "text",
        "disabled": true
       },
       {
        "key": "caption",
        "description": "Optional. Document caption (may also be used when resending documents by file_id), 0-1024 characters after entities parsing",
        "type": "text",
        "disabled": true
       },
       {
        "key": "parse_mode",
        "description": "Optional. Mode for parsing entities in the document caption. See formatting options for more details.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "caption_entities",
        "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_content_type_detection",
        "description": "Optional. Disables automatic server-side content type detection for files uploaded using multipart/form-data",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendDocument",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendDocument"
      ]
     },
     "description": "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 i
    },
    "response": []
   },
   {
    "name": "sendVideo",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "video",
        "description": "Required. 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
        "type": "text"
       },
       {
        "key": "duration",
        "description": "Optional. Duration of sent video in seconds",
        "type": "text",
       "disabled": true
      },
      {
       "key": "width",
       "description": "Optional. Video width",
       "type": "text",
       "disabled": true
      },
      {
       "key": "height",
       "description": "Optional. Video height",
       "type": "text",
       "disabled": true
      },
      {
       "key": "thumbnail",
       "description": "Optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should
       "type": "text",
       "disabled": true
      },
      {
       "key": "caption",
       "description": "Optional. Video caption (may also be used when resending videos by file_id), 0-1024 characters after entities parsing",
       "type": "text",
       "disabled": true
      },
      {
       "key": "parse_mode",
       "description": "Optional. Mode for parsing entities in the video caption. See formatting options for more details.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "caption_entities",
       "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
       "type": "text",
       "disabled": true
      },
      {
       "key": "has_spoiler",
       "description": "Optional. Pass True if the video needs to be covered with a spoiler animation",
       "type": "text",
       "disabled": true
      },
      {
       "key": "supports_streaming",
       "description": "Optional. Pass True if the uploaded video is suitable for streaming",
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendVideo",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendVideo"
     ]
    },
    "description": "Use this method to send video files, Telegram clients support MPEG4 videos (other formats may be sent as Document). On success, the sent Me
   },
   "response": []
  },
  {
   "name": "sendAnimation",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
    {
     "key": "message_thread_id",
     "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
     "type": "text",
     "disabled": true
    },
    {
     "key": "animation",
     "description": "Required. Animation to send. Pass a file_id as String to send an animation that exists on the Telegram servers (recommended), pass an HT
     "type": "text"
    },
    {
     "key": "duration",
     "description": "Optional. Duration of sent animation in seconds",
     "type": "text",
     "disabled": true
    },
    {
     "key": "width",
     "description": "Optional. Animation width",
     "type": "text",
     "disabled": true
    },
    {
     "key": "height",
     "description": "Optional. Animation height",
     "type": "text",
     "disabled": true
    },
    {
     "key": "thumbnail",
     "description": "Optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should
     "type": "text",
     "disabled": true
    },
    {
     "key": "caption",
     "description": "Optional. Animation caption (may also be used when resending animation by file_id), 0-1024 characters after entities parsing",
     "type": "text",
     "disabled": true
    },
    {
     "key": "parse_mode",
     "description": "Optional. Mode for parsing entities in the animation caption. See formatting options for more details.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "caption_entities",
     "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
     "type": "text",
     "disabled": true
    },
    {
     "key": "has_spoiler",
     "description": "Optional. Pass True if the animation needs to be covered with a spoiler animation",
     "type": "text",
     "disabled": true
    },
    {
     "key": "disable_notification",
     "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "protect_content",
     "description": "Optional. Protects the contents of the sent message from forwarding and saving",
     "type": "text",
     "disabled": true
    },
    {
     "key": "reply_parameters",
     "description": "Optional. Description of the message to reply to",
     "type": "text",
     "disabled": true
    },
    {
     "key": "reply_markup",
     "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/sendAnimation",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "sendAnimation"
   ]
  },
  "description": "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 c
 },
 "response": []
},
{
 "name": "sendVoice",
 "request": {
  "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "voice",
        "description": "Required. Audio file to send. Pass a file_id as String to send a file that exists on the Telegram servers (recommended), pass an HTTP UR
        "type": "text"
       },
       {
        "key": "caption",
        "description": "Optional. Voice message caption, 0-1024 characters after entities parsing",
        "type": "text",
        "disabled": true
       },
       {
        "key": "parse_mode",
        "description": "Optional. Mode for parsing entities in the voice message caption. See formatting options for more details.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "caption_entities",
        "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
        "type": "text",
        "disabled": true
       },
       {
        "key": "duration",
        "description": "Optional. Duration of the voice message in seconds",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendVoice",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendVoice"
      ]
     },
     "description": "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 a
    },
    "response": []
   },
   {
    "name": "sendVideoNote",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "video_note",
        "description": "Required. Video note to send. Pass a file_id as String to send a video note that exists on the Telegram servers (recommended) or upload
        "type": "text"
       },
       {
        "key": "duration",
        "description": "Optional. Duration of sent video in seconds",
        "type": "text",
        "disabled": true
       },
       {
        "key": "length",
        "description": "Optional. Video width and height, i.e. diameter of the video message",
        "type": "text",
        "disabled": true
       },
       {
        "key": "thumbnail",
        "description": "Optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendVideoNote",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendVideoNote"
      ]
     },
     "description": "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 succes
    },
    "response": []
   },
   {
    "name": "sendMediaGroup",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "media",
        "description": "Required. A JSON-serialized array describing messages to be sent, must include 2-10 items",
        "type": "text"
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends messages silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent messages from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
      "disabled": true
     }
    ]
   },
   "url": {
    "raw": "https://api.telegram.org/bot{{token}}/sendMediaGroup",
    "protocol": "https",
    "host": [
     "api",
     "telegram",
     "org"
    ],
    "path": [
     "bot{{token}}",
     "sendMediaGroup"
    ]
   },
   "description": "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 alb
  },
  "response": []
 },
 {
  "name": "sendLocation",
  "request": {
   "method": "POST",
   "header": [],
   "body": {
    "mode": "formdata",
    "formdata": [
     {
      "key": "chat_id",
      "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
      "type": "text"
     },
     {
      "key": "message_thread_id",
      "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
      "type": "text",
      "disabled": true
     },
     {
      "key": "latitude",
      "description": "Required. Latitude of the location",
      "type": "text"
     },
     {
      "key": "longitude",
      "description": "Required. Longitude of the location",
      "type": "text"
     },
     {
      "key": "horizontal_accuracy",
      "description": "Optional. The radius of uncertainty for the location, measured in meters; 0-1500",
      "type": "text",
      "disabled": true
     },
     {
      "key": "live_period",
      "description": "Optional. Period in seconds for which the location will be updated (see Live Locations, should be between 60 and 86400.",
      "type": "text",
      "disabled": true
     },
     {
      "key": "heading",
      "description": "Optional. For live locations, a direction in which the user is moving, in degrees. Must be between 1 and 360 if specified.",
      "type": "text",
      "disabled": true
     },
     {
      "key": "proximity_alert_radius",
      "description": "Optional. For live locations, a maximum distance for proximity alerts about approaching another chat member, in meters. Must be between
      "type": "text",
      "disabled": true
     },
     {
      "key": "disable_notification",
      "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
      "type": "text",
      "disabled": true
     },
     {
      "key": "protect_content",
      "description": "Optional. Protects the contents of the sent message from forwarding and saving",
      "type": "text",
      "disabled": true
     },
     {
      "key": "reply_parameters",
      "description": "Optional. Description of the message to reply to",
      "type": "text",
      "disabled": true
     },
     {
      "key": "reply_markup",
      "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
      "type": "text",
      "disabled": true
     }
    ]
   },
   "url": {
    "raw": "https://api.telegram.org/bot{{token}}/sendLocation",
    "protocol": "https",
    "host": [
     "api",
     "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendLocation"
     ]
    },
    "description": "Use this method to send point on the map. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "sendVenue",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "latitude",
       "description": "Required. Latitude of the venue",
       "type": "text"
      },
      {
       "key": "longitude",
       "description": "Required. Longitude of the venue",
       "type": "text"
      },
      {
       "key": "title",
       "description": "Required. Name of the venue",
       "type": "text"
      },
      {
       "key": "address",
       "description": "Required. Address of the venue",
       "type": "text"
      },
      {
       "key": "foursquare_id",
       "description": "Optional. Foursquare identifier of the venue",
       "type": "text",
       "disabled": true
      },
      {
       "key": "foursquare_type",
       "description": "Optional. Foursquare type of the venue, if known. (For example, â€œarts_entertainment/defaultâ€ , â€œarts_entertainment/aquariumâ€  or â
       "type": "text",
       "disabled": true
      },
      {
       "key": "google_place_id",
       "description": "Optional. Google Places identifier of the venue",
       "type": "text",
       "disabled": true
      },
      {
       "key": "google_place_type",
       "description": "Optional. Google Places type of the venue. (See supported types.)",
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendVenue",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendVenue"
     ]
    },
    "description": "Use this method to send information about a venue. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "sendContact",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "phone_number",
