Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №3

по дисциплине "Технологии программирования"

ТЕМА РАБОТЫ:

Разработка микросервиса как элемента микросервисной архитектуры

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

Ассистент кафедры ИСПИ

Данилов В.В.

Владимир 2022 г.

Цель работы:

Познакомиться с микросервисами, как с базовыми элементами микросервисных архитектур и освоить принципы разработки микросервиса на платформе .NET

Выполненине работы:

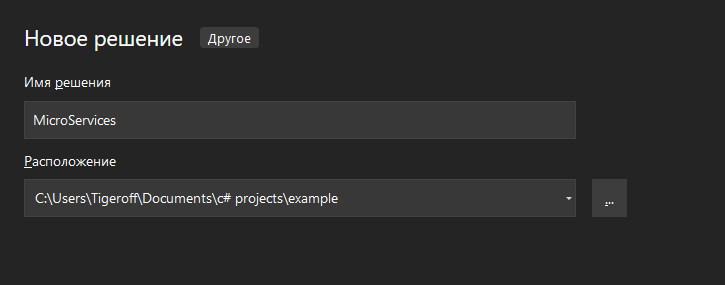


Рисунок 1. Создание нового решения MicroServices

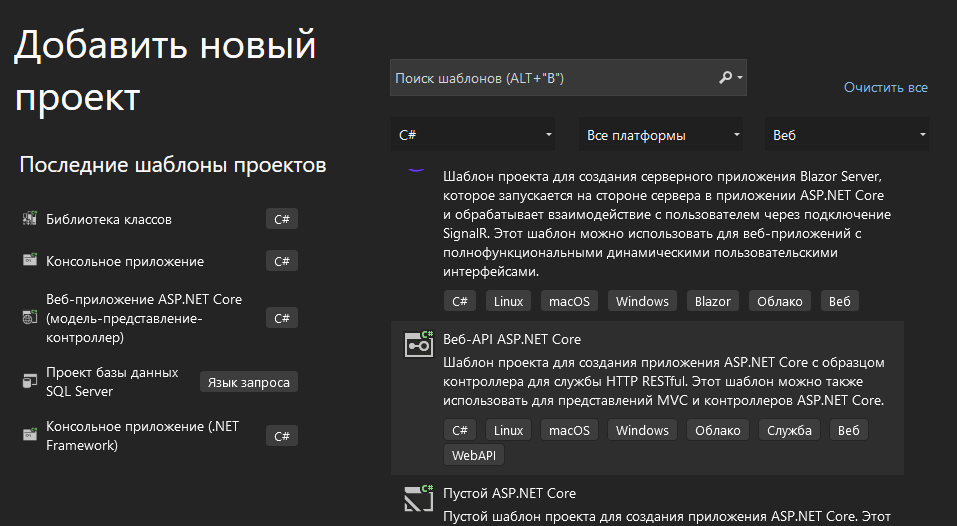


Рисунок 2. Добавление нового проекта шаблона ASP.NET Core

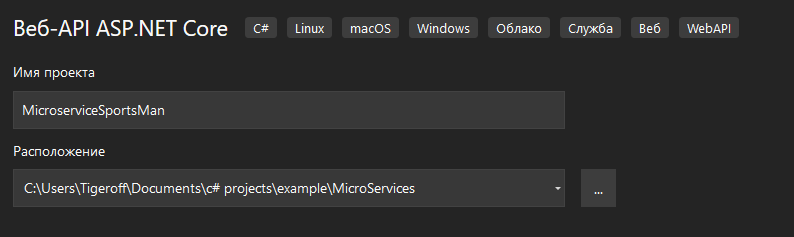


Рисунок 3. Проект 1 под названием MicroServiceSportsMan

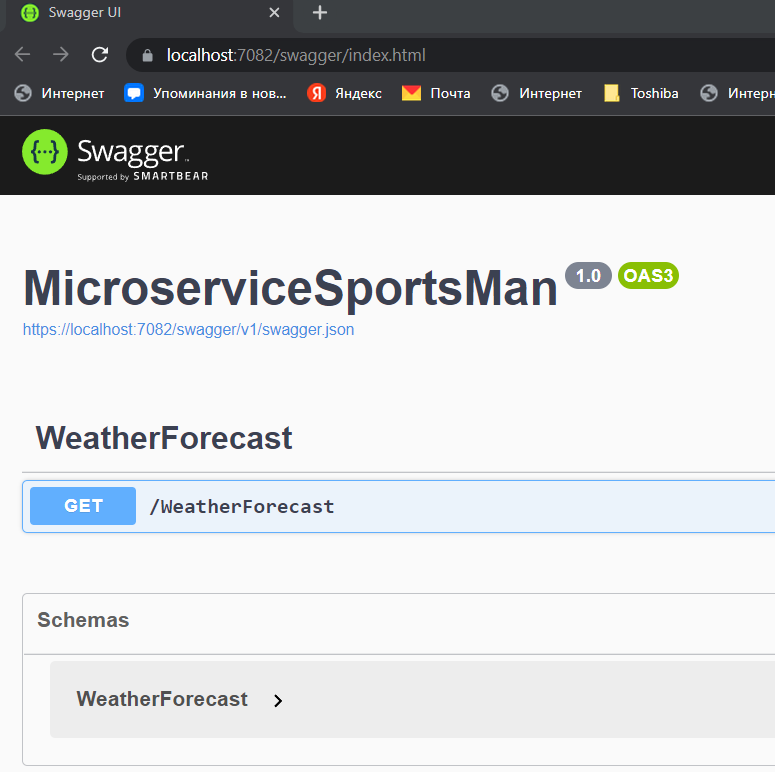


Рисунок 4. Запуск созданного проекта

(с заранее настроенным SSL-сертификатом)

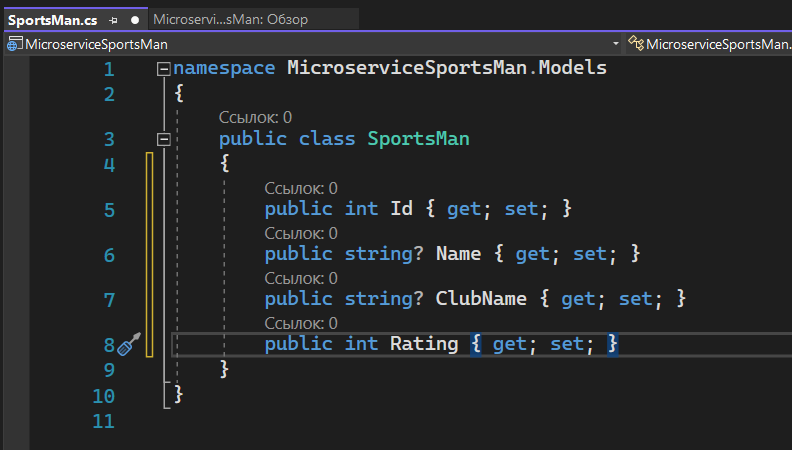


Рисунок 5. Создание модели SportsMan

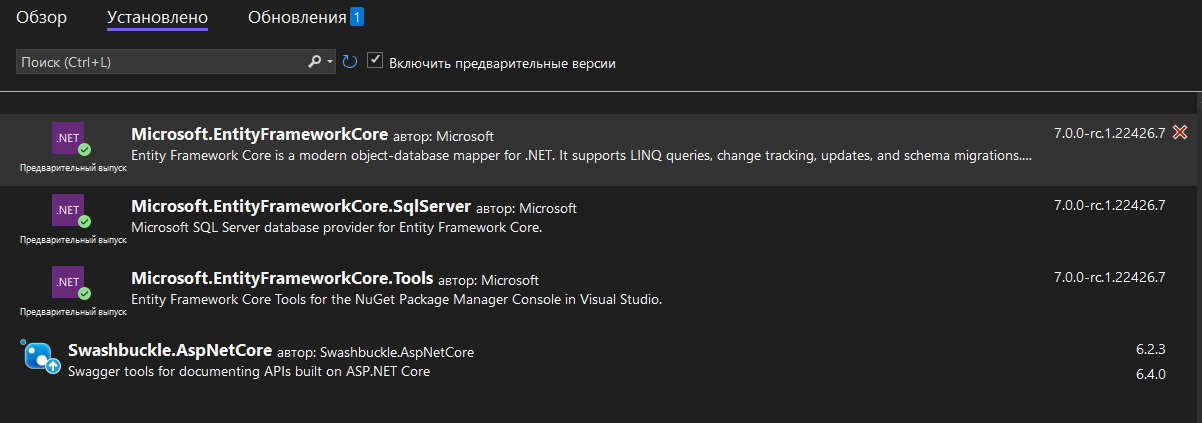


Рисунок 6. Установка пакетов NuGet для проекта

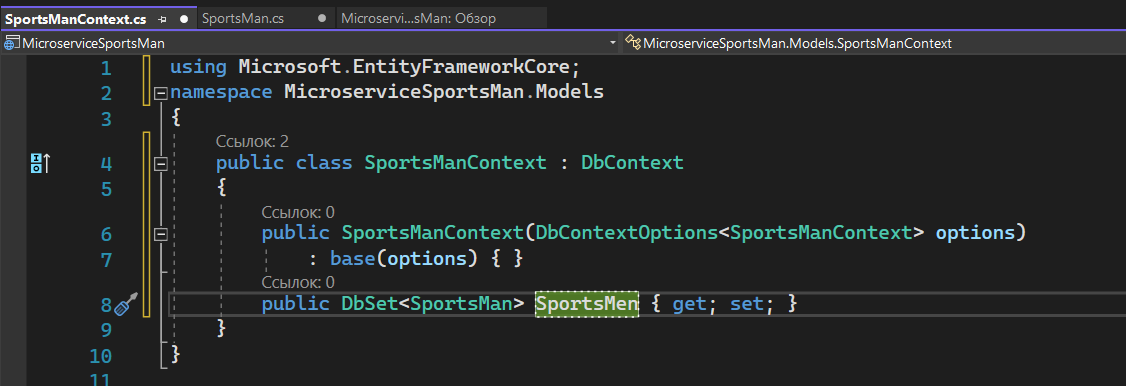


Рисунок 7. Создание контекста для модели SportsMan

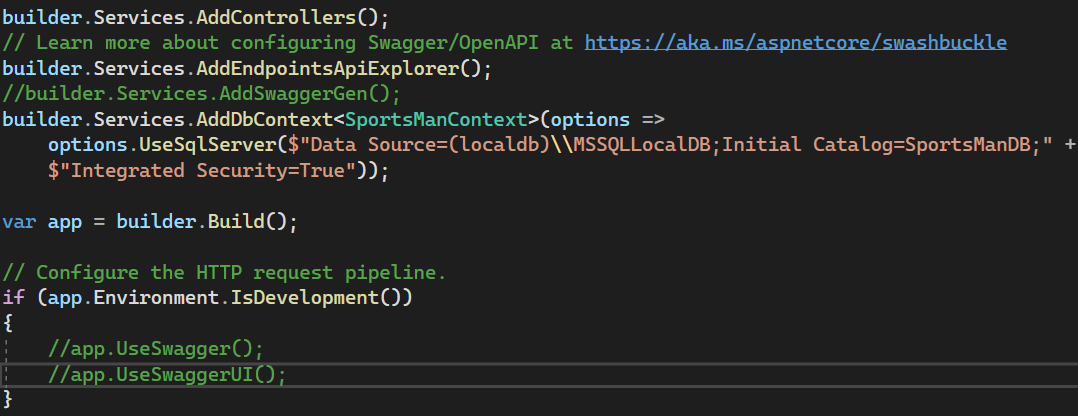


Рисунок 8. Комментирование ненужных строк и добавление контекста в сервисы в файле Program.cs

