|  |  |
| --- | --- |
|  | **МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ**  Федеральное государственное Бюджетное образовательное учреждение высшего образования  **«российский государственный аграрный университет –**  **МСха имени К.А. Тимирязева»**  **(ФГБОУ ВО ргау - МСХА имени К.А. Тимирязева)** |

**Отчёт по теме:**

**Разработка алгоритма парсинга чарта Яндекс Музыки с применением Телеграм-бота.**

Выполнил (а)

студент (ка) 3 курса 306группы Бука Артём Русланович

**Москва, 2024**

**Содержание**

[**Введение** 3](#_Toc183599974)

[**1. Парсинг страницы чарта топ 100 «Яндекс Музыки»** 4](#_Toc183599975)

[**2. Написание алгоритма вывода нужной нам информации в командную строку.** 5](#_Toc183599976)

[**3. Написание программы, которая конвертирует полученные данные в формат Excel таблицы.** 8](#_Toc183599977)

[**4. Создание Телеграмм-бота.** 10](#_Toc183599978)

[**Заключение** 13](#_Toc183599979)

[**Список литературы** 14](#_Toc183599980)

# **Введение**

В данной работе представлен проект по разработке алгоритма с применением Telegram-бота, предназначенного для предоставления пользователям актуальной информации о музыкальных чартах российского сервиса «Яндекс Музыка». Актуальность информации подкрепляется самим запросом, т.е. информация актуальна после выполнения запроса. Основная цель проекта — обеспечить доступ к списку 100 самых популярных треков на данный момент.

Выбор Telegram в качестве платформы обусловлен его широкой популярностью и удобством использования для обмена информацией. Разработка бота позволяет предоставить пользователям быстрый и простой способ получения информации о музыкальных трендах без необходимости посещать веб-сайт «Яндекс Музыки».

Проект направлен на решение задачи предоставления пользователям актуальных данных о музыкальных предпочтениях, что может быть полезно широкому кругу людей, интересующихся современной музыкой. В рамках работы будет рассмотрена архитектура бота, алгоритмы взаимодействия с API (или веб-скрапинга, если API отсутствует) «Яндекс Музыки» (или альтернативного метода получения данных, если API недоступен или ограничен) и методы обработки и представления данных пользователям. Результатом проекта станет функциональный Telegram-бот, предоставляющий актуальные данные о 100 лучших треках «Яндекс Музыки».

Проект будет разделён на 3 ключевые части:

1. Парсинг страницы чарта топ 100 «Яндекс Музыки».

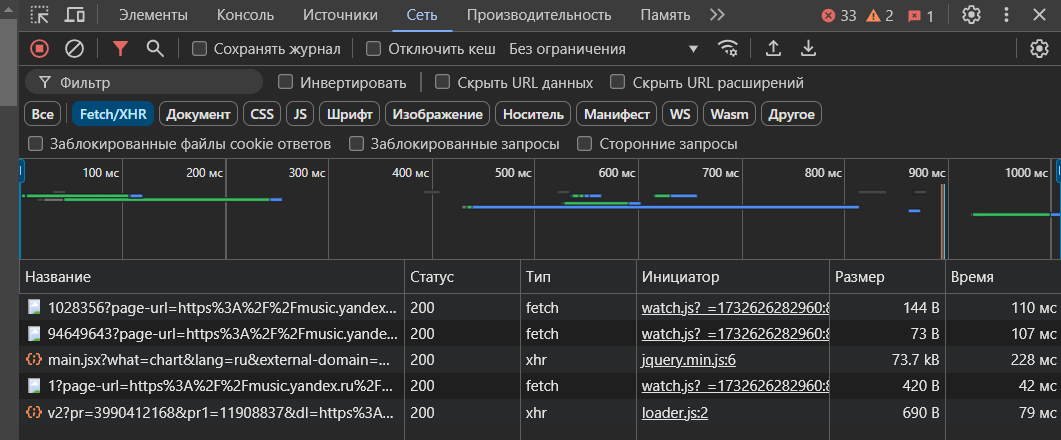
2. Написание алгоритма вывода нужной нам информации в командную строку.

3. Написание программы, которая конвертирует полученные данные в формат Excel таблицы.

