
АКТУАЛЬНОСТЬ НАПИСАНИЯ СОБСТВЕННОГО TELEGRAM БОТА

Свищёв Андрей Владимирович

Старший преподаватель кафедры практической и прикладной информатики
МИРЭА-Российский технологический университет (РТУ МИРЭА), г. Москва.

Беликов Илья Владиславович

Студент магистратуры, 1 курс
МИРЭА-Российский технологический университет (РТУ МИРЭА), г. Москва

Аннотация

В статье приведены итоги исследования направлений государственной политики поддержки конкуренции в России, от эффективности которой в настоящее время зависит темп экономического развития страны. Выполнен анализ направлений и инструментов государственной поддержки конкурентных рынков, определены приоритетные сферы.

Ключевые слова: telegram, java, chat bot, bot, development.

THE RELEVANCE OF WRITING YOUR OWN TELEGRAM BOT

Svishchev Andrey Vladimirovich

Senior Lecturer at the Department of Practical and Applied Informatics
MIREA - Russian Technological University (RTU MIREA), Moscow.

Belikov Ilya Vladislavovich

Master's student, 1st year
MIREA - Russian Technological University (RTU MIREA), Moscow

ABSTRACT

This article discusses the relevance of creating and ways to use your own telegram bot. The advantages and disadvantages of development in the java language, which can be used in application development, are analyzed.

Keywords: telegram, java, чат-бот, бот, разработка.

В сети большое кол-во ресурсов, где дают возможность реализовать графическим интерфейсом личного бота, который подготовлен к деплою. Проблема заключается в том, что чаще всего люди получают ботов, которые предоставляют исключительно базовый набор команд, который вы изначально в него загрузили, ни работы с БД [1], ни с сетью там не будет. Далее мы кратко рассмотрим возможности что дает нам написание своими руками,

это будет – цельное веб-приложение, при желании к которому можно привязать ни одну базу данных, заниматься парсингом сайтов, просчитывать сложные вычисления, а также выполнять запросы к различным API компаний или своим и прочее. Ограничение - только ваша фантазия.

Первым делом, нужно нашего бота зарегистрировать в телеграм, это делается в чате BotFather. После регистрации, нам требуется узнать имя бота, его мы указываем самостоятельно, и его токен, который будет получен при регистрации. По факту наш чат-бот будет консольным веб-приложением [2]. Графики, то есть фронтенда, в данном случае не будет, исключительно базовая обработка введенных команд. Конечно же, в том же Telegram есть WebApi, однако хоть это и сильно интереснее и визуально более приятно, на данном момент мы его опустим. На рисунке 1, можно увидеть, как произошла успешная регистрация нашего бота и наш токен.

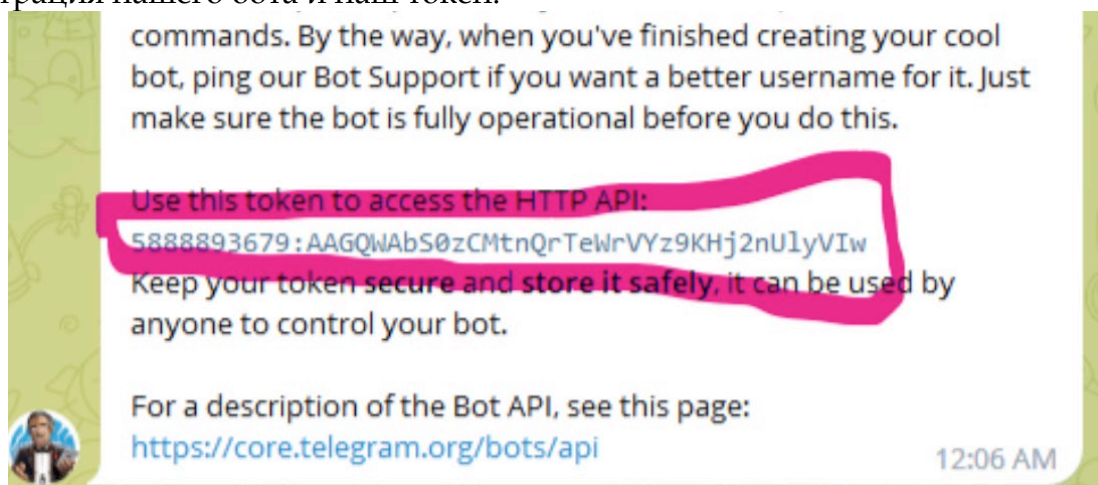


Рисунок 1. Успешная регистрация бота

Далее нам требуется подключить библиотеку в наш файл зависимостей, в приведенном примере это pom.xml (вы так же можете применять Maven), пример предоставлен ниже [3]:

```
<dependency>
  <groupId>org.telegram</groupId>
  <artifactId>telegrambots</artifactId>
  <version>3.7</version>
</dependency>
```

