# 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

#### Запуск приложения.

Скачайте репозиторий. Например:

git clone https://github.com/KIchkinevVladislav/VK\_test.git

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

#### Использование АРІ

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

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

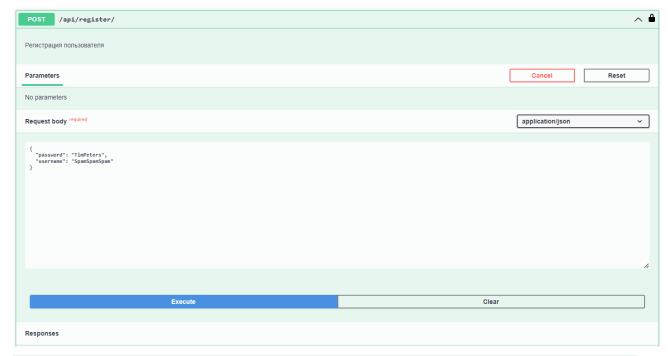
GET http://localhost:8000/api/friends/ - посмотреть список друзей

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"}



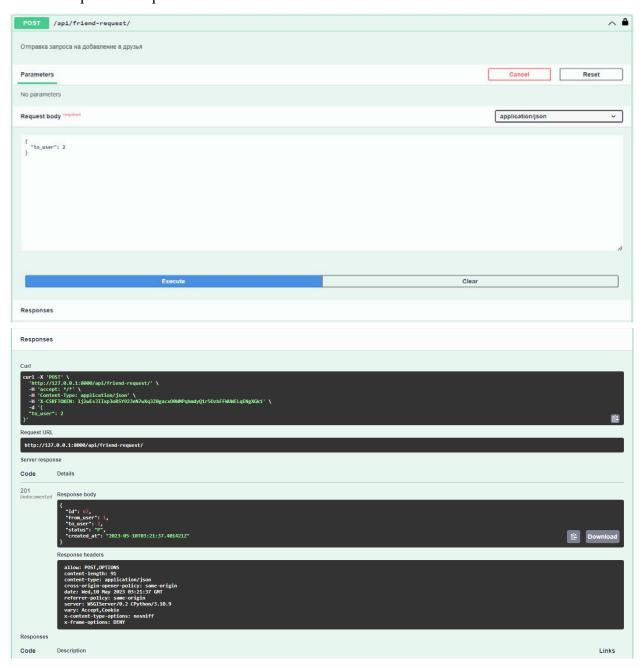


Если имеется встречная заявка от пользователя, вы автоматически станете друзьями



#### 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. Не аутентифицированный пользователь не может отправить запрос.



Если такого пользователя не существует, приложение сообщит об этом.



Если Вы отправляете запрос сами, приложение сообщит об этом.

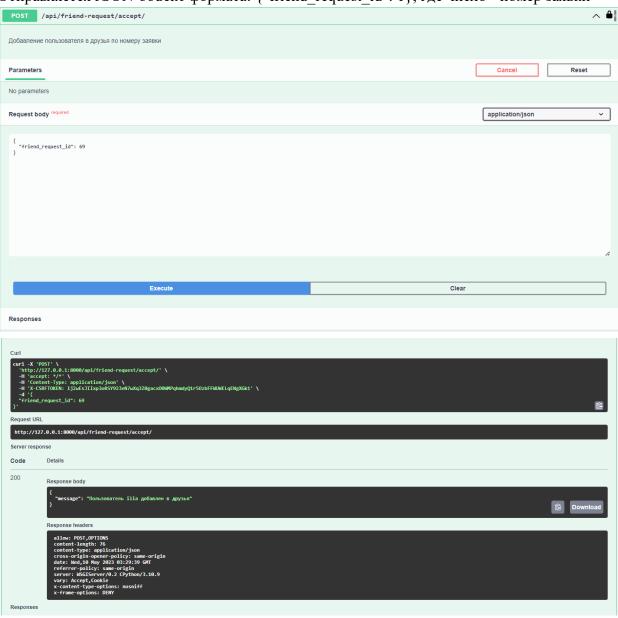


Если Вы ранее направляли запрос, и он не рассмотрен, приложение сообщит об этом.

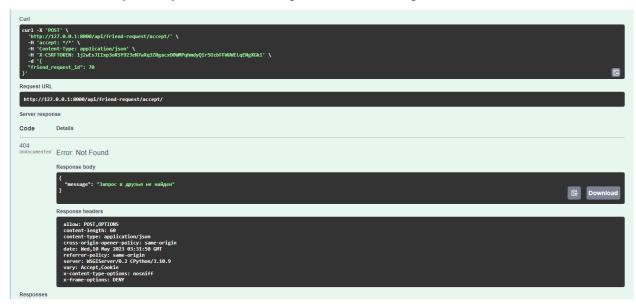


## http://localhost:8000/api/friend-request/accept/ - одобрить заявку в друзья

Отправляется JSON-объект формата: {"friend\_request\_id": 1}, где число - номер заявки



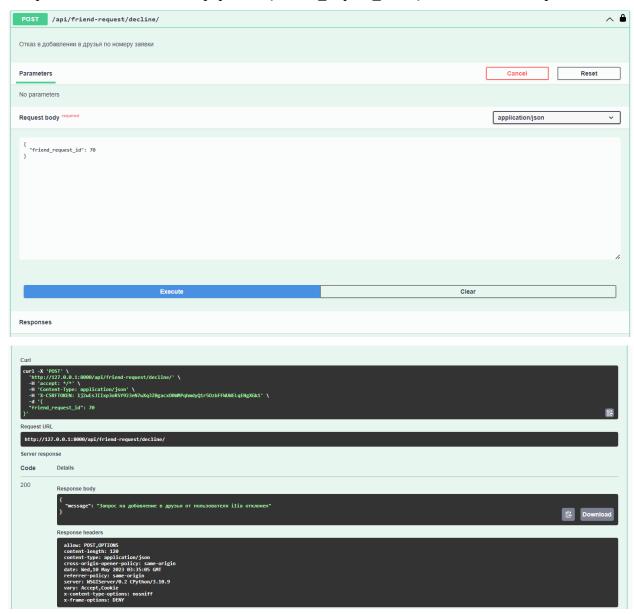
Если заявки не существует, либо она направлена не Вам, приложение сообщит об этом.



