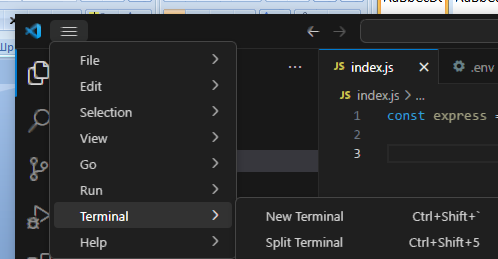
Первым делом – надо установить visualstudiocode, nodejs, git и зарегистрироваться на github.

Создать директорию с проектом, в ней создать еще две директории – clientи server.

Открыть папку проекта в visualstudiocode и запустить терминал в visualstudiocode.



В терминале перейти в директорию server, инициализировать nodejs командой:

**npminit**

Установить зависимости, которые нам потребуются.

**Npm install cors**

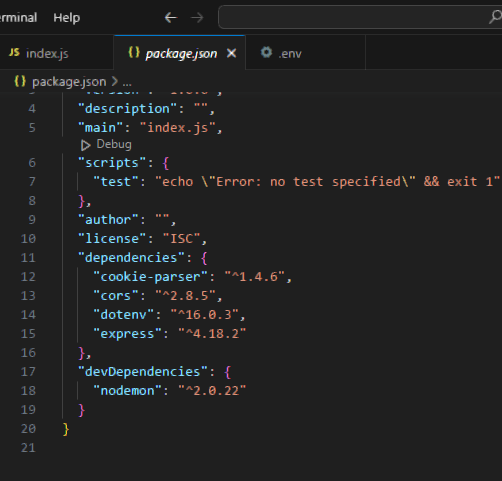
**Npm install express**

**Npm install cookie-parser**

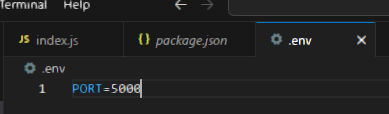
**Npm install dotenv**

**Npm install nodemon --save-dev**

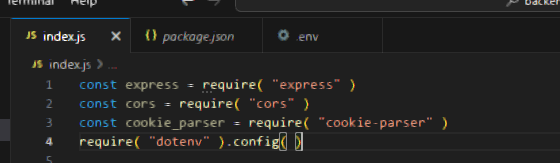
Долженпоявитьсяфайликpackage.jsonследующеговида:



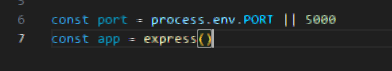
Создать файл .envв директории server. Заполнить его следующим образом:



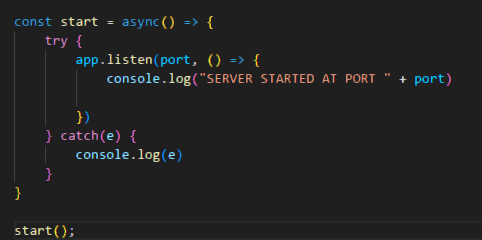
Создать файл index.js. В нем в первых строчка нужно импортировать пакеты с помощью функции require().



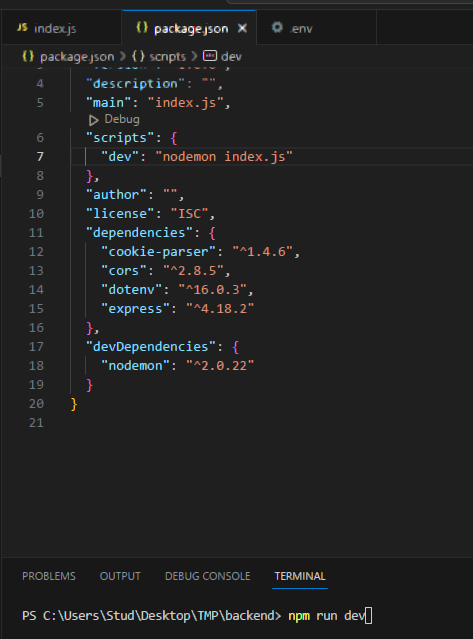
Далее нужно объявить две переменные, одна из них – это порт, а вторая – express.



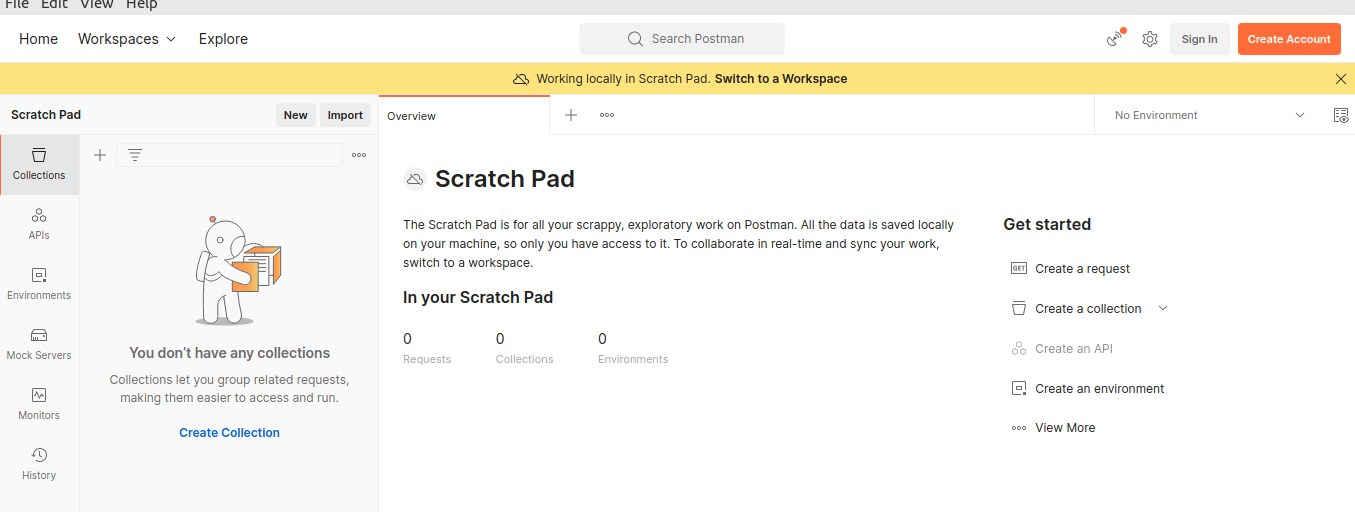
Создаем асинхронную функцию для запуска сервера.



Играем с nodemon – открываемpackage.json и редактируем объект scripts.