       "description": "Required. Contact's phone number",
       "type": "text"
      },
      {
       "key": "first_name",
       "description": "Required. Contact's first name",
       "type": "text"
      },
      {
       "key": "last_name",
       "description": "Optional. Contact's last name",
       "type": "text",
       "disabled": true
      },
      {
       "key": "vcard",
       "description": "Optional. Additional data about the contact in the form of a vCard, 0-2048 bytes",
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendContact",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendContact"
     ]
    },
    "description": "Use this method to send phone contacts. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "sendPoll",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "question",
       "description": "Required. Poll question, 1-300 characters",
       "type": "text"
      },
      {
       "key": "options",
       "description": "Required. A JSON-serialized list of answer options, 2-10 strings 1-100 characters each",
       "type": "text"
      },
      {
       "key": "is_anonymous",
       "value": "1",
       "description": "Optional. True, if the poll needs to be anonymous, defaults to True",
       "type": "text",
       "disabled": true
      },
      {
       "key": "type",
       "value": "regular",
       "description": "Optional. Poll type, â€œquizâ€  or â€œregularâ€ , defaults to â€œregularâ€ ",
       "type": "text",
       "disabled": true
      },
      {
       "key": "allows_multiple_answers",
       "description": "Optional. True, if the poll allows multiple answers, ignored for polls in quiz mode, defaults to False",
       "type": "text",
       "disabled": true
      },
      {
       "key": "correct_option_id",
       "description": "Optional. 0-based identifier of the correct answer option, required for polls in quiz mode",
       "type": "text",
       "disabled": true
      },
      {
       "key": "explanation",
       "description": "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 wit
       "type": "text",
       "disabled": true
      },
      {
       "key": "explanation_parse_mode",
       "description": "Optional. Mode for parsing entities in the explanation. See formatting options for more details.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "explanation_entities",
       "description": "Optional. A JSON-serialized list of special entities that appear in the poll explanation, which can be specified instead of parse_mode",
       "type": "text",
       "disabled": true
      },
      {
       "key": "open_period",
       "description": "Optional. Amount of time in seconds the poll will be active after creation, 5-600. Can't be used together with close_date.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "close_date",
       "description": "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 
       "type": "text",
       "disabled": true
      },
      {
       "key": "is_closed",
       "description": "Optional. Pass True if the poll needs to be immediately closed. This can be useful for poll preview.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendPoll",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendPoll"
     ]
    },
    "description": "Use this method to send a native poll. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "sendDice",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "emoji",
       "value": "ðŸŽ²",
       "description": "Optional. Emoji on which the dice throw animation is based. Currently, must be one of â€œðŸŽ²â€ , â€œðŸŽ¯â€ , â€œðŸ €â€ , â€œâš½â€ , â€œ
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendDice",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendDice"
     ]
    },
    "description": "Use this method to send an animated emoji that will display a random value. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "sendChatAction",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread; supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "action",
       "description": "Required. Type of action to broadcast. Choose one, depending on what the user is about to receive: typing for text messages, upload_phot
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendChatAction",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendChatAction"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "setMessageReaction",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_id",
       "description": "Required. Identifier of the target message. If the message belongs to a media group, the reaction is set to the first non-deleted messag
       "type": "text"
      },
      {
       "key": "reaction",
       "description": "Optional. New list of reaction types to set on the message. Currently, as non-premium users, bots can set up to one reaction per message
       "type": "text",
       "disabled": true
      },
      {
       "key": "is_big",
       "description": "Optional. Pass True to set the reaction with a big animation",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setMessageReaction",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setMessageReaction"
     ]
    },
    "description": "Use this method to change the chosen reactions on a message. Service messages can't be reacted to. Automatically forwarded messages from a
   },
   "response": []
  },
  {
   "name": "getUserProfilePhotos",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      },
      {
       "key": "offset",
       "description": "Optional. Sequential number of the first photo to be returned. By default, all photos are returned.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "limit",
       "value": "100",
       "description": "Optional. Limits the number of photos to be retrieved. Values between 1-100 are accepted. Defaults to 100.",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getUserProfilePhotos",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "getUserProfilePhotos"
     ]
    },
    "description": "Use this method to get a list of profile pictures for a user. Returns a UserProfilePhotos object."
   },
   "response": []
  },
  {
   "name": "getFile",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "file_id",
       "description": "Required. File identifier to get information about",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getFile",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "getFile"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "banChatMember",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target group or username of the target supergroup or channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      },
      {
       "key": "until_date",
       "description": "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 curr
       "type": "text",
       "disabled": true
      },
      {
       "key": "revoke_messages",
       "description": "Optional. Pass True to delete all messages from the chat for the user that is being removed. If False, the user will be able to see mess
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/banChatMember",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "banChatMember"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "unbanChatMember",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target group or username of the target supergroup or channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      },
      {
       "key": "only_if_banned",
       "description": "Optional. Do nothing if the user is not banned",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/unbanChatMember",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "unbanChatMember"
     ]
    },
    "description": "Use this method to unban a previously banned user in a supergroup or channel. The user will not return to the group or channel automaticall
   },
   "response": []
  },
  {
   "name": "restrictChatMember",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      },
      {
       "key": "permissions",
       "description": "Required. A JSON-serialized object for new user permissions",
       "type": "text"
      },
      {
       "key": "use_independent_chat_permissions",
       "description": "Optional. Pass True if chat permissions are set independently. Otherwise, the can_send_other_messages and can_add_web_page_previews perm
       "type": "text",
       "disabled": true
      },
      {
       "key": "until_date",
       "description": "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 se
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/restrictChatMember",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "restrictChatMember"
     ]
    },
    "description": "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 a
   },
   "response": []
  },
  {
   "name": "promoteChatMember",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      },
      {
       "key": "is_anonymous",
       "description": "Optional. Pass True if the administrator's presence in the chat is hidden",
       "type": "text",
       "disabled": true
       },
       {
        "key": "can_manage_chat",
        "description": "Optional. Pass True if the administrator can access the chat event log, boost list in channels, see channel members, report spam message
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_delete_messages",
        "description": "Optional. Pass True if the administrator can delete messages of other users",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_manage_video_chats",
        "description": "Optional. Pass True if the administrator can manage video chats",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_restrict_members",
        "description": "Optional. Pass True if the administrator can restrict, ban or unban chat members, or access supergroup statistics",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_promote_members",
        "description": "Optional. Pass True if the administrator can add new administrators with a subset of their own privileges or demote administrators that
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_change_info",
        "description": "Optional. Pass True if the administrator can change chat title, photo and other settings",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_invite_users",
        "description": "Optional. Pass True if the administrator can invite new users to the chat",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_post_messages",
        "description": "Optional. Pass True if the administrator can post messages in the channel, or access channel statistics; channels only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_edit_messages",
        "description": "Optional. Pass True if the administrator can edit messages of other users and can pin messages; channels only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_pin_messages",
        "description": "Optional. Pass True if the administrator can pin messages, supergroups only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_post_stories",
        "description": "Optional. Pass True if the administrator can post stories in the channel; channels only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_edit_stories",
        "description": "Optional. Pass True if the administrator can edit stories posted by other users; channels only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_delete_stories",
        "description": "Optional. Pass True if the administrator can delete stories posted by other users; channels only",
        "type": "text",
        "disabled": true
       },
       {
        "key": "can_manage_topics",
        "description": "Optional. Pass True if the user is allowed to create, rename, close, and reopen forum topics, supergroups only",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/promoteChatMember",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "promoteChatMember"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "setChatAdministratorCustomTitle",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "user_id",
        "description": "Required. Unique identifier of the target user",
        "type": "text"
       },
       {
        "key": "custom_title",
        "description": "Required. New custom title for the administrator; 0-16 characters, emoji are not allowed",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setChatAdministratorCustomTitle",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setChatAdministratorCustomTitle"
      ]
     },
     "description": "Use this method to set a custom title for an administrator in a supergroup promoted by the bot. Returns True on success."
