Краткая документация к проекту VK_test_friends (с примерами запуска сервиса и вызова арі)

ФУНКЦИОНАЛ:

Приложение используется для создания статуса дружбы между пользователями. Имеются следующие возможности

- регистрация пользователя
- отправка заявки на добавление в друзья
- прием/отклонение заявок
- получать пользователю статус дружбы с другим пользователей или наличие заявок
- удалить пользователя из друзей
- автоматическое добавление в друзья при наличии встречных заявок

Используемые сущности:

- Пользователи (User) стандартная модель фреймворка Django
- Заявки (FriendRequest) хранит заявки в друзья и их статус. Заявка содержит информацию об отправителе и получателе, статус, а также дату создания.
- -Друзья (Friend) хранит связку пользователей, которая определяет их как друзей.

ЗАПУСК СЕРВИСА.

Скачайте репозиторий. Например:

git clone https://github.com/KIchkinevVladislav/VK_test.git

Вариант 1

Установите зависимости из requirements.txt:

pip install -r requirements.txt

Выполните все необходимые миграции:

python manage.py makemigrations

python manage.py migrate

Для доступа к панели администратора создайте администратора:

python manage.py createsuperuser

Запустите приложение:

python manage.py runserver

Вариант 2

Путем запуска контейнера docker на основе файлов docker и docker-compose в проекте.

Сервис будет доступен на http://localhost:8000/.

Тестирование можно производить либо через графический интерфейс Django REST Framework API, либо посредством Postman. Для удобства примеры будут представлены с использование swagger-панели автособираемой документации по адресу: http://localhost:8000/api/schema/swagger-ui/.

Доступные конечные точки

POST http://localhost:8000/api/register/ - регистрация нового пользователя

POST http://localhost:8000/api/api-auth/login/ - стандартный эндпоит DRF для входа в аккаунт

POST http://localhost:8000/api/api-auth/logout/ - - стандартный эндпоит DRF для выхода из аккаунта

POST http://localhost:8000/api/friend-request/ - отправить заявку в друзья

POST http://localhost:8000/api/friend-request/accept/ - одобрить заявку в друзья

POST http://localhost:8000/api/friend-request/decline/ - отклонить заявку в друзья

GET http://localhost:8000/api/friend-requests/ - посмотреть список входящих и исходящих заявок

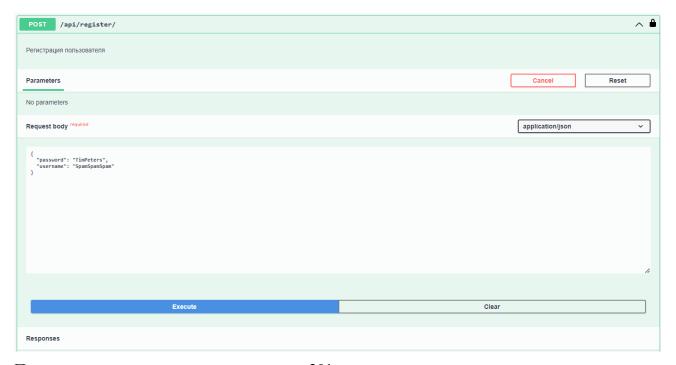
GET http://localhost:8000/api/friends/ - посмотреть список друзей

GET http://localhost:8000/api/friend-status/<int:user_id>/ - проверить наличие заявок от пользователя или его статус как друга

DELETE http://localhost:8000/ remove-friend/<int:user_id>/ - удалить пользователя из друзей

http://localhost:8000/api/register/ - регистрация нового пользователя

Достаточно передать JSON-объект формата: {"username": "test", "password": "testpassword"}



При корректных данных ответ со статусов 201 и данные пользователя.



При некоренных данных возвращает ответ со статусом 400 и замечаниями по содержанию полей



Стандартные эндпоинты для авторизации пользователей

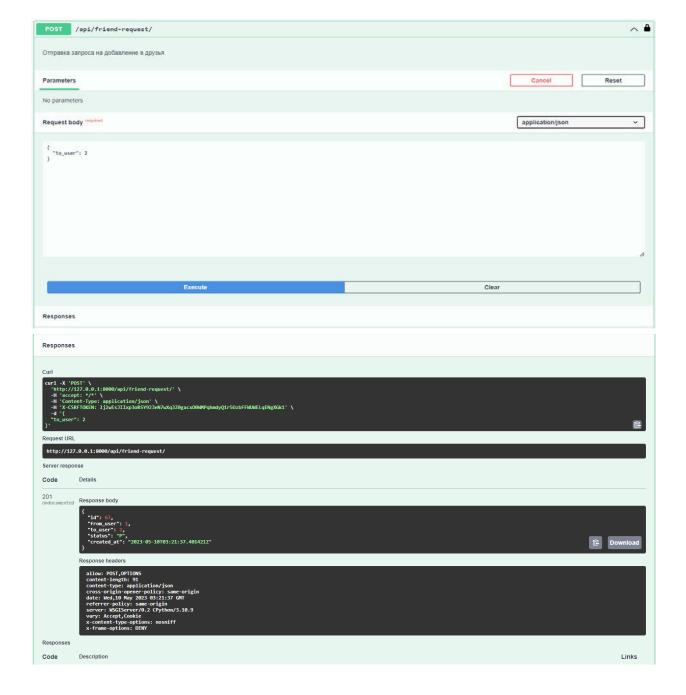
http://localhost:8000/api/api-auth/login/ - вход в аккаунт