Установить postman – софт для отправки тестовых запросов без веба (можно использовать curl, если знаете что это такое) (snapinstallpostman)



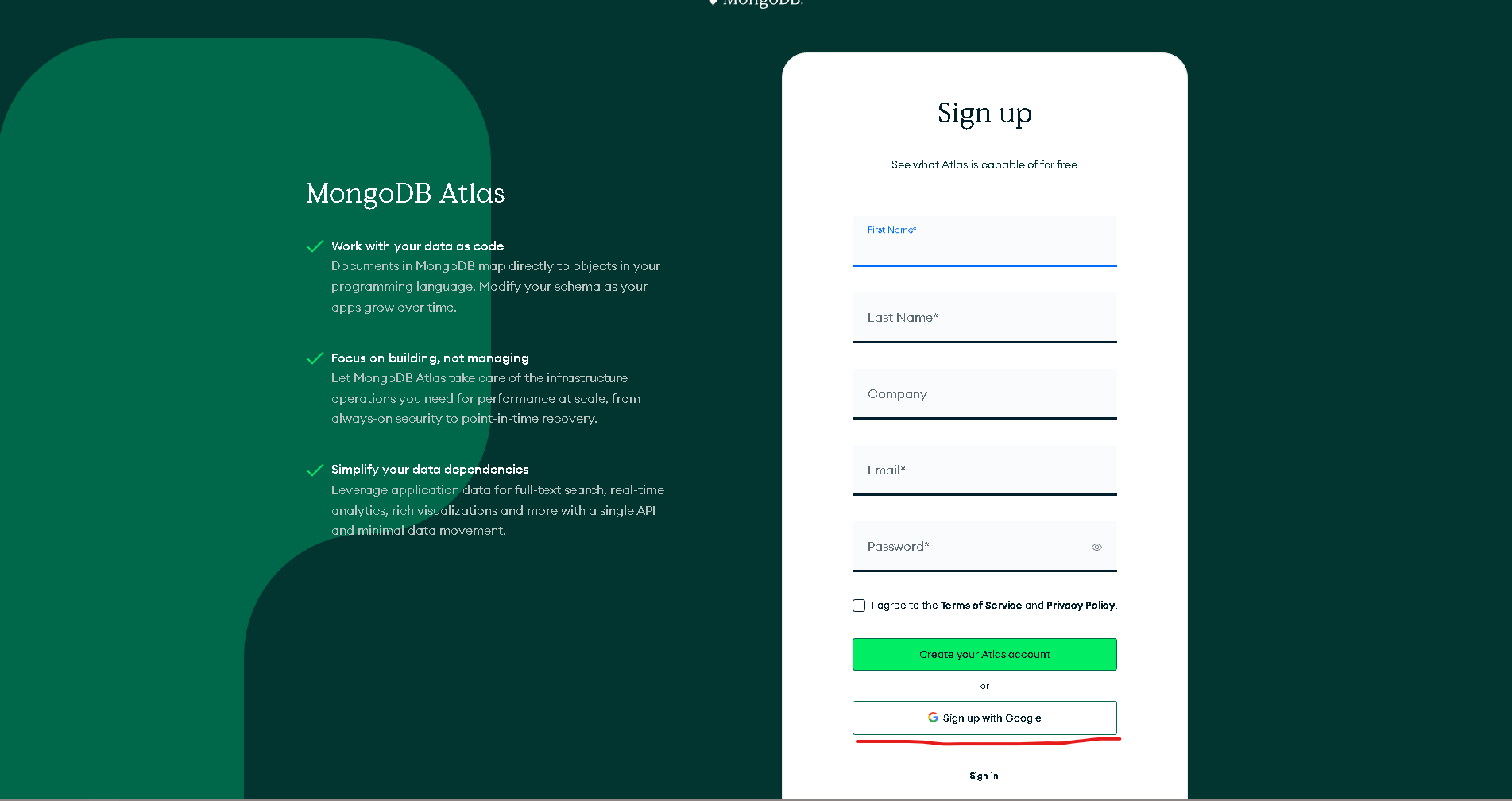
УСТАНАВЛИВАЕМ ВПН НА БРАУЗЕР! САНКЦИИ!!! (Не вздумайте вставить этот пункт в отчете, Дед побьет)

Ставимmongoseиmongodb. Mongodb – облачная база данных, а mongoose– штука, необходимая для работы с ней