    },
    "response": []
   },
   {
    "name": "banChatSenderChat",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "sender_chat_id",
        "description": "Required. Unique identifier of the target sender chat",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/banChatSenderChat",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "banChatSenderChat"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "unbanChatSenderChat",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "sender_chat_id",
        "description": "Required. Unique identifier of the target sender chat",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/unbanChatSenderChat",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "unbanChatSenderChat"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "setChatPermissions",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "permissions",
        "description": "Required. A JSON-serialized object for new default chat permissions",
        "type": "text"
       },
       {
        "key": "use_independent_chat_permissions",
        "description": "Optional. Pass True if chat permissions are set independently. Otherwise, the can_send_other_messages and can_add_web_page_previews perm
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setChatPermissions",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setChatPermissions"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "exportChatInviteLink",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/exportChatInviteLink",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "exportChatInviteLink"
      ]
     },
     "description": "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 admi
    },
    "response": []
   },
   {
    "name": "createChatInviteLink",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "name",
        "description": "Optional. Invite link name; 0-32 characters",
        "type": "text",
        "disabled": true
       },
       {
        "key": "expire_date",
        "description": "Optional. Point in time (Unix timestamp) when the link will expire",
        "type": "text",
        "disabled": true
       },
       {
        "key": "member_limit",
        "description": "Optional. The maximum number of users that can be members of the chat simultaneously after joining the chat via this invite link; 1-9999
        "type": "text",
        "disabled": true
       },
       {
        "key": "creates_join_request",
        "description": "Optional. True, if users joining the chat via the link need to be approved by chat administrators. If True, member_limit can't be specif
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/createChatInviteLink",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "createChatInviteLink"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "editChatInviteLink",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "invite_link",
        "description": "Required. The invite link to edit",
        "type": "text"
       },
       {
        "key": "name",
        "description": "Optional. Invite link name; 0-32 characters",
        "type": "text",
        "disabled": true
       },
       {
        "key": "expire_date",
        "description": "Optional. Point in time (Unix timestamp) when the link will expire",
        "type": "text",
        "disabled": true
       },
       {
        "key": "member_limit",
        "description": "Optional. The maximum number of users that can be members of the chat simultaneously after joining the chat via this invite link; 1-9999
        "type": "text",
        "disabled": true
       },
       {
        "key": "creates_join_request",
        "description": "Optional. True, if users joining the chat via the link need to be approved by chat administrators. If True, member_limit can't be specif
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/editChatInviteLink",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "editChatInviteLink"
      ]
     },
     "description": "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 mus
    },
    "response": []
   },
   {
    "name": "revokeChatInviteLink",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier of the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
      },
      {
       "key": "invite_link",
       "description": "Required. The invite link to revoke",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/revokeChatInviteLink",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "revokeChatInviteLink"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "approveChatJoinRequest",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/approveChatJoinRequest",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "approveChatJoinRequest"
     ]
    },
    "description": "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_
   },
   "response": []
  },
  {
   "name": "declineChatJoinRequest",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. Unique identifier of the target user",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/declineChatJoinRequest",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "declineChatJoinRequest"
     ]
    },
    "description": "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_
   },
   "response": []
  },
  {
   "name": "setChatPhoto",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "photo",
        "description": "Required. New chat photo, uploaded using multipart/form-data",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setChatPhoto",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setChatPhoto"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "deleteChatPhoto",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/deleteChatPhoto",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "deleteChatPhoto"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "setChatTitle",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "title",
        "description": "Required. New chat title, 1-128 characters",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setChatTitle",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setChatTitle"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "setChatDescription",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "description",
        "description": "Optional. New chat description, 0-255 characters",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setChatDescription",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setChatDescription"
      ]
     },
     "description": "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 w
    },
    "response": []
   },
   {
    "name": "pinChatMessage",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_id",
        "description": "Required. Identifier of a message to pin",
        "type": "text"
       },
       {
        "key": "disable_notification",
        "description": "Optional. Pass True if it is not necessary to send a notification to all chat members about the new pinned message. Notifications are al
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/pinChatMessage",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "pinChatMessage"
      ]
     },
     "description": "Use this method to add a message to the list of pinned messages in a chat. If the chat is not a private chat, the bot must be an administra
    },
    "response": []
   },
   {
    "name": "unpinChatMessage",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "message_id",
        "description": "Optional. Identifier of a message to unpin. If not specified, the most recent pinned message (by sending date) will be unpinned.",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/unpinChatMessage",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "unpinChatMessage"
     ]
    },
    "description": "Use this method to remove a message from the list of pinned messages in a chat. If the chat is not a private chat, the bot must be an admin
   },
   "response": []
  },
  {
   "name": "unpinAllChatMessages",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/unpinAllChatMessages",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "unpinAllChatMessages"
     ]
    },
    "description": "Use this method to clear the list of pinned messages in a chat. If the chat is not a private chat, the bot must be an administrator in the
   },
   "response": []
  },
  {
   "name": "leaveChat",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup or channel (in the format @channelusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/leaveChat",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "leaveChat"
     ]
    },
    "description": "Use this method for your bot to leave a group, supergroup or channel. Returns True on success."
   },
   "response": []
  },
  {
   "name": "getChat",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup or channel (in the format @channelusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getChat",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "getChat"
     ]
    },
    "description": "Use this method to get up to date information about the chat. Returns a Chat object on success."
   },
   "response": []
  },
  {
   "name": "getChatAdministrators",
   "request": {
    "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup or channel (in the format @channelusername)",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getChatAdministrators",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getChatAdministrators"
   ]
  },
  "description": "Use this method to get a list of administrators in a chat, which aren't bots. Returns an Array of ChatMember objects."
 },
 "response": []
},
{
 "name": "getChatMemberCount",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup or channel (in the format @channelusername)",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getChatMemberCount",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getChatMemberCount"
   ]
  },
  "description": "Use this method to get the number of members in a chat. Returns Int on success."
