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 «Проектирование и разработка серверных частей интернет-ресурсов»

Студент группы ИКБО-21-23 Муравьев А.О.

_____

(подпись студента)

Руководитель практической работы Благирев М.М.

_____

(подпись руководителя)

Работа представлена «___»_____2025 г.

Допущен к работе «___» _____2025 г.

Москва 2025

СОДЕРЖАНИЕ

ЦЕЛЬ РАБОТЫ

Предполагается реализация интерфейса прикладного программирования для доступа к некоторым данным по варианту. Предполагается реализация серверной части обработки запросов и тестирование данного интерфейса с использованием программы Postman. Для реализации данного сервиса предлагается использовать серверную конфигурацию, модернизированную в течение первых трех практических работ. Важной частью данной практической работы является сохранение функциональности реализованной в практической работе №3. То есть интерфейс предлагается создать уже в существующем веб-приложении. Также предполагается использование темы практической работы №3 для продолжения модернизирования собственной системы. Изменение темы согласовывается отдельно с преподавателем. Хранение данных предполагается уже в существующей базе данных.

Тема проекта: Сервис для контроля выполнения задач.

Парадигма API: REST.

ХОД РАБОТЫ

Структура веб-приложения

Проект написан на ASP.NET. Использует СУБД PostgreSQL для хранения данных. Roik для преобразования данных. Swagger для тестирования API. Entity Framework для работы с реляционной базой данных. MediatR для выполнения шаблона «посредник» и др.

Реализация веб-приложения

Для реализации CRUD были созданы команды для работы с бордами (Board), показанные на рисунке 1.