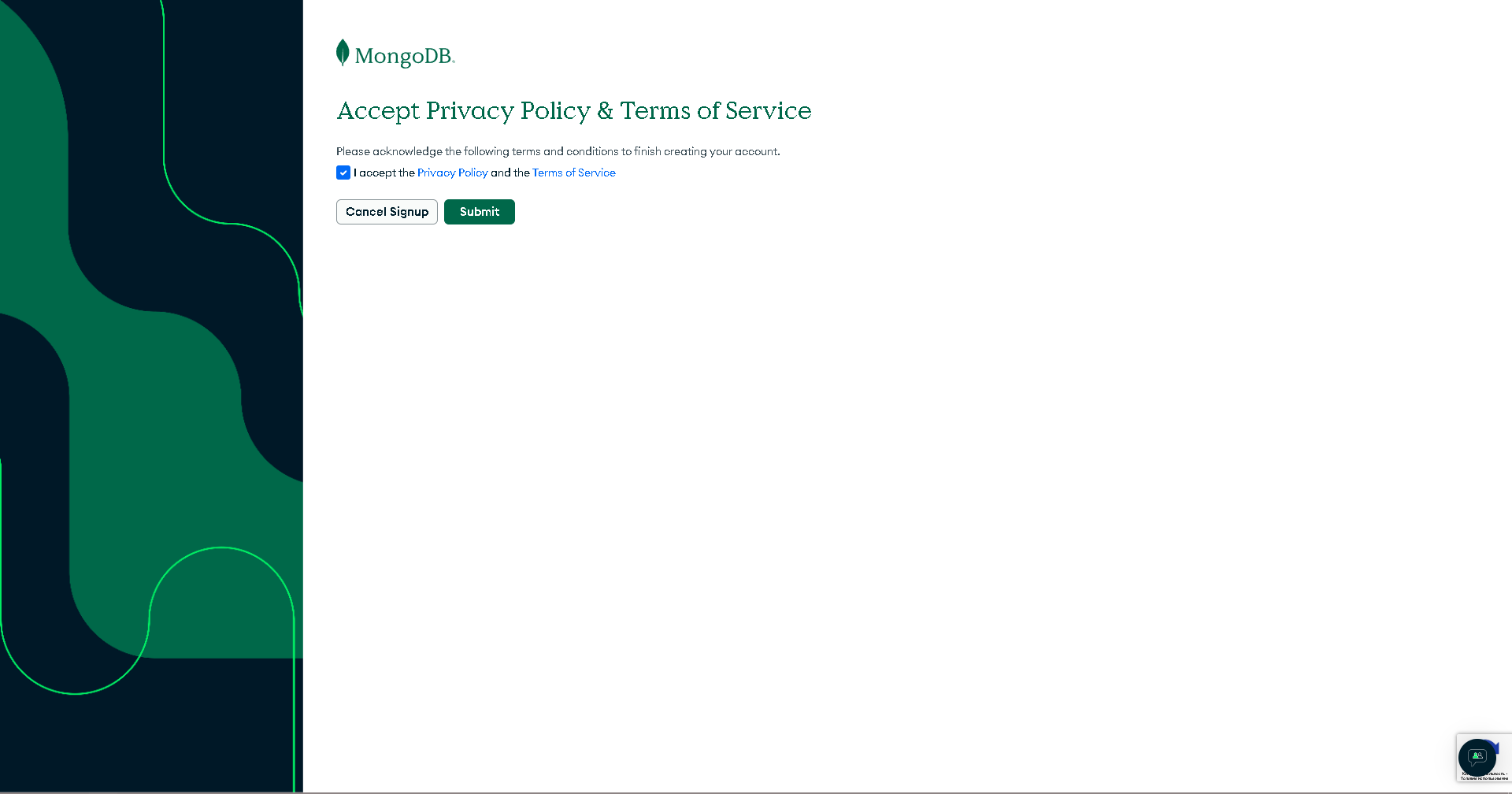


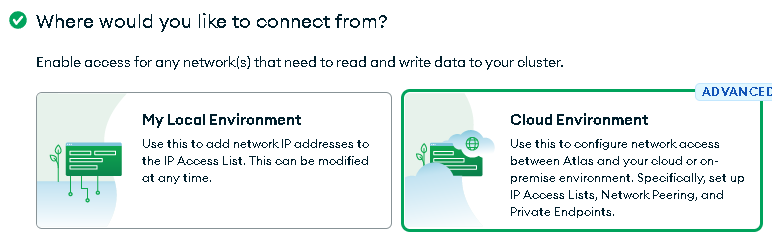
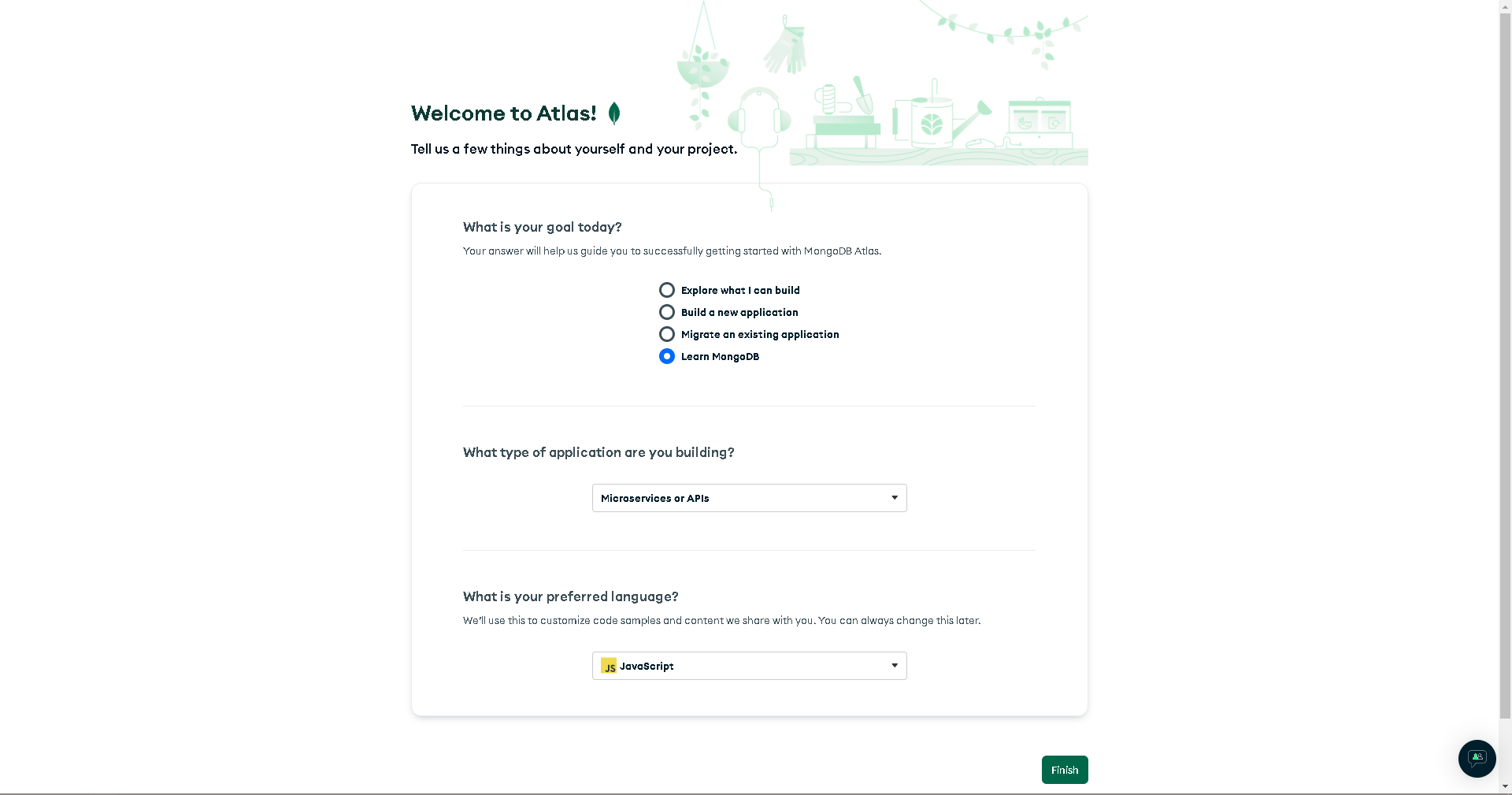
ОБЯЗАТЕЛЬНО УСТАНОВИТЕ NODEJSИМЕННО С ОФИЦИАЛЬНОГО САЙТА! Иначе могут возникнуть проблемы

