

УДК 004.428.4

https://doi.org/10.33619/2414-2948/109/20

РАЗРАБОТКА И ЗАПУСК TELEGRAM-БОТА – АГРЕГАТОРА СООБЩЕНИЙ

©Сафина Г. Ф., ORCID: 0000-0002-7326-0896, SPIN-код: 4562-2453,
канд. физ.-мат. наук, Уфимский университет науки и технологий, Нефтекамский филиал,
г. Нефтекамск, Россия, safinagf@mail.ru

©Коняев Ю. С., Уфимский университет науки и технологий, Нефтекамский филиал,
г. Нефтекамск, Россия, iury.conyaev2016@yandex.ru

DEVELOPMENT AND LAUNCH OF THE TELEGRAM BOT – MESSAGE AGGREGATOR

©Safina G., ORCID: 0000-0002-7326-0896, SPIN-code: 4562-2453, Ph.D., Ufa University of
Science and Technology, Neftekamsk Branch, Neftekamsk, Russia, safinagf@mail.ru

©Konayev Yu., Ufa University of Science and Technology, Neftekamsk Branch,
Neftekamsk, Russia, iury.conyaev2016@yandex.ru

Аннотация. Приведен процесс создания телеграм-бота для агрегации сообщений с целью упрощения коммуникации между самоуправлением и студентами общежития вуза (на примере Нефтекамского филиала УУНиТ). Код бота написан на языке программирования Python в среде программирования PyCharm. Для запуска бота используется хостинг, специализированный под Python-проекты: PythonAnywhere.

Abstract. Describes the process of creating a telegram bot for aggregating messages in order to simplify communication between local government and students of the university dormitory. The bot code is written in the Python programming language in the PyCharm programming environment. To launch the bot, hosting is used, specialized for Python projects: PythonAnywhere.

Ключевые слова: telegram-бот, разработка, язык Python, онлайн-хостинг.

Keywords: telegram bot, development, Python language, online hosting.

Telegram — мессенджер, получивший широкое распространение во многих странах. Сегодня телеграм-канал стал практически обязательным атрибутом любой компании, медийной личности, свои каналы есть у СМИ и государственных учреждений. Многие каналы в мессенджере для упрощения взаимодействия со своей аудиторией прибегают к использованию ботов [1]. Область применения ботов ограничена лишь возможностями современного программирования.

Рассмотрим процесс создания своего бота для Telegram на языке программирования Python. Все студенты, проживающие в общежитии Нефтекамского филиала УУНиТ состоят в группе в Телеграме. Помимо объявлений самоуправления и предложений (просьб) студентов в чате также происходит постоянное общение, что затрудняет работу самоуправления. Для решения этой проблемы разработали чат-бота, который может принимать жалобы и предложения, адресованные старостам общежития, и пересылать их в отдельную беседу, в которой состоят только члены самоуправления.

Для начала нужно зарегистрировать нового бота через другого бота – BotFather. Для этого достаточно следовать простым инструкциям, придумать имя бота и его уникальный логин. После этого получим токен нашего бота — строку из различных символов,

используемую для доступа к нашему боту. Токен вашего бота необходимо хранить в секрете.

Теперь, когда бот создан, добавляем его в беседу, куда он будет пересылать сообщения. Для этого необходимо узнать id беседы. Сделать это можно несколькими способами, один из них — добавить в участники беседы бота LeadConverter и отправить туда же сообщение: “/get_chat_id”.

Переходим к написанию кода бота. В программе PyCharm открываем новый проект и создаём файл «main.py». Проект можно разделить на несколько файлов, содержащих главное тело кода, функции и переменные, но программа будет компактной, и находиться целиком в “main.py”.

В написании кода будем использовать библиотеку pyTelegramBotAPI: import telebot.