 },
 "response": []
},
{
 "name": "getChatMember",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup or channel (in the format @channelusername)",
     "type": "text"
    },
    {
     "key": "user_id",
     "description": "Required. Unique identifier of the target user",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getChatMember",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getChatMember"
   ]
  },
  "description": "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 adminis
 },
 "response": []
},
{
 "name": "setChatStickerSet",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
     "type": "text"
    },
    {
     "key": "sticker_set_name",
     "description": "Required. Name of the sticker set to be set as the group sticker set",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/setChatStickerSet",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "setChatStickerSet"
   ]
  },
  "description": "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 hav
 },
 "response": []
},
{
 "name": "deleteChatStickerSet",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/deleteChatStickerSet",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "deleteChatStickerSet"
   ]
  },
  "description": "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 hav
 },
 "response": []
},
{
 "name": "getForumTopicIconStickers",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": []
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getForumTopicIconStickers",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getForumTopicIconStickers"
   ]
  },
  "description": "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
 },
 "response": []
},
{
 "name": "createForumTopic",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
     "type": "text"
    },
    {
     "key": "name",
     "description": "Required. Topic name, 1-128 characters",
     "type": "text"
    },
    {
     "key": "icon_color",
     "description": "Optional. Color of the topic icon in RGB format. Currently, must be one of 7322096 (0x6FB9F0), 16766590 (0xFFD67E), 13338331 (0xCB86DB),
     "type": "text",
     "disabled": true
    },
    {
     "key": "icon_custom_emoji_id",
     "description": "Optional. Unique identifier of the custom emoji shown as the topic icon. Use getForumTopicIconStickers to get all allowed custom emoji i
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/createForumTopic",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "createForumTopic"
   ]
  },
  "description": "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 t
 },
 "response": []
},
{
 "name": "editForumTopic",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
     "type": "text"
    },
    {
     "key": "message_thread_id",
     "description": "Required. Unique identifier for the target message thread of the forum topic",
     "type": "text"
    },
    {
     "key": "name",
     "description": "Optional. New topic name, 0-128 characters. If not specified or empty, the current name of the topic will be kept",
     "type": "text",
     "disabled": true
    },
    {
     "key": "icon_custom_emoji_id",
     "description": "Optional. New unique identifier of the custom emoji shown as the topic icon. Use getForumTopicIconStickers to get all allowed custom emo
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/editForumTopic",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "editForumTopic"
   ]
  },
  "description": "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
 },
 "response": []
},
{
 "name": "closeForumTopic",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
     "type": "text"
    },
    {
     "key": "message_thread_id",
     "description": "Required. Unique identifier for the target message thread of the forum topic",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/closeForumTopic",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
      "path": [
       "bot{{token}}",
       "closeForumTopic"
      ]
     },
     "description": "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 h
    },
    "response": []
   },
   {
    "name": "reopenForumTopic",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Required. Unique identifier for the target message thread of the forum topic",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/reopenForumTopic",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "reopenForumTopic"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "deleteForumTopic",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Required. Unique identifier for the target message thread of the forum topic",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/deleteForumTopic",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "deleteForumTopic"
      ]
     },
     "description": "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 cha
    },
    "response": []
   },
   {
    "name": "unpinAllForumTopicMessages",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "message_thread_id",
        "description": "Required. Unique identifier for the target message thread of the forum topic",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/unpinAllForumTopicMessages",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "unpinAllForumTopicMessages"
      ]
     },
     "description": "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 mu
    },
    "response": []
   },
   {
    "name": "editGeneralForumTopic",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       },
       {
        "key": "name",
        "description": "Required. New topic name, 1-128 characters",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/editGeneralForumTopic",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "editGeneralForumTopic"
      ]
     },
     "description": "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 t
    },
    "response": []
   },
   {
    "name": "closeGeneralForumTopic",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/closeGeneralForumTopic",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "closeGeneralForumTopic"
      ]
     },
     "description": "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 
    },
    "response": []
   },
   {
    "name": "reopenGeneralForumTopic",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/reopenGeneralForumTopic",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
     "path": [
      "bot{{token}}",
      "reopenGeneralForumTopic"
     ]
    },
    "description": "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 wor
   },
   "response": []
  },
  {
   "name": "hideGeneralForumTopic",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/hideGeneralForumTopic",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "hideGeneralForumTopic"
     ]
    },
    "description": "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 m
   },
   "response": []
  },
  {
   "name": "unhideGeneralForumTopic",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/unhideGeneralForumTopic",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "unhideGeneralForumTopic"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "unpinAllGeneralForumTopicMessages",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target supergroup (in the format @supergroupusername)",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/unpinAllGeneralForumTopicMessages",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "unpinAllGeneralForumTopicMessages"
     ]
    },
    "description": "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 wor
   },
   "response": []
  },
  {
```

```json
    "name": "answerCallbackQuery",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "callback_query_id",
        "description": "Required. Unique identifier for the query to be answered",
        "type": "text"
       },
       {
        "key": "text",
        "description": "Optional. Text of the notification. If not specified, nothing will be shown to the user, 0-200 characters",
        "type": "text",
        "disabled": true
       },
       {
        "key": "show_alert",
        "description": "Optional. If True, an alert will be shown by the client instead of a notification at the top of the chat screen. Defaults to false.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "url",
        "description": "Optional. URL that will be opened by the user's client. If you have created a Game and accepted the conditions via @BotFather, specify t
        "type": "text",
        "disabled": true
       },
       {
        "key": "cache_time",
        "description": "Optional. The maximum amount of time in seconds that the result of the callback query may be cached client-side. Telegram apps will supp
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/answerCallbackQuery",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "answerCallbackQuery"
      ]
     },
     "description": "Use this method to send answers to callback queries sent from inline keyboards. The answer will be displayed to the user as a notification
    },
    "response": []
   },
   {
    "name": "getUserChatBoosts",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Required. Unique identifier for the chat or username of the channel (in the format @channelusername)",
        "type": "text"
       },
       {
        "key": "user_id",
        "description": "Required. Unique identifier of the target user",
        "type": "text"
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getUserChatBoosts",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "getUserChatBoosts"
      ]
     },
     "description": "Use this method to get the list of boosts added to a chat by a user. Requires administrator rights in the chat. Returns a UserChatBoosts ob
    },
    "response": []
   },
   {
    "name": "setMyCommands",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "commands",
        "description": "Required. 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.",
        "type": "text"
       },
       {
```

```
     "key": "scope",
     "value": "BotCommandScopeDefault",
     "description": "Optional. A JSON-serialized object, describing scope of users for which the commands are relevant. Defaults to BotCommandScopeDefault.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "language_code",
     "description": "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
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/setMyCommands",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "setMyCommands"
   ]
  },
  "description": "Use this method to change the list of the bot's commands. See this manual for more details about bot commands. Returns True on success."
 },
 "response": []
},
{
 "name": "deleteMyCommands",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "scope",
     "value": "BotCommandScopeDefault",
     "description": "Optional. A JSON-serialized object, describing scope of users for which the commands are relevant. Defaults to BotCommandScopeDefault.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "language_code",
     "description": "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
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/deleteMyCommands",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "deleteMyCommands"
   ]
  },
  "description": "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
 },
 "response": []
},
{
 "name": "getMyCommands",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "scope",
     "value": "BotCommandScopeDefault",
     "description": "Optional. A JSON-serialized object, describing scope of users. Defaults to BotCommandScopeDefault.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "language_code",
     "description": "Optional. A two-letter ISO 639-1 language code or an empty string",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getMyCommands",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getMyCommands"
   ]
```

```
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "setMyName",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Optional. New bot name; 0-64 characters. Pass an empty string to remove the dedicated name for the given language.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "language_code",
       "description": "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 n
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setMyName",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setMyName"
     ]
    },
    "description": "Use this method to change the bot's name. Returns True on success."
   },
   "response": []
  },
  {
   "name": "getMyName",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "language_code",
       "description": "Optional. A two-letter ISO 639-1 language code or an empty string",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getMyName",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "getMyName"
     ]
    },
    "description": "Use this method to get the current bot name for the given user language. Returns BotName on success."
   },
   "response": []
  },
  {
   "name": "setMyDescription",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "description",
       "description": "Optional. New bot description; 0-512 characters. Pass an empty string to remove the dedicated description for the given language.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "language_code",
       "description": "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 de
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setMyDescription",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
      ],
      "path": [
       "bot{{token}}",
       "setMyDescription"
      ]
     },
     "description": "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."
