

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №1: Boilerplate на express +
sequelize + typescript

Выполнила:

Лорс Хава

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задачи

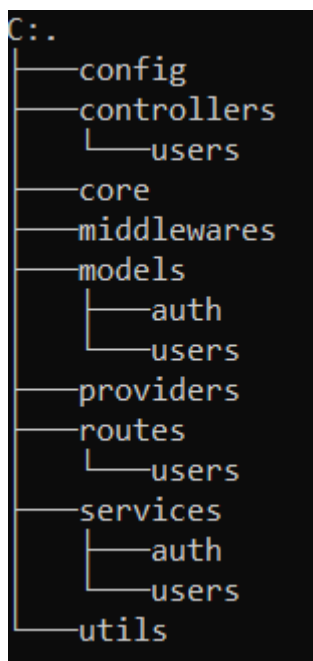
Нужно написать свой boilerplate на express + sequelize / TypeORM + typescript.

Явное разделение на:

- модели
- контроллеры
- роуты
- сервисы для работы с моделями (реализуем паттерн “репозиторий”)

Ход работы

Структура приложения



Здесь:

config: содержит файлы конфигурации для базы данных, других сторонних сервисов и приложения в целом.

controllers: содержит классы контроллеров, которые обрабатывают запросы от клиента и возвращают ответы.

core: содержит основные компоненты приложения, такие как класс App для конфигурирования и запуска express-приложения, server для запуска сервера, и database для настройки и синхронизации базы данных.

middlewares: содержит промежуточные обработчики, такие как middleware аутентификации.

migrations: содержит файлы миграций базы данных.

models: содержит модели, описывающие таблицы в базе данных.

providers: содержит классы, которые предоставляют методы для работы с моделями.

routes: содержит классы, описывающие маршруты API и использующие соответствующие контроллеры.

seeders: содержит файлы для заполнения базы данных начальными данными.

services: содержит классы, реализующие бизнес-логику и работу с моделями.

utils: содержит утилитарные функции, такие как функции хеширования и проверки паролей.

Модель пользователя состоит из *username*, *firstName*, *lastName*, *email*, *password*. Также создана модель для хранения токенов.

Реализованные методы класса UserController:

- POST /users/signup - метод для регистрации нового пользователя в системе. Принимает на вход данные пользователя (из модели), сохраняет их в базе данных и возвращает информацию о созданном пользователе.
- POST /users/login - метод для аутентификации пользователя в системе. Проверяет их на корректность и, если они верны, генерирует токен доступа, который в последующем будет использоваться для авторизации пользователя.
- GET /users/me - метод используется для получения информации о текущем авторизованном пользователе.

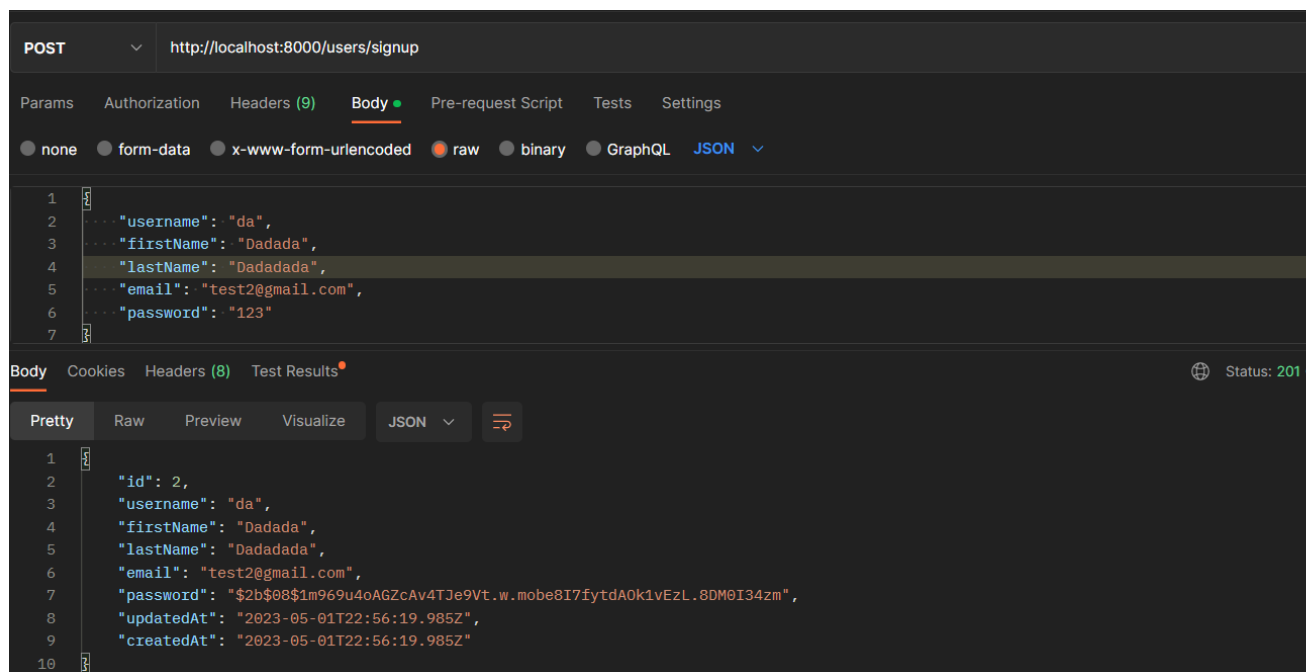
- POST /users/refresh - этот метод используется для обновления токена доступа, который может просрочиться через определенное время.

