SECTION 4. Computer science, computer engineering and automation.

Vadim Andreevich Kozhevnikov
Senior Lecturer
Peter the Great St. Petersburg Polytechnic University
vadim.kozhevnikov@gmail.com

Oleg Yurievich Sabinin
Candidate o fEngineering Sciences, Associate Professor
Peter the Great St. Petersburg Polytechnic University
olegsabinin@mail.ru

Julia Efimovna Shats
student
Peter the Great St. Petersburg Polytechnic University
julia7476@gmail.comn

LIBRARY DEVELOPMENT FOR CREATING BOTS ON SLACK, TELEGRAM AND FACEBOOK MESSENGERS

Abstract: *Article contains analyze of existing libraries for messenger bot creation. Also this article describes creation of library that take into consideration advantages and disadvantages of existing solutions.*
Key words: *Bot, library, software development, Java, Telegram, Facebook, Slack, messenger.*
Language: *English*
Citation: Kozhevnikov VA, Sabinin OY, Shats JE (2017) LIBRARY DEVELOPMENT FOR CREATING BOTS ON SLACK, TELEGRAM AND FACEBOOK MESSENGERS. ISJ Theoretical & Applied Science, 06 (50): 59-62.
Soi: http://s-o-i.org/1.1/TAS-06-50-4   *Doi*: crossref https://dx.doi.org/10.15863/TAS.2017.06.50.4

UDC 004.4'2

Introduction

In nowadays more and more people are using messengers. Messenger is a desktop, web, or mobile application for communication over the internet. It must be said that the concept of communication over the internet does not imply only text messaging. Modern messengers in addition to messaging implement voice and video, file sharing, web conferencing. [1]

Today companies are paying special attention to instant messengers, which are becoming serious competitors to other mobile applications. This is facilitated by chatbots - conversational-program, which are already near to become full-fledged interlocutors and assistants. They help to learn a news, exchange rates, order goods, pick up a suitable flight, music, book, and even can act as a distance teacher.

Motivation

The purpose of this work is to develop a library that will allow developers to create bots for instant messengers such as Slack, Facebook Messenger, Telegram in Java.

There is a large number of users that have difficulties installing a set of mobile applications for every company. Analyzing this situation, many companies come to the conclusion that such users are more likely to use one messenger with several chatbots that replace multiple applications. Because each person uses what is more suitable for him, companies need to support as many platforms as possible, thereby increase the number of clients. Therefore, it would be simpler to be able to communicate with users on multiple platforms without developing for each platform. Hence, there is a demand for existing libraries for fast and simple chatbots development. Therefore, the main task for us was the creation of the library that supports Slack, Facebook Messenger, Telegram platforms. [8]

Implementation

Chat bots can be developed in any programming language with which you can create a web API. The back-end receives the message, thinks what to answer, and returns the response to the user.

For the library that must run on multiple platforms, we need to choose appropriate language. Java is one of most popular language that can run on a huge number platforms. (By Tiobe, PyPl and IEEE Spectrum)

API (application programming interface) is set of classes, procedures, functions, structures and constant provided by application, library, service or operation system for use in external software. In this case, the API is provided to developers in the public domain.

Your web server will install so-called webhooks - URL-based connections between your bot and the chat platform. There are two mutually exclusive ways of receiving updates for bot — the getUpdates method on one hand and Webhooks on the other. Incoming updates are stored on the server until the bot receives them either way, but they will not be kept longer than 24 hours. Webhooks will allow you to safely send and receive messages through simple HTTP requests. [5, 6]

Our library will allow developers to call Java methods to send and receive data from messengers.

```java
@Override
public void sendMessage(Message message) {
    SendMessage libraryMessage = new SendMessage();
    libraryMessage.setChatId(message.getChat().id);
    if (message.getType() == MessageType.TEXT_MESSAGE) {
        libraryMessage.setText(((TextMessage)message).text);
    }
    try {
        sendMessage(libraryMessage);
    } catch (TelegramApiException e) {
        e.printStackTrace();
    }
}
```

Figure 1 - Sending message demonstration.

Platforms features

Each platform has its own individual characteristics and features, which must be taken into account. Examples of features:

1. Telegram support new types of input, keyboards that integrated into a message.

Unlike the regular keyboard, clicking on the button of the integrated keyboard will not send messages to the bot - instead, the data will be transferred in the background, which mirrors the interaction with a normal application. [2]

2. Facebook messages can be either just text or Structured Text, which are divided into:

□ button:

This type is used to send messages, which require user response.

Buttons can be one of two types:

o Sending response to bot;
o Redirecting to internet address;

□ generic:

Elements are intended to send cards of goods or other items having a similar structure.

Each member may have title, subtitle, description, image and buttons.

□ receipt:

A receipt may contain information about goods, cost, delivery address, discounts. [3]

3. Slack allows you to use additional types of keyboards and mark individual user messages. [4]

Overview of existing libraries

Already exist a sufficient number of libraries that implement the integration of messenger APIs to create bots on various programming languages. But we did not find any that would unite these popular platforms in one, with full functionality, and would be implemented in the Java language. (Table 1)

Table 1

Compare existing and existing libraries.

