

Создание бота с вебхуками на NODE JS

Введение.

Учебная статья начинающим разработчикам от начинающего разработчика.

Необходимая для закрепления пройденного материала.

Вся суть бота это прием/запрос (в зависимости от способа) структуры данных, называемые апдейтами (Updates/обновлений). В настоящее время существует два способа получения обновлений. Мы можем использовать long polling (длинный опрос) или Webhooks.

long polling - данные не присылают по заранее указанному адресу, а мы должны сами посылать на сервера телеграм запросы на их получение используя метод getUpdates. Опрос должен происходить не реже 1 раза в сутки, поскольку дольше Telegram не согласен хранить апдейты. Кроме того, воспользоваться методом getUpdates можно только в том случае, если для бота не установлен webhook.

Создание бота для Telegram на Webhooks предпочтительнее, чем использования метода getUpdates. Webhook держит постоянное соединение, но запросы отправляются только когда произошло событие, например, пользователь ввел сообщение.

Для примера буду использовать наработки своего бота.

Этап 1: Подготовительный.

Для удобства использования терминала рекомендую для всех своих проектов создать папку в корне диска.

Например:



Скачиваем и устанавливаем все необходимые программы и дополнения

1. <https://desktop.github.com> – программа для создания и обновления репозитория прямо с устройства без использования браузера. Необходима для дистанционного просмотра выполненных операций. Неплохо заходит как блокнот с выполненными действиями где можно просмотреть выполненные операции. Необходима регистрация на <https://github.com/desktop>
2. <https://atom.io> – текстовый редактор.
3. `atom://settings-view/show-package?package=atom-terminal-panel` – панель терминала или командной строки в ATOMe
4. <https://nodejs.org/en/> - среда для управления событиями javascript.
5. Для запуска YARN (менеджер пакетов для вашего кода) открыть терминал, прописать путь к необходимой папке «`cd C:\GIT\https`», прописать `yarn install`, проверить установку командой `yarn ver`, в ответ выйдет сообщение с версией файла в противном случае ответа не будет.

```
PS C:\GIT\123> yarn install
yarn install v1.15.2
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
Done in 2.71s.
PS C:\GIT\123> yarn ver
yarn run v1.15.2
```

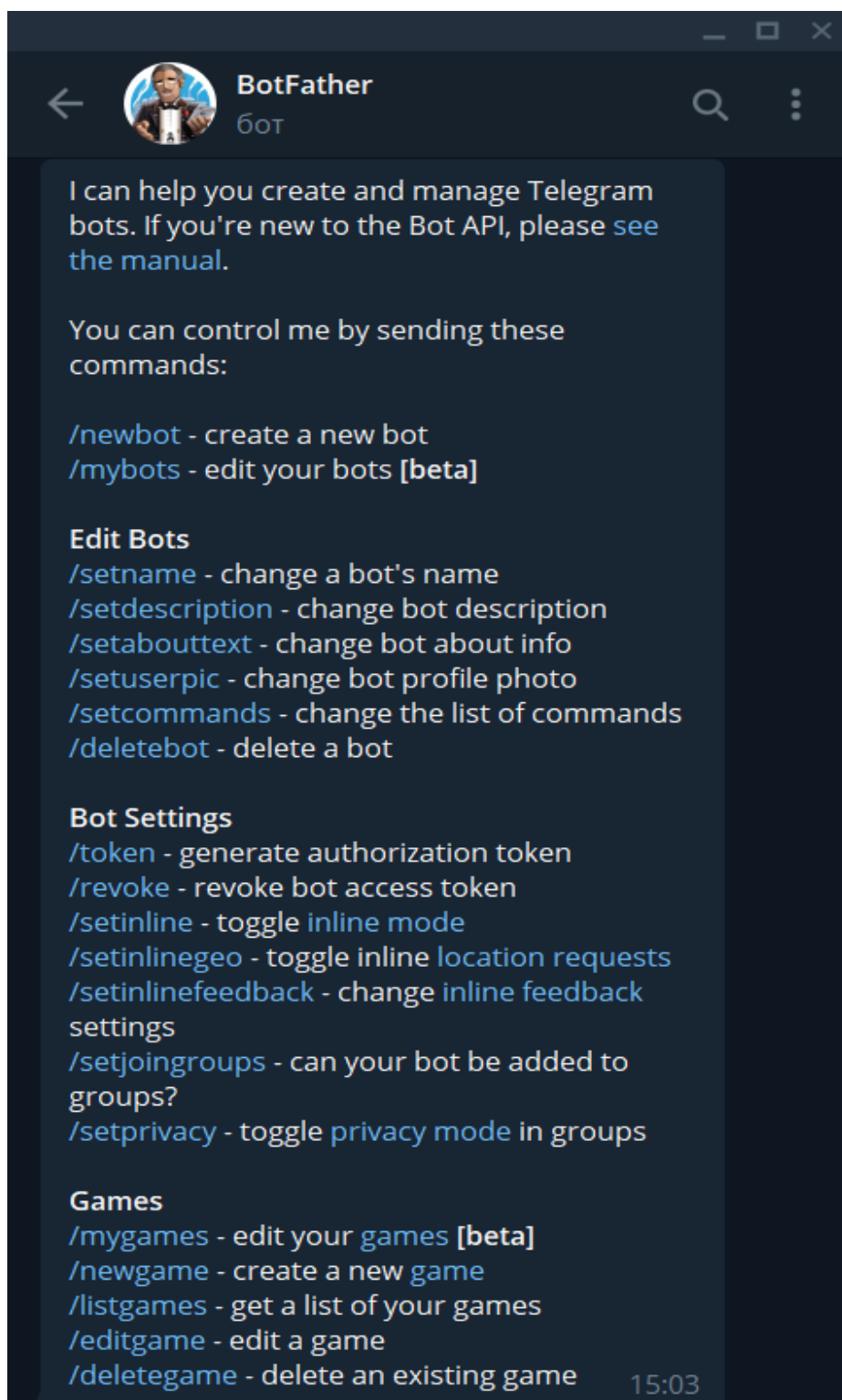
6. <https://ngrok.com> – безопасный URL к вашему серверу localhost (действие не более 8 часов)
Необходим для проверки полученных апдейтов от бота.