Далее создаём глобальные переменные, содержащие токен бота и id чата, в который будут пересылаться сообщения:

```
BOT = telebot.TeleBot(<токен бота>)
```

```
CHAT_ID = <id чата>
```

Теперь можно написать главную функцию-обработчика бота, которая будет проверять полученное от пользователя сообщение и вызывать соответствующую функцию ответа с использованием специального декоратора message_handler из ранее установленной библиотеки:

```
@BOT.message_handler()
def get_text_messages(message):
    if message.text.upper() == "ПРИВЕТ" or message.text == "/start":
        greeting(message)
    elif message.text == "/help":
        bot_help(message)
    elif message.text.split()[0] == "problem:":
        forward(message)
    else:
        exception(message)
```

Далее реализуем все вышеуказанные функции:

“greeting” — функция приветствия, вызывается при получении слова «привет» или стандартной команды “/start”;

“bot_help” — стандартная функция справки, вызываемая по команде “/help”;

“forward” — основная функция нашего бота, которая пересылает сообщение в отдельный чат, если оно начинается со слова «problem: »;

“exception” — функция исключения, вызываемая, если не были вызваны остальные функции.

```
def forward(message):
    BOT.forward_message(CHAT_ID, message.chat.id, message.message_id)
    BOT.send_message(message.from_user.id, f'Сообщение отправлено: «{message.text}»')
def greeting(message):
```

BOT.send_message(message.from_user.id, «Привет, я бот общежития. Если у тебя есть замечания к работе управления, твоим соседям или самому общежитию, напиши: “problem: <сообщение>”. Твоё сообщение увидят только старосты”).

```
def bot_help(message):
```

BOT.send_message(message.from_user.id, «Если хочешь отправить сообщение о проблеме, напиши: «problem: <сообщение>»»)

```
def exception(message):
```

```
BOT.send_message(message.from_user.id, «Я тебя не понимаю. Напиши /help»)
```

Основная часть программы завершена, осталось вызвать функцию «infinity_polling», которая позволит боту получать обновления и запускать собственный код неограниченное количество раз: BOT.infinity_polling()

Наш бот готов. Его можно включить, запустив файл «main.py», но, когда выполнение файла прекратиться, бот остановится. Решить эту проблему предлагаем запуском главного файла на сервере. Для этого мы будем использовать сервис “PythonAnywhere”. Этот сервис предназначен для работы с Python-проектами на сервере.

