|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №4**

по дисциплине «Интерфейсы прикладного программирования»

**Тема практической работы:**

**Создание простейшего имитационного REST API с помощью NodeJS**

**Студент группы** ИКБО-01-21 Маров Герман Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** Зарипов Е.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

**ОГЛАВЛЕНИЕ**

[Цель работы 3](#_Toc115999382)

[Ход работы 3](#_Toc115999383)

[Вывод 6](#_Toc115999384)

# Цель работы

Целью данной практической работы является знакомство с принципами REST и создание простейшего имитационного REST API.

Задание:

1. Проверить, установлен ли NodeJS. В случае отсутствия установить с официального сайта разработчика.

2. По аналогии с вышеуказанным примером, необходимо, с помощью json-server, создать имитационный API для сервиса поиска книг в библиотеке. Для этого необходимо создать, как минимум три конечных точки: авторы, книги и библиотеки.

3. Протестировать работу с локальным сервисом, путём соответствующих запросов. Дать объяснение каждому запросу и возвращаемому ответу. Для этого желательно использовать curl или Postman.

# Ход работы

Node.js уже был установлен данном компьютере, чтобы убедиться в этом, введём соответствующую команду в консоль (рисунок 1). Далее в соответствии с заданием создадим базу данных в виде json-файла (рисунок 2). После этого мы можем установить node.js сервер через консоль, что можно наблюдать на рисунке 3. Затем необходимо подключить созданную БД к нашему серверу, то как это было сделано представлено на рисунке 4.

