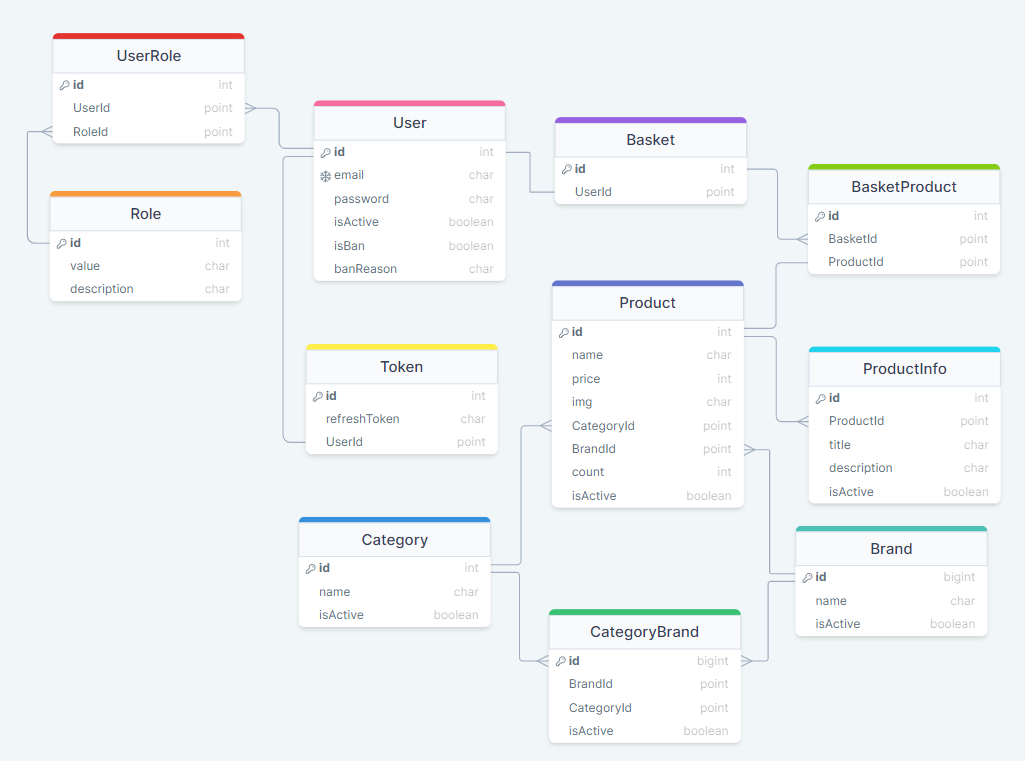
Приветствую вас дорогие мои читатели. Это документ по работе с моим чудо-API. Сначала была предпринята попытка создания документа с использованием SWAGGER, но из-за великой тайны, покрытая странными обстоятельствами, ничего не получилось. Поэтому был создан этот прекрасный и супер мега ультра понятный документ. Приятного ознакомления! PS: номер скорой помощи ***03***

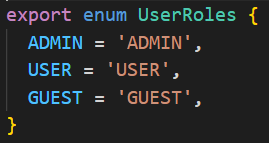
СХЕМА БД ДЛЯ ТЕХ, КТО **НЕ СМОГ** ПОСМОТРЕТЬ САМ!!!!



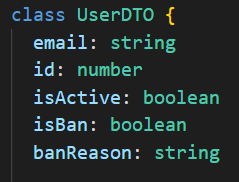
(у Product больше нет rating)

**ДЛЯ ВСЕХ ЗАПРОСОВ ДОБАВЛЯЕМ HEADER С `Bearer ${ACCESSTOKEN}`!!!**

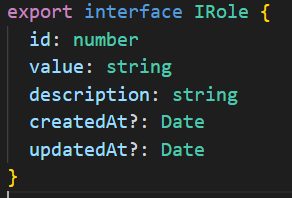
ЕНАМ РОЛЕЙ ПОЛЬЗОВАТЕЛЯ (Теперь только админ!!! Все остальное создает этот админ!!!)