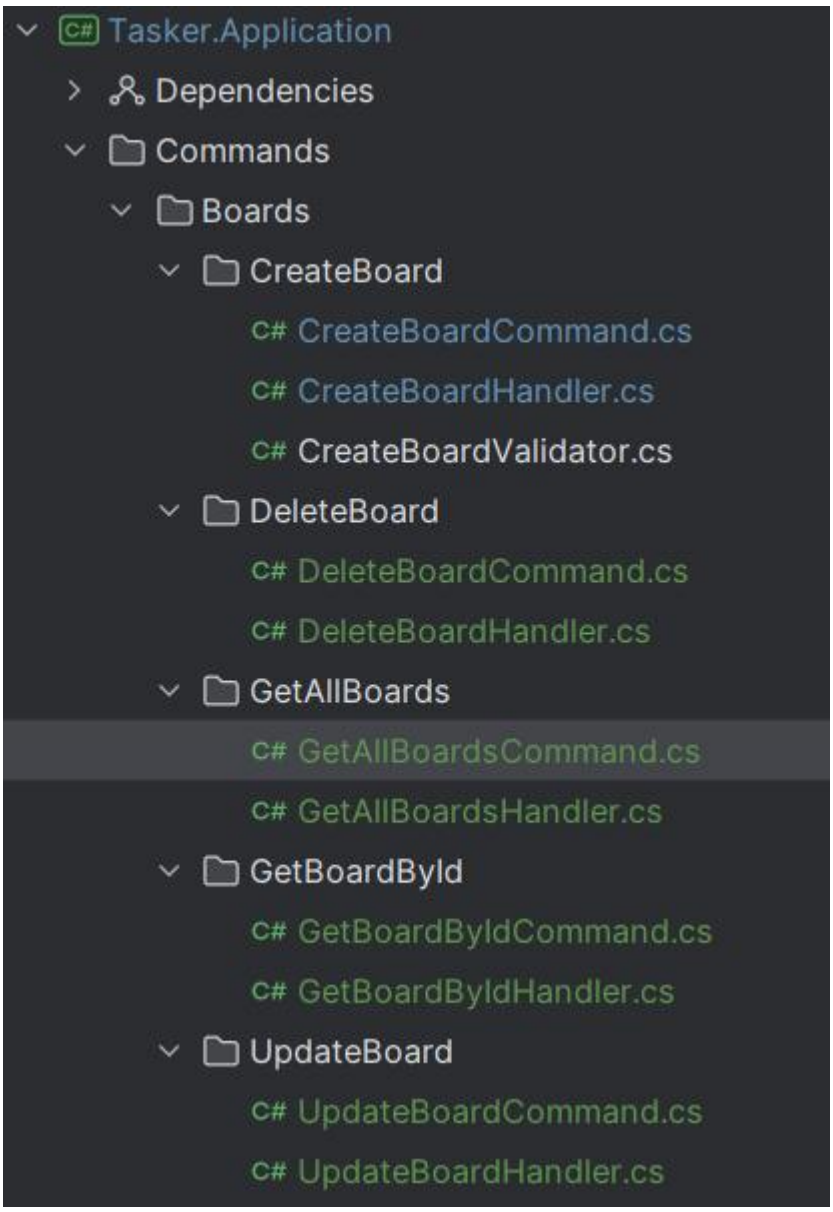

Рисунок 1 – Файлы созданных команд

Каждая команда состоит из Command-файла (объявление команды), Handler-файла (описание обработки его вызова) и может иметь Validator-файл (для валидации запроса).

Каждый вид запроса необходимо утвердить в MediatR и в самом ASP.NET. Для этого нужно создать API-контроллер и его основу (рисунок 2).

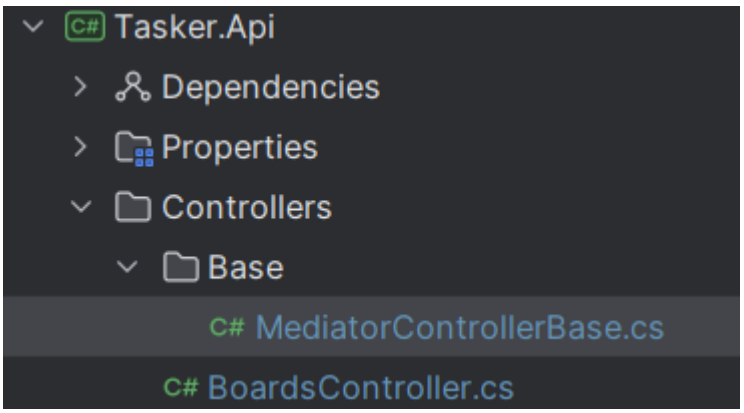

Рисунок 2 – Файлы контроллеров

Все команды регистрируются в контроллере, что показано на рисунке 3.