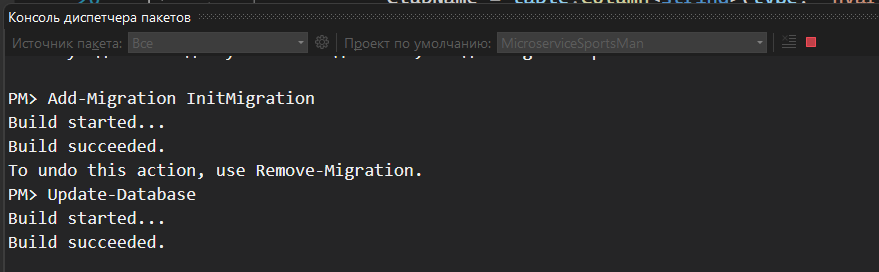


Рисунок 9. Выполнение миграции и создание базы данных EF

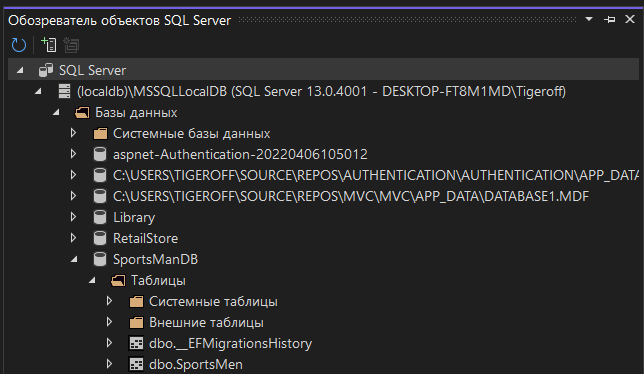


Рисунок 10. Появление созданной БД в обозревателе объектов SQL Server

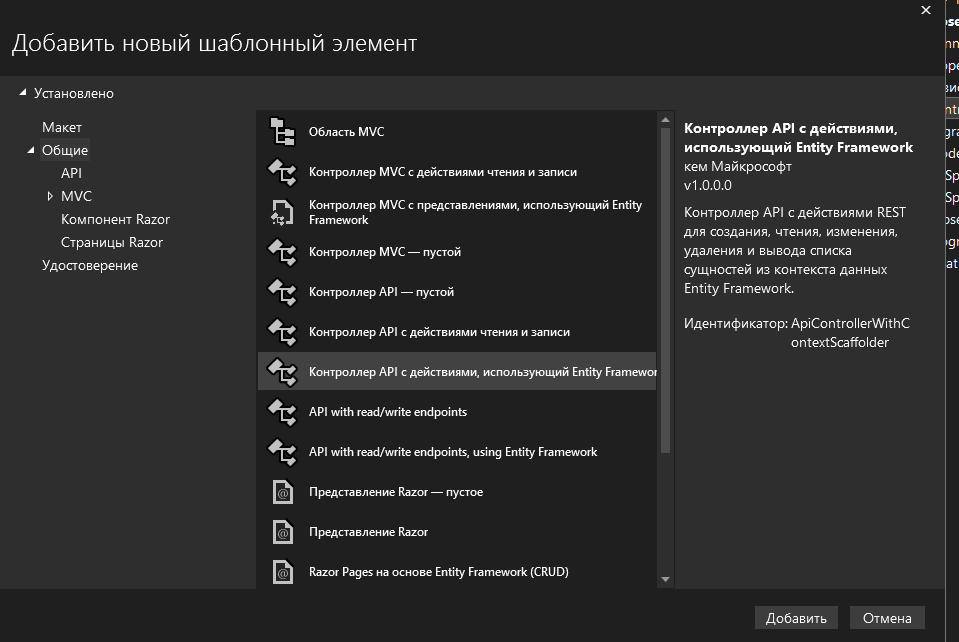


Рисунок 11. Добавление нового контроллера для данного микросервиса MicroserviceSportMan

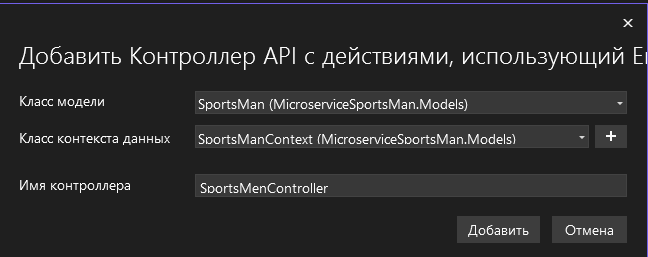


Рисунок 12. Заполнение информации о классах модели и контекса данных контроллера SportMenController

