**Bots**

Bots are Telegram accounts operated by programs. They can respond to messages or mentions, can be invited into groups and can be integrated into other programs.

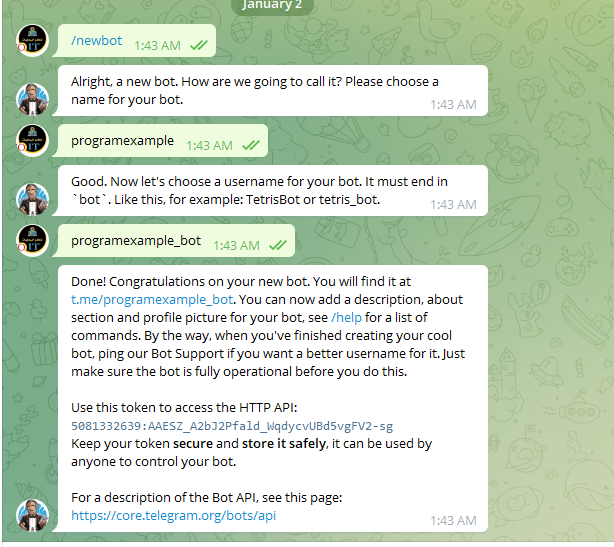
[**Bot Father**](https://telegrambots.github.io/book/1/quickstart.html#bot-father)

**Before you start, you need to talk to**[**@BotFather**](https://t.me/botfather)**on Telegram.**[**Create a new bot**](https://core.telegram.org/bots#6-botfather)**, acquire the bot token and get back here.**



**Bot token is a key that required to authorize the bot and send requests to the Bot API. Keep your token secure and store it safely, it can be used to control your bot. It should look like this:**

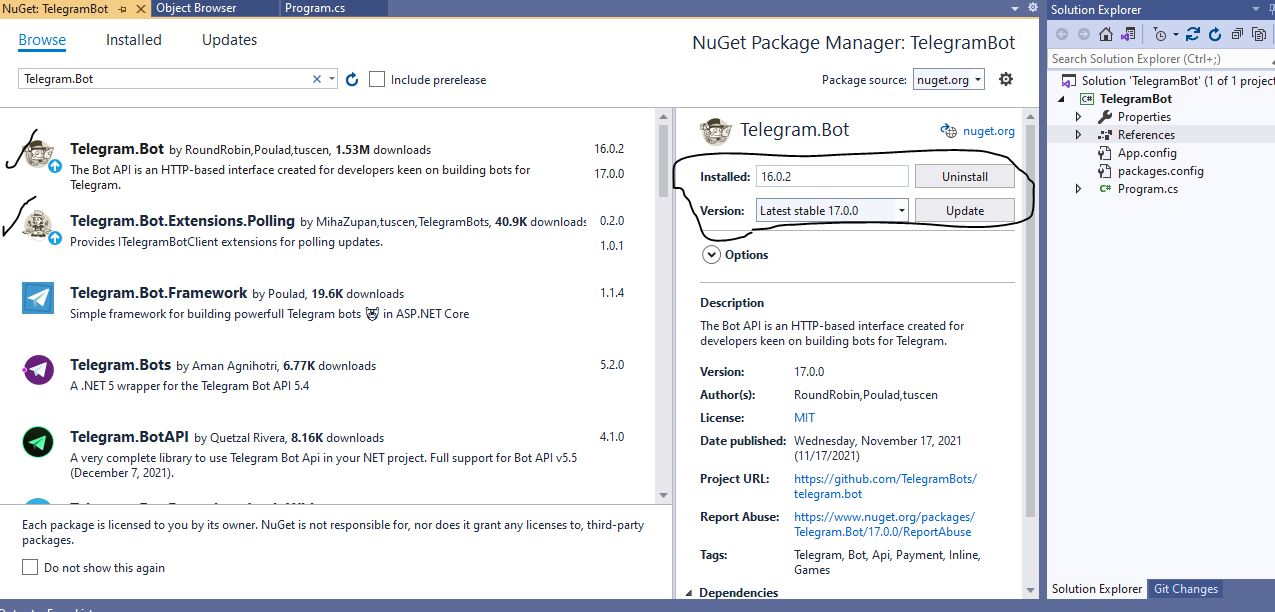
Search Telegram @BotFather



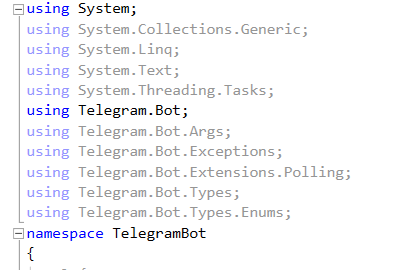
Copy -> HTTP API

Add a reference to [Telegram.Bot](https://www.nuget.org/packages/Telegram.Bot) package:

