УДК 004

Разработка телеграмм бота с помощью WebHook

Еровлева Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье будет представлен пример разработки простого телеграмм бота с использованием WebHook. Работа будет происходить в среде программирования IntelijIdea с использованием языка программирования Java.
Ключевые слова: Java, Telegram, Bot

Telegram bot development with WebHook

Еровлева Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Abstract
This article will present an example of developing a simple telegram bot with WebHook. The work will take place in the IntelijIdea programming environment using the Java programming language.
Keywords: Java, Telegram, Bot

В современном мире большинство компаний внедряют в свои мессенджеры различных ботов. Боты служат для решения вопросов, возникающих у клиентов, начиная от заказа товара, заканчивая решение какой-либо проблемы.

WebHook – это метод при котором сервера телеграмм, когда на них поступает запрос, сами отправляют его на сервер бота.

Цель работы – реализовать телеграмм бота с использований WebHook.

Исследованиями в данной теме занимались следующие авторы.

А.Д.Ананьев рассмотрел реализацию телеграмм бота с нуля на языке программирования Java [1]. Л.А.Наединский реализовал в своей работе библиотеку позволяющую работать с telegrambotapi на языке Java [2]. С.В.Мельников в своей статье описал принцип работы с отладочным интерфейсом Java [3]. Д.И.Самохавалов, Л.В.Дворжанский разработали автоматическую генерацию кода от вложенных сетей петри к событным системам на платформе telegram [4]. М.К.Ермаков и С.П.Вартанов рассмотрели при помощи динамического анализа подход к поиску состояний гонки в многопоточных программах на языке Java. [5].

Для того, чтобы создать телеграмм бота необходимо сперва получить токен самого бота. Для этого в телеграмме необходимо найти «@BotFather». Далее следуя подсказкам бота, создаем нового бота и получаем токен (рис.1).

