УДК 004.45

Абенова Самал Казымовна, Михно Сергей Валерьевич,
Бержицкая Лилия Анатольевна
Международный Университет «Астана»
(Нур-Султан, Казахстан)

TELEGRAM – БОТ КАК ПРОСТОЙ И УДОБНЫЙ СПОСОБ ПОЛУЧЕНИЯ ИНФОРМАЦИИ

Аннотация: *в статье рассматривается один из наиболее удобных способов быстрого и качественного получения необходимой информации посредством диалога пользователя с Телеграм-ботом.*

Ключевые слова: *информация, мессенджер, Телеграм, бот, приложение, диалог, Интернет, учебные занятия.*

Abenova Samal K., Mikhno Sergey V., Berzhitskaya Liliya A.
«Astana» International University
(Nur-Sultan, Kazakhstan)

THE TELEGRAM – BOT AS A SIMPLE AND CONVENIENT WAY TO OBTAIN INFORMATION

Abstract: *this article presents one of the most convenient ways to quickly and competently obtain the necessary information using a dialogue between a user and Telegram bot.*

Keywords: *information, messenger, Telegram, bot, application, dialogue, the Internet, training sessions.*

The pace of modern life is extremely dynamic, not least because of a rampant development of information technology. Currently, a wide range of gadgets and applications are designed to simplify as much as possible the lives of their ordinary users in solving everyday tasks, and to provide sufficiently quick access to any information of interest.

According to statistics, the largest group of people who actively use both the Internet and the functionalities of electronic devices and various applications are schoolchildren and college students. Any modern educational institution strives to develop and improve the quality of the educational process, its accessibility and interactivity. At the moment, most universities and secondary specialized educational institutions have their website within the boundlessness of World Wide Web, where the most relevant information about the educational institution, its history, personnel and faculty members is posted, student life and various events get covered. Generally, the main goal of such web services is to draw the attention of intending students, their parents, potential employers, firstly, by providing information on the current projects, research, and other scientific and extracurricular activities conducted in a particular educational institution. However, the students themselves may not always receive quick and convenient access to the information related to their academic activities, for example, updates in their schedules of training sessions.

There are several options for solving this kind of problems, for example, to create a mobile application, which displays the schedule for each group and faculty members. The second option is to create of a Telegram bot, through which information from the site is accessed. For the purposes of this article, we will consider the second option.

The first user of Telegram was presented in 2013, and since then the messenger is growing in popularity both in Kazakhstan and worldwide due to its usability, safety, high-speed operation and accessibility. An interesting distinctive feature of Telegram is that anyone can use all the functionalities of the service to create their own programs – the bots. Such a program performs the functions of an indispensable assistant, which, through the most convenient interface and set of commands, allows solving an elementary operation without interrupting the communication. Communication with the bot is carried out through a dialogue, as well as with an ordinary user. In this context, purpose of the research is to create a simple and convenient Telegram bot, which would provide fast and safe access to the necessary information, and would have a convenient and simplest interface.

To achieve this purpose, the following tasks have been performed:

- Studying the ways to obtain information on schedules of training sessions in the University
- Development of the architecture and implementation of the software design that allows, through a dialogue with the user, identifying the specific parameters of the query and provide a fast and reliable way to obtain a list of training sessions of a group.
- Testing the final application.

The simplest way to obtain a list of training sessions of a specific group for a specific week was the HTTP GET method at a certain address, which considers the group ID and the timestamp of the start of the week in question. As a response from the server, a JSON array is presented, the elements of which are the training sessions of the group within a week, containing the following data:

- ID of the group.
- Subgroup number.
- Name of the discipline.
- Type of the training session.
- Number of the training session.
- Week day.
- Academic building.
- Classroom number.
- ID of the group.
- Full name of the faculty member.

The project developed consists of five interconnected modules: main module; module of interaction with the training sessions database; module of auxiliary tools; script for updating the training sessions database; module with the necessary project configuration. Also, the following files are connected with the application: SQLite file of the training sessions database, the Vedis file of the user profiles repository.

The framework for the entire project is the main module. This file may be divided into four logical subdivisions:

- Setting up the logger.

- Web server class description.
- Various auxiliary tools.
- User’s Message Handlers.

Logging the errors and collection of notifications from various modules of the project are integral parts of any good application that claims to be reliable under the continuous operation conditions. When operation of the logger is properly managed, even in the most unforeseen situations, the developer will see the problem and be able to quickly correct the error.