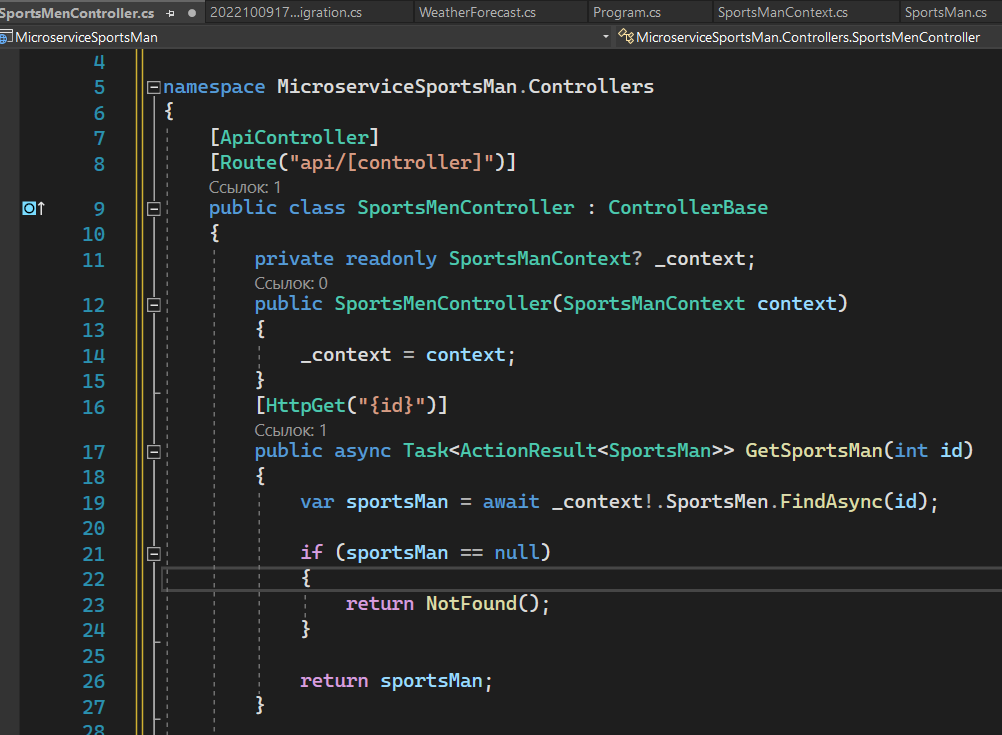


Рисунок 13. Листинг метода HTTP-GET контроллера SportsMenController



Рисунок 14. Замена launchUrl в файле launchSettings.json

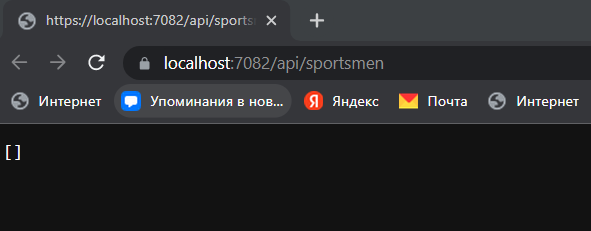


Рисунок 15. Запуск приложения по пути /api/sportsmen

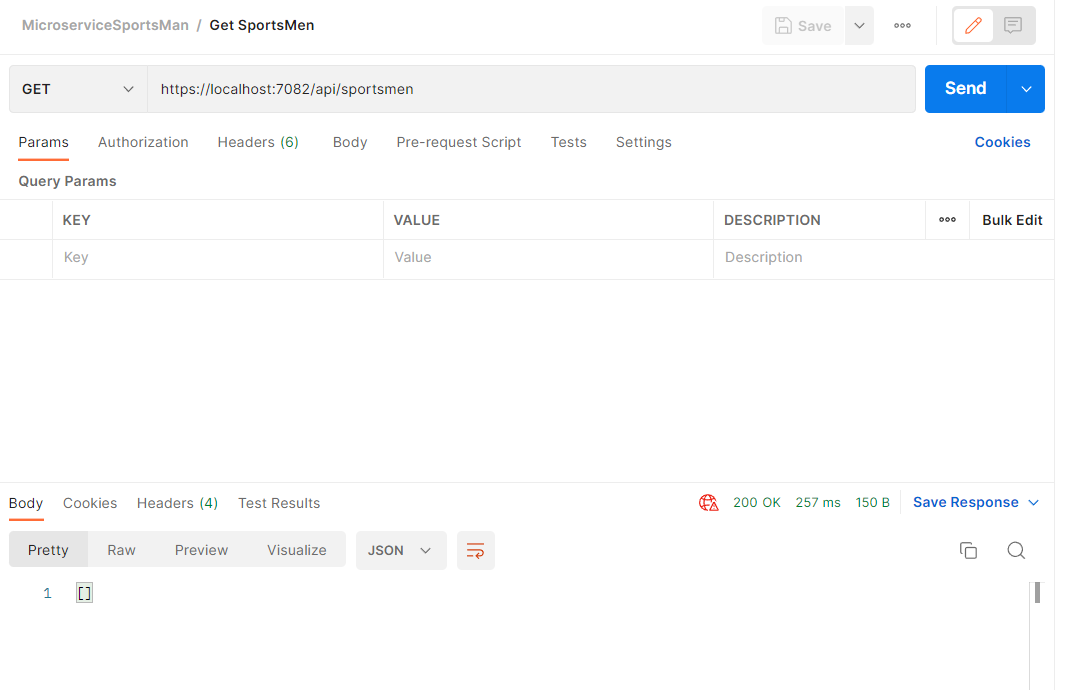


Рисунок 16. Получение ответа на GET - запрос в приложении Postman

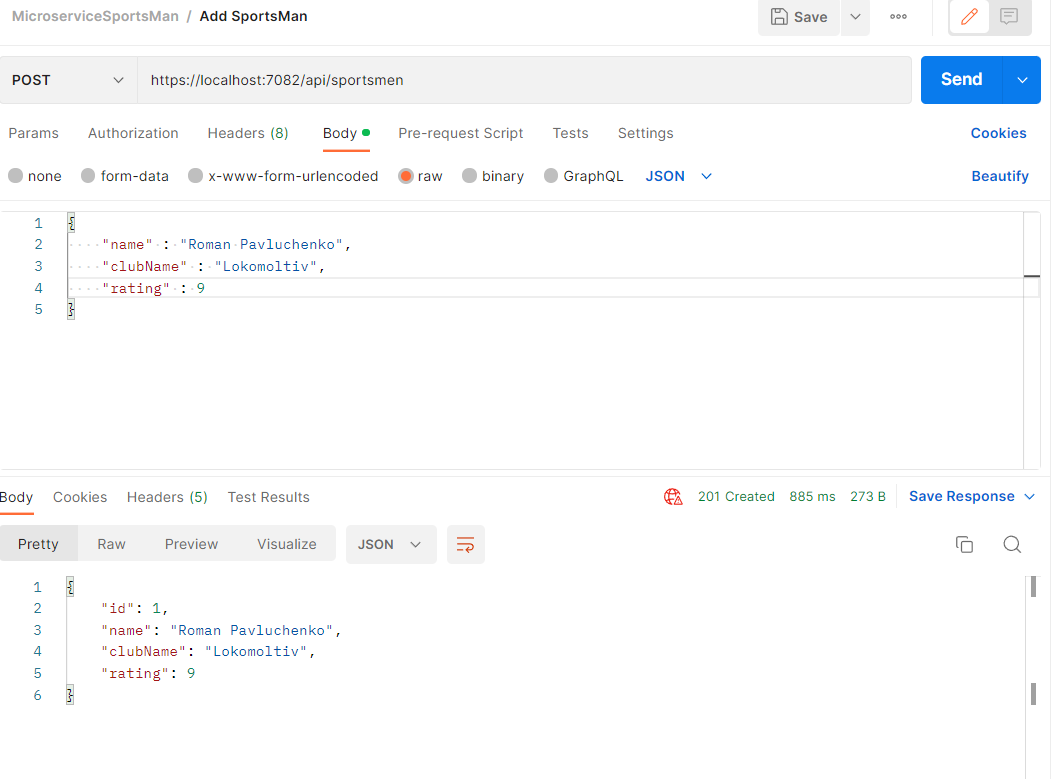


Рисунок 17. Результат добавление спортсмена в БД с помощью POST – запроса

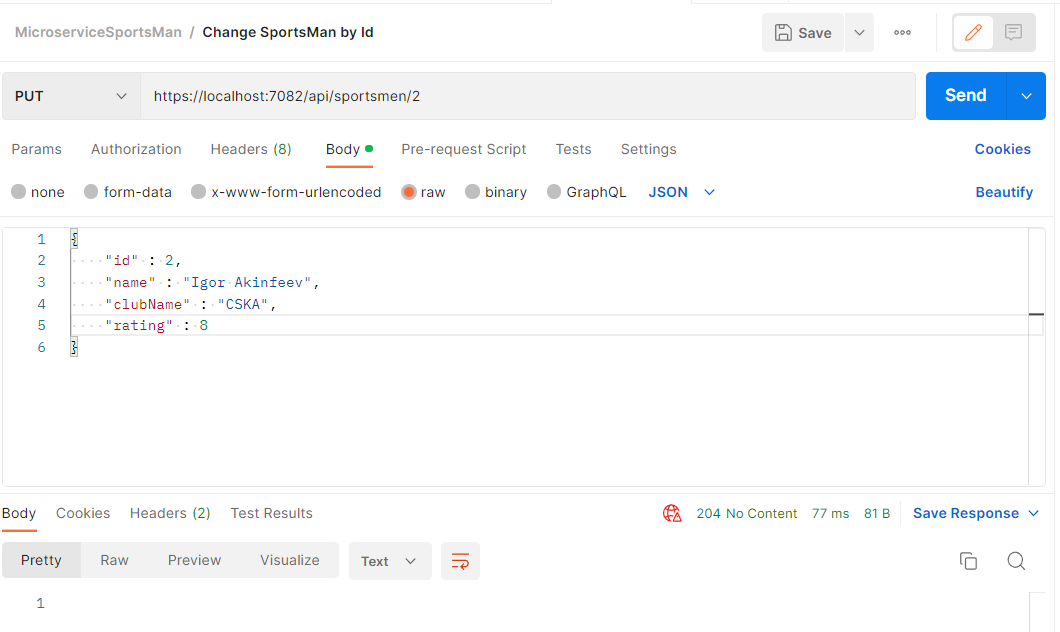


Рисунок 18. Результат выполнения PUT – запроса на изменение студента

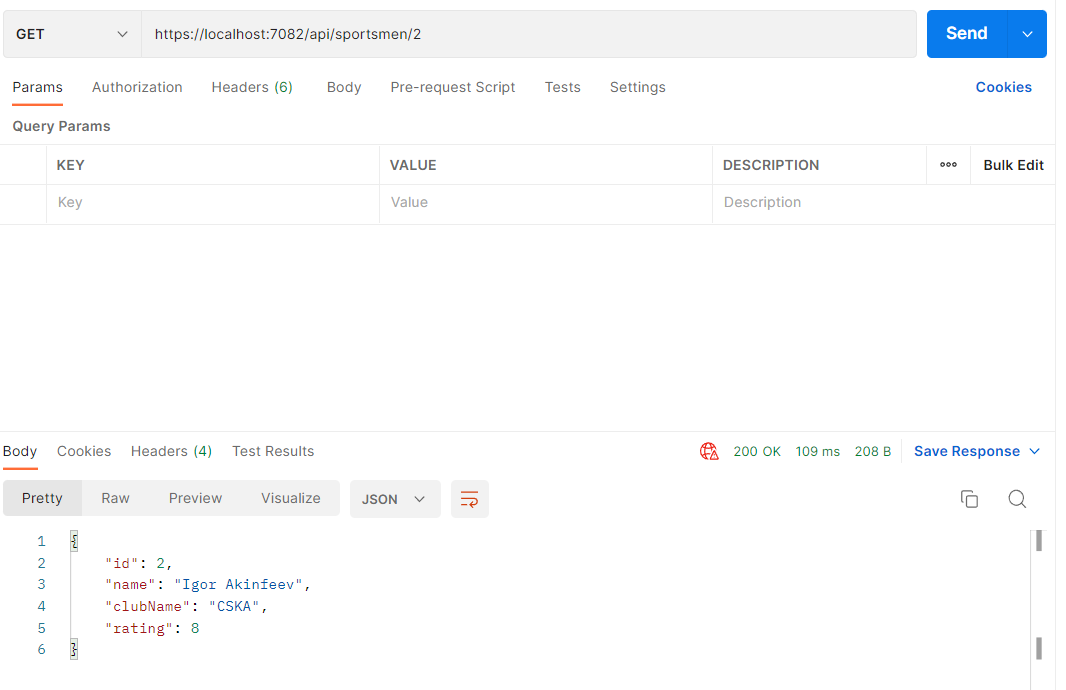
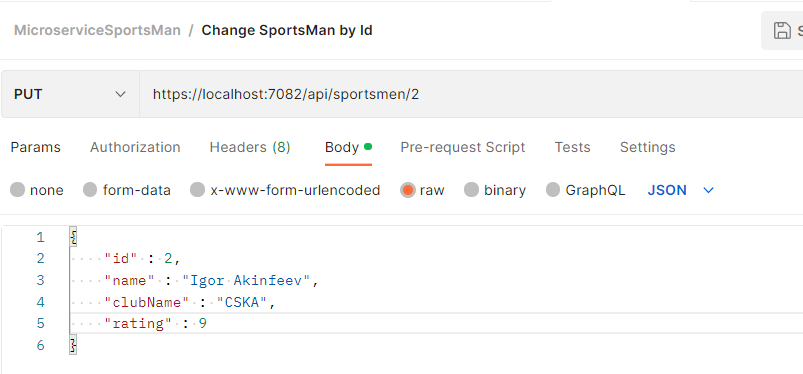


Рисунок 19. Проверка выполнения PUT – запроса при помощи измененного GET – запроса



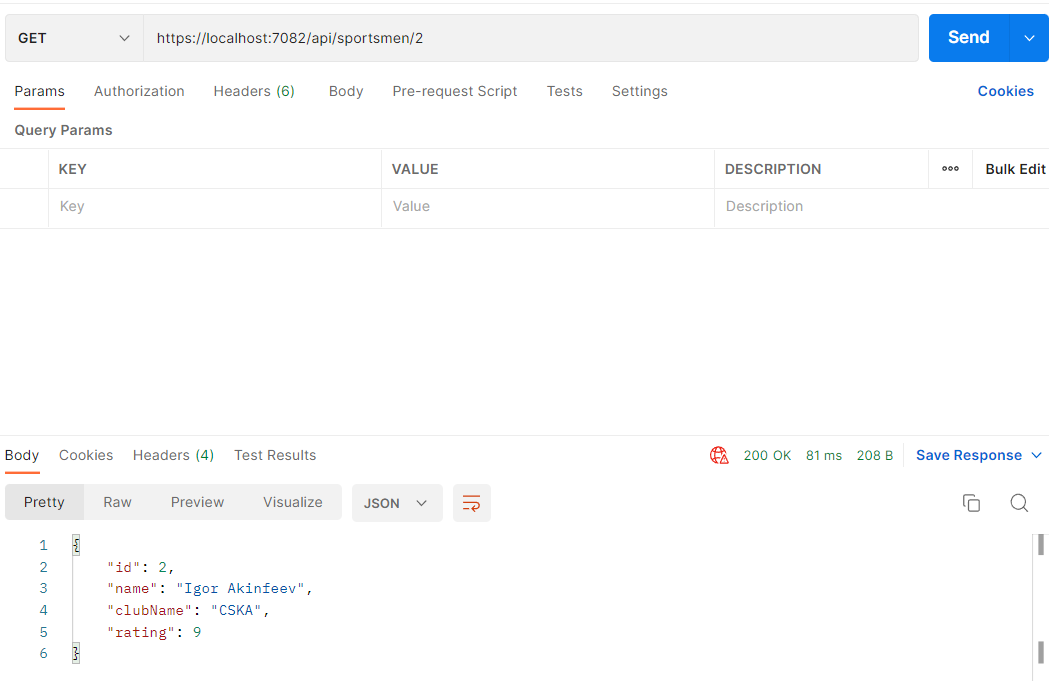
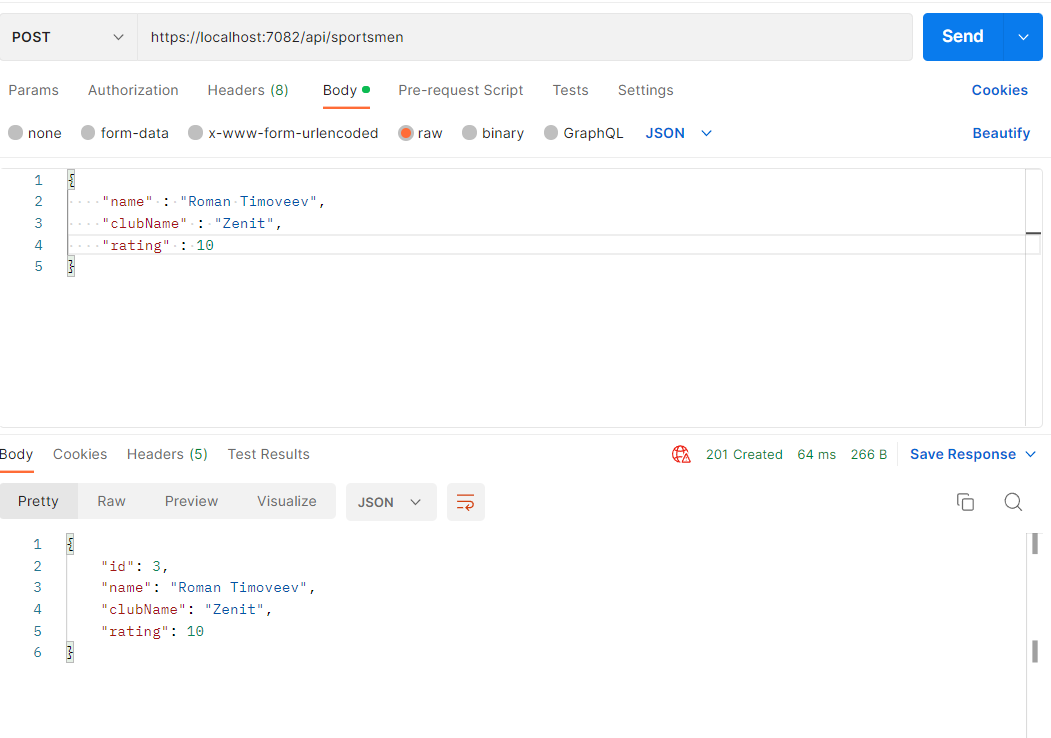
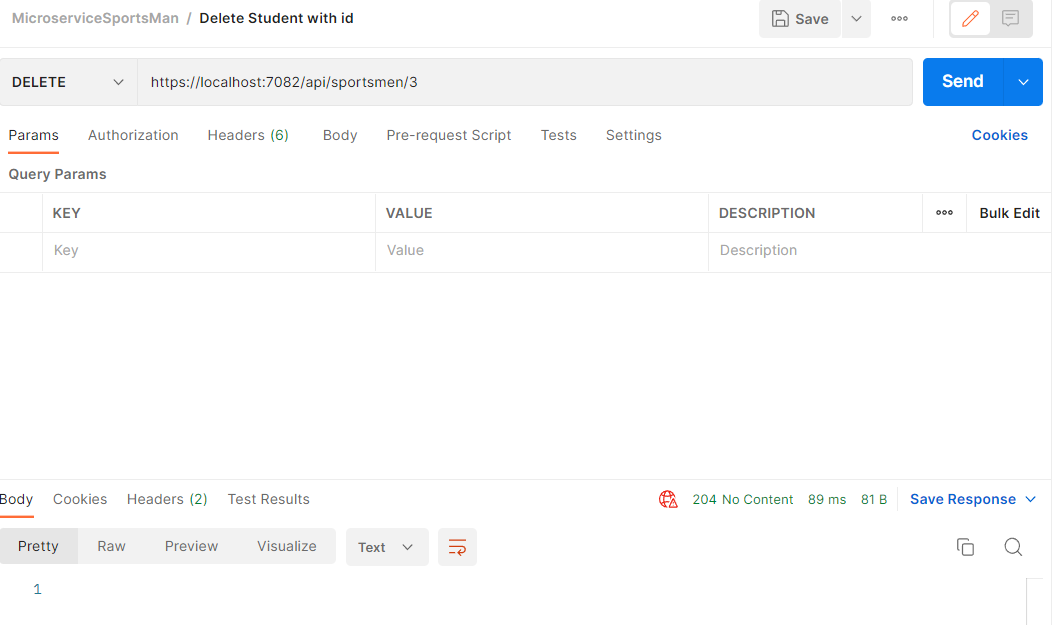