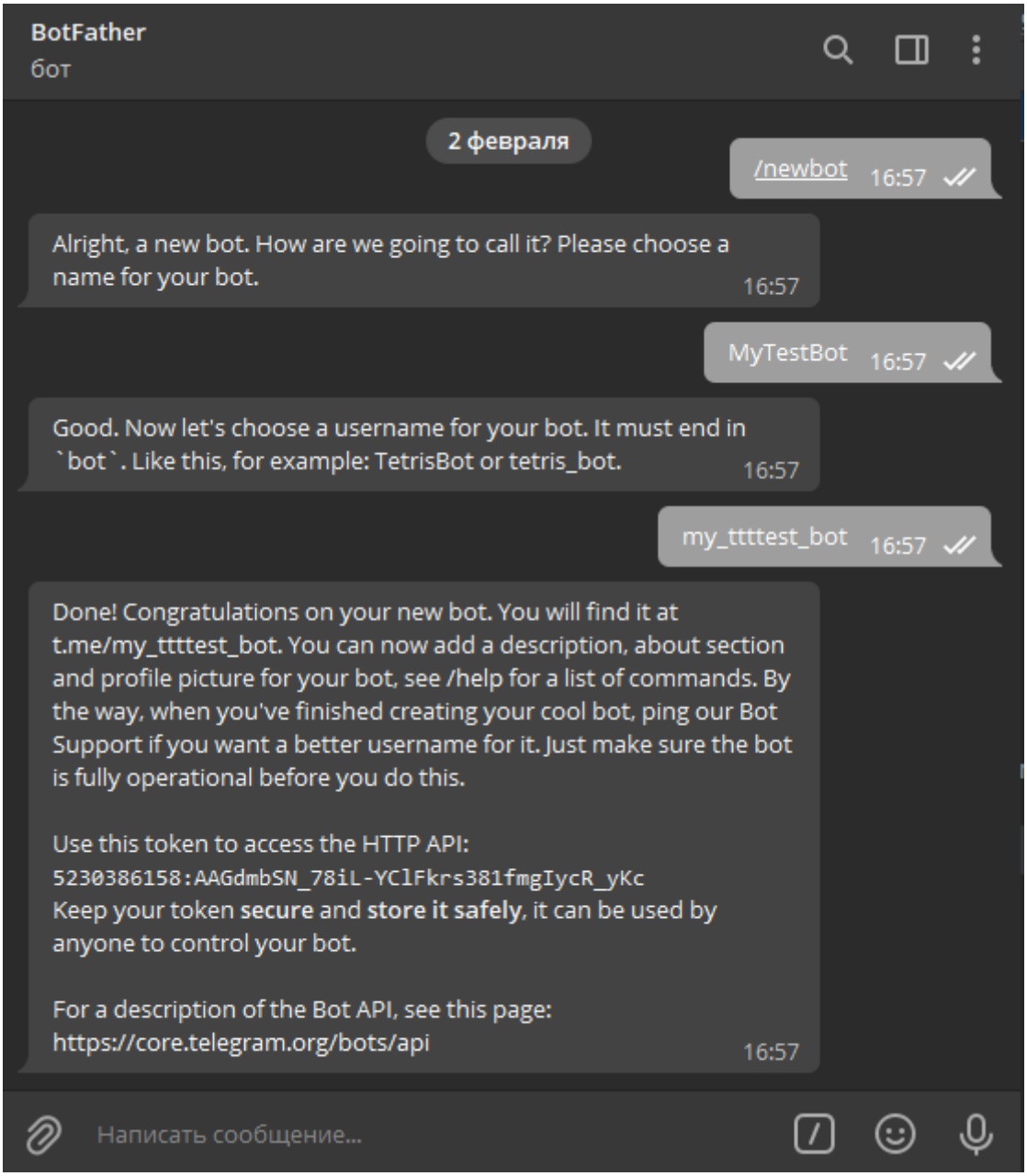


Рисунок 1- Получение токена

Теперь создаем проект в IntelijIdea и в файле конфигурации прописываем данные бота, которые указали при его регистрации (рис.2).

application.properties

```
server.port=8008
telegrambot.userName = my_ttttest_bot
telegrambot.botToket = 5230386158:AAGdmbSN_78iL-YClFkrs381fmgIycR_yKc
telegrambot.webHookPath = https://2aac-62-249-129-16.ngrok.io
```