Реализованные методы UserService:

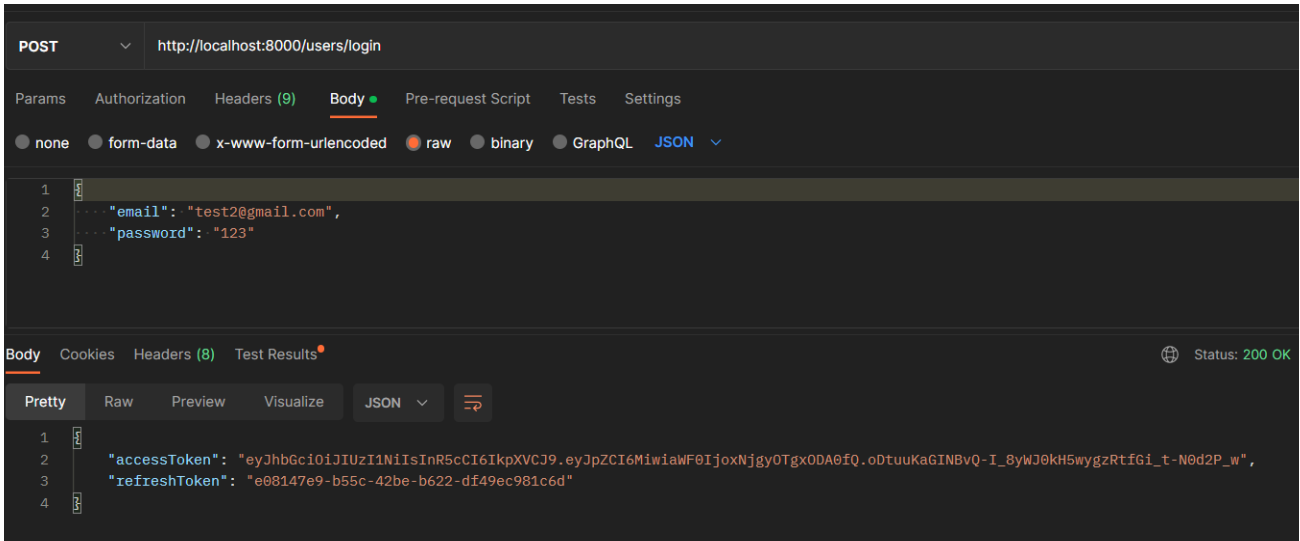
1. create: создает нового пользователя в базе данных на основе переданных параметров.
2. findById: ищет пользователя в базе данных по его id
3. checkPassword: ищет пользователя в базе данных по email и проверяет, соответствует ли введенный пароль хэшу пароля пользователя.