Рисунок 20. Результат PUT – запроса на изменение рейтинга спортсмена с id = 2





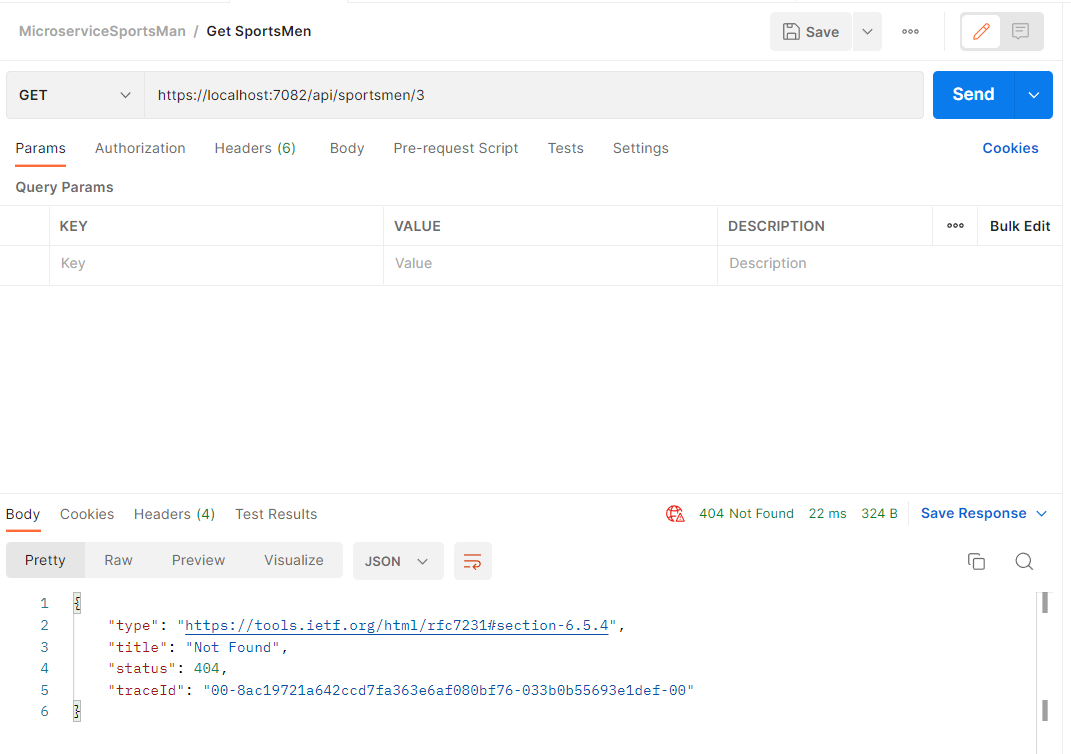


Рисунок 21. Проверка выполнения DELETE – запроса на удаление спортсмена

с id = 3

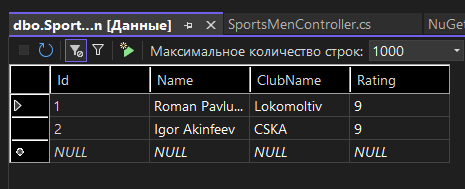


Рисунок 22. Содержимое бд EF

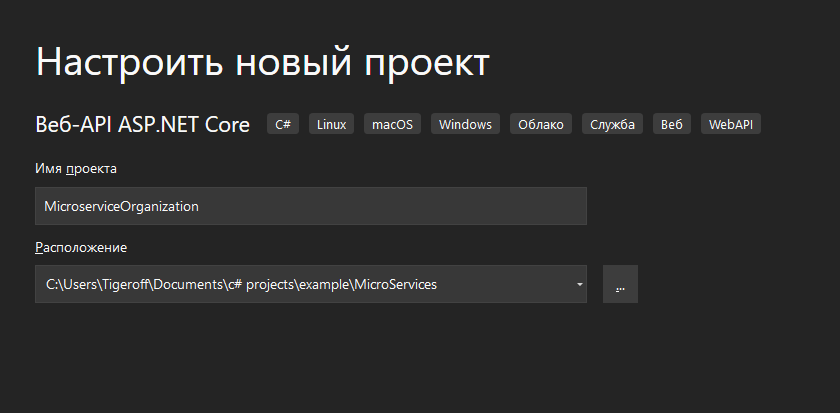


Рисунок 23. Добавление нового проекта MicroserviceOrganization в решение

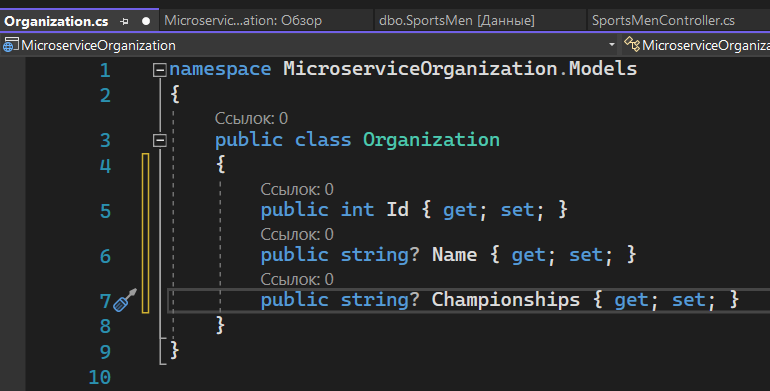


Рисунок 24. Создание модели Organization

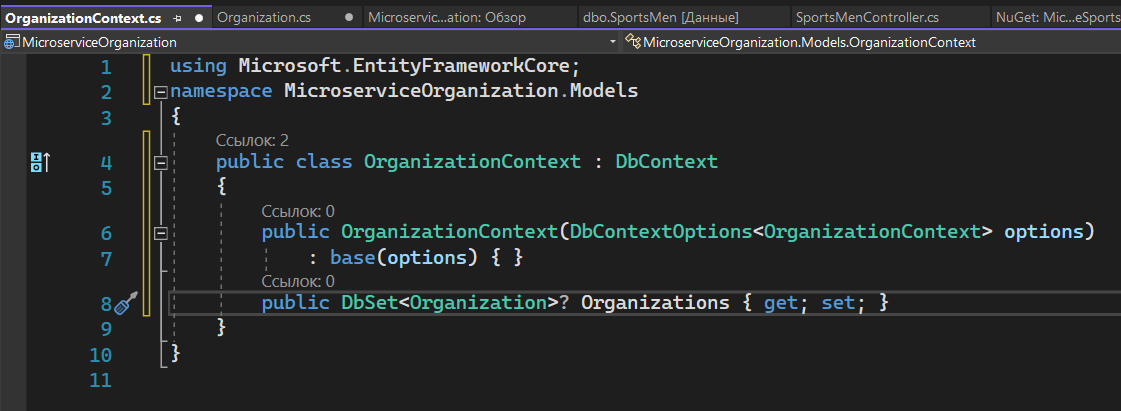


Рисунок 25. Создание контекста данных OrganizationContext

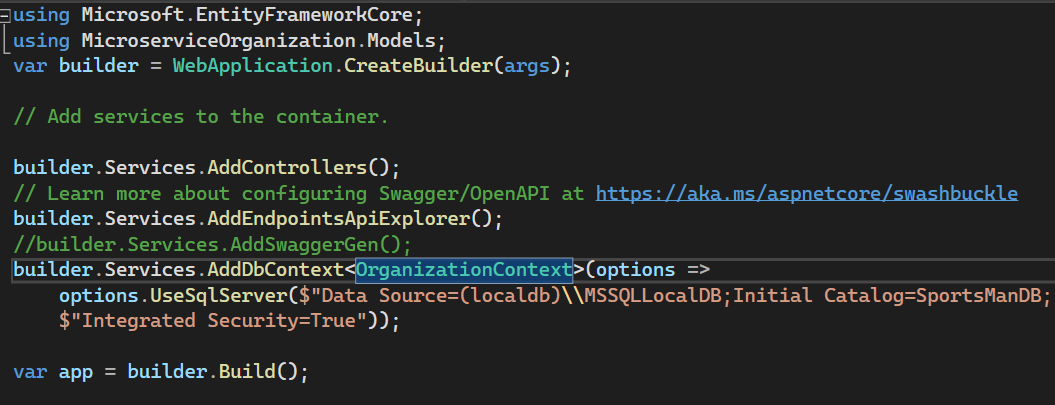


Рисунок 26. Изменение Program.cs нового проекта

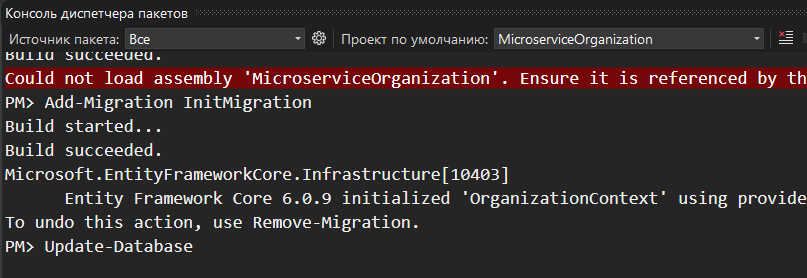


Рисунок 27. Добавление стартовой миграции в проекте MicroserviceOrganization

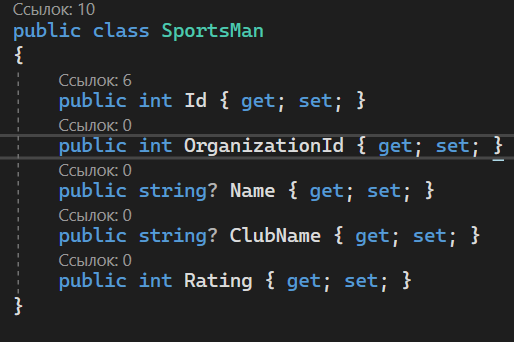


Рисунок 28. Добавление нового поля OrganiztionId в модель SportsMan

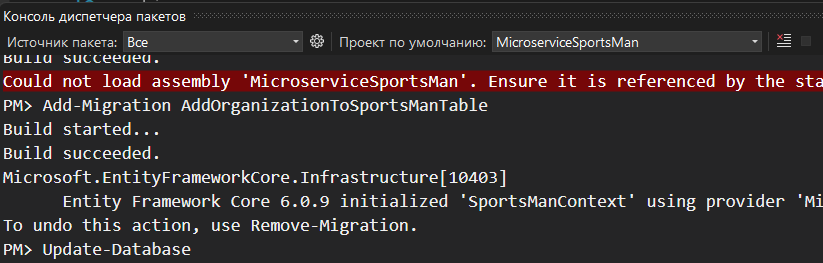


Рисунок 29. Добавление новой миграции в проект MicroserviceSportsMan

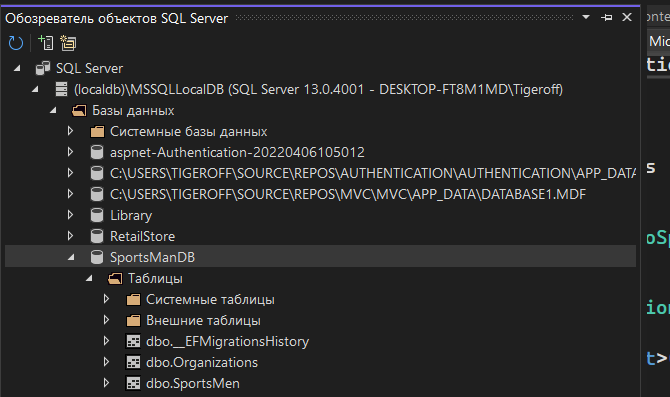


Рисунок 30. Изменение общей для двух сервисов БД (SportsManDB)

