**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бек-энд разработка

Отчет

Лабораторная работа 1: Typescript: основы языка

Выполнила:

Пронина Александра

Группа

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Цель работы:

Целью данной лабораторной работы является разработка RESTful API для управления пользователями с использованием технологий Node.js, Express.js и Sequelize ORM.

Используемый стек технологии:

Node.js - среда выполнения JavaScript на сервере.

Express.js - фреймворк для создания веб-приложений на Node.js.

Sequelize - ORM (Object-Relational Mapping) для работы с базами данных в Node.js.

Ход работы:

Настройка проекта:

В начале работы был создан проект с использованием Node.js. Были установлены необходимые зависимости, такие как Express, Sequelize, и другие. Проект был настроен с использованием TypeScript для типизации кода.

Настройка базы данных:

Для работы с базой данных была выбрана SQLite, так как она легко интегрируется в проекты Node.js и не требует отдельного сервера баз данных. Был создан файл конфигурации для Sequelize, где указаны параметры подключения к базе данных.

Создание модели пользователей:

Была создана модель User, описывающая структуру данных пользователей. В модели были определены поля id, name и email, а также их типы и ограничения.

Разработка репозитория пользователей:

Был разработан репозиторий UserRepository, отвечающий за выполнение операций с данными пользователей (создание, чтение, обновление, удаление) в базе данных. Репозиторий использует модель User для взаимодействия с базой данных.

Разработка сервиса пользователей:

Был разработан сервис UserService, предоставляющий бизнес-логику для управления пользователями. Сервис использует репозиторий UserRepository для выполнения операций с данными пользователей.

Настройка маршрутов:

Были настроены маршруты Express для обработки запросов от клиента. Для управления пользователями были созданы маршруты POST /users (создание пользователя), GET /users/:id (получение пользователя по ID) и DELETE /users/:id (удаление пользователя).

Настройка сервера Express:

Был настроен Express сервер для прослушивания входящих запросов на определенном порту. В качестве middleware был добавлен body-parser для обработки JSON данных в запросах.

Тестирование:

После завершения разработки было проведено тестирование API с использованием инструментов для отправки HTTP запросов (например, Postman). Были протестированы все реализованные функциональности, чтобы убедиться в их корректной работе.

**Выводы:**

В результате выполнения лабораторной работы было разработано RESTful API для управления пользователями с использованием Node.js, Express и Sequelize. Полученный API обладает функциональностью создания, чтения, обновления и удаления пользователей, что позволяет эффективно управлять данными пользователей через HTTP запросы. Процесс разработки позволил понять основы работы с Node.js, фреймворком Express и ORM Sequelize, а также научиться создавать веб-приложения и API на их основе.