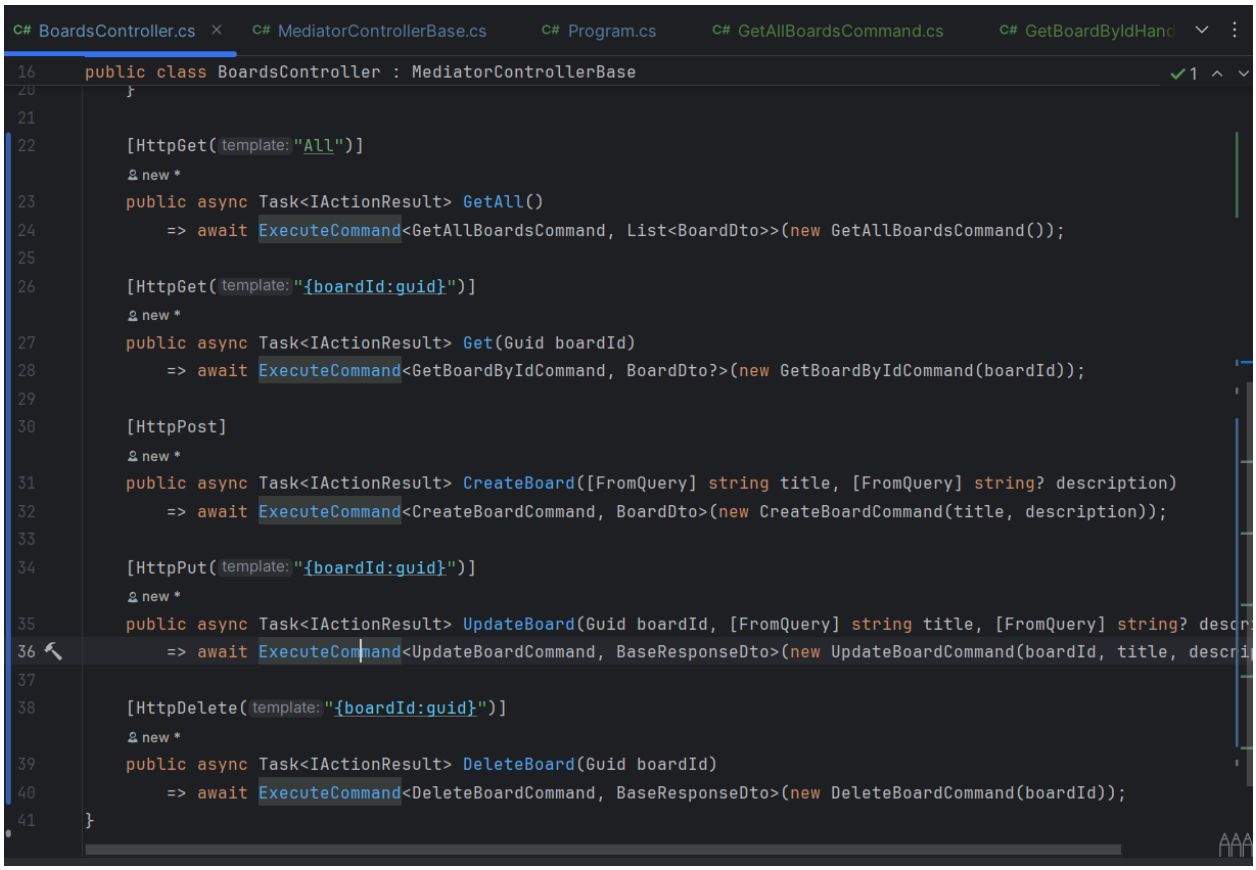

Рисунок 3 – Регистрация команд в контроллере

Тестирование веб-приложения

Для тестирования API был использован Swagger UI.

На рисунке 4 показан результат GET запроса по адресу /Api/Boards/All.

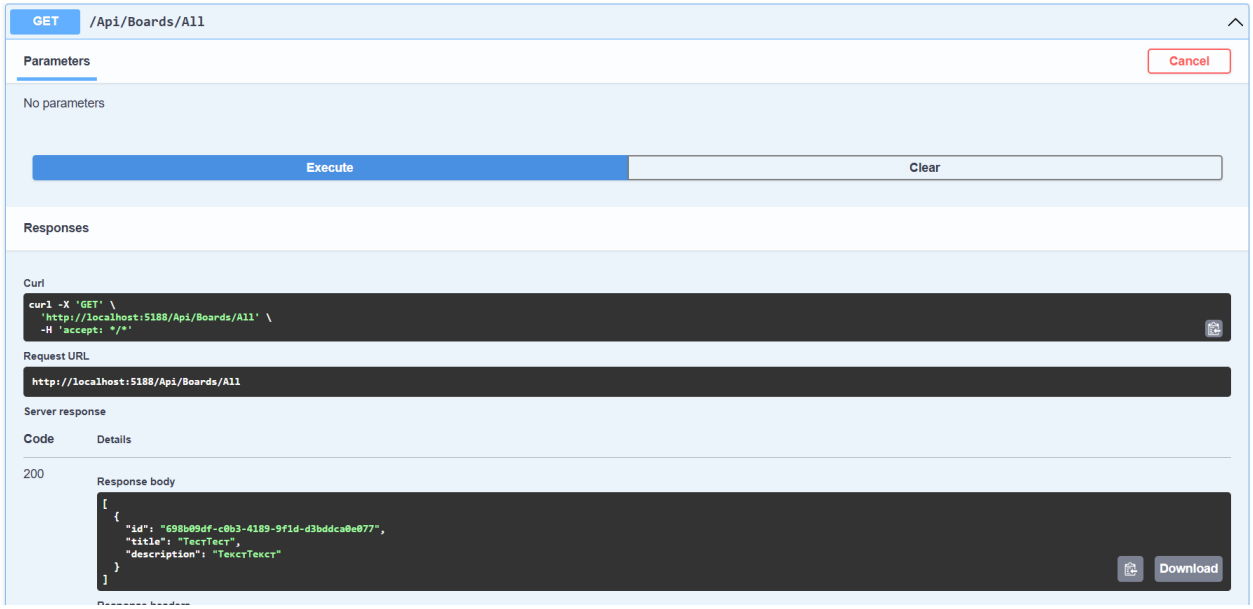

Рисунок 4 – Запрос всех Boards

На рисунке 5 показан результат запроса по адресу /Api/Boards/{boardId}.