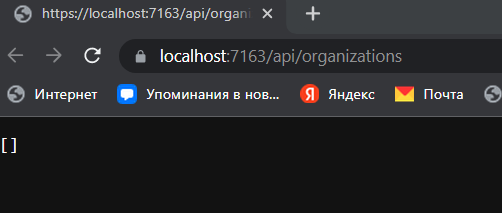
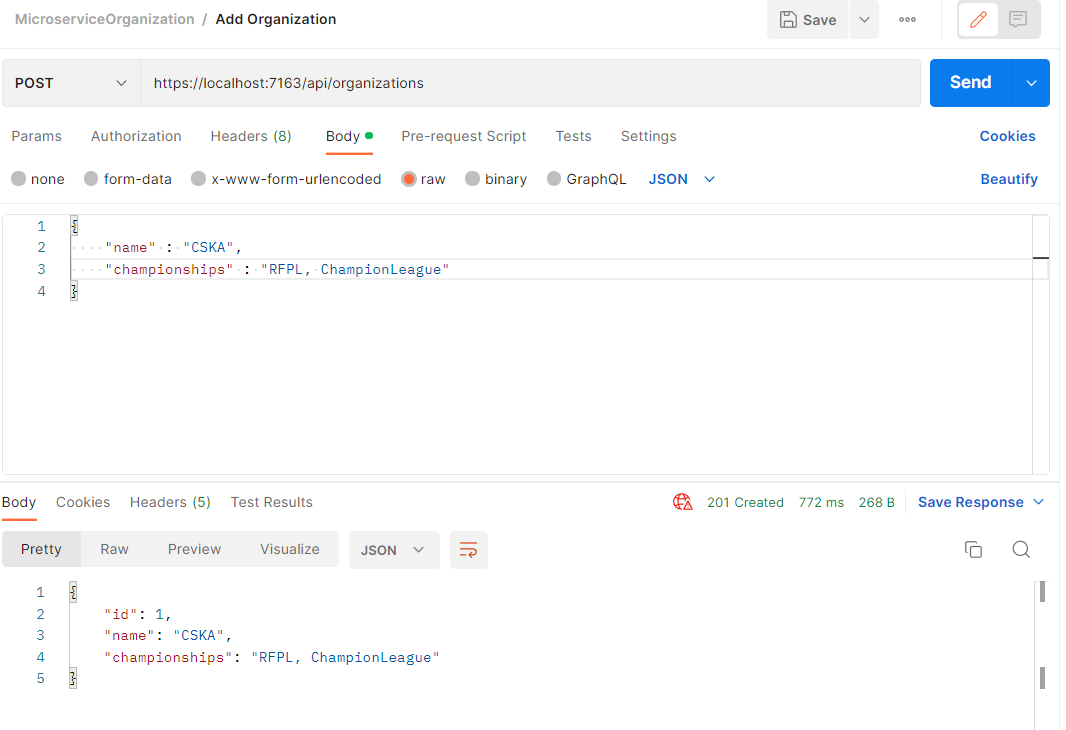
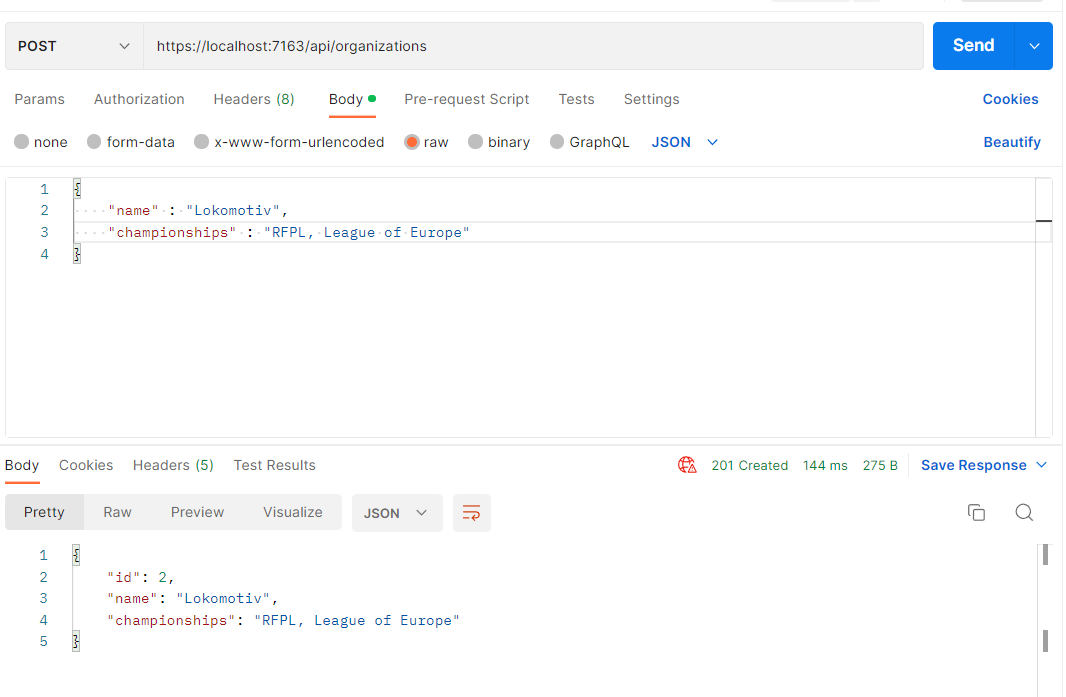
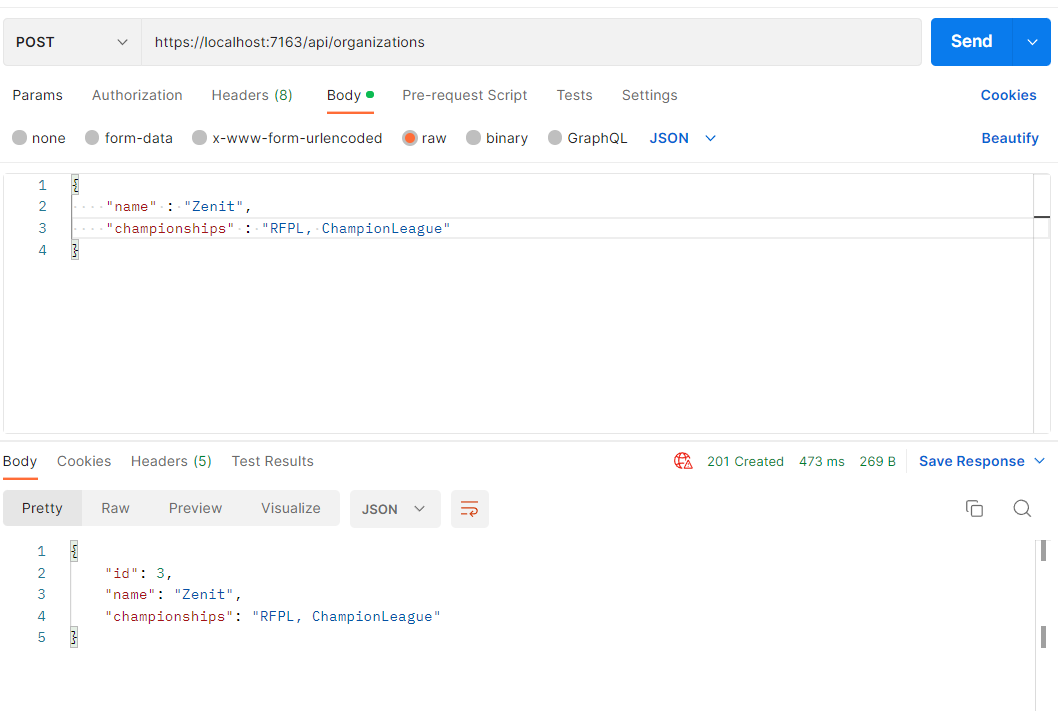
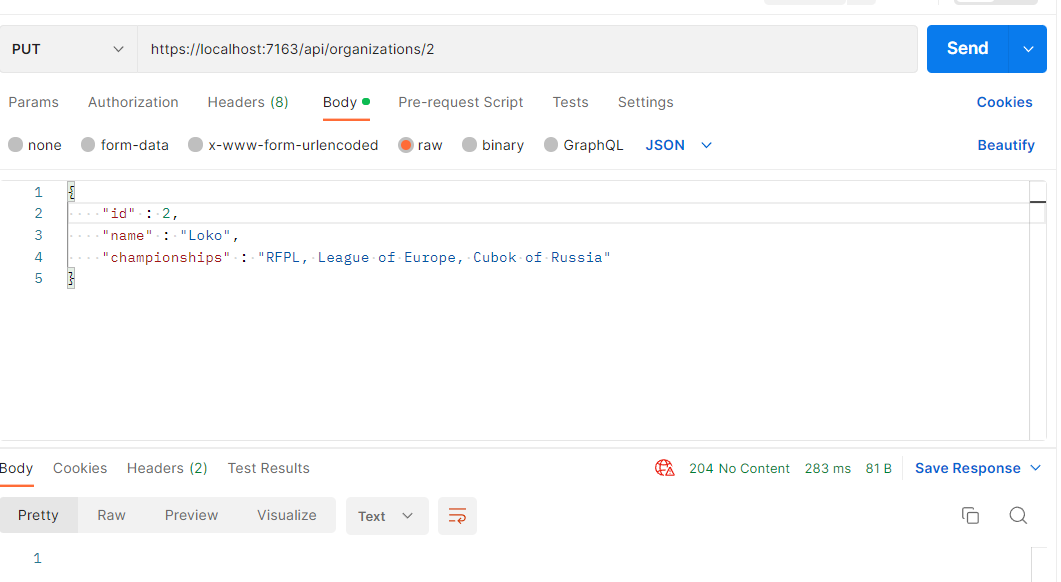


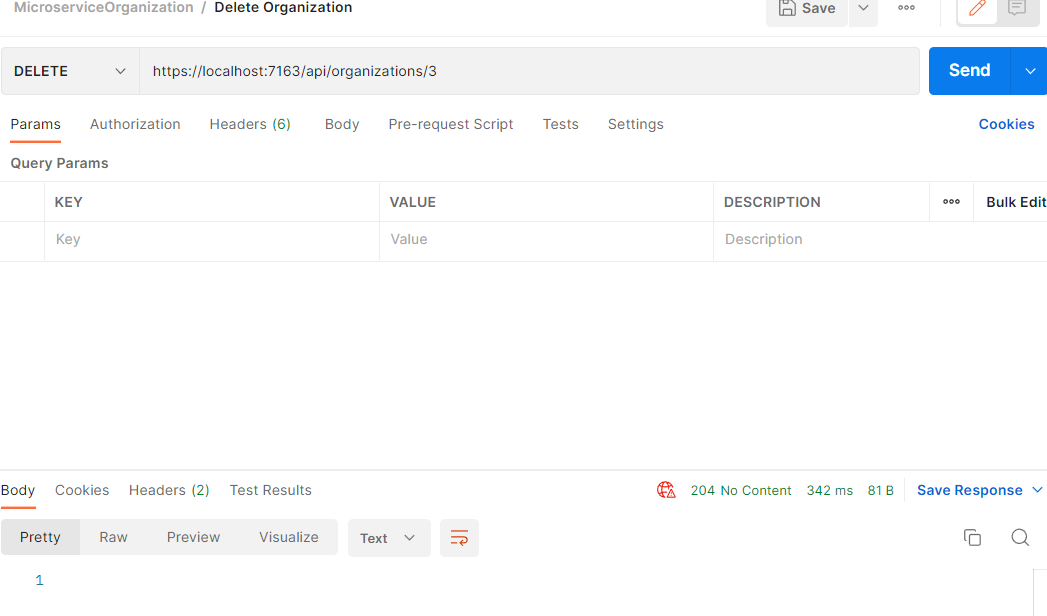
Рисунок 31. Запуск приложения MicroServiceOrganization

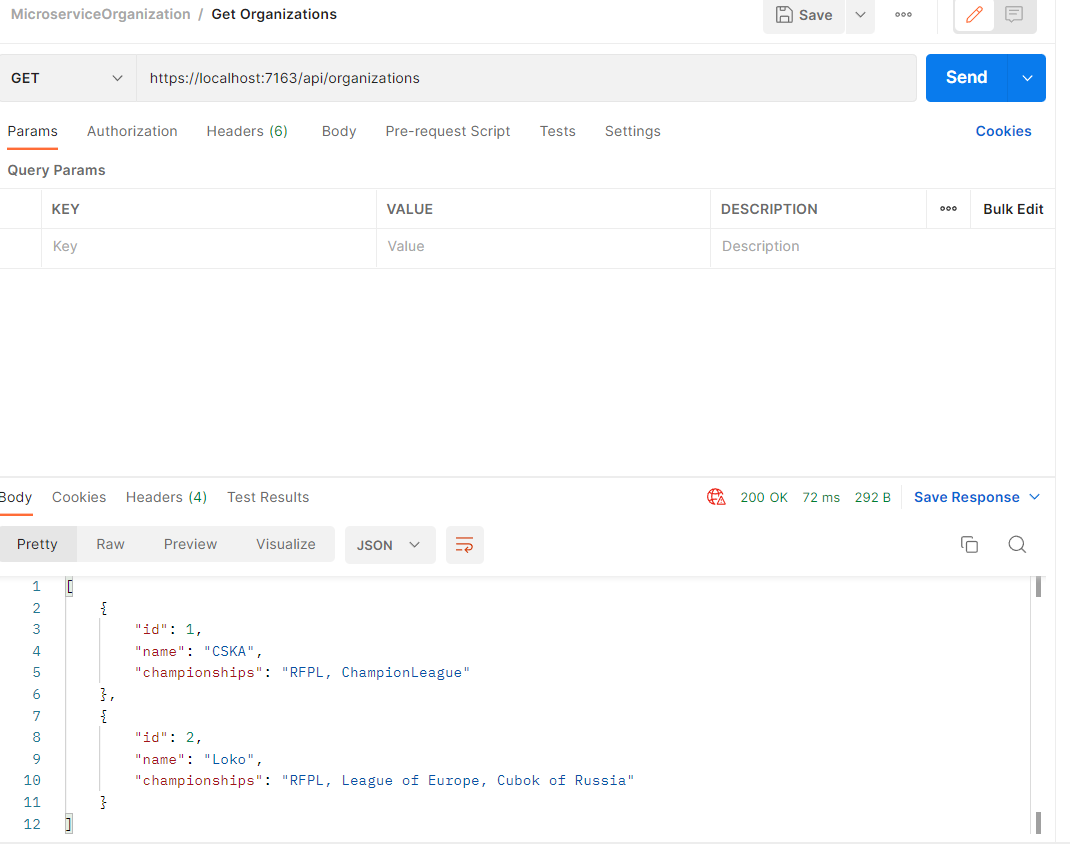












Рисунки 32.1 – 32.6. Тестирование POST, PUT, DELETE и GET – запросов для запущенного проекта MicroserviceOrganization

