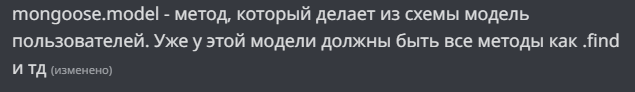
Нужно хранить пользователей в базе даных. Роуты все те же, суть и та же, просто теперь все юзеры должны храниться в базе данных.

У тебя уже должен быть скачан MongoDB Compass. Его используй для того, чтобы посмотреть, какие данные сейчас лежат в базе данных. База данных у нас облачная, мы уже с тобой регистрировались, и у тебя должна остаться ссылка для подключения к базе. Используем библиотеку mongoose.



const express = require("express") //подключаем модуль express

const mongoose = require('mongoose');

mongoose

.connect('mongodb+srv://MihailKrasota:xegfrf,hf567@33333.xo3esxv.mongodb.net/?retryWrites=true&w=majority', { useNewUrlParser: true })

.then(() => {

    const app = express() //вызываем express

    const SchemaUser = mongoose.Schema({

        email: String,

        password: String,

        id: Number,

    });

    const model = mongoose.model('restoration',SchemaUser);

        app.get("/users", async (req, res) => {

            const users = await model.find({});

            res.send(users)// ответ

        })

        app.post("/users", async (req, res) => {

            const user = model ({

                email: req.query.email,

                password: req.query.password,

                id: Math.round(Math.random() \* 1000),

            });

            await user.save()

            res.send(user)//ответ

        })

        app.patch("/users", async (req, res) => {

            const userPatch = await model.findOne({ id: req.query.id })

            userPatch.email = req.query.email

            await userPatch.save()

            res.send(userPatch)// ответ

        })

        app.delete("/users", async (req, res) => {

            await model.deleteOne({ id: req.query.id })

            res.send("пользователь удален")// ответ

        })

        app.listen(3000, () => {

            console.log('Server has started!');

        });

    });