Чтобы наш код мог корректно работать с библиотекой Telegram, нам требуется унаследоваться от класса, который предоставлен подключенной зависимостью, TelegramLongPollingBot, обязательно подключив/ переопределив его методы:

```
public class BotTG extends TelegramLongPollingBot {
  @Override
  public void onUpdateReceived(Update superUpdate) {
    String ourMessage = superUpdate.getMessage().getText();
    sendMsg(superUpdate.getMessage().getChatId().toString(), message);
  }

  public synchronized void sendMessage(String ourChatId, String str) {
    SendMessage sendOurMessage = new SendMessage();
    sendMessage.enableMarkdown(true);
    sendMessage.setChatId(ourChatId);
    sendMessage.setText(str);
    try {
```

```
        sendMessage(sendMessage);
    } catch (TelegramApiException exemption) {
        log.log(Level.SEVERE, "Ошибка: ", exemption.toString());
    }
}
@Override
public String getOurBotUsername(){return "ourBotName";}
@Override
public String getOurBotToken() {return "ourBotToken";}
}
```

Далее представлен пример содержимого класса main:

```
public static void main(String[] args) {
    ApiContextInitializer.init();
    TelegramBotsApi telegramBotsApi = new TelegramBotsApi();
    try {
        telegramBotsApi.registerBot(Bot.getBot());
    } catch (TelegramApiRequestException exemption) {
        exemption.printStackTrace();
    }
}
```

Используя методы `getBotUsername()` и `getBotToken()`, а так же имя нашего чат-бота плюс его токен, мы сможем запустить и посмотреть на его работу. На данный момент, чат-бот пересылает текст тех сообщений, что мы отправим ему, то есть, мы получаем зеркальный ответ. Происходит это так: когда происходит запуск приложения, начинается взаимодействие с сервером Telegram, происходит GET отправка запросов в раз определенное время по данному пути `“https://api.telegram.org/BotToken/getMe”`, тут `BotToken` – тот самый токен(строка), что мы получили в начале. Ответ будет получен в JSON формате, там будут находиться все сообщения/информация по запросу. Они будут обработаны библиотекой телеграмма и после попадут в метод `OnUpdateReceived` в виде объекта `Update` [4]. С данным объектом далее мы и будем производить все наши манипуляции. Главным плюсом любых Telegram-ботов, они могут работать на любом устройстве, смартфон, ПК, даже чайник, если на него можно будет поставить приложение телеграмма, а для тестирования требуется запустить приложение и написать что-либо боту. Мы избегаем, деплоя после каждого коммита. Это максимально удобно.

В заключении, чат-боты на данный момент используются повсеместно, иметь свой личный – отличная возможность потренироваться в написании кода, а также оптимизировать какой ни будь рутинный процесс. В последствии, можно всегда переключиться на разработку WebApi приложений в Telegram, что еще больше расширит ваши возможности.

Список литературы:

1. Федоричев, Л. А. Реализация многопоточности в языке Java / Л. А. Федоричев, О. В. Букунова. – Санкт-Петербург: Лань, 2024. – 72 с. – ISBN 978-5-507-48153-8.
2. Инструментальное программное обеспечение разработки и проектирования информационных систем: учебное пособие / А. А. Куликов, В. Т. Матчин, А. В. Сеницын, В. В. Литвинов. – Москва: РТУ МИРЭА, 2022.

3. Волков, М. Ю. Разработка серверных частей интернет-ресурсов: учебное пособие / М. Ю. Волков, В. В. Литвинов, А. А. Лобанов. — Москва: РТУ МИРЭА, 2021. — 188 с.
4. Архитектурные решения информационных систем / А. И. Водяхо, Л. С. Выговский, В. А. Дубенецкий, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург: Лань, 2023. — 356 с. — ISBN 978-5-507-46063-2.

References:

1. Fedorichev, L. A. Implementation of multithreading in the Java language / L. A. Fedorichev, O. V. Bukunova. - St. Petersburg: Lan, 2024. - 72 p. — ISBN 978-5-507-48153-8.
2. Tool software for the development and design of information systems: textbook / A. A. Kulikov, V. T. Matchin, A. V. Sinitsyn, V. V. Litvinov. — Moscow: RTU MIREA, 2022.
3. Volkov, M. Yu. Development of server parts of Internet resources: textbook / M. Yu. Volkov, V. V. Litvinov, A. A. Lobanov. — Moscow: RTU MIREA, 2021. — 188 p.
4. Architectural solutions of information systems / A. I. Vodyakho, L. S. Vygovsky, V. A. Dubenetsky, V. V. Tsekhanovsky. — 3rd ed., erased. - St. Petersburg: Lan, 2023. - 356 p. — ISBN 978-5-507-46063-2.