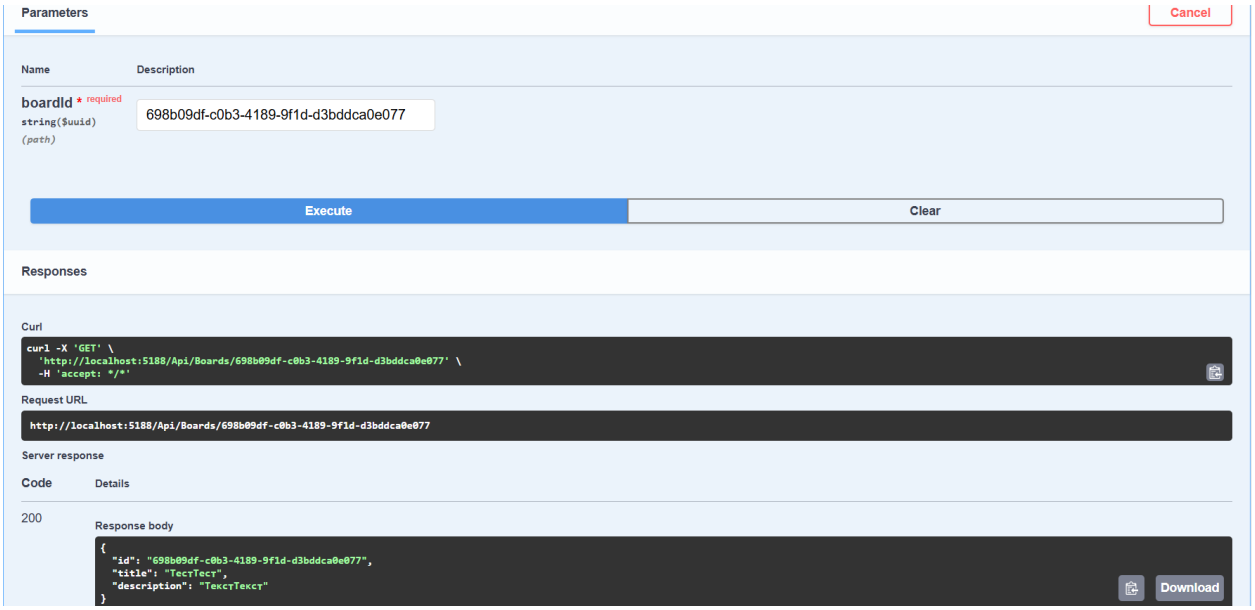

Рисунок 5 – Запрос определённого Board

На рисунке 6 показан результат POST запроса по адресу /Api/Boards/.

