



FREE eBook

LEARNING telegram-bot

Free unaffiliated eBook created from
Stack Overflow contributors.

#telegram-
bot

Table of Contents

About.....	1
Chapter 1: Getting started with telegram-bot.....	2
Remarks.....	2
Examples.....	2
Create a bot with the BotFather.....	2
Chapter 2: Basic methods and how to use them.....	4
Introduction.....	4
Examples.....	4
SendMessage method.....	4
Chapter 3: Webhooks.....	5
Introduction.....	5
Examples.....	5
Setup the webhook.....	5
Credits.....	6

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [telegram-bot](#)

It is an unofficial and free telegram-bot ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official telegram-bot.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with telegram-bot

Remarks

This section provides an overview of what telegram-bot is, and why a developer might want to use it.

It should also mention any large subjects within telegram-bot, and link out to the related topics. Since the Documentation for telegram-bot is new, you may need to create initial versions of those related topics.

Examples

Create a bot with the BotFather

Open a chat with [@BotFather](#) in Telegram and click the `/start` command.

After you get a list of commands Select the command `/newbot` to get this Message:

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

Enter a name for your bot, which can be anything, and send it. After that BotFather will ask for a username for your bot:

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

That must end with "bot". If your desired username is taken or not available, try again with another name.

Finally your bot is created and the following message is shown. You must use the bot token to communicate with the Telegram, so make sure you write it down.

Done! Congratulations on your new bot. You will find it at [telegram.me/???bot](#). You can now add a description, about section and profile picture for your bot, see `/help` for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API: xxx:xxx

With the gained token you now can send a test message by calling the website

https://api.telegram.org/botBOTTOKEN/sendmessage?chat_id=YOURCHATID&text=YOURTEXT

There are two general methods for the telegram bots for interaction: a push and a pull method. Using the pull method you have to call the `/getupdates` every once in a while to check if there are new messages send to your bot. The push method uses a webhook (`/setwebhook` method) to your

script which is called every time a user sends a message to your bot.

The provided information is a [JSON formatted file](#) with all the info (of the sender and the message) you need.

You can write your bot in many languages (PHP, Python, Lua...) as long as they can handle the webhook as an input and can call websites.

For more info you can always use the BOT API documentation which you can find [here](#).

Read [Getting started with telegram-bot](#) online: <https://riptutorial.com/telegram-bot/topic/7624/getting-started-with-telegram-bot>

Chapter 2: Basic methods and how to use them

Introduction

Different methods and their arguments are used to achieve the wished behaviour of your telegram bot, which you created by now (hopefully).

The basic structure of a query is (as mentioned in "Create a bot with the BotFather"):

`https://api.telegram.org/bot*BOTTOKEN*/METHOD?*ARGUMENT1*=*VALUE1*&*ARGUMENT2*=*VALUE2*` You will use this structure with as many arguments you need/ want and this documentation will show you the basic methods and how to use them.

Examples

SendMessage method

`https://api.telegram.org/bot*BOTTOKEN*/sendMessage?chat_id=exampleID&text=exampleText&parse_mode=HTML`

What does this do? With your parameters adjusted correctly this call will send a message to the **exampleID** user with the **exampleText** as message with an HTML encoding. Standard encoding is markdown (see [source](#)) but in some cases you may use HTML.

Next to the *chat_id*, *text* and *parse_mode* you can use the following parameters:

- *disable_web_page_preview* - For disabling the standard preview if you send a link
- *disable_notification* - For disabling the notification on userside (Android users will still get a notification, but without sound)
- *reply_to_message_id* - Sends the message as a reply to another message (with this ID)
- *reply_markup* - Send your custom keyboard with this parameter

Only the *chat_id* and the *text* parameter are required to send a simple message, all other operators are optional. The *disable_web_page_preview* and the *disable_notification* method will need a boolean operator (true or false) to work. All of the parameters are case sensitive so watch out!

For more information jump to the [sendMessage part of the bot api doku](#).

Read [Basic methods and how to use them online](#): <https://riptutorial.com/telegram-bot/topic/10007/basic-methods-and-how-to-use-them>

Chapter 3: Webhooks

Introduction

There are two methods to get messages and informations from your bot:

The `/getUpdates` function wich is documented pretty straight forward in the [specific section of the documentation](#)

Next: The webhook function which is a bit more complex and often causes issues.

The difference basically is that you use the first alternative to **pull** changes and react to them and the second one acts more like a **push** method, by sending a json document.

Examples

Setup the webhook

Documented [here](#) in the API you see that the syntax of the call is

`https://api.telegram.org/bot<token>/setwebhook?url=<yoururl>`. For the most part this is it. If you need more information, take a look at [this guide](#) or look at the requirements (like using HTTPS).

After you set this, you can use the `/getwebhookinfo` method to get information about your webhook (JSON file). The attributes mentioned in this file are explained [in the documentation](#).

Note that there are several ways to set up your webhook. If you have problems with your webhook always describe how you set up your webhook (with custom certificate/without ...). Furthermore if you got problems always check if you met the requirements of the webhooks and if your HTTPS certificate is valid.

Read Webhooks online: <https://riptutorial.com/telegram-bot/topic/10736/webhooks>

Credits

S. No	Chapters	Contributors
1	Getting started with telegram-bot	Community , creyD , Jimmy , Naser Yousefi
2	Basic methods and how to use them	creyD
3	Webhooks	creyD