    },
    "response": []
   },
   {
    "name": "getMyDescription",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "language_code",
        "description": "Optional. A two-letter ISO 639-1 language code or an empty string",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getMyDescription",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "getMyDescription"
      ]
     },
     "description": "Use this method to get the current bot description for the given user language. Returns BotDescription on success."
    },
    "response": []
   },
   {
    "name": "setMyShortDescription",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "short_description",
        "description": "Optional. New short description for the bot; 0-120 characters. Pass an empty string to remove the dedicated short description for the gi
        "type": "text",
        "disabled": true
       },
       {
        "key": "language_code",
        "description": "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
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/setMyShortDescription",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setMyShortDescription"
      ]
     },
     "description": "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 use
    },
    "response": []
   },
   {
    "name": "getMyShortDescription",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "language_code",
        "description": "Optional. A two-letter ISO 639-1 language code or an empty string",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getMyShortDescription",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
   "path": [
    "bot{{token}}",
    "getMyShortDescription"
   ]
  },
  "description": "Use this method to get the current bot short description for the given user language. Returns BotShortDescription on success."
 },
 "response": []
},
{
 "name": "setChatMenuButton",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Optional. Unique identifier for the target private chat. If not specified, default bot's menu button will be changed",
     "type": "text",
     "disabled": true
    },
    {
     "key": "menu_button",
     "value": "MenuButtonDefault",
     "description": "Optional. A JSON-serialized object for the bot's new menu button. Defaults to MenuButtonDefault",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/setChatMenuButton",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "setChatMenuButton"
   ]
  },
  "description": "Use this method to change the bot's menu button in a private chat, or the default menu button. Returns True on success."
 },
 "response": []
},
{
 "name": "getChatMenuButton",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Optional. Unique identifier for the target private chat. If not specified, default bot's menu button will be returned",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getChatMenuButton",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getChatMenuButton"
   ]
  },
  "description": "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 succ
 },
 "response": []
},
{
 "name": "setMyDefaultAdministratorRights",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "rights",
     "description": "Optional. A JSON-serialized object describing new default administrator rights. If not specified, the default administrator rights will
     "type": "text",
     "disabled": true
    },
    {
     "key": "for_channels",
     "description": "Optional. Pass True to change the default administrator rights of the bot in channels. Otherwise, the default administrator rights of th
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/setMyDefaultAdministratorRights",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "setMyDefaultAdministratorRights"
      ]
     },
     "description": "Use this method to change the default administrator rights requested by the bot when it's added as an administrator to groups or channels.
    },
    "response": []
   },
   {
    "name": "getMyDefaultAdministratorRights",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "for_channels",
        "description": "Optional. Pass True to get default administrator rights of the bot in channels. Otherwise, default administrator rights of the bot for g
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/getMyDefaultAdministratorRights",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "getMyDefaultAdministratorRights"
      ]
     },
     "description": "Use this method to get the current default administrator rights of the bot. Returns ChatAdministratorRights on success."
    },
    "response": []
   },
   {
    "name": "editMessageText",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "chat_id",
        "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
        "type": "text",
        "disabled": true
       },
       {
        "key": "message_id",
        "description": "Optional. Required if inline_message_id is not specified. Identifier of the message to edit",
        "type": "text",
        "disabled": true
       },
       {
        "key": "inline_message_id",
        "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
        "type": "text",
        "disabled": true
       },
       {
        "key": "text",
        "description": "Required. New text of the message, 1-4096 characters after entities parsing",
        "type": "text"
       },
       {
        "key": "parse_mode",
        "description": "Optional. Mode for parsing entities in the message text. See formatting options for more details.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "entities",
        "description": "Optional. A JSON-serialized list of special entities that appear in message text, which can be specified instead of parse_mode",
        "type": "text",
        "disabled": true
       },
       {
        "key": "link_preview_options",
        "description": "Optional. Link preview generation options for the message",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. A JSON-serialized object for an inline keyboard.",
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
```

```
   "raw": "https://api.telegram.org/bot{{token}}/editMessageText",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "editMessageText"
   ]
  },
  "description": "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,
 },
 "response": []
},
{
 "name": "editMessageCaption",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
     "type": "text",
     "disabled": true
    },
    {
     "key": "message_id",
     "description": "Optional. Required if inline_message_id is not specified. Identifier of the message to edit",
     "type": "text",
     "disabled": true
    },
    {
     "key": "inline_message_id",
     "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
     "type": "text",
     "disabled": true
    },
    {
     "key": "caption",
     "description": "Optional. New caption of the message, 0-1024 characters after entities parsing",
     "type": "text",
     "disabled": true
    },
    {
     "key": "parse_mode",
     "description": "Optional. Mode for parsing entities in the message caption. See formatting options for more details.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "caption_entities",
     "description": "Optional. A JSON-serialized list of special entities that appear in the caption, which can be specified instead of parse_mode",
     "type": "text",
     "disabled": true
    },
    {
     "key": "reply_markup",
     "description": "Optional. A JSON-serialized object for an inline keyboard.",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/editMessageCaption",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "editMessageCaption"
   ]
  },
  "description": "Use this method to edit captions of messages. On success, if the edited message is not an inline message, the edited Message is returned, o
 },
 "response": []
},
{
 "name": "editMessageMedia",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
     "type": "text",
     "disabled": true
    },
    {
     "key": "message_id",
     "description": "Optional. Required if inline_message_id is not specified. Identifier of the message to edit",
     "type": "text",
     "disabled": true
    },
    {
     "key": "inline_message_id",
     "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
     "type": "text",
     "disabled": true
    },
    {
     "key": "media",
     "description": "Required. A JSON-serialized object for a new media content of the message",
     "type": "text"
    },
    {
     "key": "reply_markup",
     "description": "Optional. A JSON-serialized object for a new inline keyboard.",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/editMessageMedia",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "editMessageMedia"
   ]
  },
  "description": "Use this method to edit animation, audio, document, photo, or video messages. If a message is part of a message album, then it can be edite
 },
 "response": []
},
{
 "name": "editMessageLiveLocation",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "chat_id",
     "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
     "type": "text",
     "disabled": true
    },
    {
     "key": "message_id",
     "description": "Optional. Required if inline_message_id is not specified. Identifier of the message to edit",
     "type": "text",
     "disabled": true
    },
    {
     "key": "inline_message_id",
     "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
     "type": "text",
     "disabled": true
    },
    {
     "key": "latitude",
     "description": "Required. Latitude of new location",
     "type": "text"
    },
    {
     "key": "longitude",
     "description": "Required. Longitude of new location",
     "type": "text"
    },
    {
     "key": "horizontal_accuracy",
     "description": "Optional. The radius of uncertainty for the location, measured in meters; 0-1500",
     "type": "text",
     "disabled": true
    },
    {
     "key": "heading",
     "description": "Optional. Direction in which the user is moving, in degrees. Must be between 1 and 360 if specified.",
     "type": "text",
     "disabled": true
    },
    {
     "key": "proximity_alert_radius",
     "description": "Optional. The maximum distance for proximity alerts about approaching another chat member, in meters. Must be between 1 and 100000 if sp
     "type": "text",
     "disabled": true
    },
    {
     "key": "reply_markup",
     "description": "Optional. A JSON-serialized object for a new inline keyboard.",
     "type": "text",
     "disabled": true
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/editMessageLiveLocation",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
      "bot{{token}}",
      "editMessageLiveLocation"
     ]
    },
    "description": "Use this method to edit live location messages. A location can be edited until its live_period expires or editing is explicitly disabled by
   },
   "response": []
  },
  {
   "name": "stopMessageLiveLocation",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
       "type": "text",
       "disabled": true
      },
      {
       "key": "message_id",
       "description": "Optional. Required if inline_message_id is not specified. Identifier of the message with live location to stop",
       "type": "text",
       "disabled": true
      },
      {
       "key": "inline_message_id",
       "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. A JSON-serialized object for a new inline keyboard.",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/stopMessageLiveLocation",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "stopMessageLiveLocation"
     ]
    },
    "description": "Use this method to stop updating a live location message before live_period expires. On success, if the message is not an inline message, t
   },
   "response": []
  },
  {
   "name": "editMessageReplyMarkup",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat or username of the target channel (in th
       "type": "text",
       "disabled": true
      },
      {
       "key": "message_id",
       "description": "Optional. Required if inline_message_id is not specified. Identifier of the message to edit",
       "type": "text",
       "disabled": true
      },
      {
       "key": "inline_message_id",
       "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. A JSON-serialized object for an inline keyboard.",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/editMessageReplyMarkup",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "editMessageReplyMarkup"
     ]
    },
    "description": "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 i
   },
   "response": []
  },
  {
   "name": "stopPoll",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_id",
       "description": "Required. Identifier of the original message with the poll",
       "type": "text"
      },
      {
       "key": "reply_markup",
       "description": "Optional. A JSON-serialized object for a new message inline keyboard.",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/stopPoll",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "stopPoll"
     ]
    },
    "description": "Use this method to stop a poll which was sent by the bot. On success, the stopped Poll is returned."