Создадим запросы, чтобы проверить работу boilerplate

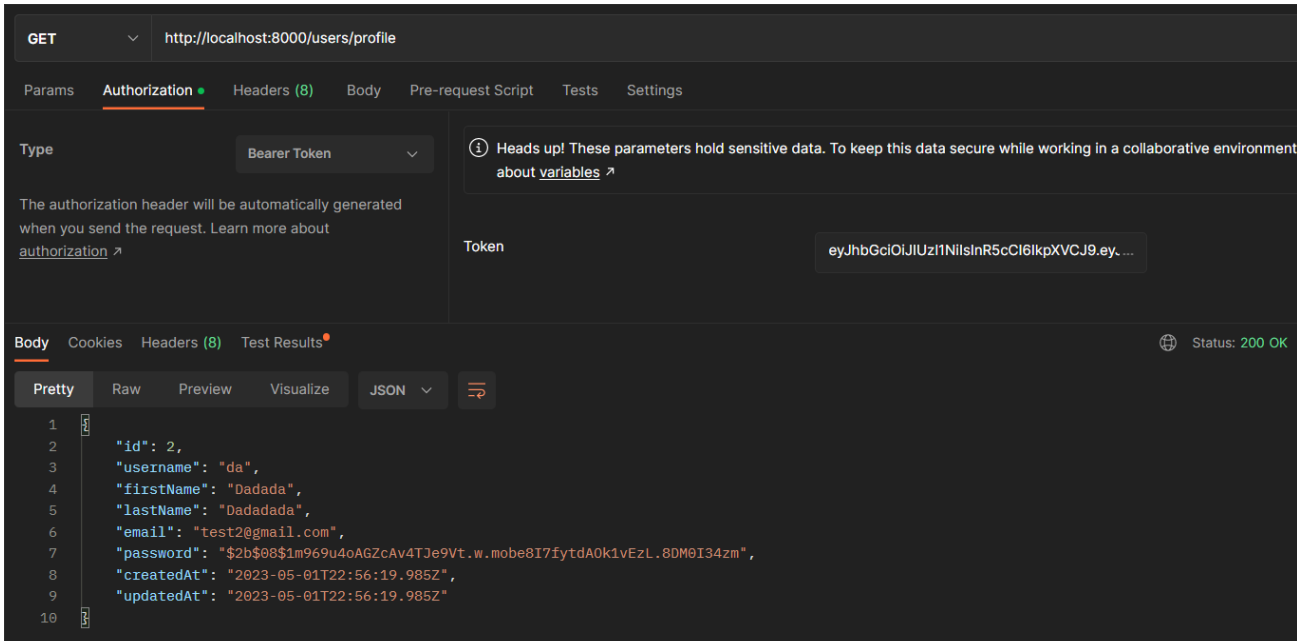
Регистрация пользователя:



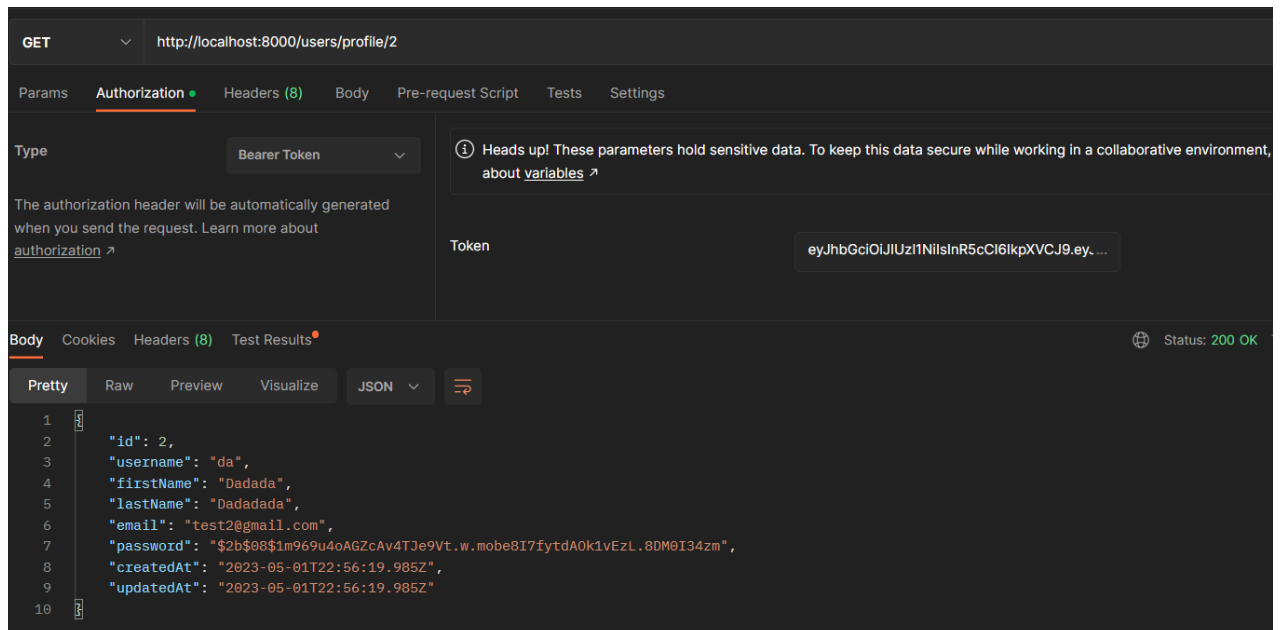
Вход для получения токенов:



Авторизация по токену:



Получение пользователя по *id*:



Вывод

В ходе выполнения лабораторной работы были получены практические навыки работы с express, sequelize, typescript в связке для создания своего готового шаблона boilerplate.