Choosing between Longpolling and installing Webhook is obvious.

In the first case, connection is opened for a short time and the Telegram servers are polled for changes. This method is not a good choice for several reasons. First, polls occur from time to time, which causes delays. Secondly, with this method of polling, the Telegram servers start to return the errors from time to time, which is unacceptable under the continuous operation conditions of the application. Thirdly, an excessive system load is caused. In the second case, by installing Webhook, we subscribe for certain events, i.e. changes appearing on the Telegram servers that are automatically transmitted to the “subscribing” system in case of appearing. The price for this way to obtain updates is the necessity to find a full-fledged web server on a computer designed to run the application.

The auxiliary tools mainly include the features of building and forwarding various user keyboards with the variety of prompts and notifications to the user.

As message handlers, features are presented that are framed in the special decorators. Each such “attendee” accepts the messages of a well-defined format, which allows building reliable program logic and making the project code easy to understand.

Regarding the training sessions database, it is worth noting that only one table is used, which has the following structure:

- ID of the group.
- Week day.
- Number of the training session.
- Classroom number.
- Name of the training session.
- Number of the academic building.
- Type of the training session.
- Full name of the faculty member.

Since the application gives access to training sessions for the current week only, the number of entries in the training sessions table is averagely equal to one thousand.

As for the module of interaction with the training sessions database, it contains a number of features that provide access to the training sessions of a specific group on a particular day of the week and allow updating the list of the training sessions of a certain group.

Special attention should be paid to the repository of Vedis user profiles, which is of the key-value type. In this case, by the user's chat ID, we store a serialized JSON string that contains the following user data:

- Training course.
- ID of the group.
- Current status of the user.

The states and transitions between them will be discussed later.

The advantages of using key-value repository are reflected in the absence of the necessity to constantly make queries on the database. All information on the user profiles will be reserved even when the application is restarted. It also provides a sufficiently fast and reliable connection with the repository to record and obtain data.

In the test version of the application, the script for updating the training sessions database is launched manually. The final version of the application launched and operating on a separate server will automatically launch this module for performance at a certain time every day. When updating information, the lists of training sessions of a specific group are downloaded from the website of the university through an http query. Then the obtained information is processed and the table data is modified. Updating a database takes about nine seconds on average.

An integral part of the application is the configuration file which presents the objects used in the application code. This module allows the developer to change the values of certain properties of the application without directly changing the program code.

After a brief review of the project architecture, it is necessary to:

- Understand how the dialogue between the user and the developed software system is carried out.
- Perform testing of all the features of the end product.

Let us assume that a user decided to use this software product as a daily assistant. The student is left wondering how to find and to write a message the bot account in the Telegram messenger. The search can be performed quite simply. It suffices to go to the search section by selecting the magnifying glass icon in the upper right corner of the application window. The next step is to enter the “@” symbol in the search box. After that, start entering the bot account name “printschedule_test_bot” (Fig. 1).

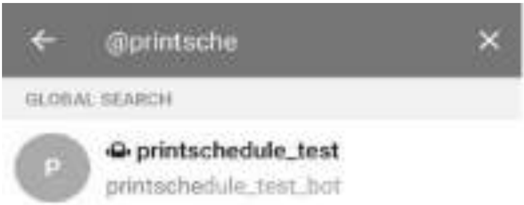


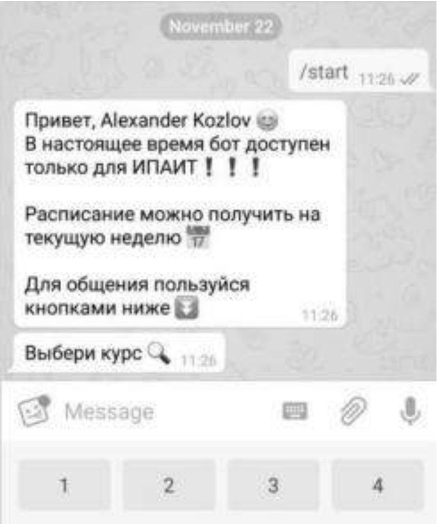
Figure 1 - Searching for a bot account in the Telegram messenger

The next step is to select the bot account presented in the GlobalSearch section. Upon successful completion of this action, the user reaches the startup page of the dialogue. At the bottom of this page there is the “Start” button (Fig. 2). To start a dialogue with the bot, it is necessary to click on the button displayed.

