**Пояснительная записка к приложению «Музыкальный бот»**

**Цель проекта:** создать telegram-бота, который способен взаимодействовать с API Яндекс.Музыки и осуществлять поиск и скачивание треков и плейлистов

**Исполнитель:** Ятченко Кирилл Вячеславович, Русанов Евгений Васильевич

**Используемые библиотеки:**

yandex\_music==2.0.1

python-telegram-bot==20.2

sqlalchemy==1.4.27

requests==2.28.1

Все функции, отвечающие за взаимодействие бота с API, вынесены в отдельный файл music\_functions\_async.py

Функции в music\_functions\_async.py:

* authorize() – авторизация бота в клиенте API для осуществления запросов
* search() – осуществляет поиск по данному вопросу и возвращает результат
* process\_search() – обрабатывает результат поиска треков и приводит их в понятный пользователю вид
* get\_name\_for\_file() – возвращает имя файла, которое будет использовано, когда данный трек будет скачиваться
* get\_track\_name() – возвращает название трека в понятном пользователю виде
* get\_playlist\_name() – возвращает название плейлиста в понятном пользователю виде
* download() – скачивает данный обьект для дальнейшей передачи пользователю
* get\_user\_playlists() – по логину пользователя возвращает его плейлисты
* process\_user\_playlist\_search() – обрабатывает результат функции get\_user\_playlists() и приводит их в понятный вид
* search\_random\_track() – находит случайные 10 треков
* process\_search\_random\_track() – обрабатывает результат функции search\_random\_track() и приводит их в понятный вид
* get\_user\_yandex\_login() – по id чата возвращает логин данного пользователя в Яндекс.Музыке
* save\_login() – сохраняет логин пользователя из Яндекс.Музыки в базу данных

Функции и команды в основном файле Telegram bot.py:

1. Команда /search\_user\_playlists, функции диалога поиска плейлистов:
   1. start\_dialog\_search\_playlists() – обработчик начала диалога
   2. search\_users\_playlists() – обработчик, выполняющий сам поиск плейлиста
   3. ask\_for\_playlist\_download() – обработчик, задающий вопрос пользователю о скачивании плейлиста
   4. download\_playlist() – обработчик, отвечающий за скачивание плейлиста
2. Команда /playlists, функции диалога быстрого поиска плейлистов (не требуется вводить логин пользователя, сразу выводит плейлисты того пользователя, логин которого сохранен в базе данных с этим id чата):
   1. start\_dialog\_fast\_search\_playlists() – обработчик начала диалога, отправляет список плейлистов
   2. ask\_for\_playlist\_download() – аналогично пункту 1.c.
   3. download\_playlist() – аналогично пункту 1.d.
3. Команда /search\_track, функции диалога поиска треков:
   1. start\_dialog\_search\_track() – обработчик начала диалога
   2. search\_track() – обработчик, выполняющий поиск
   3. ask\_for\_track\_download() – обработчик, спрашивающий пользователя про скачивание трека
   4. download\_track() – обработчик, отвечающий за скачивание трека
4. Команда /random\_track, функции диалога поиска случайного трека
   1. start\_dialog\_search\_random\_track() – обработчик начала диалога
   2. ask\_for\_track\_download() – аналогично пункту 2.c.
   3. download\_track() – аналогично пункту 2.d.
5. Команда /register, функции диалога регистрации:
   1. register() – обработчик начала диалога
   2. get\_user\_login() – обработчик, сохраняющий логин пользователя в базу данных
6. Команда /subscription, функции диалога оформления подписки:
   1. start\_dialog\_making\_subscription() – обработчик начала диалога
   2. adding\_account() – обработчик, добавляющий введенный логин в базу данных
7. Команда /help – печатает руководство по использованию бота, обработчик – функция help\_info()
8. Команда /account\_info – позволяет узнать логин, сохраненный в базе данных с id чата пользователя, обработчик – функция get\_my\_info()
9. Функция stop() – отвечает за выход из диалогов