Add a reference to [Telegram.Bot.Extensions.Polling](https://www.nuget.org/packages/Telegram.Bot.Extensions.Polling/) package:



Add Library Bot



**Paste Http API key BotTelegram**

static TelegramBotClient botClient = new TelegramBotClient("5081332639:AAESZ\_A2bJ2Pfald\_WqdycvUBd5vgFV2-sg");

static void Main(string[] args)

{

botClient.OnMessage += Bot\_OnMessage;

botClient.StartReceiving();

Console.ReadLine();

botClient.StopReceiving();

}

It runs waiting for text messages unless forcefully stopped by pressing Enter. Open a private chat with your bot in Telegram and send a text message to it. Bot should reply in no time.

By invoking StartReceiving(...) bot client starts fetching updates using getUpdates method for the bot from Telegram servers. This operation does not block the caller thread, because it is done on the ThreadPool. We use Console.ReadLine() to keep the app running.

When user sends a message, the HandleUpdateAsync(...) method gets invoked with the Update object passed as an argument. We check Message.Type and skip the rest if it is not a text message. Finally, we send a text message back to the same chat we got the message from.

# [Sending Messages](https://telegrambots.github.io/book/2/send-msg/index.html#sending-messages)

There are many different types of message that a bot can send. Fortunately, methods for sending such messages are similar. Take a look at these examples:

if (e.Message.Text == "Hello")

botClient.SendTextMessageAsync(e.Message.Chat.Id, " Hello ");

# [Photo and Sticker Messages](https://telegrambots.github.io/book/2/send-msg/photo-sticker-msg.html#photo-and-sticker-messages)

You can provide the source file for almost all multimedia messages (e.g. photo, video) in 3 ways:

* Uploading a file with the HTTP request
* HTTP URL for Telegram to get a file from the internet
* file\_id of an existing file on Telegram servers (recommended)

Examples in this section show all three. You will learn more about them later on when we discuss file upload and download.

else if (e.Message.Text == "pic")

{

botClient.SendStickerAsync(e.Message.Chat.Id, "https://github.com/TelegramBots/book/raw/master/src/docs/sticker-fred.webp");

}

# [Audio and Voice Messages](https://telegrambots.github.io/book/2/send-msg/audio-voice-msg.html#audio-and-voice-messages)

These two types of messages are pretty similar. An audio is in MP3 format and gets displayed in music player. A voice file has OGG format and is not shown in music player.

else if (e.Message.Text == "Audio")

{

botClient.SendAudioAsync(e.Message.Chat.Id, "https://github.com/TelegramBots/book/raw/master/src/docs/audio-guitar.mp3");

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Telegram.Bot;

using Telegram.Bot.Args;

using Telegram.Bot.Exceptions;

using Telegram.Bot.Extensions.Polling;

using Telegram.Bot.Types;

using Telegram.Bot.Types.Enums;

namespace TelegramBot

{

class Program

{

static TelegramBotClient botClient = new TelegramBotClient("5081332639:AAESZ\_A2bJ2Pfald\_WqdycvUBd5vgFV2-sg");

static void Main(string[] args)

{

botClient.OnMessage += Bot\_OnMessage;

botClient.StartReceiving();

Console.ReadLine();

botClient.StopReceiving();

}

private static void Bot\_OnMessage(object sender,Telegram.Bot.Args. MessageEventArgs e)

{

if (e.Message.Type == Telegram.Bot.Types.Enums.MessageType.Text)

{

if (e.Message.Text == "Hello")

botClient.SendTextMessageAsync(e.Message.Chat.Id, "Hello ");

else if (e.Message.Text == "pic")

{

botClient.SendStickerAsync(e.Message.Chat.Id, "https://github.com/TelegramBots/book/raw/master/src/docs/sticker-fred.webp");

}

else if (e.Message.Text == "date")

{

botClient.SendTextMessageAsync(e.Message.Chat.Id, DateTime.Now.ToString("yyyy-MM-dd"));

}

else if (e.Message.Text == "time")

{

botClient.SendTextMessageAsync(e.Message.Chat.Id, DateTime.Now.ToString("hh:mm:ss tt"));

}

else if (e.Message.Text == "Audio")

{

botClient.SendAudioAsync(e.Message.Chat.Id, "https://github.com/TelegramBots/book/raw/master/src/docs/audio-guitar.mp3");

}

}

} }}