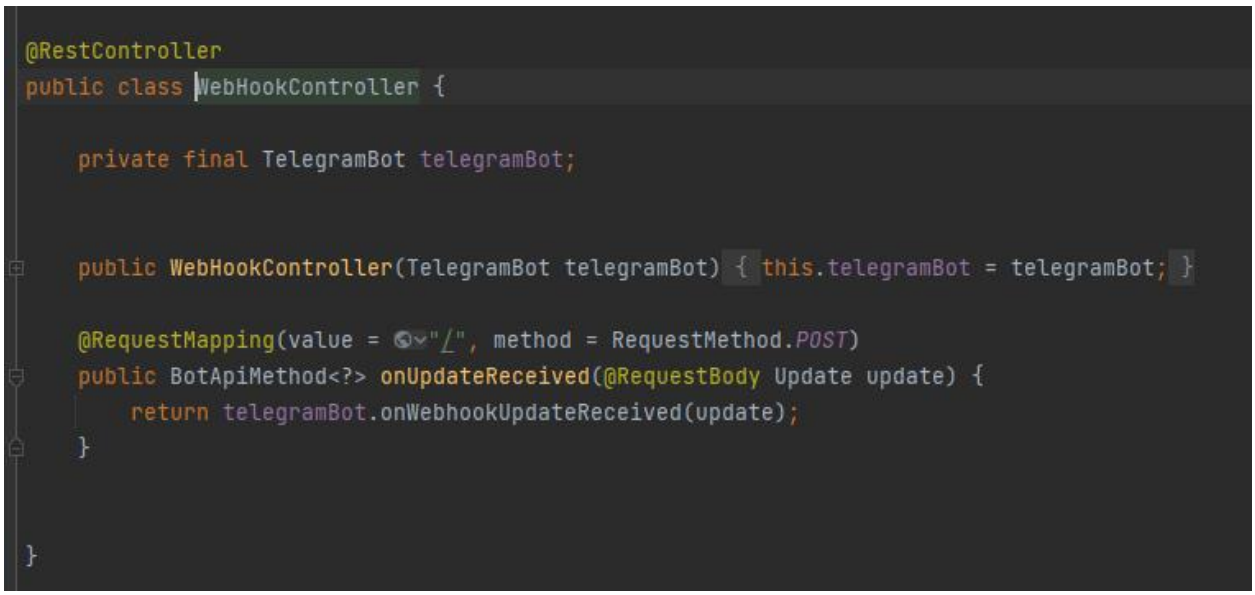
Рисунок 2 – Данные для подключения

После этого создаем класс, который будет получать эти значения и передавать в аунтификацию при попытке отправить сообщения в телеграмм (рис.3).

```
public class TelegramBot extends TelegramWebhookBot {

    private String webHookPath;
    private String botUserName;
    private String botToken;
    private TelegramFacade telegramFacade;

    public TelegramBot (TelegramFacade telegramFacade) { this.telegramFacade = telegramFacade; }

    @Override
    public String getBotToken() { return botToken; }

    @Override
    public String getBotUsername() { return botUserName; }

    @Override
    public String getBotPath() { return webHookPath; }

    public void setWebHookPath(String webHookPath) throws MalformedURLException {...}
    public void setBotUserName(String botUserName) { this.botUserName = botUserName; }
    public void setBotToken(String botToken) { this.botToken = botToken; }
```

Рисунок 3 – Класс для передачи данных

Когда пользователь будет посылать сообщения боту, то бот будет получать каждый раз «update». И что бы перехватить это сообщение необходимо реализовать контроллер куда будут поступать эти обновления (рис.4).



```
@RestController
public class WebHookController {

    private final TelegramBot telegramBot;


    public WebHookController(TelegramBot telegramBot) { this.telegramBot = telegramBot; }

    @RequestMapping(value = "/", method = RequestMethod.POST)
    public BotApiMethod<?> onUpdateReceived(@RequestBody Update update) {
        return telegramBot.onWebhookUpdateReceived(update);
    }


}
```

Рисунок 4 – Контролер-перехватчик

Теперь все поступающие сообщения перехватываются и их можно обработать, для этого создадим метод проверки сообщений (рис.5).

```
public BotApiMethod<?> handleUpdate(Update update) throws TelegramApiException {


    if(update.getMessage().getText().equals("Привет")){
        return new SendMessage(update.getMessage().getChatId(), text: "Привет " + update.getMessage().getFrom().getUserName());
    }

    return new SendMessage(update.getMessage().getChatId(), text: "Вы в главном меню");
}
```

Рисунок 5 – Обработка поступающих сообщений

Теперь, когда пользователь напишет боту сообщение «Привет», то бот ответит «Привет ‘ник-пользователя’». А при написании любого другого сообщения, которое не обработается методом, бот ответит «Вы в главном меню» (рис.6).