   },
   "response": []
  },
  {
   "name": "deleteMessage",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_id",
       "description": "Required. Identifier of the message to delete",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/deleteMessage",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "deleteMessage"
     ]
    },
    "description": "Use this method to delete a message, including service messages, with the following limitations:- A message can only be deleted if it was s
   },
   "response": []
  },
  {
   "name": "deleteMessages",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_ids",
       "description": "Required. Identifiers of 1-100 messages to delete. See deleteMessage for limitations on which messages can be deleted",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/deleteMessages",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "deleteMessages"
     ]
    },
    "description": "Use this method to delete multiple messages simultaneously. If some of the specified messages can't be found, they are skipped. Returns Tru
   },
   "response": []
  },
  {
   "name": "sendSticker",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "sticker",
       "description": "Required. Sticker to send. Pass a file_id as String to send a file that exists on the Telegram servers (recommended), pass an HTTP URL a
       "type": "text"
      },
      {
       "key": "emoji",
       "description": "Optional. Emoji associated with the sticker; only for just uploaded stickers",
       "type": "text",
       "disabled": true
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove r
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendSticker",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendSticker"
     ]
    },
    "description": "Use this method to send static .WEBP, animated .TGS, or video .WEBM stickers. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "getStickerSet",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Required. Name of the sticker set",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/getStickerSet",
     "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getStickerSet"
   ]
  },
  "description": "Use this method to get a sticker set. On success, a StickerSet object is returned."
 },
 "response": []
},
{
 "name": "getCustomEmojiStickers",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "custom_emoji_ids",
     "description": "Required. List of custom emoji identifiers. At most 200 custom emoji identifiers can be specified.",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/getCustomEmojiStickers",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "getCustomEmojiStickers"
   ]
  },
  "description": "Use this method to get information about custom emoji stickers by their identifiers. Returns an Array of Sticker objects."
 },
 "response": []
},
{
 "name": "uploadStickerFile",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "user_id",
     "description": "Required. User identifier of sticker file owner",
     "type": "text"
    },
    {
     "key": "sticker",
     "description": "Required. A file with the sticker in .WEBP, .PNG, .TGS, or .WEBM format. See https://core.telegram.org/stickers for technical requiremen
     "type": "text"
    },
    {
     "key": "sticker_format",
     "description": "Required. Format of the sticker, must be one of â€œstaticâ€ , â€œanimatedâ€ , â€œvideoâ€ ",
     "type": "text"
    }
   ]
  },
  "url": {
   "raw": "https://api.telegram.org/bot{{token}}/uploadStickerFile",
   "protocol": "https",
   "host": [
    "api",
    "telegram",
    "org"
   ],
   "path": [
    "bot{{token}}",
    "uploadStickerFile"
   ]
  },
  "description": "Use this method to upload a file with a sticker for later use in the createNewStickerSet and addStickerToSet methods (the file can be used 
 },
 "response": []
},
{
 "name": "createNewStickerSet",
 "request": {
  "method": "POST",
  "header": [],
  "body": {
   "mode": "formdata",
   "formdata": [
    {
     "key": "user_id",
     "description": "Required. User identifier of created sticker set owner",
     "type": "text"
    },
    {
     "key": "name",
     "description": "Required. Short name of sticker set, to be used in t.me/addstickers/ URLs (e.g., animals). Can contain only English letters, digits and 
     "type": "text"
    },
      {
       "key": "title",
       "description": "Required. Sticker set title, 1-64 characters",
       "type": "text"
      },
      {
       "key": "stickers",
       "description": "Required. A JSON-serialized list of 1-50 initial stickers to be added to the sticker set",
       "type": "text"
      },
      {
       "key": "sticker_format",
       "description": "Required. Format of stickers in the set, must be one of â€œstaticâ€ , â€œanimatedâ€ , â€œvideoâ€ ",
       "type": "text"
      },
      {
       "key": "sticker_type",
       "description": "Optional. Type of stickers in the set, pass â€œregularâ€ , â€œmaskâ€ , or â€œcustom_emojiâ€ . By default, a regular sticker set is creat
       "type": "text",
       "disabled": true
      },
      {
       "key": "needs_repainting",
       "description": "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 use
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/createNewStickerSet",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "createNewStickerSet"
     ]
    },
    "description": "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 suc
   },
   "response": []
  },
  {
   "name": "addStickerToSet",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "user_id",
       "description": "Required. User identifier of sticker set owner",
       "type": "text"
      },
      {
       "key": "name",
       "description": "Required. Sticker set name",
       "type": "text"
      },
      {
       "key": "sticker",
       "description": "Required. A JSON-serialized object with information about the added sticker. If exactly the same sticker had already been added to the s
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/addStickerToSet",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "addStickerToSet"
     ]
    },
    "description": "Use this method to add a new sticker to a set created by the bot. The format of the added sticker must match the format of the other sticke
   },
   "response": []
  },
  {
   "name": "setStickerPositionInSet",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "sticker",
       "description": "Required. File identifier of the sticker",
       "type": "text"
      },
      {
       "key": "position",
       "description": "Required. New sticker position in the set, zero-based",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerPositionInSet",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerPositionInSet"
     ]
    },
    "description": "Use this method to move a sticker in a set created by the bot to a specific position. Returns True on success."
   },
   "response": []
  },
  {
   "name": "deleteStickerFromSet",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "sticker",
       "description": "Required. File identifier of the sticker",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/deleteStickerFromSet",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "deleteStickerFromSet"
     ]
    },
    "description": "Use this method to delete a sticker from a set created by the bot. Returns True on success."
   },
   "response": []
  },
  {
   "name": "setStickerEmojiList",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "sticker",
       "description": "Required. File identifier of the sticker",
       "type": "text"
      },
      {
       "key": "emoji_list",
       "description": "Required. A JSON-serialized list of 1-20 emoji associated with the sticker",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerEmojiList",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerEmojiList"
     ]
    },
    "description": "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
   },
   "response": []
  },
  {
   "name": "setStickerKeywords",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "sticker",
       "description": "Required. File identifier of the sticker",
       "type": "text"
      },
      {
       "key": "keywords",
       "description": "Optional. A JSON-serialized list of 0-20 search keywords for the sticker with total length of up to 64 characters",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerKeywords",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerKeywords"
     ]
    },
    "description": "Use this method to change search keywords assigned to a regular or custom emoji sticker. The sticker must belong to a sticker set created b
   },
   "response": []
  },
  {
   "name": "setStickerMaskPosition",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "sticker",
       "description": "Required. File identifier of the sticker",
       "type": "text"
      },
      {
       "key": "mask_position",
       "description": "Optional. A JSON-serialized object with the position where the mask should be placed on faces. Omit the parameter to remove the mask pos
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerMaskPosition",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerMaskPosition"
     ]
    },
    "description": "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. Return
   },
   "response": []
  },
  {
   "name": "setStickerSetTitle",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Required. Sticker set name",
       "type": "text"
      },
      {
       "key": "title",
       "description": "Required. Sticker set title, 1-64 characters",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerSetTitle",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerSetTitle"
     ]
    },
    "description": "Use this method to set the title of a created sticker set. Returns True on success."