Этап 2: Создание

Подробное и четкое описание создание ботов описание его функционала представлено на <https://core.telegram.org/bots> (возможно для просмотра потребуется VPN соединение)

Начнем:

1. Открываем Telegram, ищем @BotFather. По описанию создаем бота.



2. После создания бота /newbot и рассказываем о нём /setabouttext, даем боту имя /setname, вводим описание /setdescription, получаем ТОКЕН бота /token (уникальный идентификатор).

3. Создаем репозиторий в нашей папке через github.com или приложение на ПК
4. Инсталируем в этот репозиторий через терминал все необходимые программы и дополнения (см этап 1)
5. Создаем файл index.html, при помощи HTML заполняем сайт необходимой информацией.
6. Создаем в папке с сайтом папку с файлом .\functions\hello.js

<pre> exports.handler = function(event, context, callback) { const req = JSON.parse(event.body); console.log(req.message.text); var payload = { method: "sendMessage", chat_id: req.message.chat.id, text: "testing reply " + req.message.chat.username, parse_mode: "HTML" }; callback(null, { statusCode: 200, headers: { "Content-Type": "application/ json" }, body: JSON.stringify(payload) }); }; </pre>	<p>-Задаем функцию на событие</p> <p>-Переводим из JSON в удобно читаемый для программы формат</p> <p>-выводим запрос бота на сайт для определения переменных (ответ выводится в виде таблицы)</p> <p>-признак переменной «выгрузка»</p> <p>- свойство признака – по отправке сообщения</p> <p>- свойство признака – отправить отправителю</p> <p>-выводим тестовый текст для проверки работы сайта совместно с ботом</p> <p>- свойство признака – вид раскладки</p> <p>-опрос сайта</p> <p>при нормальной работе отображает код 200</p> <p>тип файла для выгрузки</p>
--	--

7. Открываем АТОМ. В нем терминал, переходим в расположение папки с сайтом. Вводим команду yarn start – инициализируем старт сервера, при запуске нам необходима строка «ИМЯ» server is listening on 9000 в ней нам интересен порт 9000
 8. Открываем ngrok.exe вводим ngrok http 9000 где 9000 наш порт
- видим окно с данными

```

ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Session Expires     7 hours, 59 minutes
Update              update available (version 2.3.28, Ctrl-U to update)
Version             2.3.27
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://99092212.ngrok.io -> http://localhost:9000
                    https://99092212.ngrok.io -> http://localhost:9000

Connections
  ttl    opn    rt1    rt5    p50    p90
   0      0     0.00   0.00   0.00   0.00

```

копируем строчку <https://99092212.ngrok.io> (очень важно S на конце http)

9. Активируем сайт для вебхука. Вставляем данные `https://api.telegram.org/bot[TOKEN]/setWebHook?url=https://99092212.ngrok.io/functions/hello`
в ответ увидим надпись `{"ok":true,"result":true,"description":"Webhook is already set"}`
10. По адресу localhost:4040 находится экран отображения приема/передачи запросов на сайт

11. Открываем бота в telegram вводим любое количество символов (ответ бот будет давать от одного символа).
12. В боте будет ответ в виде текста testing reply
13. По адресу localhost:4040
в графе all request – значение 200 OK
в графе POST – таблица с данными отправителя
в графе 200 OK – вывод запрошенной команды
14. В окне ngrok

```
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 43 minutes
Version             2.3.25
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://eba92606.ngrok.io -> http://localhost:9000
Forwarding           https://eba92606.ngrok.io -> http://localhost:9000

Connections
  ttl    opn    rt1    rt5    p50    p90
    5      0    0.00   0.00   5.01   5.71

HTTP Requests
-----
GET  /favicon.ico          204 No Content
GET  /.netlify/functions/hello 200 OK
POST /.netlify/functions/hello 200 OK
GET  /.netlify/functions/hello 200 OK
POST /.netlify/functions/hello 200 OK
POST /.netlify/functions/hello 200 OK
POST /.netlify/functions/hello 200 OK
```

15. Код функционирует бот выводит информацию
16. Сохраняем наши данные на github через программу с рабочего стола или вручную через сайт
17. Находим оптимальный сервис с доменным именем мне посоветовали netlify.com он принимает аккаунт с github.com и переносит все репозитории.
18. Присваиваем доменное имя по инструкции с netlify.com
19. Активируем api webhook с новым доменным именем по ссылке
https://api.telegram.org/bot/TOKEN/setWebHook?url=https://новое_доменное_имя/functions/hello
20. Проверяем работу бота посылая ему от одного символа.

ИТОГО

Данный бот является тестовой версией и далёк от окончательного варианта. Через создание этого бота познаются основы работы с приложениями для программирования.