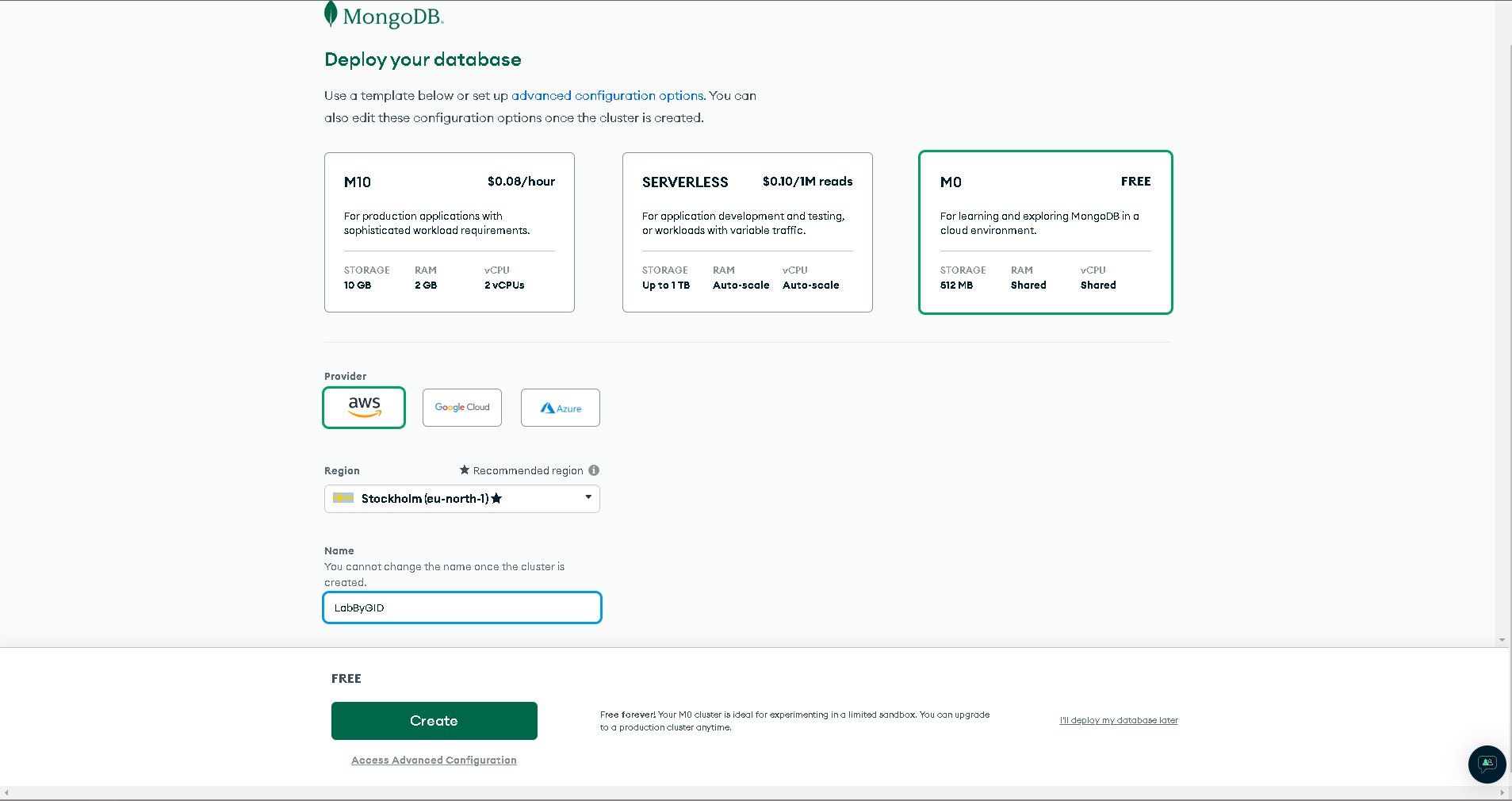
**(ПОКА НЕ ТРЕБУЕТСЯ)**Регистрируемся на MongoDB (ВКЛЮЧИТЬ ВПН, ИНАЧЕ RST)



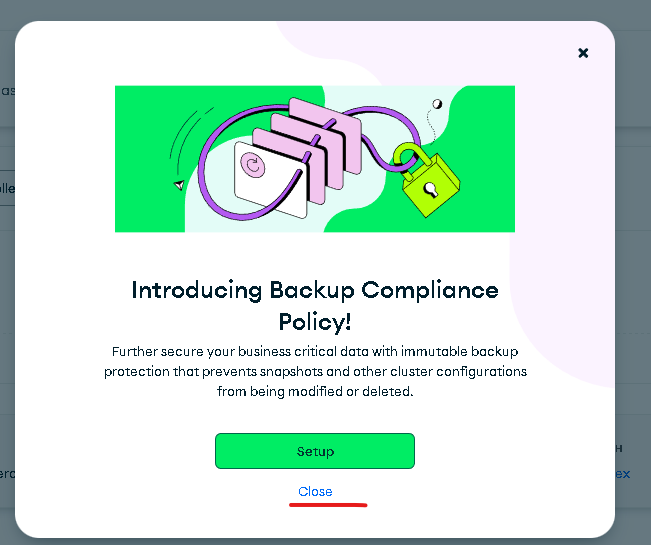