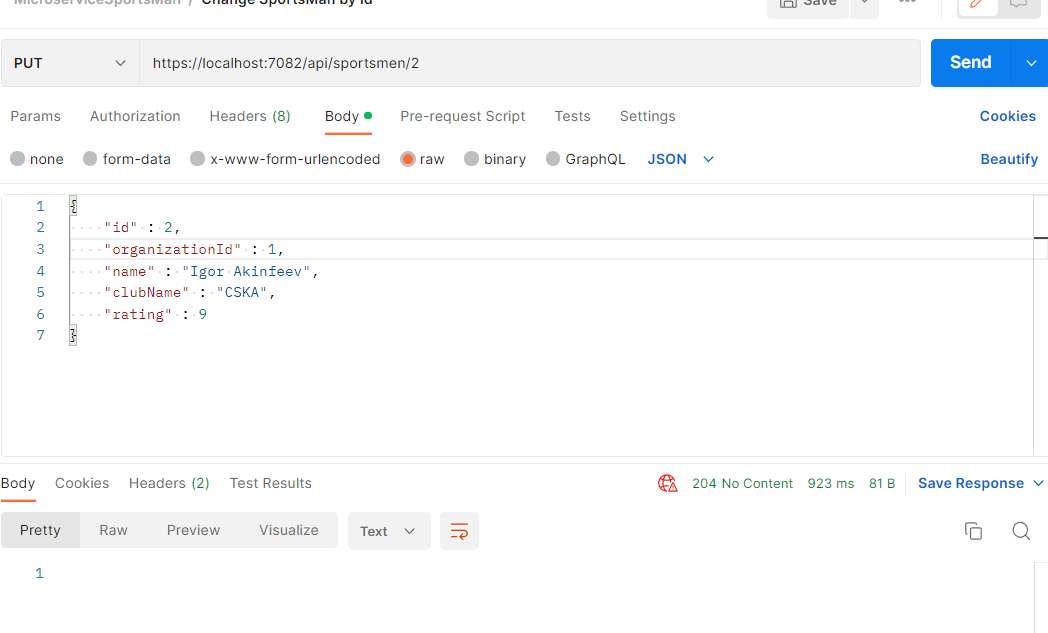


Рисунок 33. Добавление значения для поля organizationId для спортсмена с id = 2

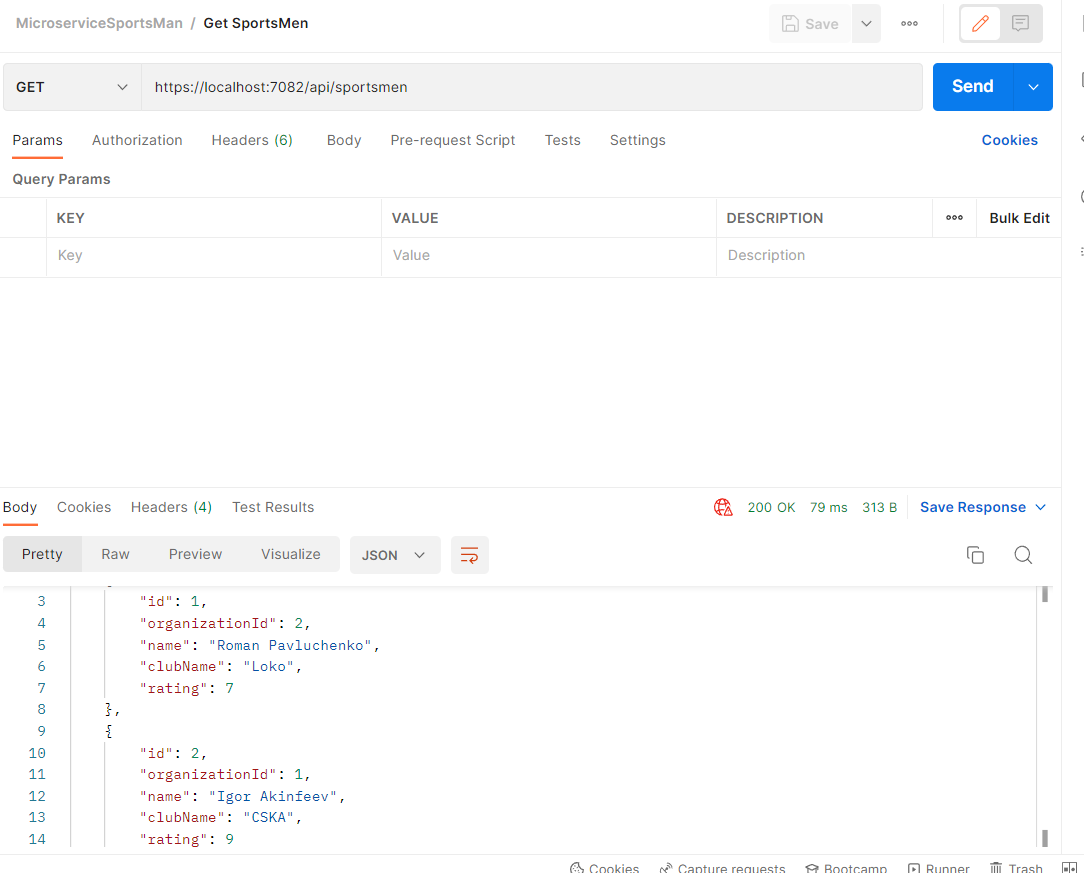


Рисунок 34. Результат добавления новых полей

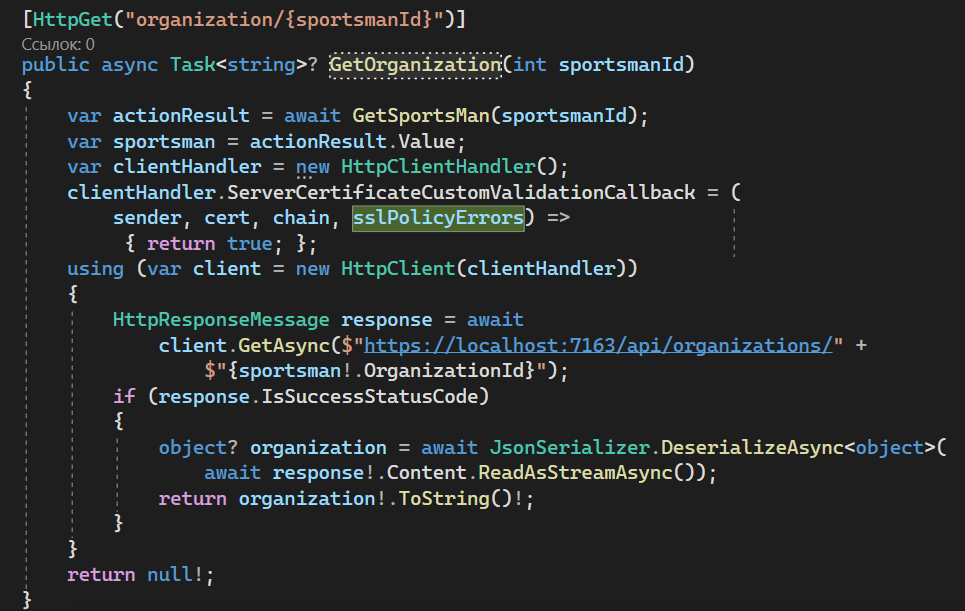


Рисунок 35. Добавление метода GetOrganization в контроллер SportsMenController

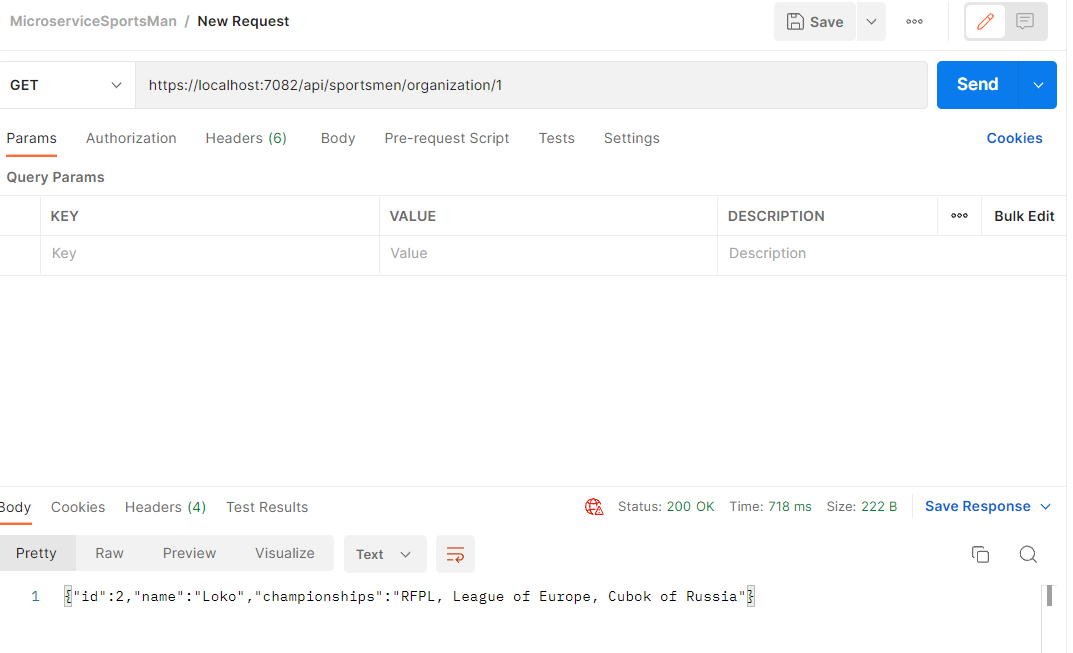


Рисунок 36. GET – запрос из микросервиса MicroserviceSportsMan к микросервису MicroserviceOrganization

**Создание композитного микросервиса с разделением баз и корректировкой модели класса SportsMan**

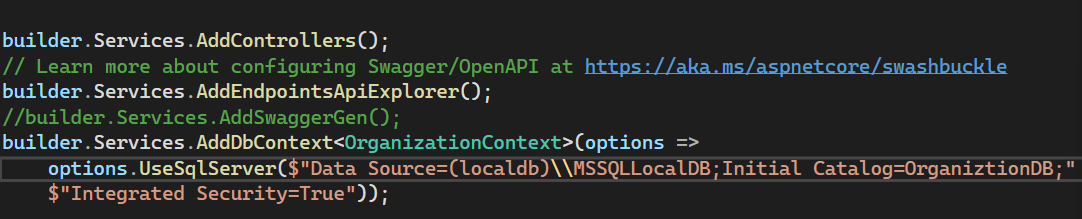
****

Рисунок 37. Замена имени БД для микросервиса MicroserviceOrganization

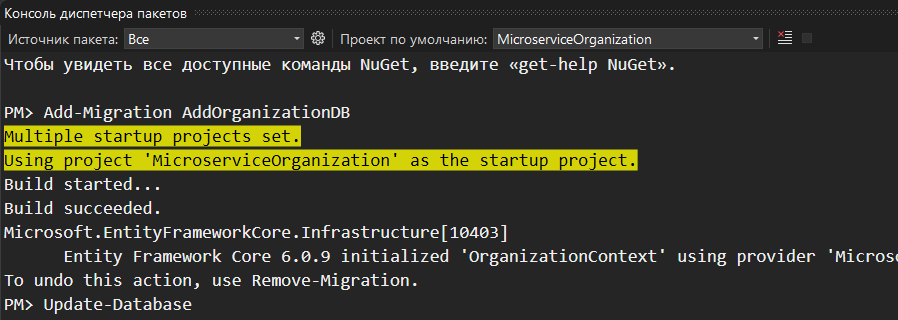


Рисунок 38. Добавление новой миграции и создание БД OrganizationDB

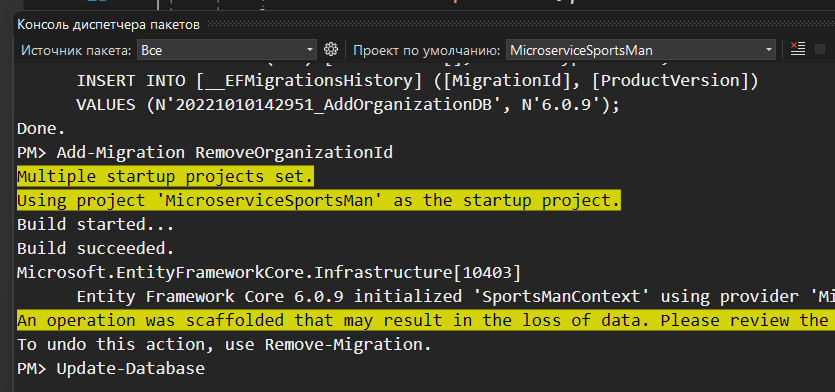


Рисунок 39. Добавление миграции на удаление из модели SportsMan поля OrganizationId