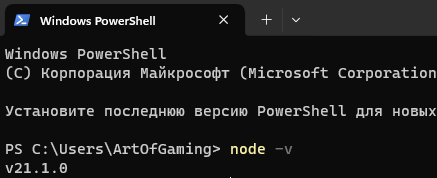


Рисунок 1 – Проверка версии установленного Node.js

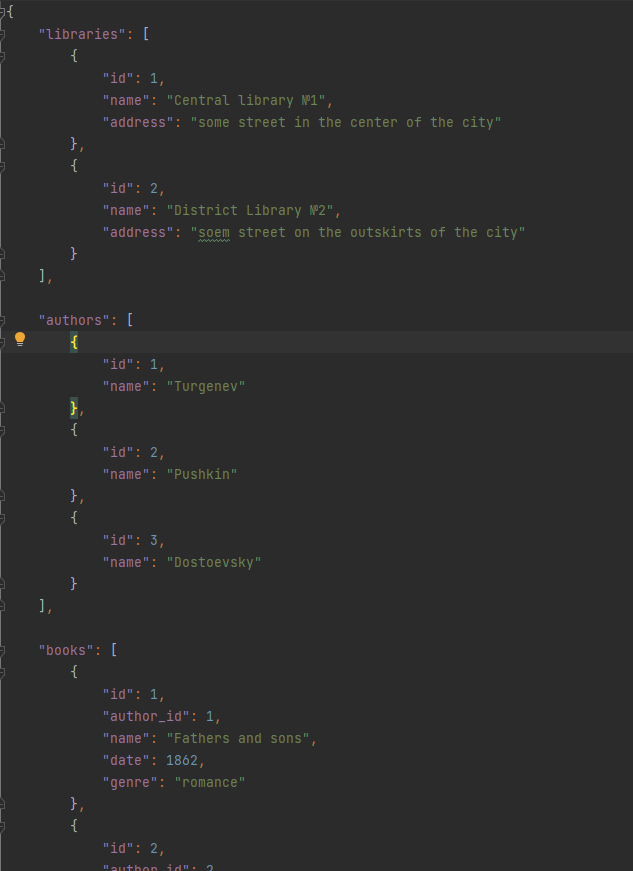


Рисунок 2 – Отрывок из созданной базы данных для будущего сервера

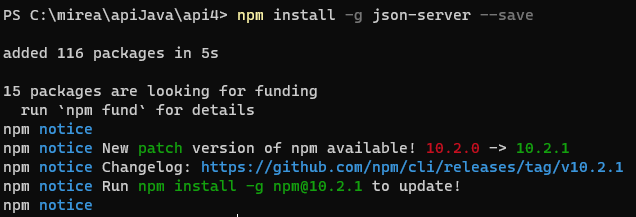


Рисунок 3 – Установка json-server

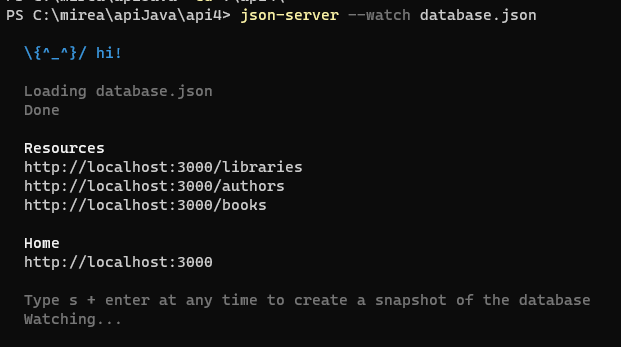


Рисунок 4 – Подключение базы данных к созданному json-server

После проведения всех вышеописанных манипуляций наш сервер полностью сконфигурирован и готов к работе. Давайте убедимся в этом и для этого выполним к нему ряд запросов через Postman (рисунки 5-9).