Принимаем TOS



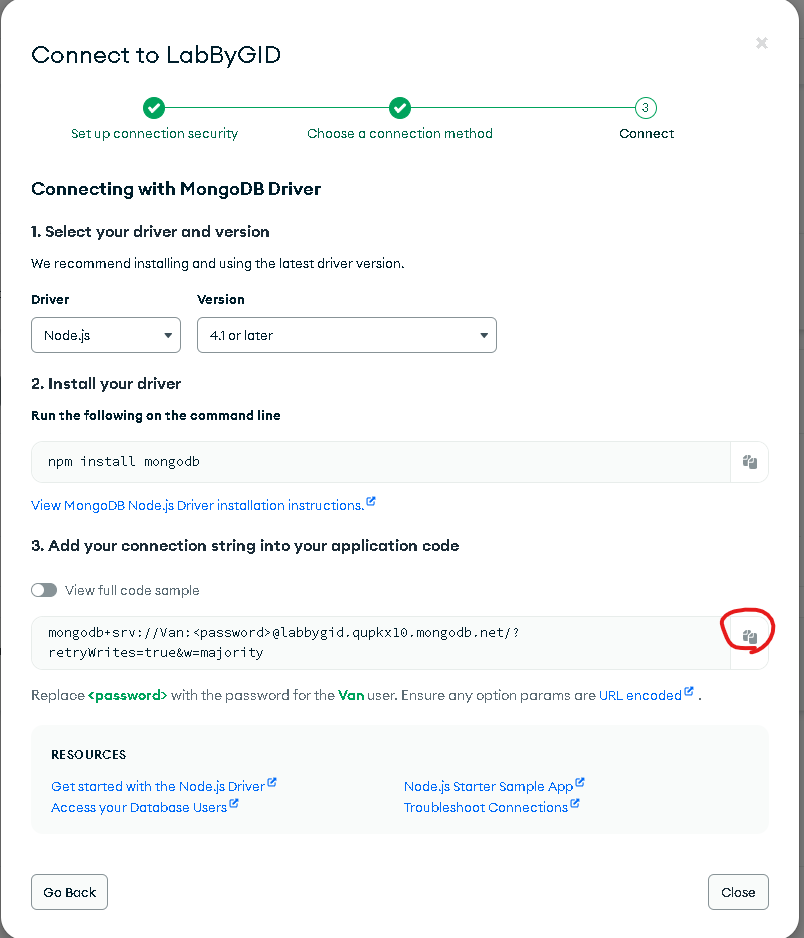
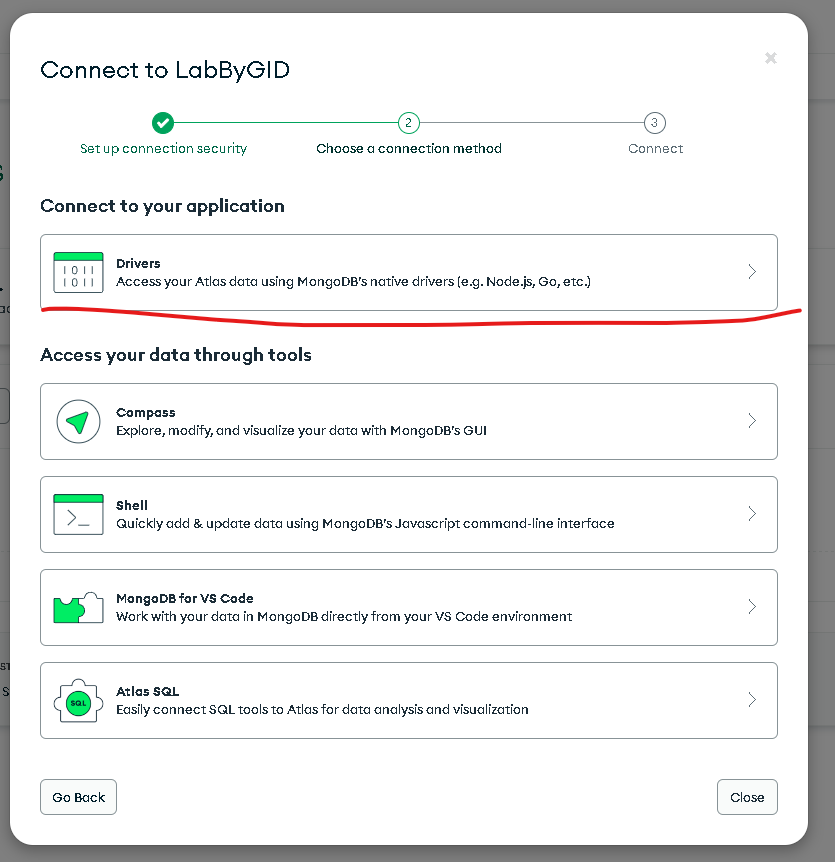
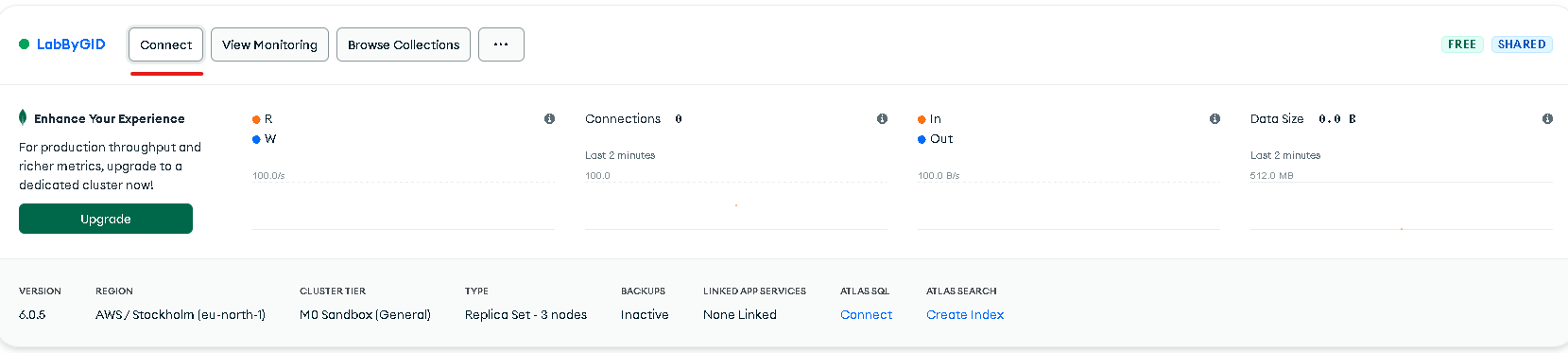
Настраиваемпрофиль