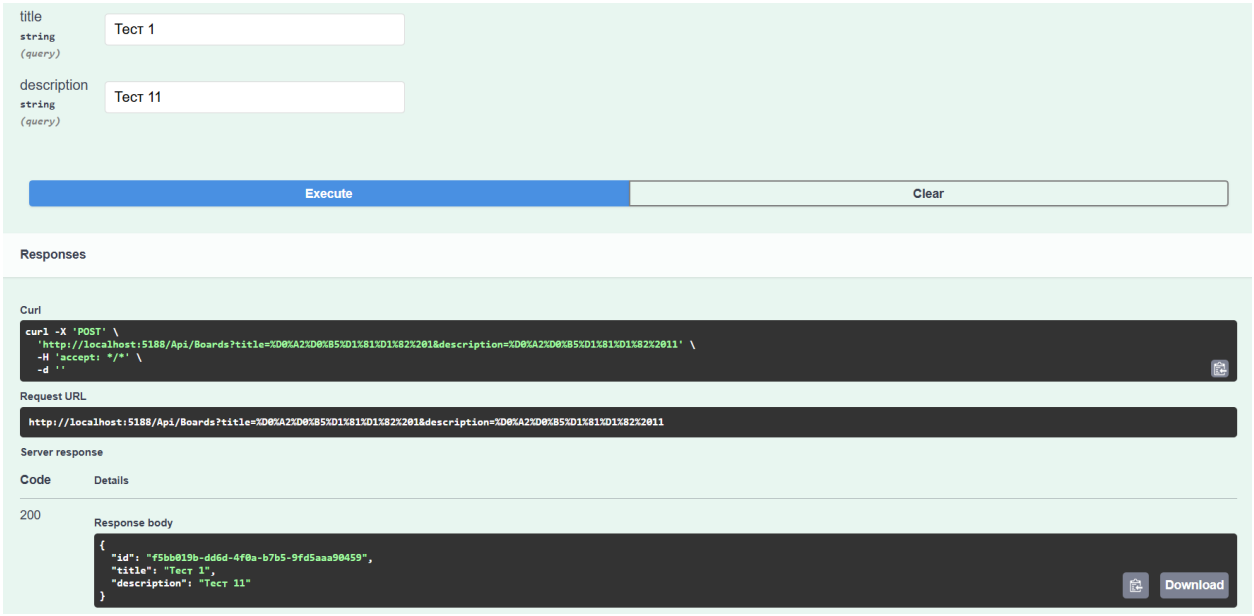

Рисунок 6 – Запрос на создание нового Board

На рисунке 7 показан результат PUT запроса по адресу /Api/Boards/{boardId}.

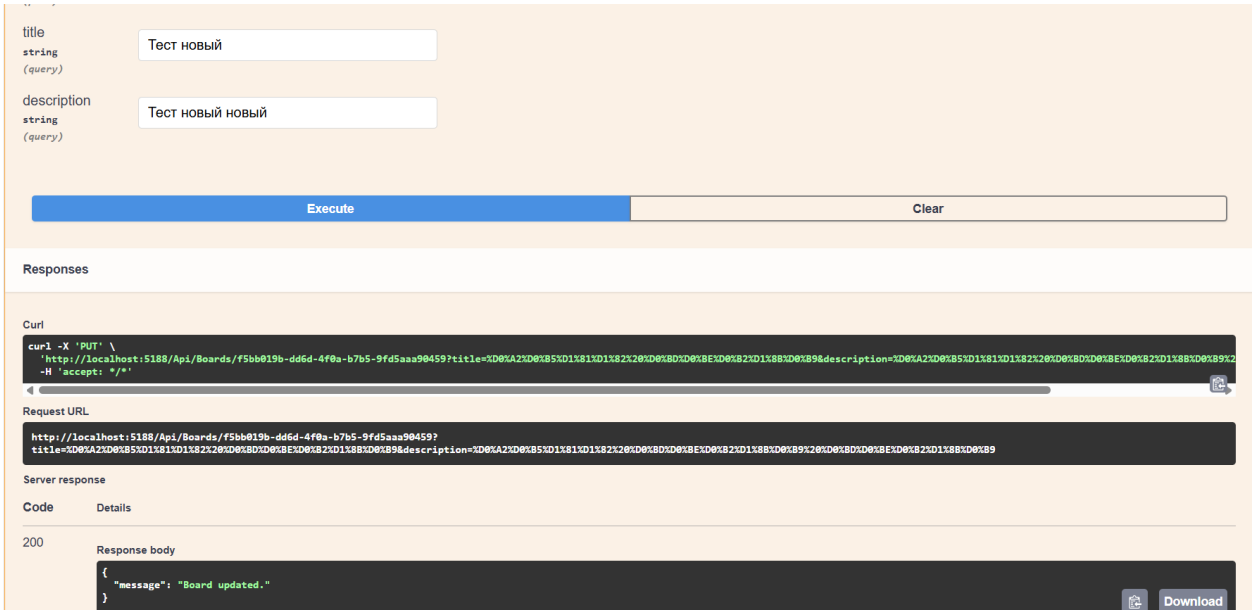

Рисунок 7 – Обновление существующего Board

Проверка изменения после PUT запроса показана на рисунке 8.

```
[
  {
    "id": "698b09df-c0b3-4189-9f1d-d3bddca0e077",
    "title": "ТестТест",
    "description": "ТекстТекст"
  },
  {
    "id": "f5bb019b-dd6d-4f0a-b7b5-9fd5aaa90459",
    "title": "Тест новый",
    "description": "Тест новый новый"
  }
]
```


Рисунок 8 – Демонстрация изменения Board

На рисунке 9 показан результат отправки DELETE запроса по адресу /Api/Boards/{boardId}.