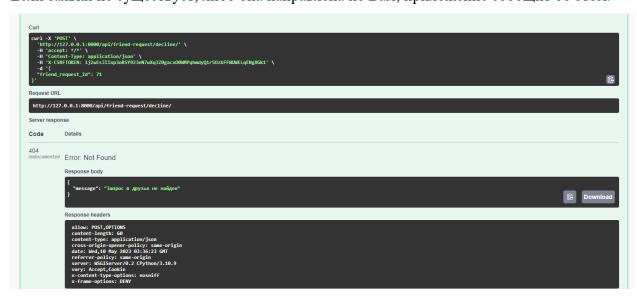
При одобрении заявка удаляется из базы данных.

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

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

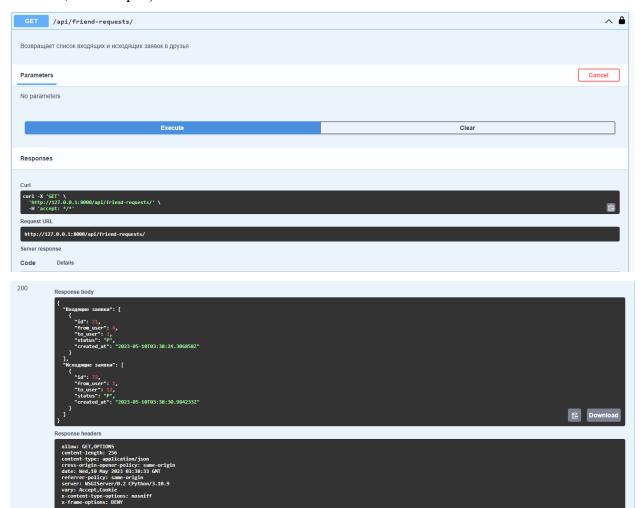


Если заявки не существует, либо она направлена не Вам, приложение сообщит об этом.



При отказе заявка удаляется из базы данных.

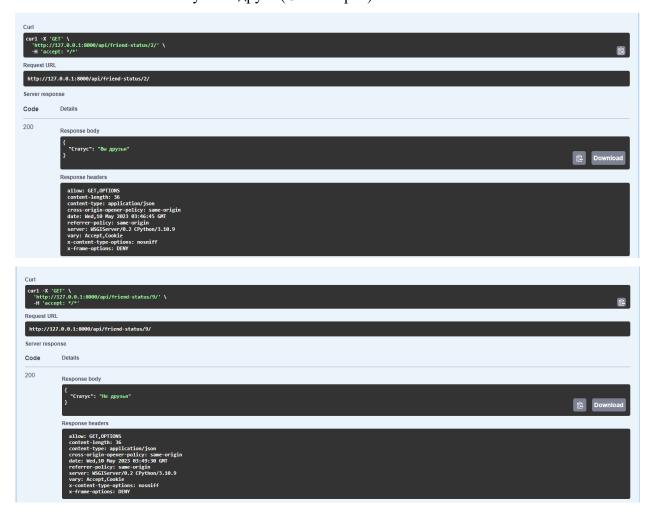
http://localhost:8000/api/friend-requests/ - посмотреть список входящих и исходящих заявок. (GET-запрос)



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



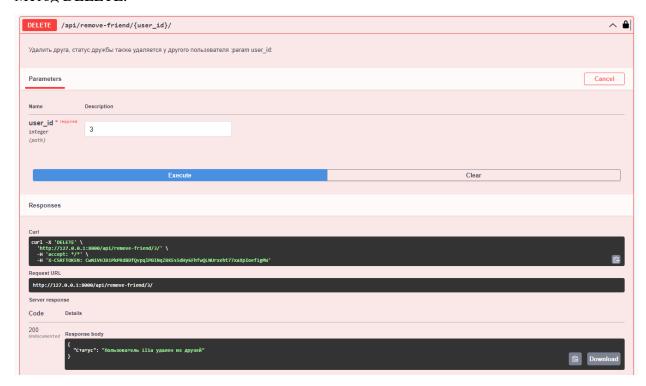
http://localhost:8000/api/friend-status/<int:user\_id>/ - проверить наличие заявок от пользователя или его статус как друга (GET-запрос)



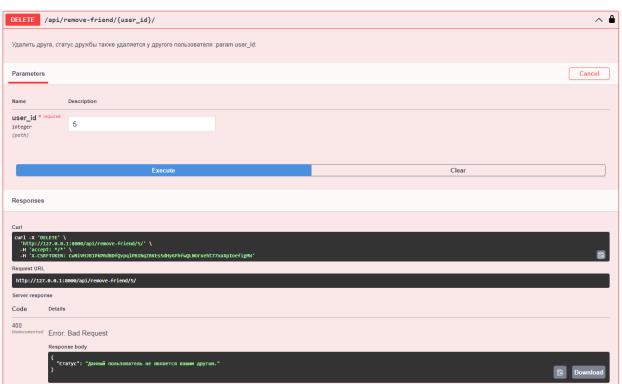
Приложение проверяет, не указываете ли Вы в запросе свой идентификатор и существует ли пользователь с таким идентификатором



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.