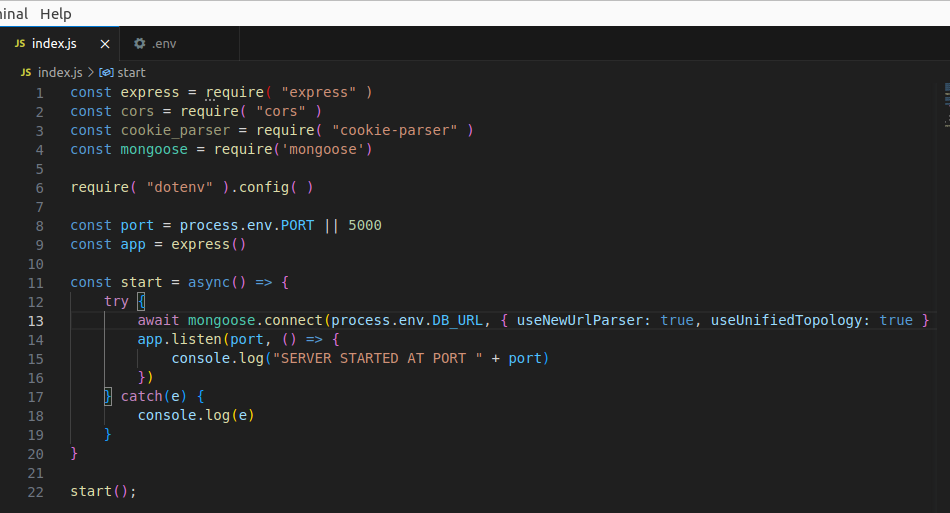
Закрываемрекламу



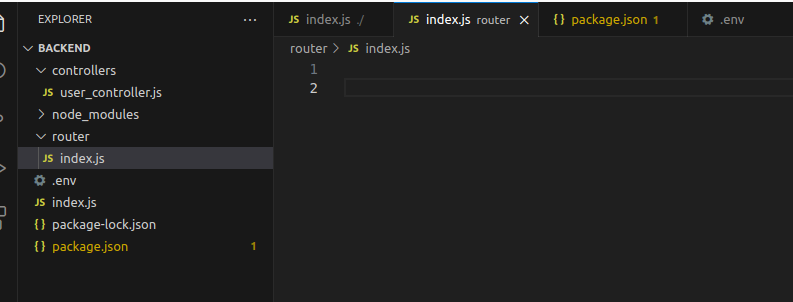
Создаем базу



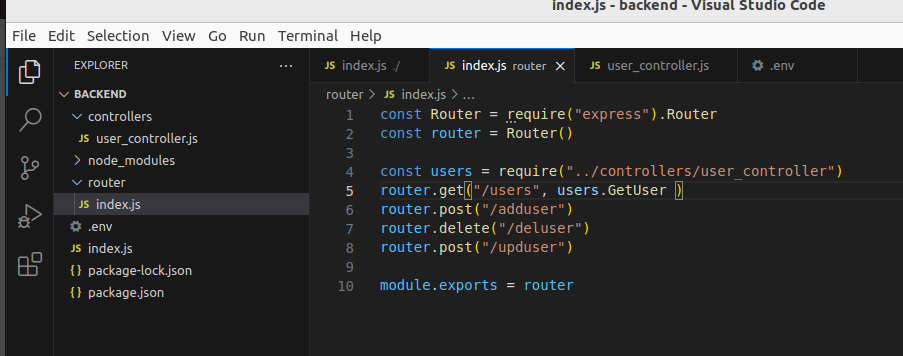
Корректируемкод



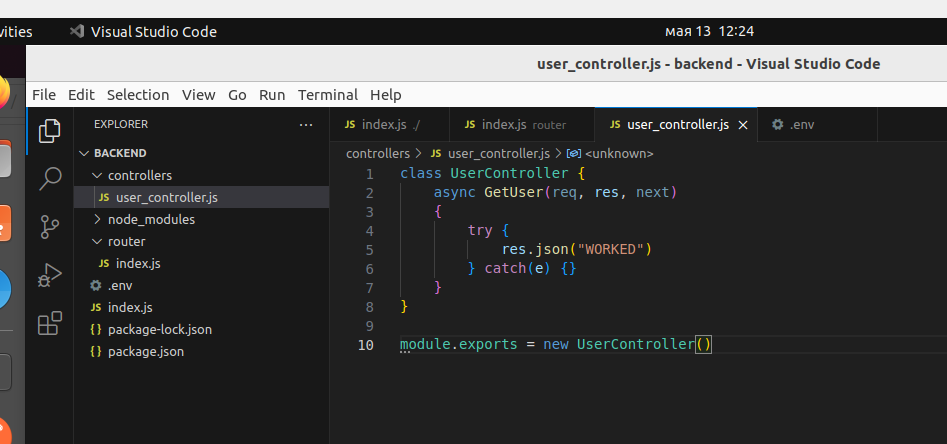
Создаемпапку controllers, router



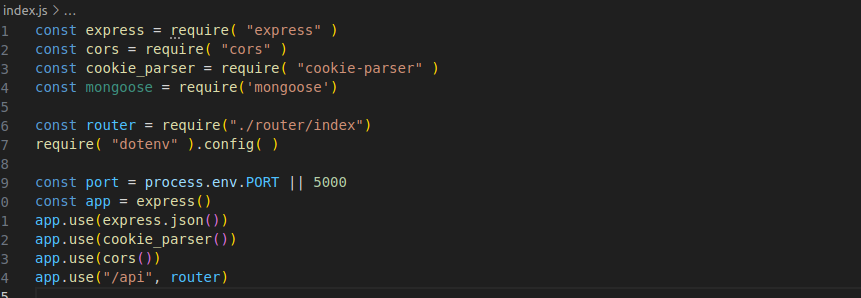
Содержимое /router/index.js



Содержимое /controllers/user\_controller.js

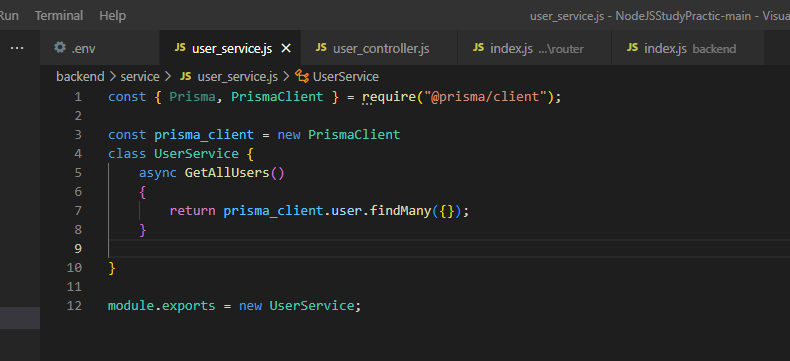


Новоесодержимое index.js

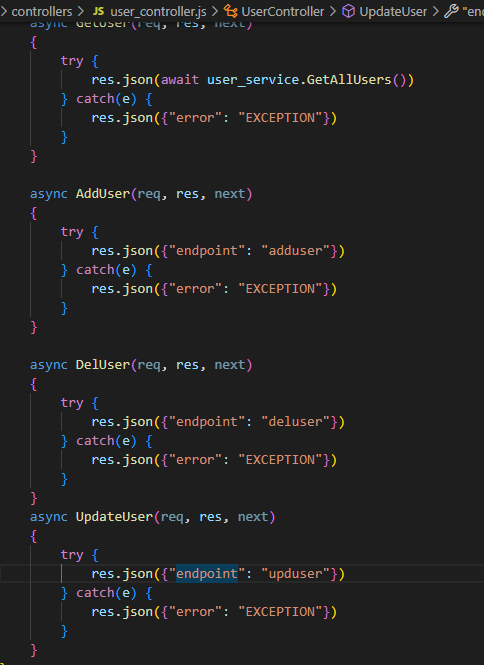


Обратиться к методичке <https://docs.google.com/document/d/e/2PACX-1vT-laa71S8z5KFWU29sZ7cV0zA_fDTYqWqIk8NtFH1Z5OxYOvcLqmfE_SdJWY7EX6ZhnnSGPNL5aGUI/pub>

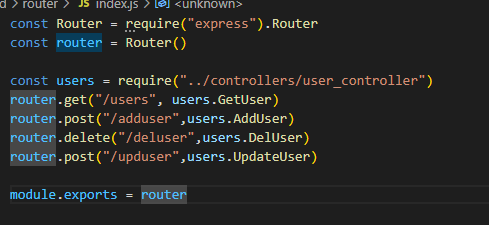
Создаем новую папку – services,в ней создаемuser\_service.js.



Дописываем все обработчики всех наших эндпоинтов



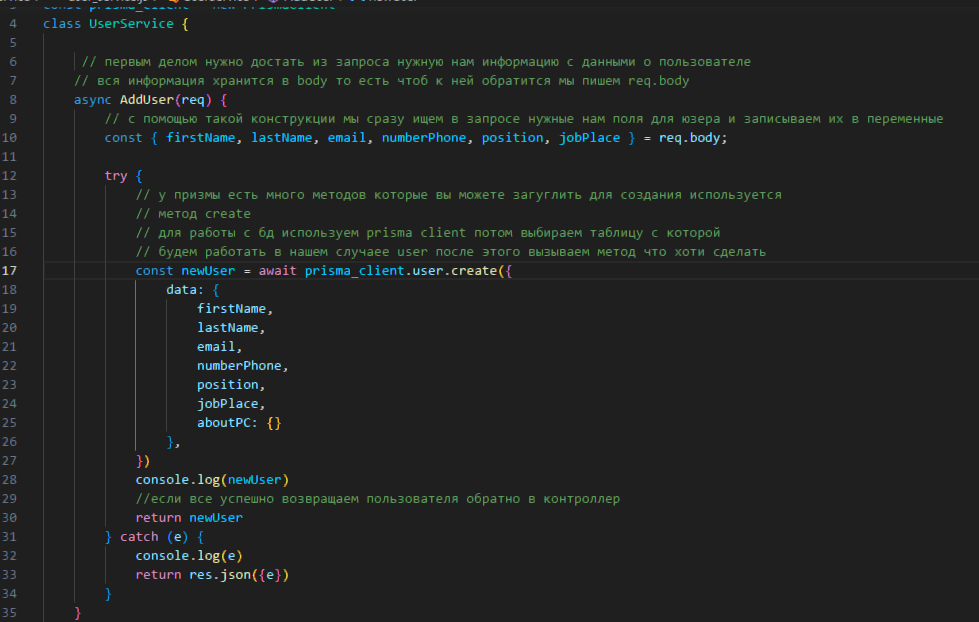
Редактируем роутер



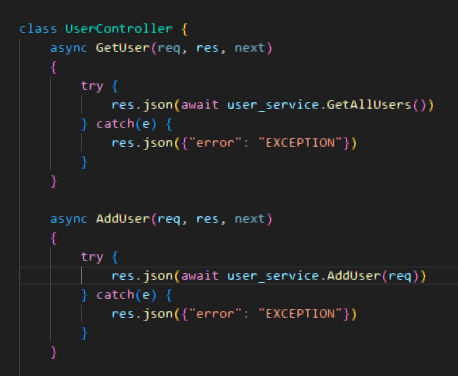
19.05 – Дописаны комментарии

Про постман <https://docs.google.com/document/d/15-HGvGAUtwz18KMCyBgIQ5lRB7jlaA2U4Rlp33szQ6w/edit?usp=sharing>