4. Создание Телеграмм-бота.

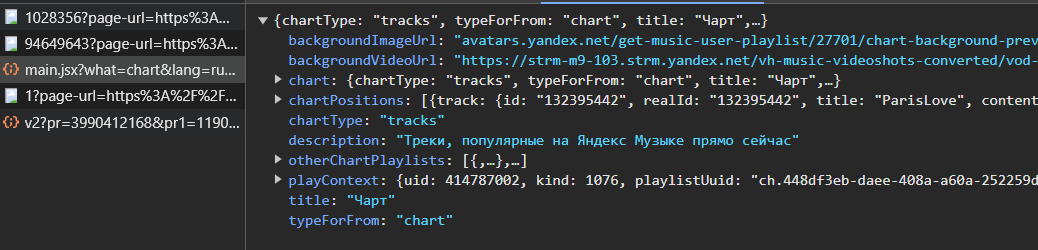
# **1. Парсинг страницы чарта топ 100 «Яндекс Музыки»**

Для парсинга воспользуемся инструментом разработчика в самом чарте. Для этого воспользуемся вкладкой «сеть», чтобы просмотреть все запросы при нажатии на чарт. Нам выдаёт данные запросы :



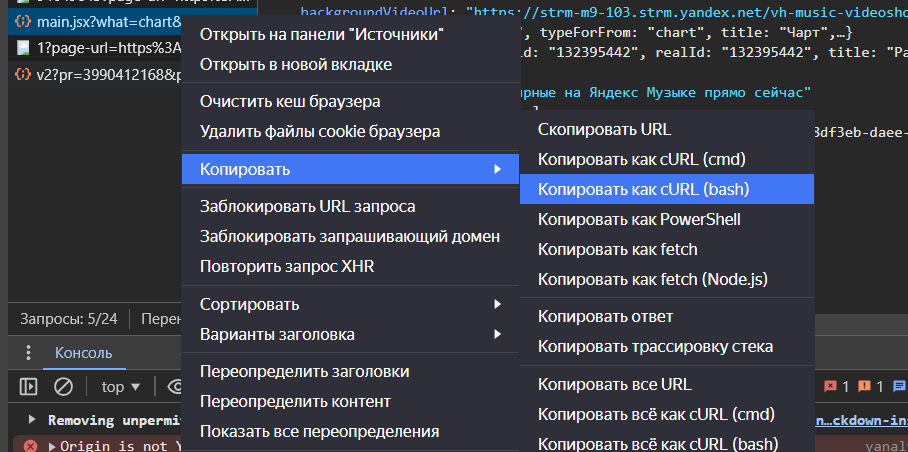
**Рисунок 1 – Запросы Яндекс музыки**

Далее переходим в запрос main.jsx и в нем ищем нужную для нас информацию:



**Рисунок 2 – Запрос main.jsx**

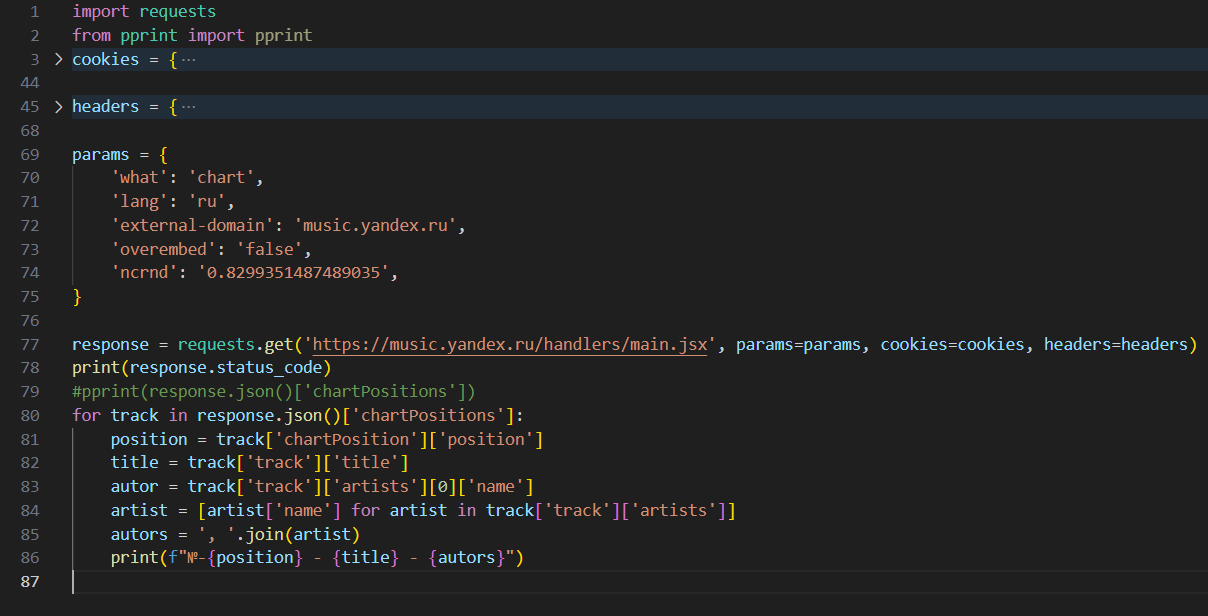
Мы видим вкладку chartPosition, переходя в которую открывается нужная нам информация с названием треков, их позициями и исполнителями. Копируем данный запрос как сURL (bash) и отправляем в конвентор для экономии нашего времени с целью получения нужных нам cookies.