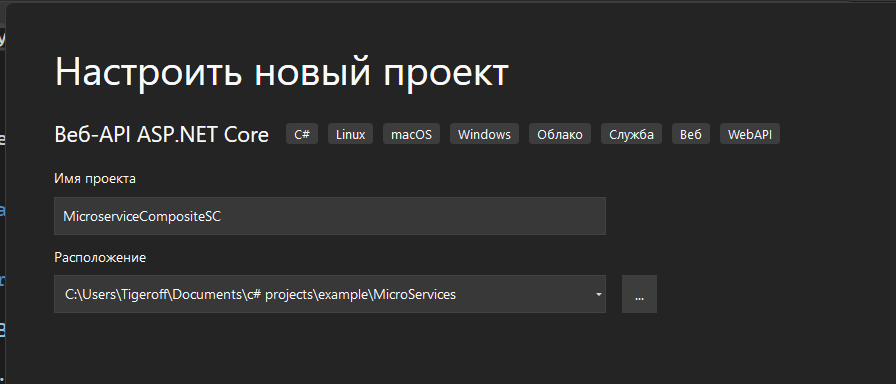


Рисунок 40. Добавление нового проекта для создания композитного микросервиса

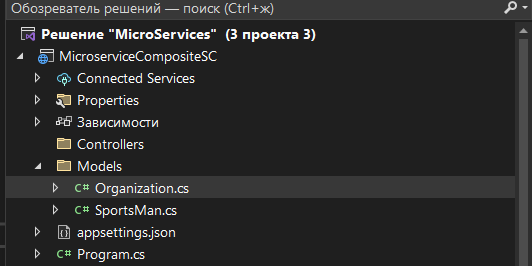


Рисунок 41. Копирование моделей

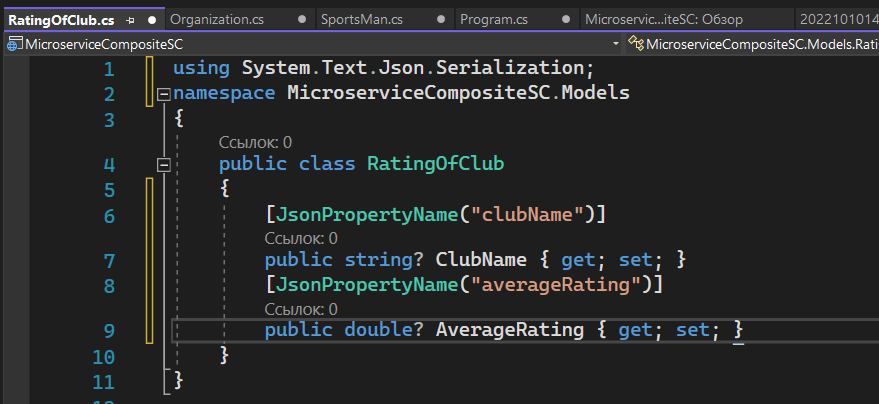


Рисунок 42. Добавление в микросервис MicroserviceCompositeSC новой модели RatingOfClub

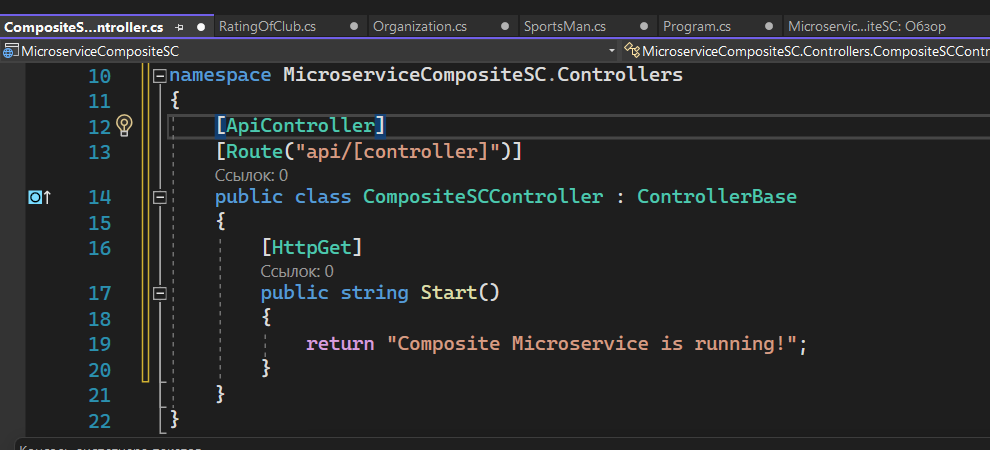


Рисунок 43. Добавление композитного контроллера

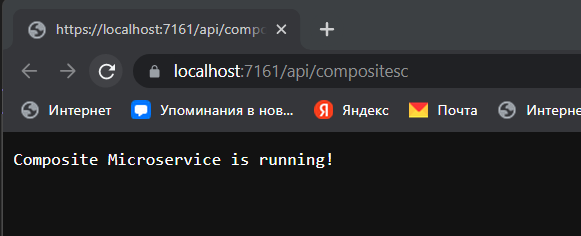


Рисунок 44. Первый запуск композитного микросервиса

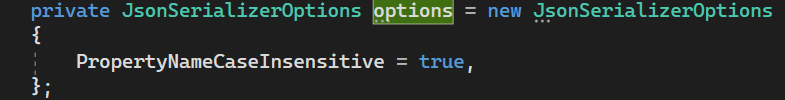


Рисунок 45. Добавление настройки учета регистра JSON

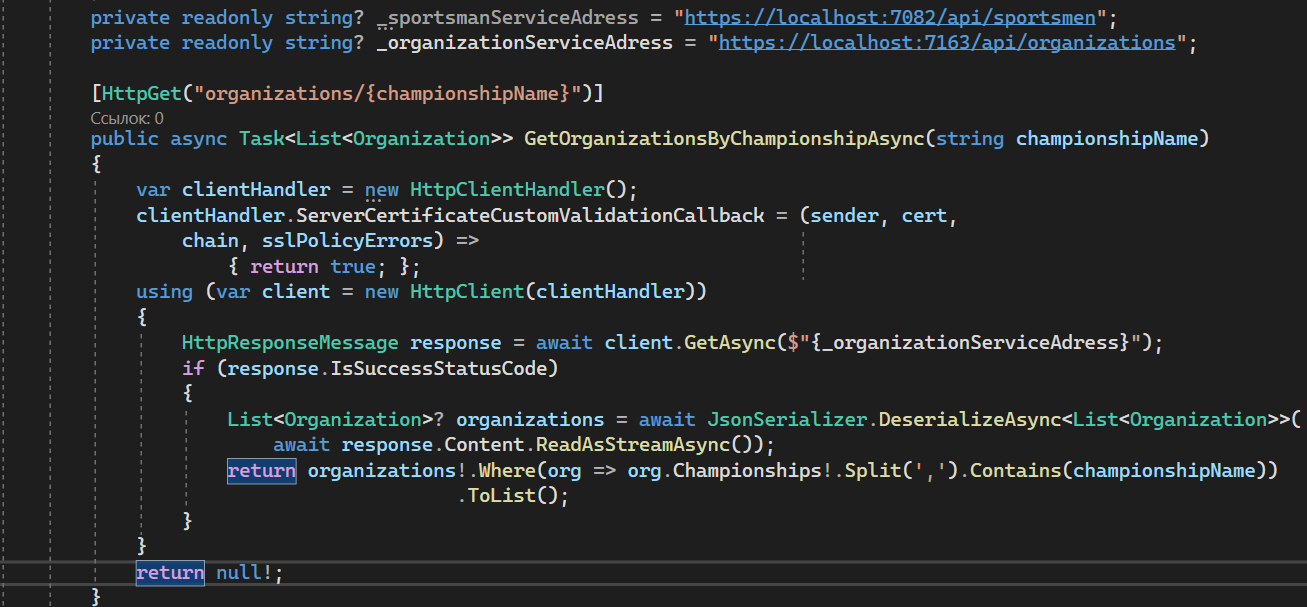


Рисунок 46.1. Добавление нового метода на получение организации по названию чемпионата среди чемпионатов этой организации в контроллере CompositeSCController

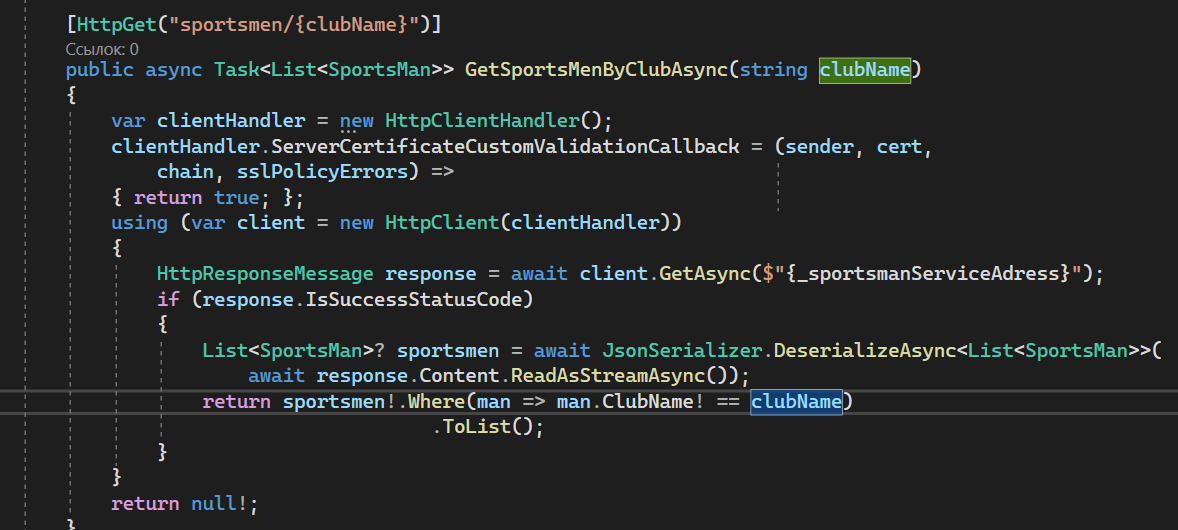


Рисунок 46.2Добавление нового метода на получение списка спортсменов, принадлежащих к клубу с заданным названием

в контроллере CompositeSCController

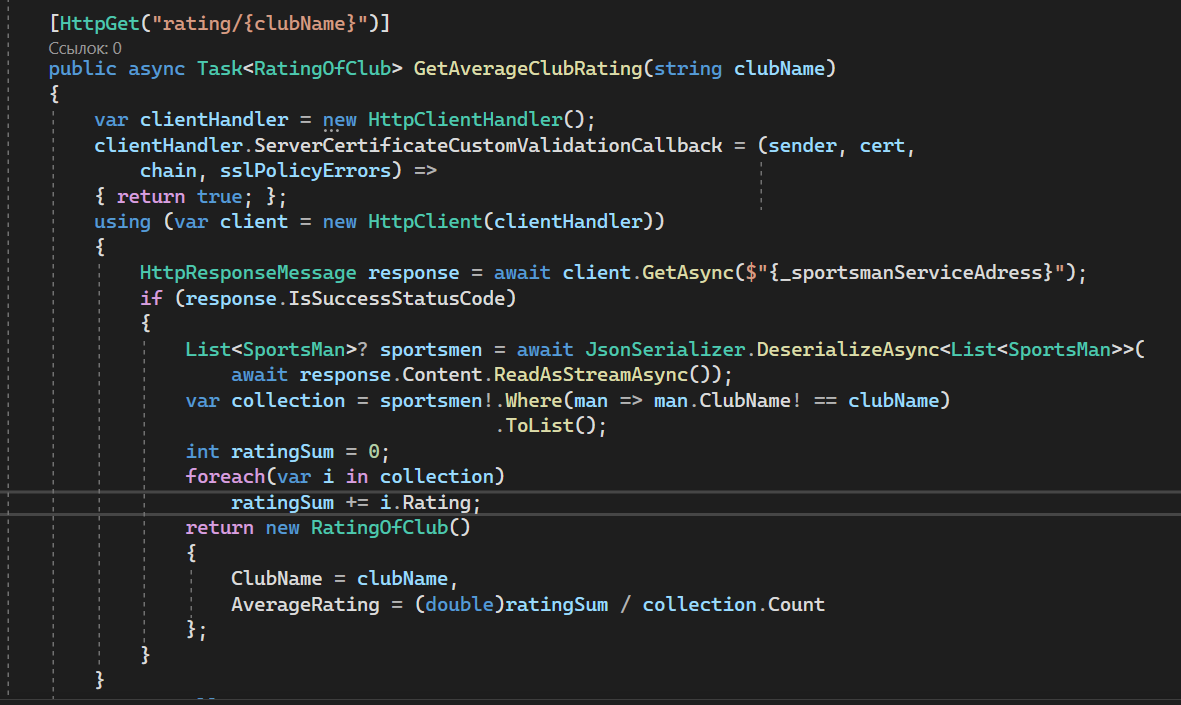


Рисунок 47. Добавление нового метода на расчет среднего рейтинга спортсменов, принадлежащих клубу