Figure 2 - Startup page of the dialogue

The dialogue with the user is obtained by using several workspaces, which are special keyboards, the interaction with which provides output of necessary information, and multiple help and prompt messages.

Upon successful start of the dialogue by the user, the bot sends a welcome message with notifications that are currently relevant, and provides access to the first workspace “Enter the training course number” (Fig. 3).


November 22

/start 11:26

Привет, Alexander Kozlov 😀
В настоящее время бот доступен только для ИПАИТ ❗❗❗

Расписание можно получить на текущую неделю 📅

Для общения пользуйся кнопками ниже ⬇️ 11:26

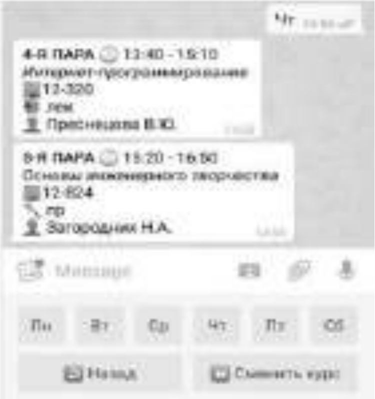
Выбери курс 🔍 11:26

Figure 3 – Entering a training course number by the user

To continue the dialogue, the user is required to select a training course and click on the relevant button. This action will entitle the bot to open the “Enter an ID of the group” workspace for the user.

This workspace consists of special buttons, either of which is entitled with a unique ID of the group of the selected training course. Also, this keyboard includes a service button "Back", when interacting with which it is possible to return to the previous workspace.

To go to the next step of the dialogue and to get access to the workspace “Selecting a week day”, the user is required to select a training group and click on the relevant button. The final workspace is a keyboard with a buttons array that allows selecting a day of the week and obtaining a list of training sessions for a selected period of time. Just like in the previous step, this workspace contains the service button “Back”. Also, the keyboard contains a unique “Change a course” button to immediately go to the initial workspace of selecting a training course.



Чт

4-Я ПАРА 13:40 - 15:10
Интернет-программирование
12-320
лек
Пресницова В.Ю.

5-Я ПАРА 15:20 - 16:50
Основы инженерного творчества
12-824
пр
Загородник Н.А.

Figure 4 - The result of the user's dialogue with the bot

The result of the user's dialogue with the bot is shown in the figure (Fig. 4). The output data contain: sequence number of the training session, time and name of the training session, academic building and classroom, type of the training session, full name of the faculty member.

Having reviewed the way the bot interacts with the user, we can conclude that certain states of the user change when transferring from one workspace to another. Indeed, there are four such states:

- Start
- Enter a course
- Enter a group
- User is registered.

In other words, the dialogue can be presented as a finite state machine. To sum up, purpose of the research was achieved. A simple and convenient bot-application was developed, which allows obtaining necessary information within the shortest period of time. In the course of performing the first task, the simplest and most convenient way, the http query specifically, was examined and selected to obtain a list of training sessions for the group. As part of the second task, solid project architecture was developed, and the application was implemented, which allows obtaining a list of training sessions for the group through a dialogue with the user.

While performing the third task, the full path from search for the bot account to obtaining the information necessary for the user has been completed.

REFERENCES

1. N.B. Akoeva. Interactive techniques to work at the museum exhibition // Cultural Life of Southern Russia. 2016. No. 3 (62). p. 112115.
2. V.I. Anikin. Information support for the international security systems // In the collection: Informatization of security systems ISB-95. Proceedings of the Fourth International Conference, International Informatization Forum. 1995. p. 1820.
3. A.A. Budaeva. Using the method of standards to improve the quality of grouping in taxonomy problems // In the collection: IT-technologies: development and

applications. Collection of reports. North Caucasian Mining and Metallurgical Institute (State Technological University). 2016. p. 3-9.

4. G.S. Osipov. Single-channel queuing systems with unlimited queue in anylogic // Bulletin of science and practice. 2016. No. 8 (9). p. 92-95.
5. A.N. Pechnikov, T.P. Avanessova, A.N.Shikov. E-learning. - St. Petersburg, 2014. – p.73
6. V.E. Chernikova. Values of culture in the modern information world. - Stavropol, 2017. – p.185.