**Рисунок 3 – Копирование запроса для конвентора**

После успешного получения данных переходим ко второму этапу.

# **2. Написание алгоритма вывода нужной нам информации в командную строку.**



**Рисунок 4 – Код парсинга для командной строки**

1. Цель программы:

Программа предназначена для извлечения и отображения данных о текущем музыкальном чарте Top 100 сервиса Яндекс Музыка. Она считывает данные с веб-сайта Яндекс Музыки, обрабатывает полученную информацию и выводит её в удобном для чтения формате.

2. Метод работы:

Программа использует библиотеку requests для отправки HTTP-запроса на сервер Яндекс Музыки. Запрос формируется с использованием параметров (params), файлов cookie (cookies) и заголовков (headers), которые необходимы для аутентификации и корректного получения данных (подробности о параметрах не указаны, предполагается наличие соответствующей настройки).

Полученный ответ (в формате JSON) анализируется программой. Цикл for перебирает список chartPositions, извлекая из каждого элемента следующие данные:

position: Номер позиции трека в чарте.

title: Название трека.

artist: Имена исполнителей (обработка случая, когда у трека несколько исполнителей).

Обработанные данные выводятся на консоль в формате: №-позиция - название трека - исполнитель(и).

Структура кода:

Код программы состоит из следующих частей:

**Запрос данных:** использование requests.get() для получения JSON-ответа от сервера Яндекс Музыки.

**Проверка статуса:** вывод кода статуса ответа (response.status\_code) для контроля успешности запроса.

**Обработка JSON:** Извлечение списка chartPositions из JSON-ответа.

**Итерация и извлечение данных:** цикл for для обработки каждого элемента списка и извлечения необходимых данных.

**Обработка исполнителей:** Специальная обработка для корректного отображения нескольких исполнителей трека.

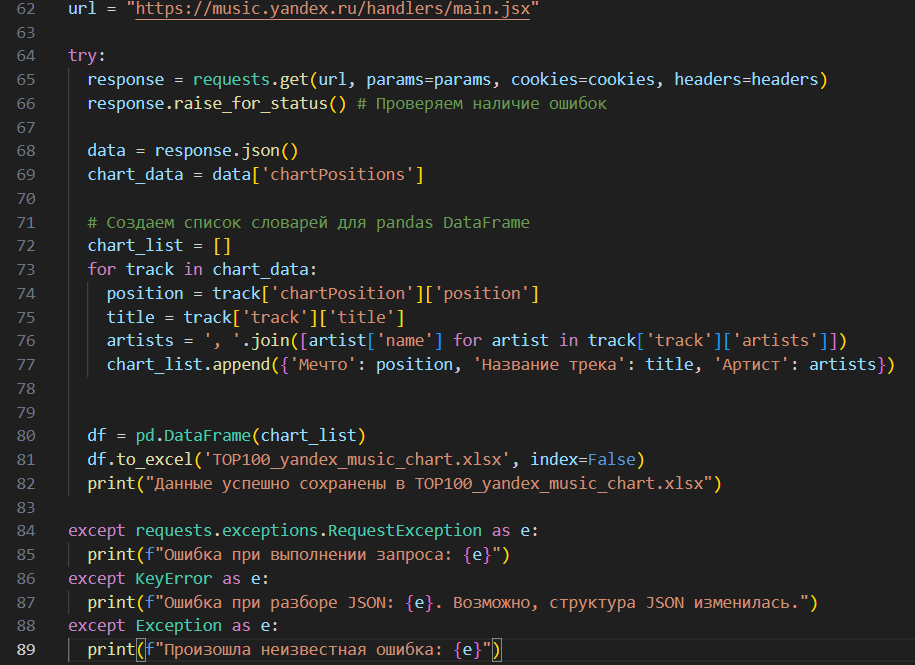
**Вывод данных:** Вывод информации на консоль в формате №-позиция - название трека - исполнитель(и).

