

СОЗДАНИЕ TELEGRAM-БОТА НА PYTHON

Зарубин В.А.¹, Семченко О.П.²

¹Томский политехнический университет, студент гр. 8К13,

²Томский политехнический университет, студент гр. 8К13, e-mail: ops5@tpu.ru

Научный руководитель: А.С. Фадеев, доцент ОИТ ИШИТР ТПУ

Введение

Расписание – это неотъемлемая часть жизни студента. Быстрый и удобный доступ к нему – это то, что необходимо каждому учащемуся. Такой доступ может обеспечить Telegram. Telegram – это не только самый популярный мессенджер в мире [1], но и целая система, в которой можно обмениваться любыми видами информацией. Одним из способов взаимодействия в Telegram являются так называемые "боты".

Боты – это специальные программы, выполняющие какие-либо заданные действия, предназначенные для пользователя. Боты передают информацию пользователю и получают её от него. Благодаря открытому API Telegram, любой пользователь может создать своего бота, который будет какие-либо действия. Какие именно, зависит от навыков и креативности программиста. При помощи "ботов" Telegram может объединить в себе функции множества различных сервисов и стать платформой для целой экосистемы.

Целью работы является создание Telegram-бота, работающего с расписанием студентов Томского Политехнического Университета.

Процесс разработки бота

Изначально, при помощи *API Telegram* была создана своеобразная заготовка бота. Далее, при помощи языка *Python* и непосредственно библиотеки *aiogram*, был создан файл *create_bot*, который является чем-то вроде телом бота на сервере.

Следующим и, по сути, главным звеном в этой цепи является файл «*main*». Это исполняющий файл, который при запуске связывается со всеми остальными файлами, отвечающими за какую-либо функцию. Например, в папке «*keyboards*» лежат файлы, отвечающие за клавиатуру бота, а в папке «*handlers*» все обработчики. Такое разделение позволяет сделать работу бота чуть более быстрой и правильной, с точки зрения разработки.

Telegram бот имеет определённую структуру работы. Клиент (*Telegram-бот*) отправляет HTTP запрос на сервер. Сервер обрабатывает полученный запрос и отвечает клиенту через сообщение с HTTP-ответом. Ответ содержит код, который демонстрирует, был ли запрос успешным. Тело успешного ответа на запрос GET будет содержать запрашиваемый ресурс. Данный проект работает по тому же принципу.

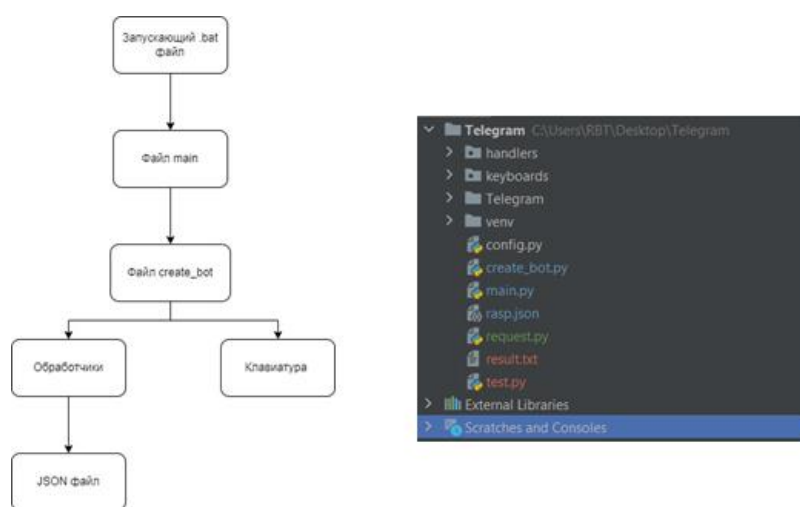


Рис. 1. Файловая структура бота (слева – в виде схемы, справа – в среде разработки)

Алгоритм работы телеграм-бота

До нажатия пользователем на кнопку, программа определяет текущую дату, а именно то, какой сегодня день недели. Это необходимо для того, чтобы взять нужную строку из JSON – файла, который получен с помощью RESTful API TPU и хранит расписание группы на текущую неделю. Данные в файле разделены по ключам занятия в хэш-таблице, которая хранит расписание на сервере.

```
"1": {
  "start": "2023-02-06 08:30:00",
  "end": "2023-02-06 10:05:00",
  "tip": null,
  "place": {
    "nomer": "002",
    "korpus": "3"
  },
  "event": null,
  "disciplina": [
    {
      "full_name": "Физика 3.2"
    }
  ],
  "gruppa": [
    {
      "title": "8К13"
    }
  ],
  "lichnost": []
},
```

Рис.2. Пример структуры JSON-файла для второго занятия на неделе

Форма взаимодействия пользователя с ботом производится посредством последовательного нажатия кнопок.

При нажатии пользователем на кнопку «/rasp» программа берет информацию по ключам «disciplina» (название дисциплины), «place» (место проведения), «gruppa» (учебная группа), «lichost» (ФИО преподавателя), «start» и «end» (начало и конец занятия). Данные из JSON-файла записываются в массивы, к элементам которых будет обращаться программа для вывода нужной информации.