USERDTO



IROLE



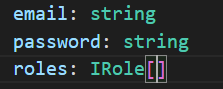
***ПОГНАЛИ 😊***

**ROUTE:** <http://localhost:8080/api/user/registration> post

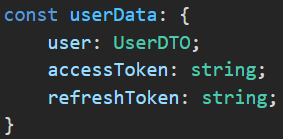
Регистрация пользователя в системе

REQUEST

req.body



RESPONSE



**ROUTE:** <http://localhost:8080/api/user/login> post

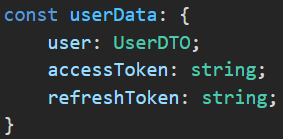
Вход пользователя в систему

REQUEST

req.body



RESPONSE



**ROUTE:** <http://localhost:8080/api/user/logout> post

выход пользователя из системы

REQUEST

req.cookies



RESPONSE



**ROUTE:** <http://localhost:8080/api/user/refresh> get

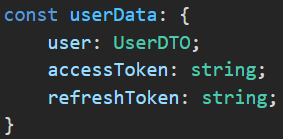
Обновление токенов пользователя

REQUEST

req.cookies



RESPONSE



**ROUTE:** <http://localhost:8080/api/user/users> get (admin)

Получение всех пользователей

REQUEST

-

RESPONSE



**ROUTE:** <http://localhost:8080/api/user/ban> post (admin)

бан пользователя

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/category/> post (admin)

Создание категории

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/category/> get

Получение всех категорий

REQUEST

-

RESPONSE



**ROUTE:** <http://localhost:8080/api/category/:id> put (admin)

Удаление категории

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/brand/> post (admin)

Создание бренда

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/brand/> get

Получение всех брендов

REQUEST

-

RESPONSE



**ROUTE:** <http://localhost:8080/api/brand/:id> put (admin)

Удаление бренда

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/product/> post (admin)

Создание бренда

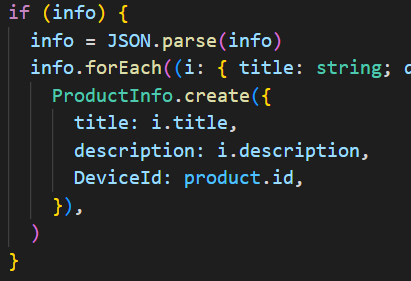
REQUEST





Info – массив характеристик.

**ЗАМЕЧЕНИЕ**: тк идет передача данных через form-data, они приходят в виде строки. Поэтому НЕОБХОДИМО на фронте парсить его в JSON строку, а на бэке обратно в js объекты!!!



По сути, это все form-data, поскольку нам нужно прикрепить file. Также BrandId и CategoryId должны быть обязательными!!!

RESPONSE



**ROUTE:** <http://localhost:8080/api/product/> get

Получение всех продуктов

REQUEST

req.**query**!!!



Limit – количество продуктов, которые будут отображаться на одной странице.

Page – страница





RESPONSE

Чтобы подсчитать количество страниц уже на фронте, нужно знать общее количество товаров, которое вернется по заданному запросу.



(Не typeId, а CategoryId)

**ROUTE:** <http://localhost:8080/api/product/:id> get

Получение одного продукта

REQUEST



RESPONSE





(Не typeId, а CategoryId)

**ROUTE:** <http://localhost:8080/api/product/:id> put (admin)

Удаление продукта

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/role> post (admin)

Создание роли

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/role/addRole> post (admin)

Добавление новых ролей определенному пользователю

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/role> get (admin)

Получение всех ролей

REQUEST

-

RESPONSE



**ROUTE:** <http://localhost:8080/api/role/getOne> get (admin)

Получение роли по имени (например ADMIN)

REQUEST



RESPONSE



**ROUTE:** <http://localhost:8080/api/role/getUserRoles> get

REQUEST





RESPONSE