http://localhost:8000/api/api-auth/logout/ - выход из аккаунта

Все остальные функции требуют авторизации пользователя, иначе возвращает ошибку со статусом 403 и телом ("detail": "Учетные данные не были предоставлены")

http://localhost:8000/api/friend-request/ - отправить заявку в друзья

Отправляется JSON-объект формата: {"to_user": id}, где id - целое число, идентификатор пользователя, к которому направляется запрос.



Если такого пользователя не существует, приложение сообщит об этом.



Если Вы отправляете запрос сами, приложение сообщит об этом.



Если Вы ранее направляли запрос, и он не рассмотрен, приложение сообщит об этом.

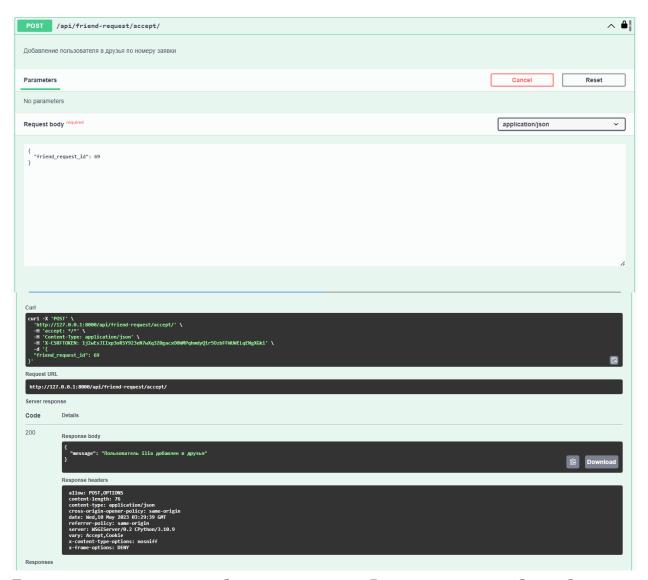


Если имеется встречная заявка от пользователя, вы автоматически станете друзьями



http://localhost:8000/api/friend-request/accept/ - одобрить заявку в друзья

Отправляется JSON-объект формата: {"friend request id": 1}, где число - номер заявки



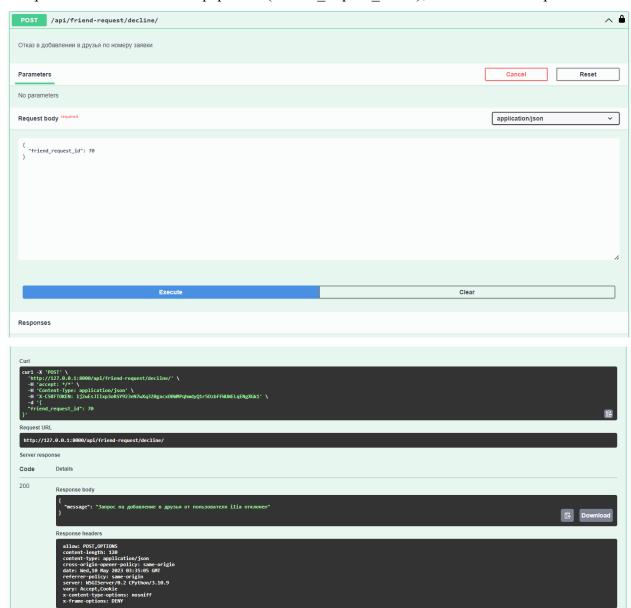
Если заявки не существует, либо она направлена не Вам, приложение сообщит об этом.



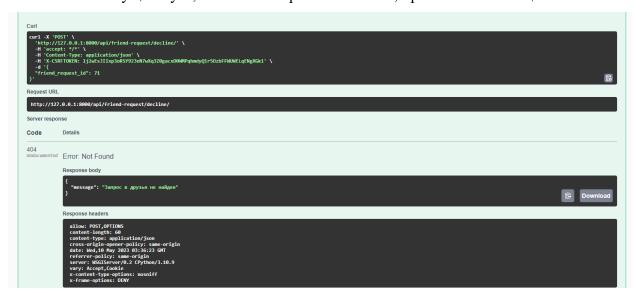
При одобрении заявка удаляется из базы данных.

http://localhost:8000/api/friend-request/decline/ - отклонить заявку в друзья

Отправляется JSON-объект формата: {"friend_request_id": 1}, где число - номер заявки