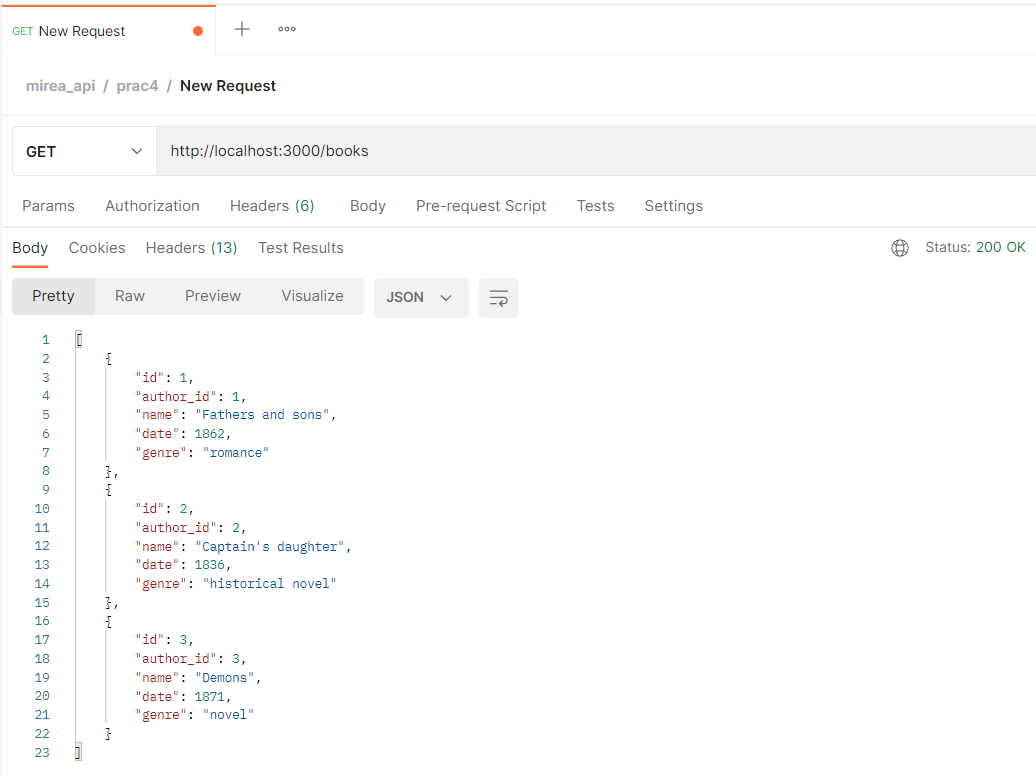


Рисунок 5 – Получение списка всех книг

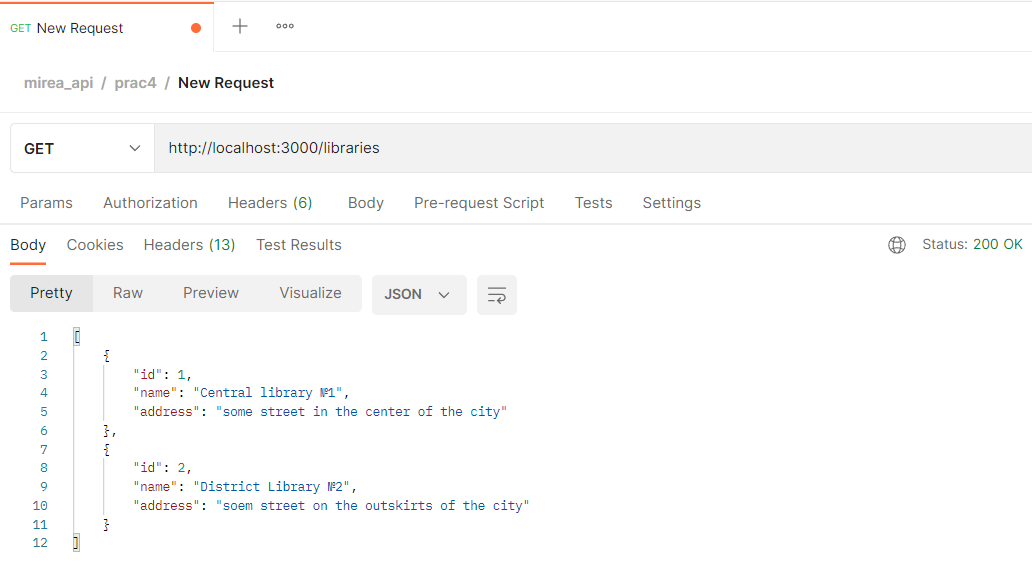


Рисунок 6 – Получение списка всех библиотек

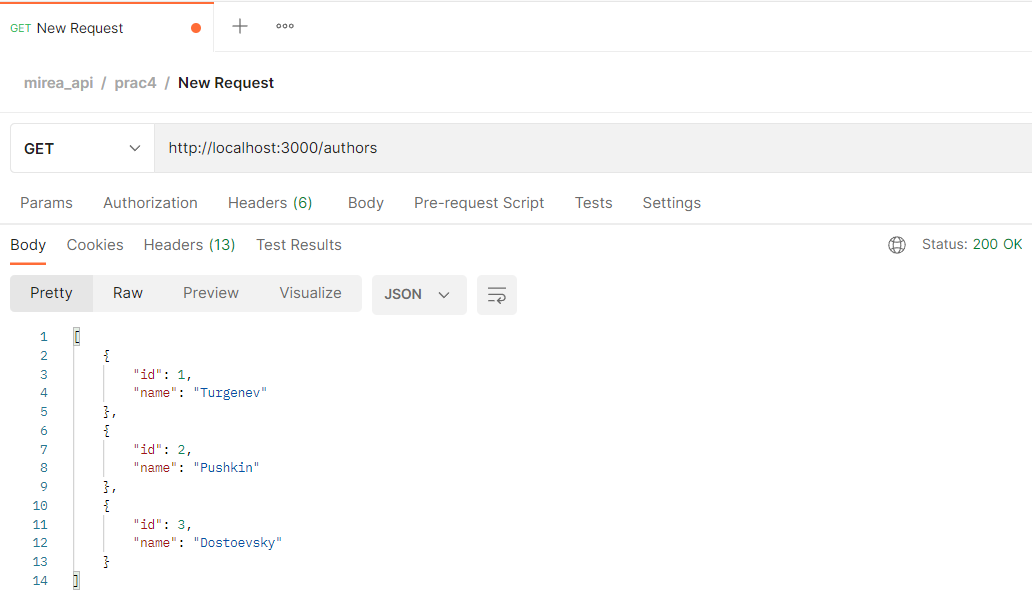


Рисунок 7 – Получение списка всех авторов

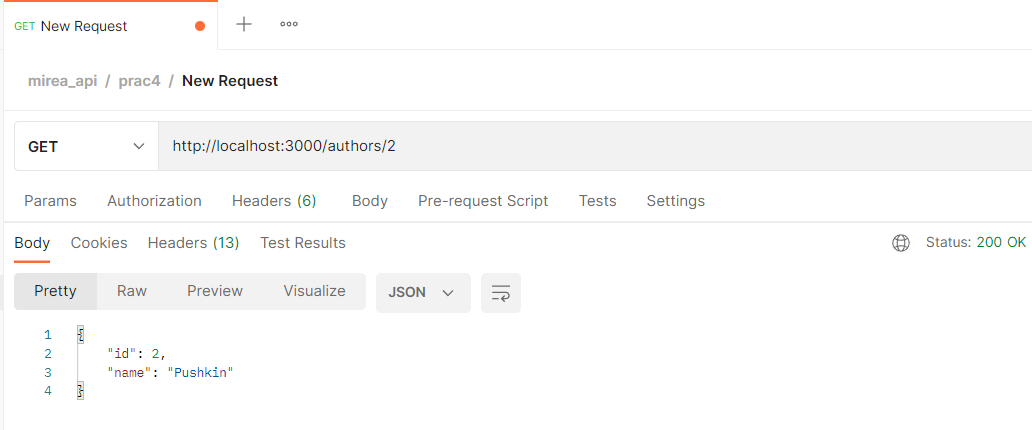


Рисунок 8 – Получение конкретного автора по id

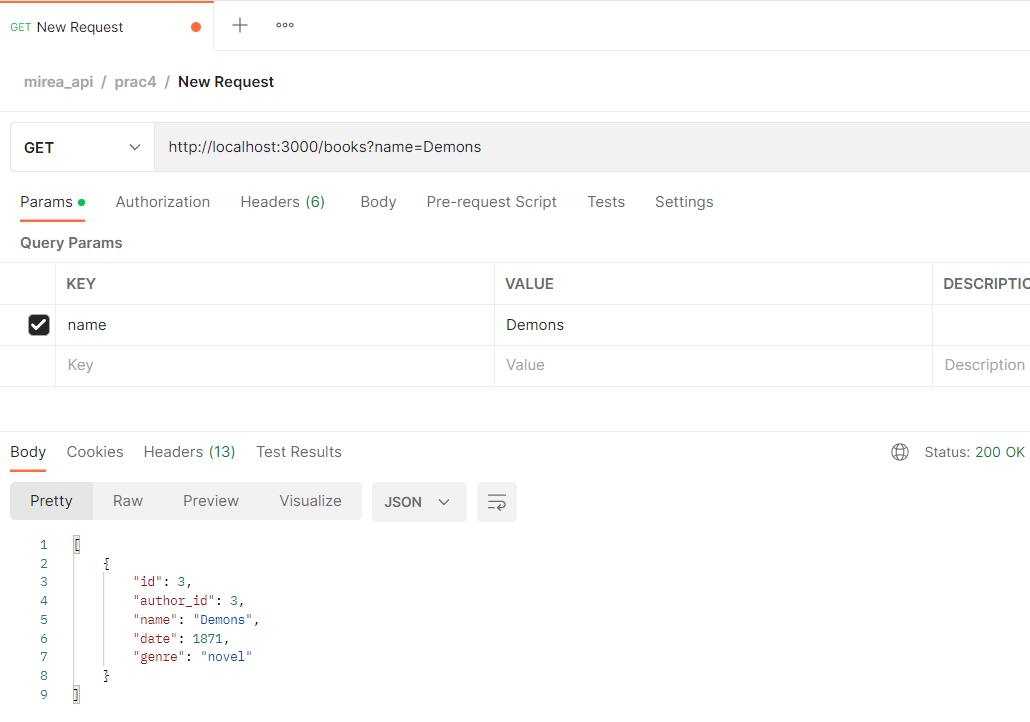


Рисунок 9 – Получение книги по названию

# Вывод

В результате выполнения данной практической работы были получены навыки по установке и работе с Node.js. Изучен формат json, составлено несколько файлов согласно заданию, использованы вложение нескольких объектов друг в друга. Создано свое API по теме «библиотека», протестированы с помощью Postman запросы на выборку, изменение, обновление. Углублено понимание HTTP-запросов и REST-архитектуры.