Libraries			
Our library	*BotKit*	*Claudia Bot Builder*	*Universal Bot Framework*
Programming languages			
Java	Java-Script	Java-Script	JavaScript
Supported systems			
Slack, Face-book Mes-senger, Tele-gram	Slack, Face-book Mes-senger, Twilio, Micro-soft Bot Frame-work	Facebook Mes-senger, Slack, Skype, Viber, Telegram, Twilio, Amazon Alexa, Line, Kik, GroupMe	Facebook Messenger,Telegram, Kik, Skype
Targeted			
Develo-pers	Deve-lopers and ordi-nary users	Develo-pers	Developers
Объединение интерфейсов			
Full support	Full support	No support for additional features	No support for additional features
Full functionality support			
Full support	Full support	Full support	No support for additional features.

It should be noted that in this table there is no comparison with open-source projects that implement the work in the java language separately for single platforms. Since they have a different task.

Testing

After the library was implemented, testing was performed on supported platforms. (Fig.2) For the tests, it was decided to create a bot that audits user actions in the PostgreSQL DBMS using created library [7].

The server was implemented through the Open Server with the PostgreSQL module. To work directly with the DBMS was used PgAdmin. [9]

The test bot responds to commands entered by the user.

After starting the bot, it connects to the server log file and starts tracking changes. Once a new entry is added to the log file and the administrator enters command the bot immediately send the message with event description.

With the command "/find_action" you can tell the bot what events you want to find in logs. For example, the command "/find_action select"

tells the bot to send data about the select requests being executed.

For example "/find_composite user1 select" - will output of select operations performed by a specific user (user1).

Also, it is possible to receive error notifications. Displaying of the necessary errors occurs through the command "/find_warning".

If you need to use a different log location you can specify it using the command "/setlog" followed by the path to the server. The path can be specified in Windows style (with backslashes) and in Unix style (with regular slashes). The path will be converted to the paths on which OS the PostgreSQL server is located. Also, the "/find" command is available for the database administrator, which is an extremely flexible way of finding specific records in the log. As arguments in this command you can use time intervals, types of records (select, insert, start, stop, etc.), the duration of the action. In the further development, additional opportunities will be added to this search.

With the development of the library, the bot will get a more user-friendly interface using buttons and lists.
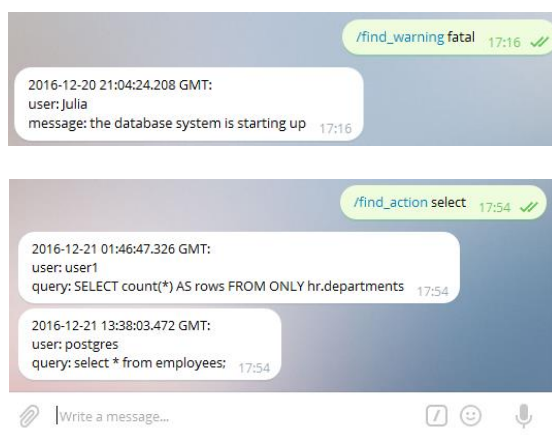
Figure 2 - Demonstration the work one of the test bots on the Telegram platform.

Practical significance

Created library is aimed at developers, who need frequent dispatches for messengers various platforms. Because, using the library, they have the opportunity not to create a separate program for each platform, but make the dispatch immediately to selected platforms, through several lines of code.

Companies that need such mailing fair amount, reiterate the importance of the development using this or similar libraries.

Conclusion

A ready-to-use Java library was created. This library allows developers create bots within the unified system for Slack, Telegram and Facebook platforms. In the future, it is planned to add new platforms, as well as to facilitate development by adding auxiliary design patterns for bots.

References:

1. Obzor Messendzherov. Available: http://www.voipoffice.ru/tags/messendzhery/. (Accessed: 07.06.17)
2. API Telegram. Available: https://core.telegram.org/bots/api. (Accessed: 07.06.17)
3. API Facebook. Available: https://developers.facebook.com/docs/messenger-platform. (Accessed: 07.06.17)
4. API Slack. Available: https://api.slack.com/bot-users. (Accessed: 07.06.17)
5. Java Documentation. Available: http://www.oracle.com/technetwork/java/javase /documentation/index.html. (Accessed: 07.06.17)
6. Massovaja rassylka. Available: https://sms-manager.ru/info/massovaya_rassylka_cherez_w hatsapp_viber_rassylka/. (Accessed: 07.06.17)
7. PostgreSQL Documentation. Available: https://www.postgresql.org/. (Accessed: 07.06.17)
8. Obzor chat-revolyucii. Available:: http://rb.ru/longread/bots-are-the-new-apps/ (Accessed: 07.06.17)
9. OpenServer. Available: https://ospanel.io/. (Accessed: 07.06.17)