Предусмотрен вывод на тот день, когда у студента отсутствуют занятия.

Работа бота

При помощи API TPU производится авторизация и делается GET запрос на сервер Томского политехнического университета. Пользователь осуществляет запрос через клавиатуру Telegram бота. Клавиатура является отдельным файлом с расширением .ру. Сервер возвращает боту тело ответа в виде JSON файла с расписанием группы на текущую неделю и бот начинает работу с ним.

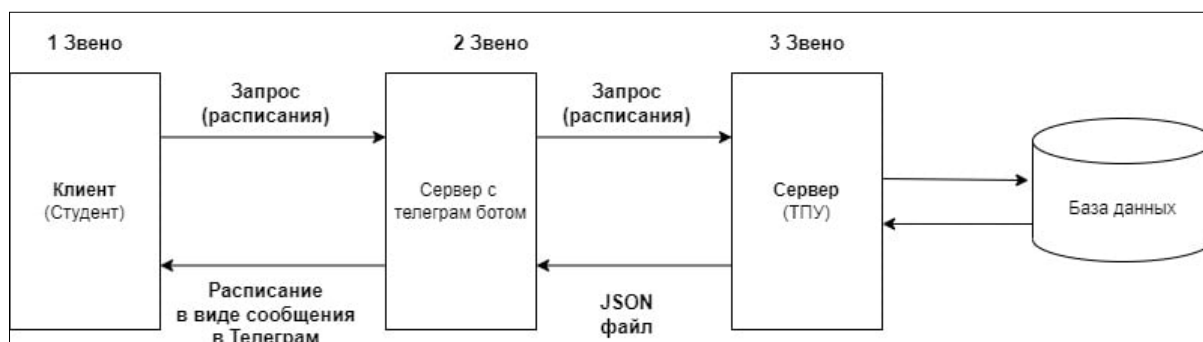


Рис. 3. Клиент-серверная архитектура

Обработка файла является самым сложным элементом в нашей цепи. Основная работа ведётся именно в этот момент. JSON файл конвертируется в словарь и из этого словаря уже берутся данные по

расписанию, которые, в свою очередь, обрабатываются ботом и выводятся студенту в сообщения от бота в *Telegram*.

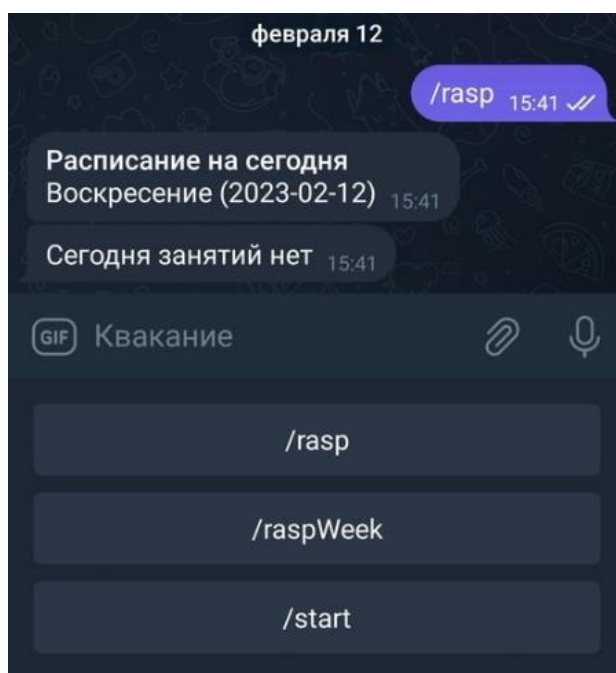


Рис. 4. Пример запроса расписания

Так как JSON файл был только с расписанием группы 8K13, то на данный момент работа с расписанием всех студентов не выполнена.

Заключение

В работе представлено выполнение работы Telegram-бота с расписанием заранее обговоренной группой студентов. JSON файл был получен при помощи RESTful API TPU. Текущее состояние бота не конечное, в будущем планируется добавление функции работы с любой группой Томского политехнического университета.

Список использованных источников

1. Самые популярные мессенджеры в России в 2022 году // Lenta.ru [Электронный ресурс]. – URL: <https://lenta.ru/news/2022/04/18/vciom/> (дата обращения 28.12.2022)
2. Telegram-бот на Python за полчаса с aiogram // tproger.ru [Электронный ресурс]. – URL: <https://tproger.ru/articles/telegram-bot-create-and-deploy/> (дата обращения 28.12.2022)
3. Как я писал telegram бот с админкой на Django// ru.hexlet.io [Электронный ресурс]. – URL: <https://ru.hexlet.io/blog/posts/kak-ya-pisal-telegram-bot-s-adminkoj-na-django> (дата обращения 28.12.2022)
4. Как я писал telegram-бота и заливал его на удаленный сервер // habr.com [Электронный ресурс]. –URL: <https://habr.com/ru/post/347106/> (дата обращения 25.01.2023)