Вывод данных:



**Рисунок 5 – Вывод кода**

# **3. Написание программы, которая конвертирует полученные данные в формат Excel таблицы.**



**Рисунок 6 – Код парсинга с переносом в Excel файл**

Принцип работы программы тот же, однако в данной интерпретации код позволяет конвертировать результат в Excel таблицу.

Структура кода:

Программа имеет структуру try...except блоков, что позволяет обеспечить надёжную работу при возможных ошибках. Основные этапы:

HTTP-запрос: Получение данных с использованием requests.get().

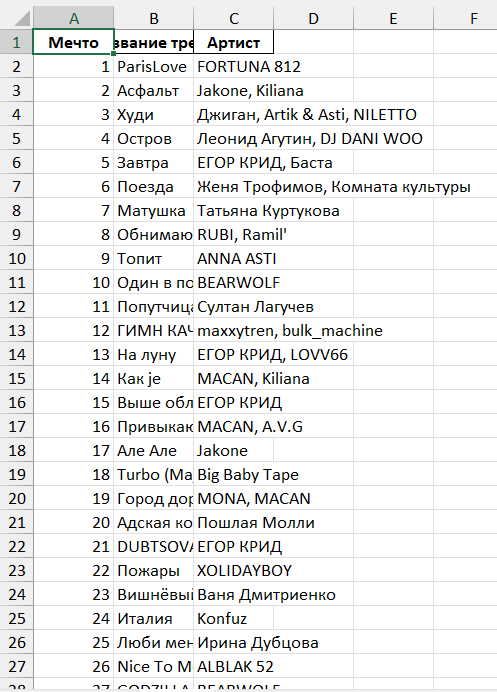
Проверка статуса запроса: response.raise\_for\_status() проверяет наличие ошибок на стороне сервера.

Парсинг JSON: Извлечение необходимых данных из JSON-ответа.

Формирование DataFrame: Создание pandas DataFrame из обработанных данных.