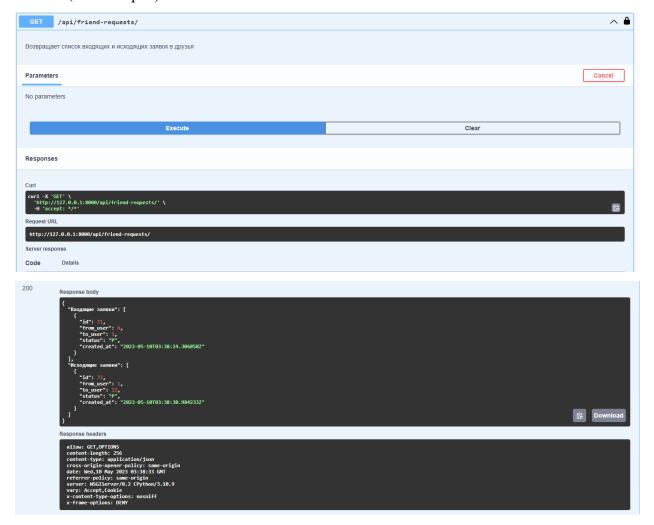


Если заявки не существует, либо она направлена не Вам, приложение сообщит об этом.



При отказе заявка удаляется из базы данных.

http://localhost:8000/api/friend-requests/ - посмотреть список входящих и исходящих заявок. (GET-запрос)

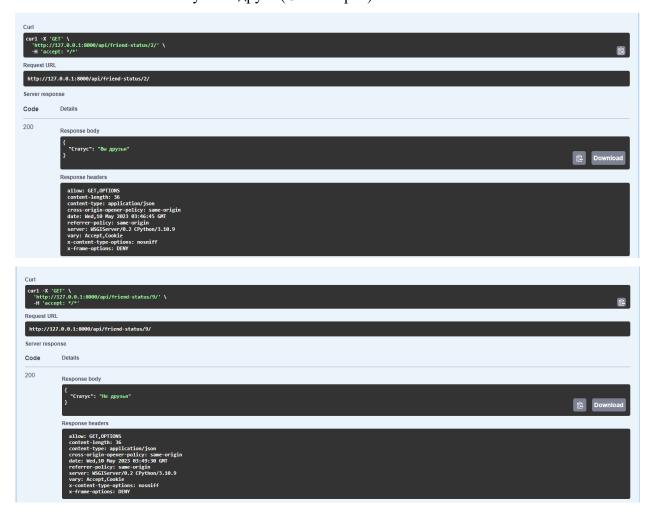


http://localhost:8000/api/friends/ - посмотреть список друзей. (GET-запрос)



Иначе возвращает: "Статус": "Нет друзей".

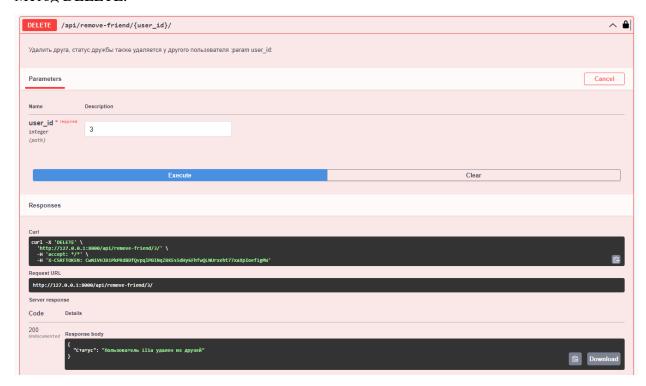
http://localhost:8000/api/friend-status/<int:user_id>/ - проверить наличие заявок от пользователя или его статус как друга (GET-запрос)



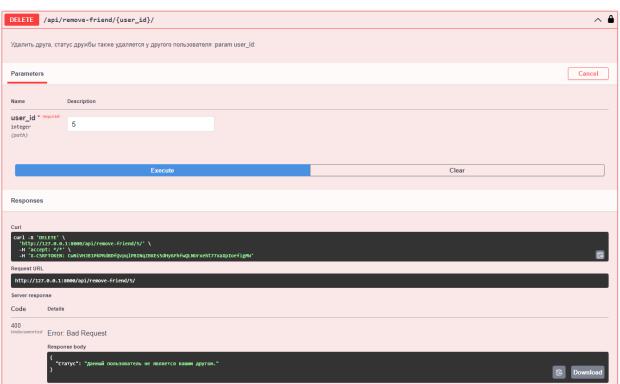
Приложение проверяет, не указываете ли Вы в запросе свой идентификатор и существует ли пользователь с таким идентификатором



http://localhost:8000/remove-friend/<int:user_id>/ - удалить пользователя из друзей Метод DELETE.



Если пользователь не является другом, приложение сообщит об этом.



Приложение проверит, что Вы указали не свой id и что, такой пользователь существует.



дополнительно:

http://localhost:8000/admin - доступ к админ-панели Django

http://127.0.0.1:8000/api/schema/swagger-ui/ - swagger-панель автособираемой

документации