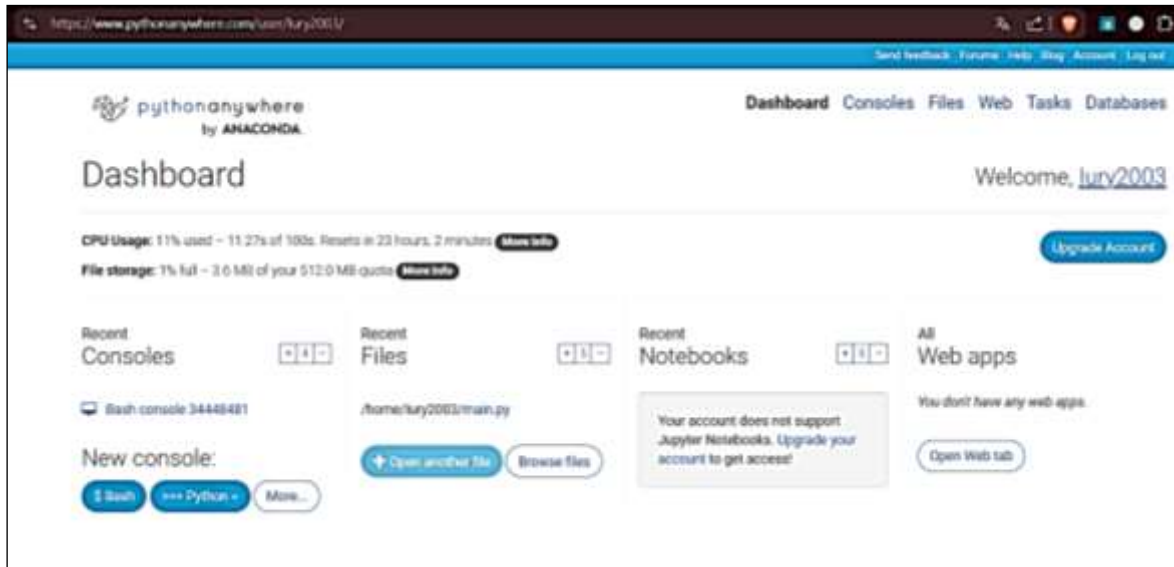


Рисунок 1. Главная страница сервиса PythonAnywhere

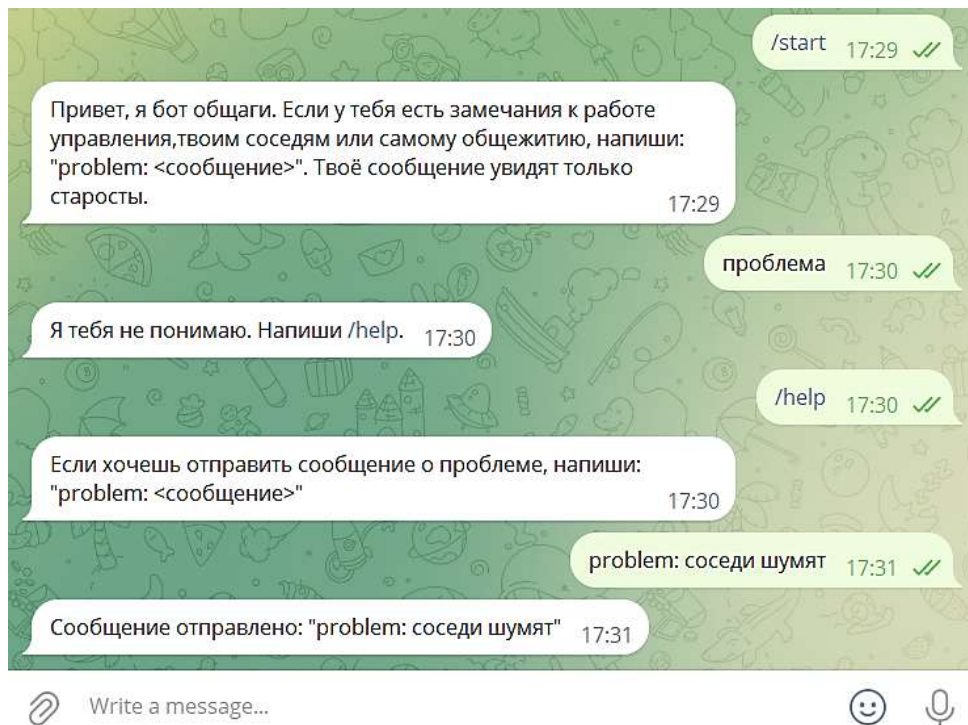


Рисунок 2. Тест бота

После простой регистрации на сервисе необходимо выгрузить наш файл: с главной страницы переходим в меню “Files”, нажимаем на кнопку “Upload a file” и выбираем наш файл. После этого возвращаемся на главную страницу и нажимаем кнопку «\$ Bash». Открывается консоль, в которой мы вводим команду для загрузки используемой в нашем боте библиотеки: `pip3 install --user telebot`.

Когда библиотека загружена на сервер, файл уже готов к работе. Для его запуска необходимо ввести команду: `python main.py`.

Теперь наш бот работает на сервере и не зависит от того, запущен он на компьютере или нет. Однако так как мы используем бесплатную версию сервера, мы не можем добавить задачу автоматического запуска файла после перезагрузки сервера или другого непредвиденного сбоя. Тем не менее, на сервере PythonAnywhere бот может работать достаточно долго, а при отключении надо будет повторить последнюю команду в консоли.

Таким образом, разработан, запущен и протестирован простой Telegram-бот, имеющий ответы на несколько базовых команд и реализующий основную функцию – пересылку сообщений в специальный чат самоуправления общежития. Внедрение этого бота может упростить работу студенческого самоуправления и наладить взаимодействие со студентами. Существует также возможность последующего расширения функционала чат-бота.

Источники:

2. TelegramBotAPI. <https://lxl.su/dzUr>
3. Python. <https://www.python.org/doc>
4. PythonAnywhere. <https://www.pythonanywhere.com>.

Список литературы:

1. Джанарсанам С. Разработка чат-ботов и разговорных интерфейсов. М.: ДМК Пресс, 2019. 340 с.

References:

1. Dzhanarsanam, S. (2019). *Razrabotka chat-botov i razgovornykh interfeisov*. Moscow. (in Russian).

*Работа поступила
в редакцию 07.11.2024 г.*

*Принята к публикации
12.11.2024 г.*

Ссылка для цитирования:

Сафина Г. Ф., Коняев Ю. С. Разработка и запуск telegram-бота – агрегатора сообщений // Бюллетень науки и практики. 2024. Т. 10. №12. С. 151-154. <https://doi.org/10.33619/2414-2948/109/20>

Cite as (APA):

Safina, G., & Konyaev, Yu. (2024). Development and Launch of the Telegram Bot – Message Aggregator. *Bulletin of Science and Practice*, 10(12), 151-154. (in Russian). <https://doi.org/10.33619/2414-2948/109/20>