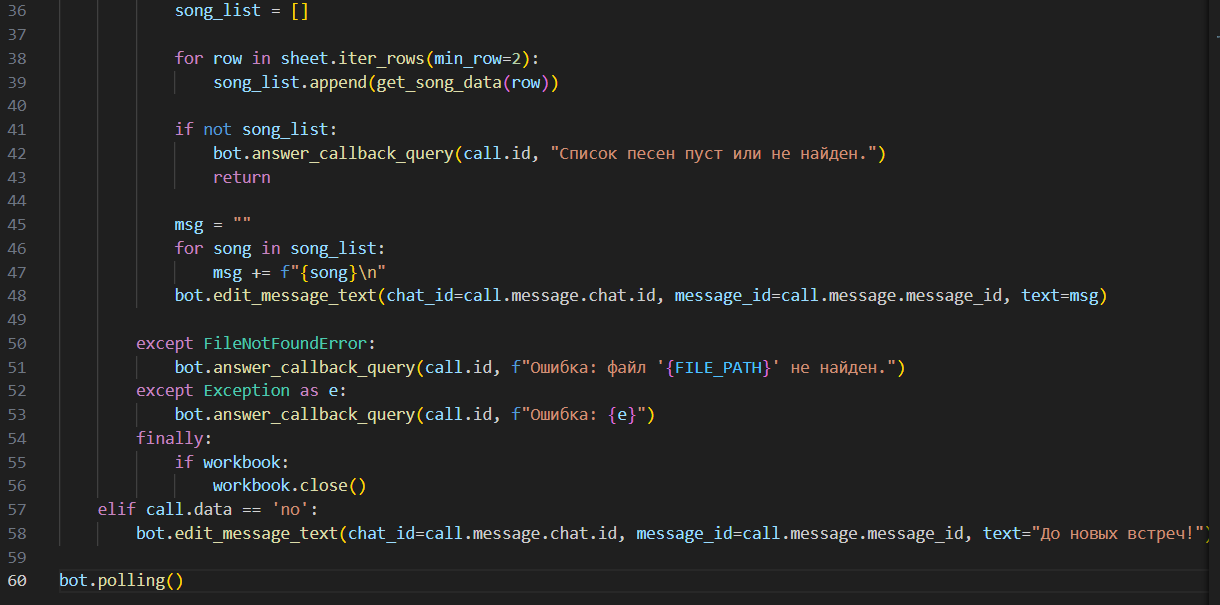
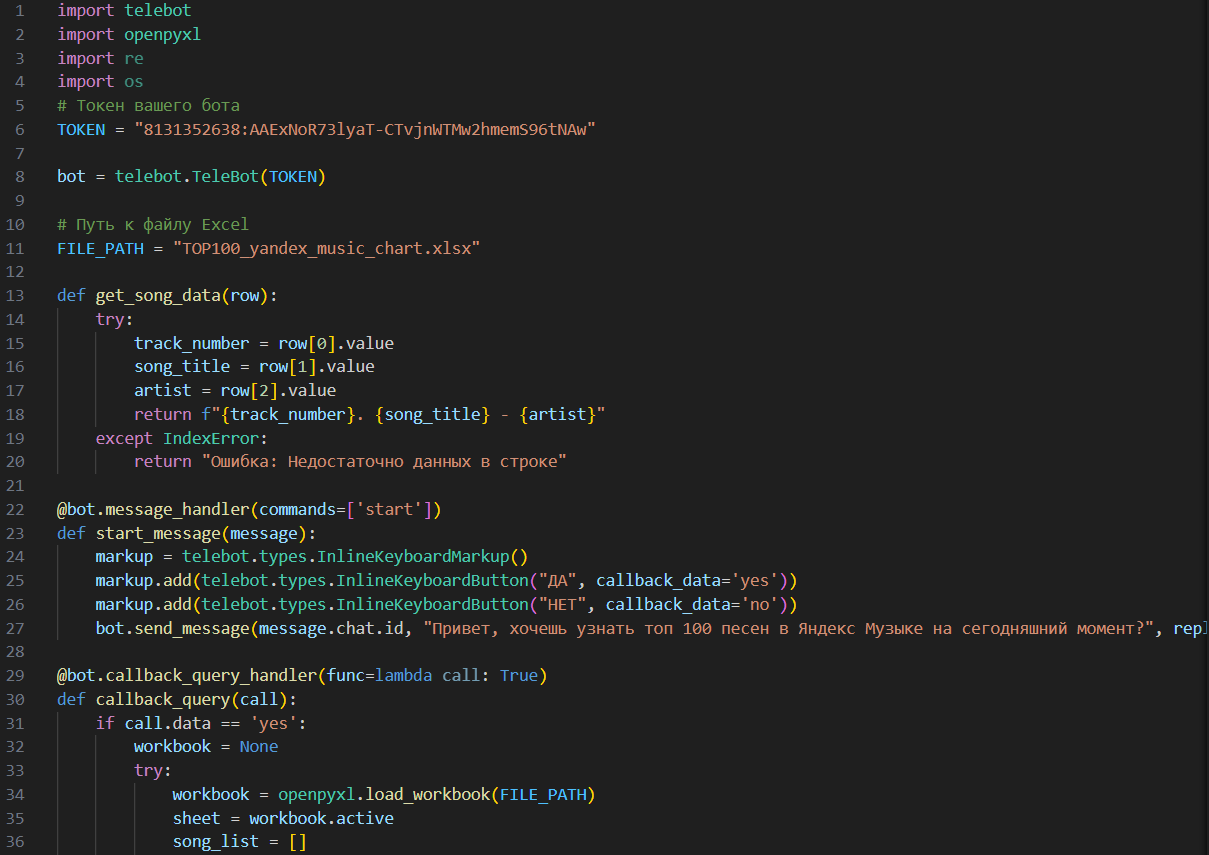
Сохранение в Excel: запись DataFrame в файл Excel с использованием to\_excel().