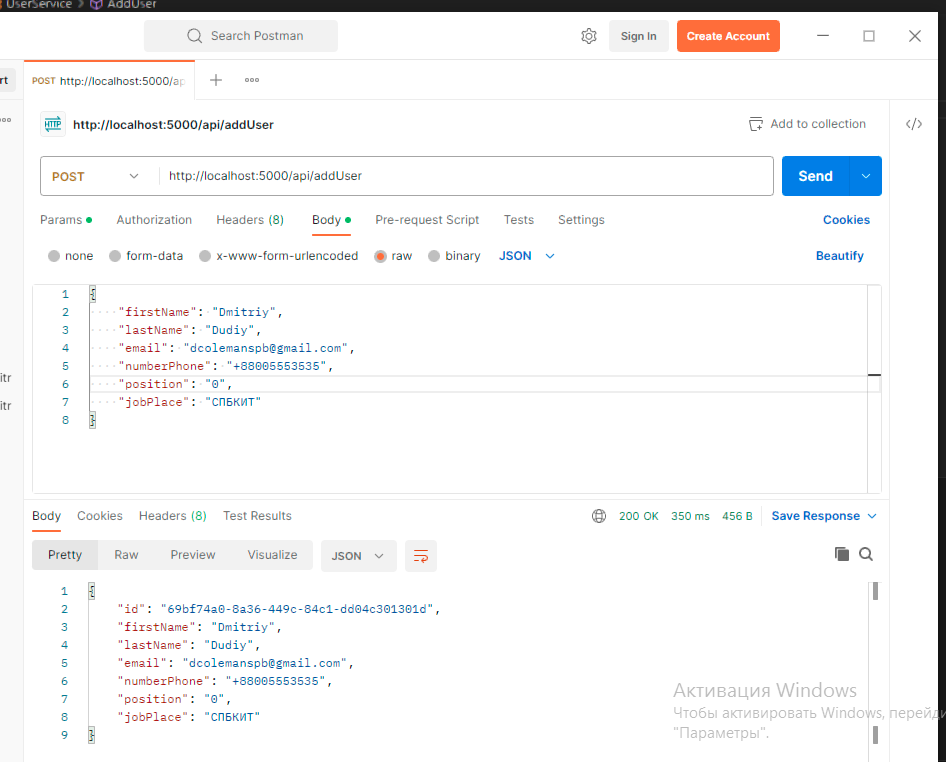
Дописываем функцию AddUser в класс UserService

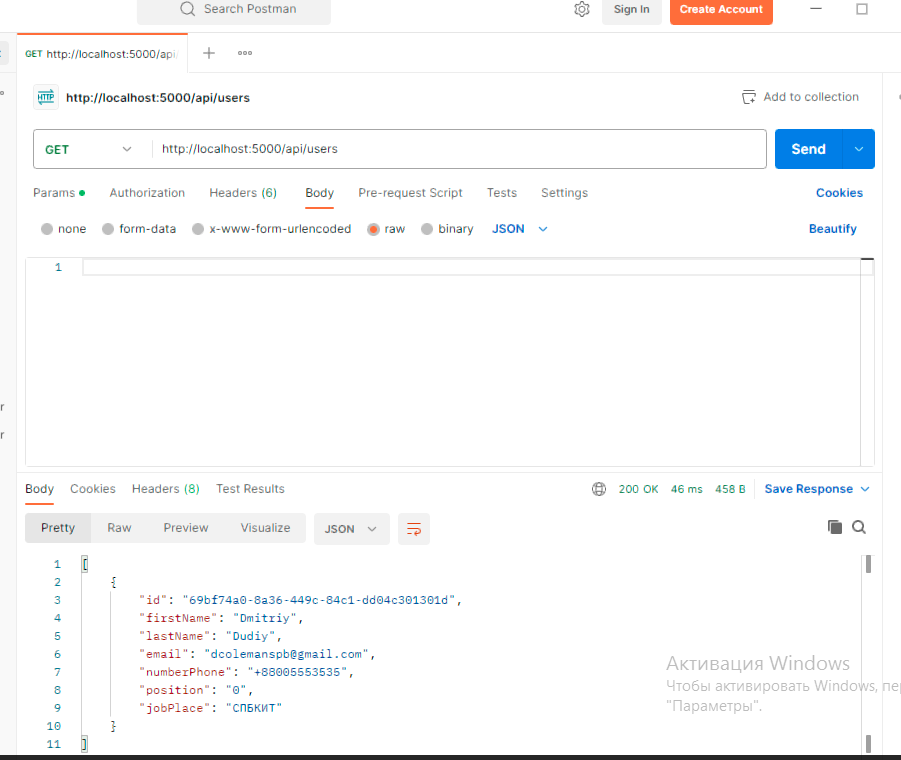


Редактируем user\_controller.js



Тестируем добавление пользователя через postman

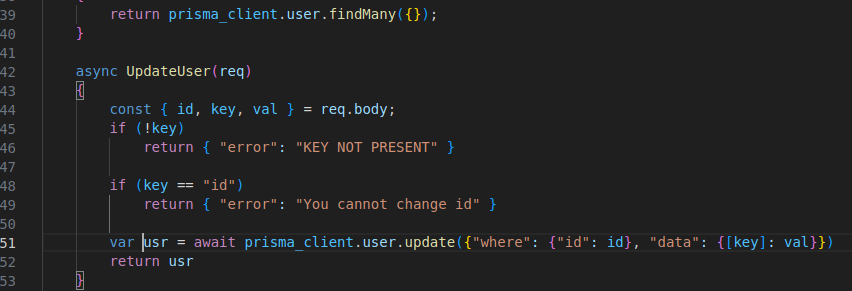


Теперь GET запрос на эндпоинт /api/users должен вернуть созданного нами пользователя

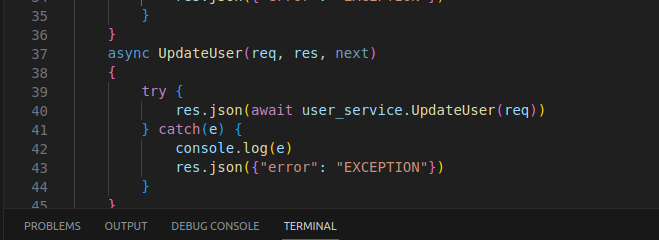
20.05 – редактирование пользователя по его ID

