

Node.JS Тестовое задание

REST API Server, технические требования:

- Метод авторизации: Bearer-token, разрешить CORS, время жизни токена 20 минут, но при каждом использовании токена в течении этих 20 минут продлить (то есть человек будет считаться не авторизованным только после 20ти минут бездействия)
- Основной формат общения - JSON
- В качестве БД можно использовать либо MongoDB либо любой SQL Server



Вступление

Задача тестового задания: реализовать упрощенный вариант взаимоотношения между людьми. Положим у нас есть сайт, где человек может авторизоваться, искать людей и добавлять себе кого либо в качестве “друга”. (под “другом” понимается вид взаимоотношений)

В зависимости от базы данных нужно реализовать связь для “друзей” (то есть один юзер может иметь несколько “друзей”, при этом, мы будем полагать, что юзеры абсолютно не честные, и если юзер **A** сказал, что он “лучший друг” для юзера **B**, то это не действует в обратном направлении)

API

- Запрос, по которому будет генерироваться токен (по паре username/password), в респонсе должен лежать непосредственно токен (токен должен храниться в бд);
- Регистрация нового пользователя (продумать варианты статусов ответа согласно модели данных);
- Запрос, который возвращает краткую инфу о пользователе-хозяине токена (id, email, firstName, lastName);
- Запрос который отмечает все активные токены текущего юзера как удаленные;
- Возвращает друзей указанного пользователя ({id, firstName, lastName, friendshipDuration, friendshipType});
- Получение всех зарегистрированных пользователей ({id, firstName, lastName});
- Добавление юзера к юзеру в качестве друга ({friendshipType: '1,2,3,4,5'}), где 1- лучший друг, 2-любовь всей жизни, 3-родственник, 4 - друг, 5-коллега;
- Запрос, который удаляет друга (по id) у пользователя.

P.S. добавление/удаление должно быть запрещено со стороны, то есть пользователь **A** не может добавить друга для пользователя **B**, только себе.

P.P.S. Все запросы кроме sign-in доступны только для авторизованных пользователей

Модель данных

- **User**
 - id
 - firstName
 - lastName
 - email
 - phoneNumber

В БД также необходимо хранить токены авторизации, схему этой таблицы/документа проработать самостоятельно..

По вопросам обращаться igor@devhouse.pro

Успехов!