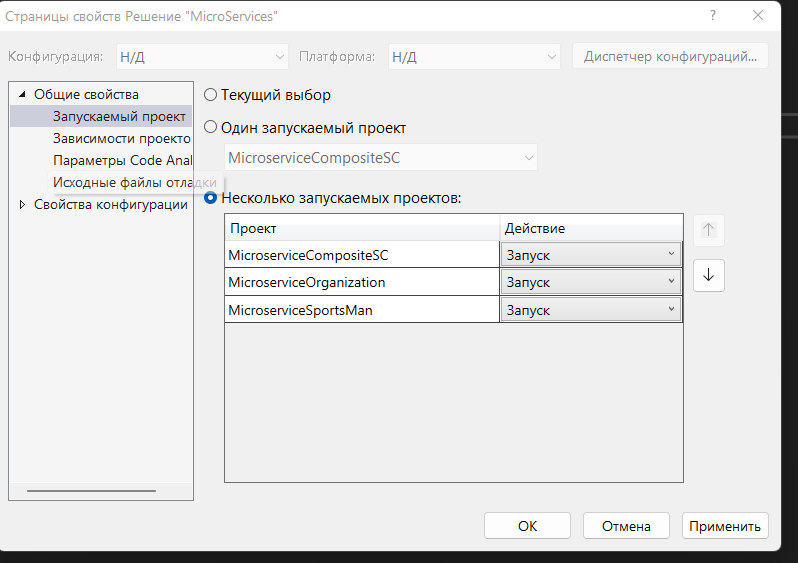


Рисунок 48. Настройка запуска всех проектов одновременно

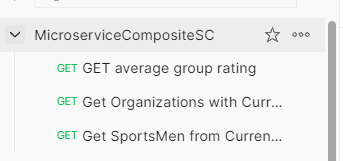


Рисунок 49. Добавление запросов композитного сервиса

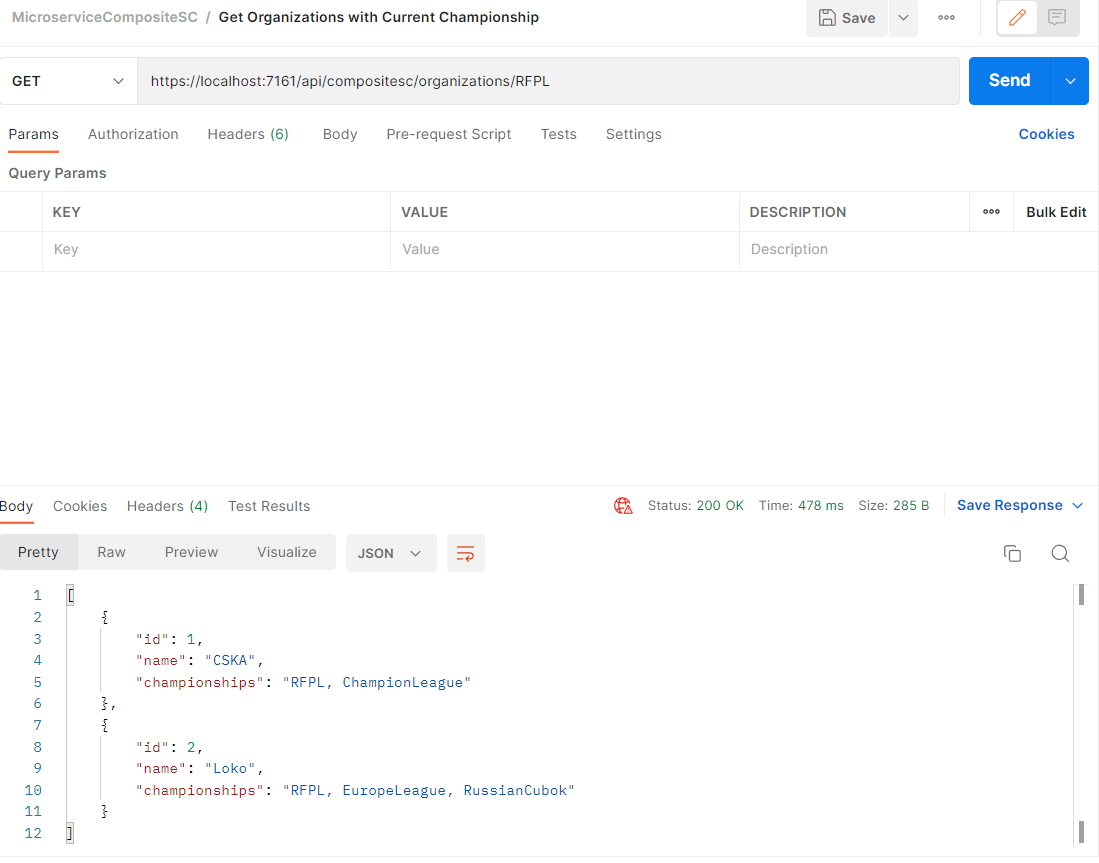


Рисунок 50. GET – запрос к композитному сервису MicroserviceCompositeSC для получения всех организаций, принимающих участие в указанном чемпионате (RFPL)

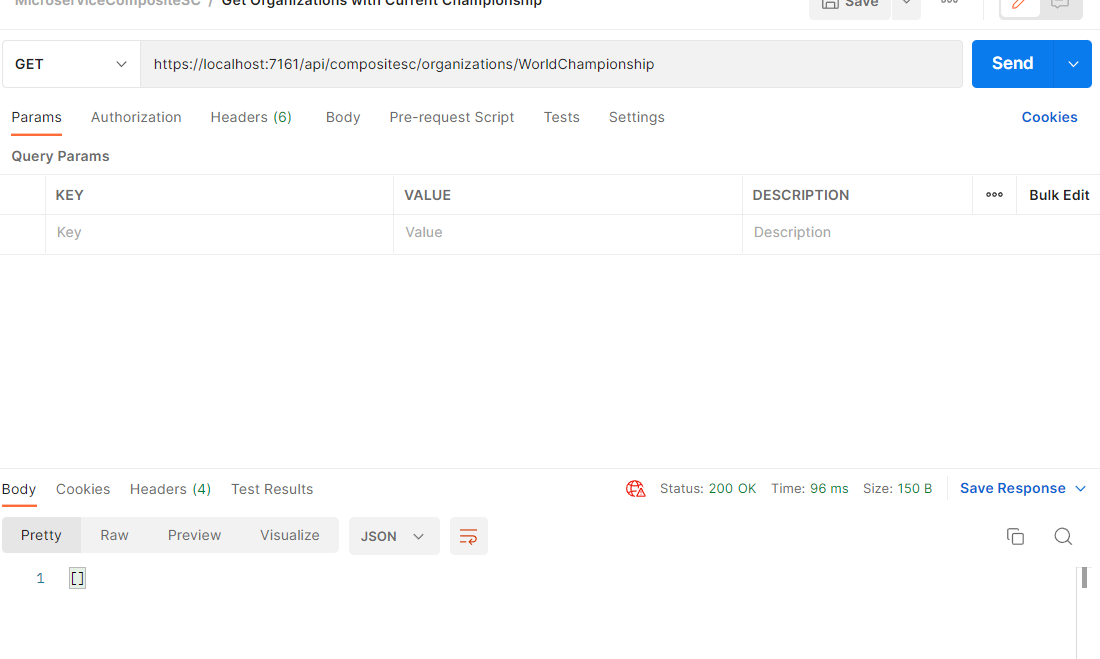


Рисунок 51. GET – запрос к композитному сервису MicroserviceCompositeSC для получения всех организаций, принимающих участие в указанном чемпионате (WorldChampionship) – таких организаций нет

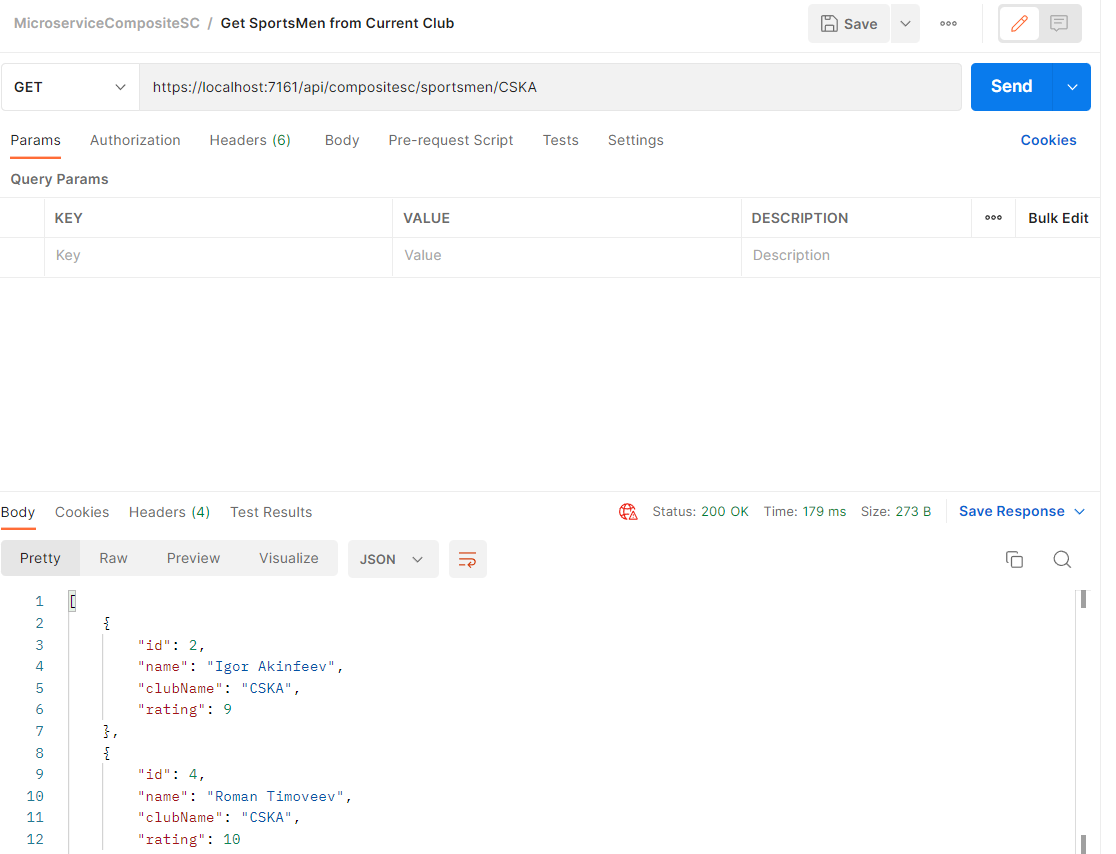


Рисунок 52. GET – запрос к композитному сервису MicroserviceCompositeSC для получения всех игроков, находящихся в клубе под названием «CSKA»

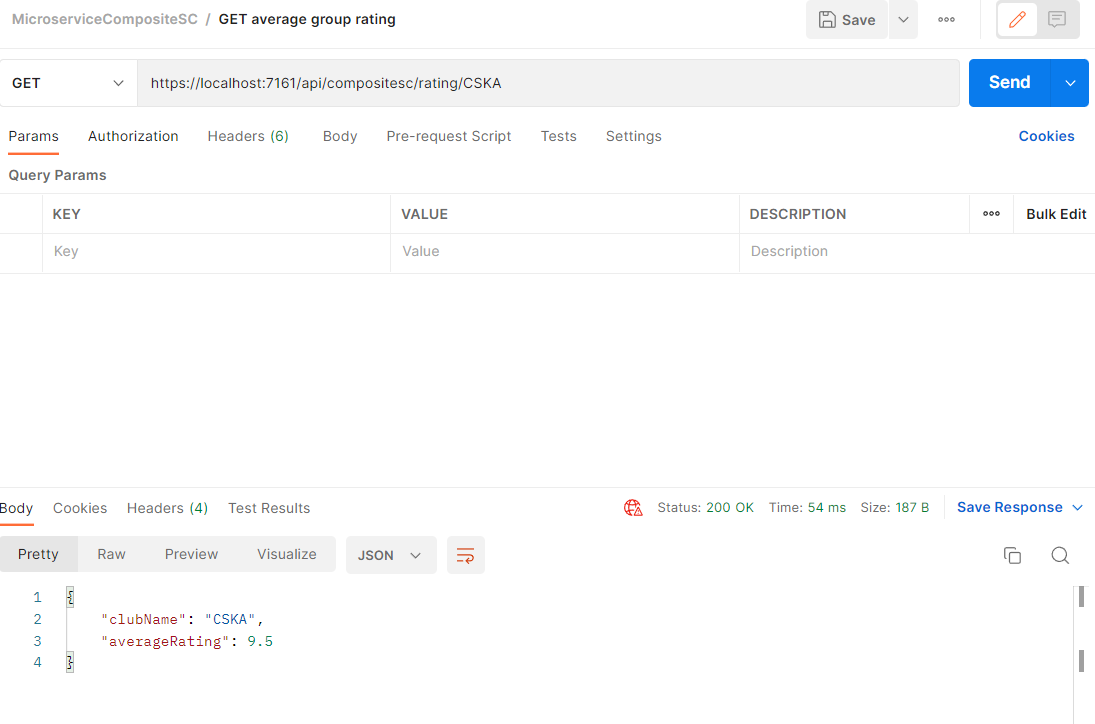


Рисунок 53. GET – запрос к композитному сервису MicroserviceCompositeSC для получения среднего рейтинга всех игроков, нахоядщихся в данном клубе

Вывод

В результате выполнения работы, я познакомился с основами микросервисной архитектуры с ее применением для создания простых композитных микросервисов на платформе .NET.