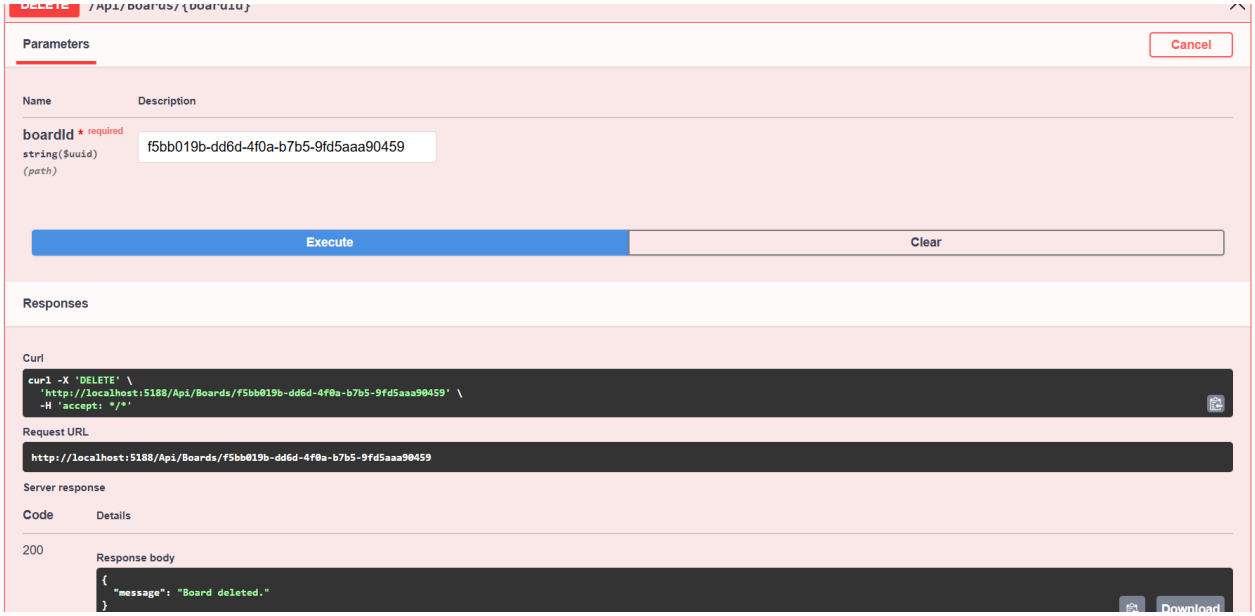

Рисунок 9 – Удаление Board

На рисунке 10 показан результат выполнения DELETE запроса.

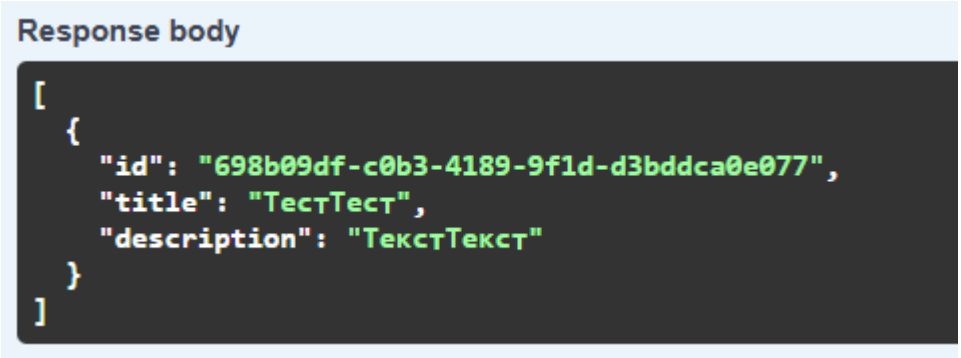

Рисунок 10 – Демонстрация удаления Board

ВЫВОД

Таким образом, был разработан API для описанного сервиса.

Исходный код проекта расположен по адресу:

https://github.com/alexomur/Tasker