   },
   "response": []
  },
  {
   "name": "setStickerSetThumbnail",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Required. Sticker set name",
       "type": "text"
      },
      {
       "key": "user_id",
       "description": "Required. User identifier of the sticker set owner",
       "type": "text"
      },
      {
       "key": "thumbnail",
       "description": "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
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setStickerSetThumbnail",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setStickerSetThumbnail"
     ]
    },
    "description": "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 sticke
   },
   "response": []
  },
  {
   "name": "setCustomEmojiStickerSetThumbnail",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Required. Sticker set name",
       "type": "text"
      },
      {
       "key": "custom_emoji_id",
       "description": "Optional. Custom emoji identifier of a sticker from the sticker set; pass an empty string to drop the thumbnail and use the first sticke
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setCustomEmojiStickerSetThumbnail",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setCustomEmojiStickerSetThumbnail"
     ]
    },
    "description": "Use this method to set the thumbnail of a custom emoji sticker set. Returns True on success."
   },
   "response": []
  },
  {
   "name": "deleteStickerSet",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "name",
       "description": "Required. Sticker set name",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/deleteStickerSet",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "deleteStickerSet"
     ]
    },
    "description": "Use this method to delete a sticker set that was created by the bot. Returns True on success."
   },
   "response": []
  },
  {
   "name": "answerInlineQuery",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "inline_query_id",
       "description": "Required. Unique identifier for the answered query",
       "type": "text"
      },
      {
       "key": "results",
       "description": "Required. A JSON-serialized array of results for the inline query",
       "type": "text"
      },
      {
       "key": "cache_time",
       "value": "300",
       "description": "Optional. The maximum amount of time in seconds that the result of the inline query may be cached on the server. Defaults to 300.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "is_personal",
       "description": "Optional. Pass True if results may be cached on the server side only for the user that sent the query. By default, results may be return
       "type": "text",
       "disabled": true
      },
      {
       "key": "next_offset",
       "description": "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 i
       "type": "text",
       "disabled": true
      },
      {
       "key": "button",
       "description": "Optional. A JSON-serialized object describing a button to be shown above inline query results",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/answerInlineQuery",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "answerInlineQuery"
     ]
    },
    "description": "Use this method to send answers to an inline query. On success, True is returned.No more than 50 results per query are allowed."
   },
   "response": []
  },
  {
   "name": "answerWebAppQuery",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "web_app_query_id",
       "description": "Required. Unique identifier for the query to be answered",
       "type": "text"
      },
      {
       "key": "result",
       "description": "Required. A JSON-serialized object describing the message to be sent",
       "type": "text"
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/answerWebAppQuery",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "answerWebAppQuery"
     ]
    },
    "description": "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 t
   },
   "response": []
  },
  {
   "name": "sendInvoice",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat or username of the target channel (in the format @channelusername)",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "title",
       "description": "Required. Product name, 1-32 characters",
       "type": "text"
      },
      {
       "key": "description",
       "description": "Required. Product description, 1-255 characters",
       "type": "text"
      },
      {
       "key": "payload",
       "description": "Required. Bot-defined invoice payload, 1-128 bytes. This will not be displayed to the user, use for your internal processes.",
       "type": "text"
      },
      {
       "key": "provider_token",
       "description": "Required. Payment provider token, obtained via @BotFather",
       "type": "text"
      },
      {
       "key": "currency",
       "description": "Required. Three-letter ISO 4217 currency code, see more on currencies",
       "type": "text"
      },
      {
       "key": "prices",
       "description": "Required. Price breakdown, a JSON-serialized list of components (e.g. product price, tax, discount, delivery cost, delivery tax, bonus,
       "type": "text"
      },
      {
       "key": "max_tip_amount",
       "description": "Optional. The maximum accepted amount for tips in the smallest units of the currency (integer, not float/double). For example, for a max
       "type": "text",
       "disabled": true
      },
      {
       "key": "suggested_tip_amounts",
       "description": "Optional. A JSON-serialized array of suggested amounts of tips in the smallest units of the currency (integer, not float/double). At mos
       "type": "text",
       "disabled": true
      },
      {
       "key": "start_parameter",
       "description": "Optional. Unique deep-linking parameter. If left empty, forwarded copies of the sent message will have a Pay button, allowing multiple u
       "type": "text",
       "disabled": true
      },
      {
       "key": "provider_data",
       "description": "Optional. JSON-serialized data about the invoice, which will be shared with the payment provider. A detailed description of required fie
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_url",
       "description": "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 bette
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_size",
       "description": "Optional. Photo size in bytes",
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_width",
       "description": "Optional. Photo width",
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_height",
       "description": "Optional. Photo height",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_name",
       "description": "Optional. Pass True if you require the user's full name to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_phone_number",
       "description": "Optional. Pass True if you require the user's phone number to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_email",
       "description": "Optional. Pass True if you require the user's email address to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_shipping_address",
        "description": "Optional. Pass True if you require the user's shipping address to complete the order",
        "type": "text",
        "disabled": true
       },
       {
        "key": "send_phone_number_to_provider",
        "description": "Optional. Pass True if the user's phone number should be sent to provider",
        "type": "text",
        "disabled": true
       },
       {
        "key": "send_email_to_provider",
        "description": "Optional. Pass True if the user's email address should be sent to provider",
        "type": "text",
        "disabled": true
       },
       {
        "key": "is_flexible",
        "description": "Optional. Pass True if the final price depends on the shipping method",
        "type": "text",
        "disabled": true
       },
       {
        "key": "disable_notification",
        "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "protect_content",
        "description": "Optional. Protects the contents of the sent message from forwarding and saving",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_parameters",
        "description": "Optional. Description of the message to reply to",
        "type": "text",
        "disabled": true
       },
       {
        "key": "reply_markup",
        "description": "Optional. A JSON-serialized object for an inline keyboard. If empty, one 'Pay total price' button will be shown. If not empty, the first
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/sendInvoice",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "sendInvoice"
      ]
     },
     "description": "Use this method to send invoices. On success, the sent Message is returned."
    },
    "response": []
   },
   {
    "name": "createInvoiceLink",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "title",
        "description": "Required. Product name, 1-32 characters",
        "type": "text"
       },
       {
        "key": "description",
        "description": "Required. Product description, 1-255 characters",
        "type": "text"
       },
       {
        "key": "payload",
        "description": "Required. Bot-defined invoice payload, 1-128 bytes. This will not be displayed to the user, use for your internal processes.",
        "type": "text"
       },
       {
        "key": "provider_token",
        "description": "Required. Payment provider token, obtained via BotFather",
        "type": "text"
       },
       {
        "key": "currency",
        "description": "Required. Three-letter ISO 4217 currency code, see more on currencies",
        "type": "text"
       },
       {
        "key": "prices",
        "description": "Required. Price breakdown, a JSON-serialized list of components (e.g. product price, tax, discount, delivery cost, delivery tax, bonus,
        "type": "text"
       },
       {
        "key": "max_tip_amount",
        "description": "Optional. The maximum accepted amount for tips in the smallest units of the currency (integer, not float/double). For example, for a max
       "type": "text",
       "disabled": true
      },
      {
       "key": "suggested_tip_amounts",
       "description": "Optional. A JSON-serialized array of suggested amounts of tips in the smallest units of the currency (integer, not float/double). At mos
       "type": "text",
       "disabled": true
      },
      {
       "key": "provider_data",
       "description": "Optional. JSON-serialized data about the invoice, which will be shared with the payment provider. A detailed description of required fie
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_url",
       "description": "Optional. URL of the product photo for the invoice. Can be a photo of the goods or a marketing image for a service.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_size",
       "description": "Optional. Photo size in bytes",
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_width",
       "description": "Optional. Photo width",
       "type": "text",
       "disabled": true
      },
      {
       "key": "photo_height",
       "description": "Optional. Photo height",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_name",
       "description": "Optional. Pass True if you require the user's full name to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_phone_number",
       "description": "Optional. Pass True if you require the user's phone number to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_email",
       "description": "Optional. Pass True if you require the user's email address to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "need_shipping_address",
       "description": "Optional. Pass True if you require the user's shipping address to complete the order",
       "type": "text",
       "disabled": true
      },
      {
       "key": "send_phone_number_to_provider",
       "description": "Optional. Pass True if the user's phone number should be sent to the provider",
       "type": "text",
       "disabled": true
      },
      {
       "key": "send_email_to_provider",
       "description": "Optional. Pass True if the user's email address should be sent to the provider",
       "type": "text",
       "disabled": true
      },
      {
       "key": "is_flexible",
       "description": "Optional. Pass True if the final price depends on the shipping method",
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/createInvoiceLink",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "createInvoiceLink"
     ]
    },
    "description": "Use this method to create a link for an invoice. Returns the created invoice link as String on success."
