Построим REST API с использованием фреймворка Express и СУБД postgres.

В первую очередь вам понадобится установить node JS на компьютере если вы этого ещё не сделали. Как СУБД установить Postgresql и чтобы это дело всё протестировать нам понадобится Postman чтобы отправлять HTTP запросы к REST API/

Итак приступим. Я сделал пустой проект в шторме и сразу же его проинициализировал. Для этого откройте терминал и напишем команду

>npm init -y

чтобы все настройки стали по умолчанию укажу флаг – y. Должен был создаться файл package.json. И теперь мы можем устанавливать зависимости. Нам понадобится само собой Express и модуль PG. Это модуль для работы с postgres. Их мы установим как основные зависимости.



В качестве зависимости для разработки установим ещё один модуль, для этого укажем флаг –D. Модуль называется nodemon



чуть дальше вы поймёте для чего он нужен. И пока модуль устанавливается создадим index.js файл. Здесь будет запускаться наш сервер и теперь сделаем базовую конфигурацию чтобы сервер мог принимать запросы для этого импортируем экспресс

const express = require('express');

и зададим порт. Порт будем в первую очередь пытаться получить из системных переменных и если ли такая переменная не задана, то по умолчанию сделаем порт 8080.

const express = require('express');

const PORT = process.env.PORT || 8090;

Далее из экспресса создадим сам сервер

const express = require('express');

const PORT = process.env.PORT || 8090;

const app = express();

и заставим этот сервер слушать порт. Для этого вызовем команду listen и первым параметром туда передаём порт, а вторым функцию callback которая будет просто в логе выводить новую информацию о том что сервер был запущен на таком-то порту

const express = require('express');

const PORT = process.env.PORT || 8090;

const app = express();

app.listen(PORT, () => console.log(`server started on port ${PORT}`));

Теперь откроем package.json и сделаем скрипт запуска сервера. Для этого находим поле scripts заменяем на start и тут пишем команду

  "scripts": {

    "start": "node index.js",

  },

В терминале пишем



и если вы увидели надпись консоли что сервер запустился значит всё хорошо.

Давайте теперь сделаем первый маршрут, по которому мы сможем отправлять запросы. Вызываем функцию, которая первым параметром принимает как раз путь, а вторым функцию callback, которая принимает также параметрами запрос и ответ. И у запроса вызываем функцию sand чтобы отправить на клиент какие-то данные. В нашем случае это просто текст.

const express = require('express');

const PORT = process.env.PORT || 8090;

const app = express();

app.get('/', (req, res) => {

    res.send('HELLO POSTGRES + NODEJS')

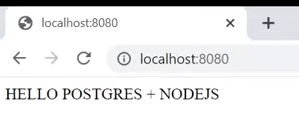
})

app.listen(PORT, () => console.log(`server started on port ${PORT}`));

Теперь попробуем перейти по url  в браузере и увидим что сервер не может обработать запрос, поскольку мы функцию написали а сервер не перезапустили



Обновим страницу и сервер обрабатывает запрос



но есть некая сложность. Если мы будем изменять как-то код, каждый раз придётся перезапускать сервер. Именно для этого мы установили nodemon. Он необходим только как dev зависимость на этапе разработки. Поэтому создадим ещё один скрипт, который будет запускать index.js-сервер, но уже через nodemon.

  "scripts": {

    "start": "node index.js",

    "dev": "nodemon index.js"

  },

При внесении изменений в код и сохранении сервер автоматически перезапускается.

3:13