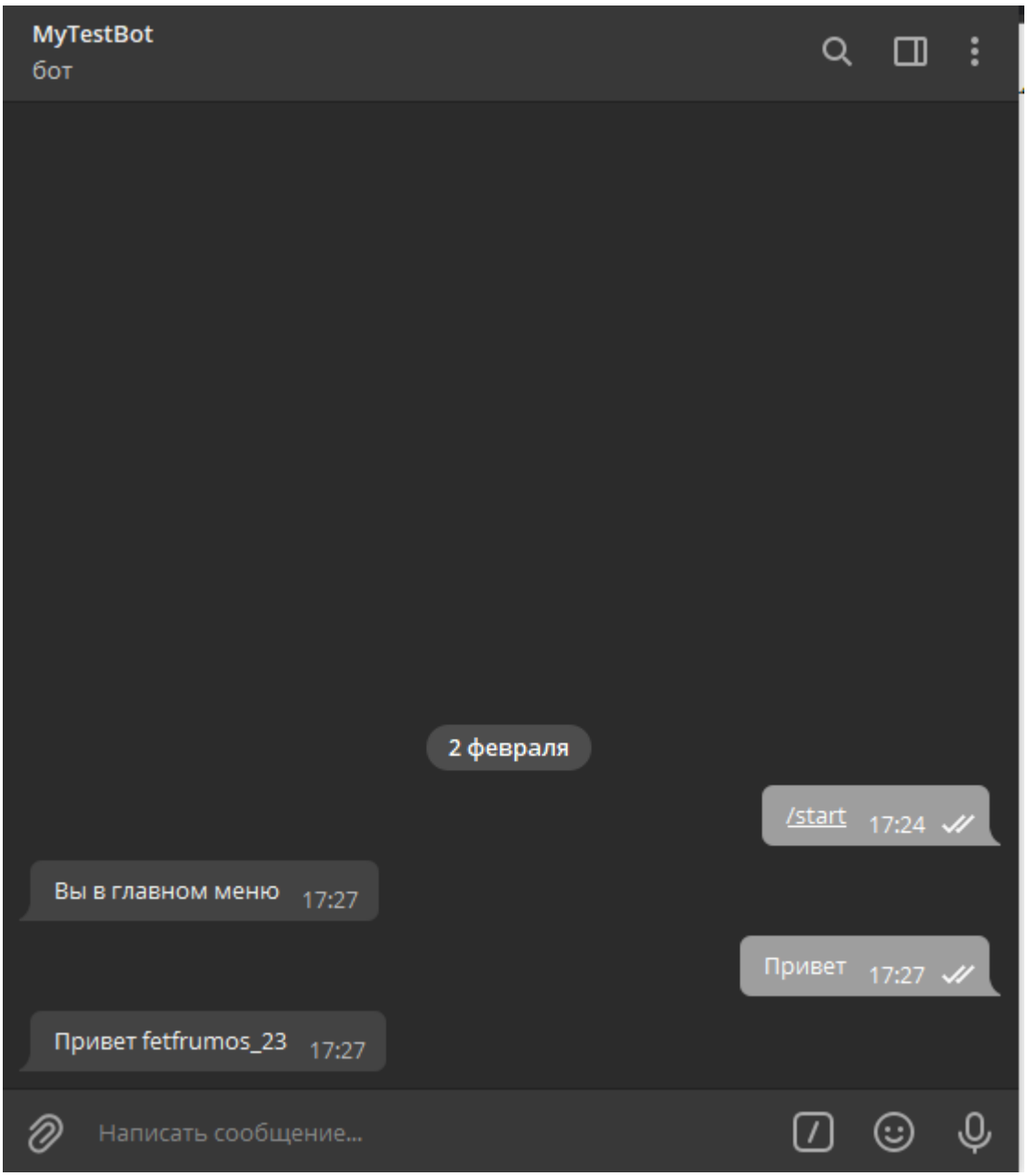


Рисунок 6 – Общение с ботом

В данной статье было рассмотрено создание телеграмм бота с использованием Webhook.

Библиографический список

1. Ананьев А.Д. Чатбот telegram на java с нуля // Новые информационные технологии как основа эффективного инновационного развития. 2020. №3. С. 17-19.
2. Наединский Л.А. Реализация библиотеки для работы с telegrambotapi на java // Информационные системы и технологии: управление и безопасность. 2017. №7. С. 48-63.
3. Мельников С.В. Обзор и применение отладочного интерфейса java (jdi) для обратимой модификации программных продуктов // Современные

проблемы науки и образования. 2018. №8. С. 8-19.
4. Самохавалов Д.И., Дворжанский Л.В. Автоматическая генерация кода от вложенных сетей Петри к событным системам на платформе telegram // Proceedings of the institute for system programming of the ras. 2016. №3. С. 65-84.
5. Ермаков М.К. и Вартанов С.П. Подход к проведению динамического анализа java-программ методом модификации виртуальной машины java // Труды института системного программирования РАН. 2015. №2. С. 39-52.