   },
   "response": []
  },
  {
   "name": "answerShippingQuery",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "shipping_query_id",
        "description": "Required. Unique identifier for the query to be answered",
        "type": "text"
       },
       {
        "key": "ok",
        "description": "Required. Pass True if delivery to the specified address is possible and False if there are any problems (for example, if delivery to th
        "type": "text"
       },
       {
        "key": "shipping_options",
        "description": "Optional. Required if ok is True. A JSON-serialized array of available shipping options.",
        "type": "text",
        "disabled": true
       },
       {
        "key": "error_message",
        "description": "Optional. Required if ok is False. Error message in human readable form that explains why it is impossible to complete the order (e.g. \
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/answerShippingQuery",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "answerShippingQuery"
      ]
     },
     "description": "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 sh
    },
    "response": []
   },
   {
    "name": "answerPreCheckoutQuery",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "pre_checkout_query_id",
        "description": "Required. Unique identifier for the query to be answered",
        "type": "text"
       },
       {
        "key": "ok",
        "description": "Required. Specify True if everything is alright (goods are available, etc.) and the bot is ready to proceed with the order. Use False if
        "type": "text"
       },
       {
        "key": "error_message",
        "description": "Optional. Required if ok is False. Error message in human readable form that explains the reason for failure to proceed with the checkou
        "type": "text",
        "disabled": true
       }
      ]
     },
     "url": {
      "raw": "https://api.telegram.org/bot{{token}}/answerPreCheckoutQuery",
      "protocol": "https",
      "host": [
       "api",
       "telegram",
       "org"
      ],
      "path": [
       "bot{{token}}",
       "answerPreCheckoutQuery"
      ]
     },
     "description": "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
    },
    "response": []
   },
   {
    "name": "setPassportDataErrors",
    "request": {
     "method": "POST",
     "header": [],
     "body": {
      "mode": "formdata",
      "formdata": [
       {
        "key": "user_id",
        "description": "Required. User identifier",
        "type": "text"
       },
       {
        "key": "errors",
        "description": "Required. A JSON-serialized array describing the errors",
        "type": "text"
       }
      ]
     },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/setPassportDataErrors",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "setPassportDataErrors"
     ]
    },
    "description": "Informs a user that some of the Telegram Passport elements they provided contains errors. The user will not be able to re-submit their Pass
   },
   "response": []
  },
  {
   "name": "sendGame",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "chat_id",
       "description": "Required. Unique identifier for the target chat",
       "type": "text"
      },
      {
       "key": "message_thread_id",
       "description": "Optional. Unique identifier for the target message thread (topic) of the forum; for forum supergroups only",
       "type": "text",
       "disabled": true
      },
      {
       "key": "game_short_name",
       "description": "Required. Short name of the game, serves as the unique identifier for the game. Set up your games via @BotFather.",
       "type": "text"
      },
      {
       "key": "disable_notification",
       "description": "Optional. Sends the message silently. Users will receive a notification with no sound.",
       "type": "text",
       "disabled": true
      },
      {
       "key": "protect_content",
       "description": "Optional. Protects the contents of the sent message from forwarding and saving",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_parameters",
       "description": "Optional. Description of the message to reply to",
       "type": "text",
       "disabled": true
      },
      {
       "key": "reply_markup",
       "description": "Optional. A JSON-serialized object for an inline keyboard. If empty, one 'Play game_title' button will be shown. If not empty, the first
       "type": "text",
       "disabled": true
      }
     ]
    },
    "url": {
     "raw": "https://api.telegram.org/bot{{token}}/sendGame",
     "protocol": "https",
     "host": [
      "api",
      "telegram",
      "org"
     ],
     "path": [
      "bot{{token}}",
      "sendGame"
     ]
    },
    "description": "Use this method to send a game. On success, the sent Message is returned."
   },
   "response": []
  },
  {
   "name": "setGameScore",
   "request": {
    "method": "POST",
    "header": [],
    "body": {
     "mode": "formdata",
     "formdata": [
      {
       "key": "user_id",
       "description": "Required. User identifier",
       "type": "text"
      },
      {
       "key": "score",
       "description": "Required. New score, must be non-negative",
       "type": "text"
      },
      {
       "key": "force",
       "description": "Optional. Pass True if the high score is allowed to decrease. This can be useful when fixing mistakes or banning cheaters",
       "type": "text",
      "disabled": true
     },
     {
      "key": "disable_edit_message",
      "description": "Optional. Pass True if the game message should not be automatically edited to include the current scoreboard",
      "type": "text",
      "disabled": true
     },
     {
      "key": "chat_id",
      "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat",
      "type": "text",
      "disabled": true
     },
     {
      "key": "message_id",
      "description": "Optional. Required if inline_message_id is not specified. Identifier of the sent message",
      "type": "text",
      "disabled": true
     },
     {
      "key": "inline_message_id",
      "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
      "type": "text",
      "disabled": true
     }
    ]
   },
   "url": {
    "raw": "https://api.telegram.org/bot{{token}}/setGameScore",
    "protocol": "https",
    "host": [
     "api",
     "telegram",
     "org"
    ],
    "path": [
     "bot{{token}}",
     "setGameScore"
    ]
   },
   "description": "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
  },
  "response": []
 },
 {
  "name": "getGameHighScores",
  "request": {
   "method": "POST",
   "header": [],
   "body": {
    "mode": "formdata",
    "formdata": [
     {
      "key": "user_id",
      "description": "Required. Target user id",
      "type": "text"
     },
     {
      "key": "chat_id",
      "description": "Optional. Required if inline_message_id is not specified. Unique identifier for the target chat",
      "type": "text",
      "disabled": true
     },
     {
      "key": "message_id",
      "description": "Optional. Required if inline_message_id is not specified. Identifier of the sent message",
      "type": "text",
      "disabled": true
     },
     {
      "key": "inline_message_id",
      "description": "Optional. Required if chat_id and message_id are not specified. Identifier of the inline message",
      "type": "text",
      "disabled": true
     }
    ]
   },
   "url": {
    "raw": "https://api.telegram.org/bot{{token}}/getGameHighScores",
    "protocol": "https",
    "host": [
     "api",
     "telegram",
     "org"
    ],
    "path": [
     "bot{{token}}",
     "getGameHighScores"
    ]
   },
   "description": "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. Re
  },
  "response": []
 }
],
"variable": [
 {
  "value": ""
 },
 {
  "value": ""
 },
 {
  "value": ""
 },
   {
    "value": ""
   },
   {
    "key": "token",
    "value": "964027989:AAHTXVzgbavAN2PSche5OG93qwoHsikCLe8"
   }
  ]
 }
```