Обработка исключений: Обработка ошибок на каждом этапе работы программы.



**Рисунок 7 – Excel файл с полученными данными**

# **4. Создание Телеграмм-бота.**



**Рисунок 8 – Создание тг-бота**

Программа реализует Telegram-бота, предоставляющего пользователям информацию о музыкальном чарте Top 100 Яндекс Музыки. Данные чарта предварительно должны быть сохранены в файле Excel (TOP100\_yandex\_music\_chart.xlsx).

Функциональность:

Бот предлагает пользователю просмотреть чарт после отправки команды /start. Взаимодействие осуществляется с помощью встроенной клавиатуры с вариантами «ДА» и «НЕТ».

**“ДА”:** Бот выводит содержимое файла Excel, содержащего чарт (номер трека, название и исполнителя), в формате списка.

**“НЕТ”:** Бот отвечает сообщением “До новых встреч!”

Архитектура:

Программа написана на Python и использует библиотеки:

telebot: Для взаимодействия с Telegram API.

openpyxl: Для чтения данных из файла Excel.

Программа состоит из:

Обработчик команды /start (start\_message): создает и отправляет сообщение с помощью встроенной клавиатуры.

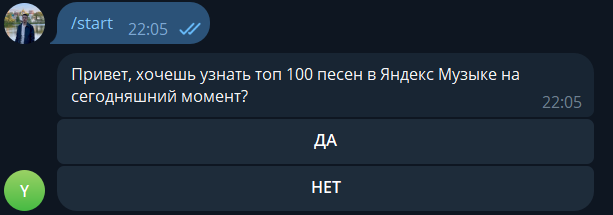
Функции get\_song\_data: Извлекает данные из строки файла Excel.

Обработчик callback-запросов (callback\_query): обрабатывает нажатия кнопок «ДА» и «НЕТ», выводя диаграмму или прощальное сообщение.

Обработчик ошибок: обрабатывает FileNotFoundError (файл не найден) и другие исключения.

Тестирование:

Программа успешно протестирована на соответствие описанной функциональности. Бот корректно реагирует на команду /start, выводит встроенную клавиатуру и обрабатывает выбор пользователя. Обработка ошибок также функционирует корректно.



**Рисунок 9 – Меню тг-бота**



**Рисунок 10 – Вывод тг-бота**

# **Заключение**

В заключение отметим, что данная работа посвящена разработке Telegram-бота, предоставляющего пользователям удобный и быстрый доступ к актуальному чарту Top 100 треков Яндекс Музыки. Проект предполагает решение трёх ключевых задач: парсинг данных с веб-сайта (или через API, если он доступен), обработку и вывод информации в консоль, а также преобразование данных в формат Excel. Объединение этих компонентов позволит создать функционального Telegram-бота, удовлетворяющего потребность пользователей в оперативном получении информации о популярных музыкальных треках.

# **Список литературы**

1. Requests в Python – Примеры выполнения HTTP запросов / [Электронный ресурс] // Python 3 : [сайт]. — URL: https://python-scripts.com/requests (дата обращения: 27.11.2024).
2. Requests: HTTP for Humans™ / [Электронный ресурс] // Requests : [сайт]. — URL: https://requests.readthedocs.io/en/latest/index.html (дата обращения: 27.11.2024).
3. Requests: HTTP for Humans™ / [Электронный ресурс] // python-telegram : [сайт]. — URL: https://docs.python-telegram-bot.org/en/stable/index.html (дата обращения: 27.11.2024).
4. Чарт / [Электронный ресурс] // Яндекс Музыка : [сайт]. — URL: https://music.yandex.ru/chart (дата обращения: 27.11.2024).