Дописываем новый член класса UserService

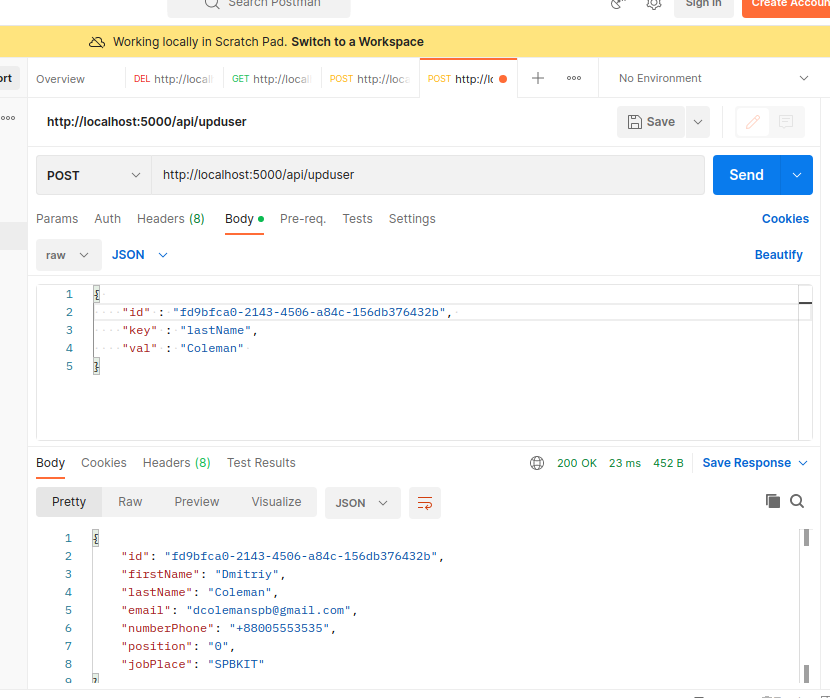
Запрещаем менять ID, чтобы не вызвать баги



Редактируем user\_controller



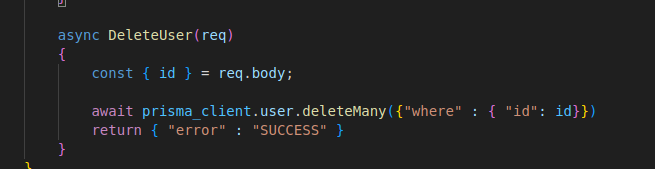
Обращаемся к POSTMAN и редактируем созданного нами ранее пользователя



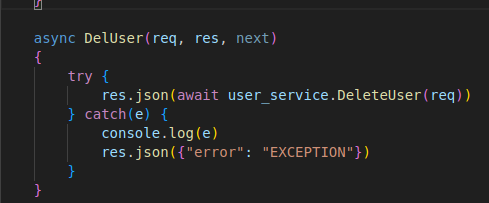
Дудий теперь Колеман…

Делаем удаление пользователя

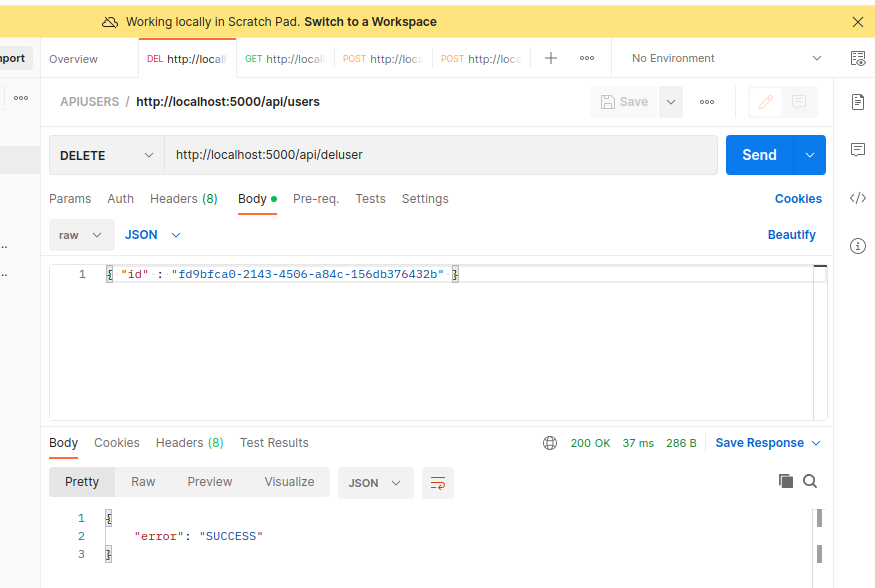
Класс UserService



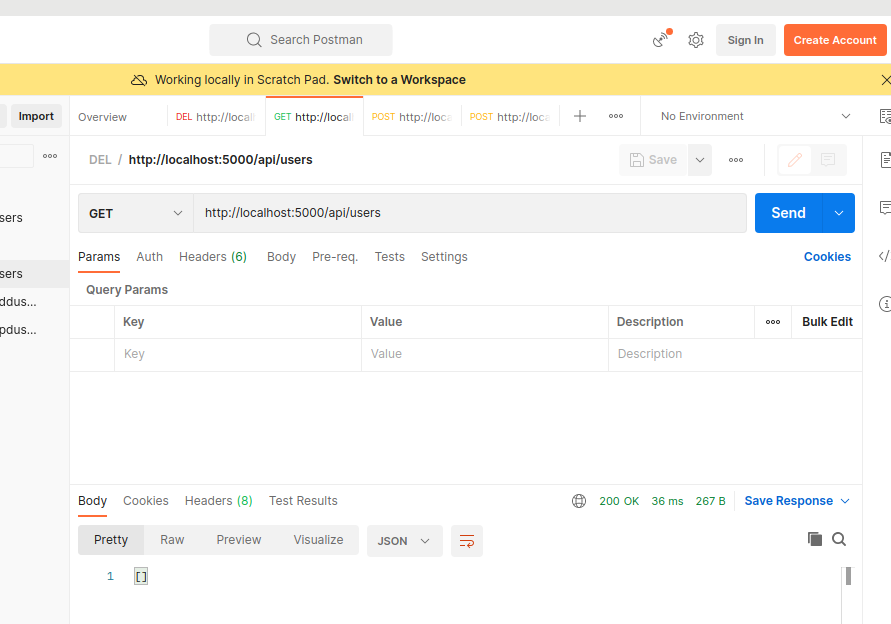
Класс UserController



Тестируем в postman

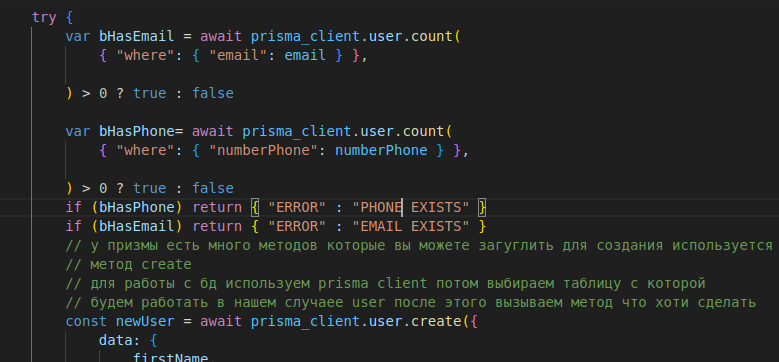


GET к /api/users должен вернуть пустоту



Добавляем проверку на существование номера телефона и email при добавлении нового юзверя

Дописать в UserService::AddUser



В UserService::UpdateUser

