



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 10

Дисциплина: «Backend»

Тема: Изучение выдачи ответов в веб-приложении на основе ASP.NET Core

Выполнил: студент группы: 231-339

Карапетян Нвер Каренович

(Фамилия И.О.)

Дата, подпись: 14.03.25

(Дата)

Н.К. (Подпись)

Проверил: _____

(Фамилия И.О., степень, звание)

(Оценка)

Дата, подпись _____

(Дата)

(Подпись)

Москва
2025

Цель:

Ознакомиться с различными способами формирования ответов в веб-приложениях на платформе ASP.NET Core.

Задачи:

- Изучить базовые способы формирования ответов в ASP.NET Core, такие как возвращение HTML, JSON, файлов и других типов данных.
- Реализовать несколько методов контроллера для возвращения различных типов ответов (например, HTML страниц, JSON данных, файлов).
- Произвести тестирование каждого метода, убедившись в корректной выдаче соответствующего типа ответа.
- Провести сравнительный анализ различных методов и их применимости в контексте конкретных ситуаций.

Ход работы

В данной лабораторной работе была реализована основа музыкального сервиса, демонстрирующая различные способы формирования ответов в веб-приложении на платформе ASP.NET Core. В процессе выполнения были изучены методы возврата HTML-страниц, JSON-данных и аудиофайлов.

Для демонстрации работы различных типов ответов был создан контроллер `MusicController`, содержащий три метода: генерация HTML-страницы со списком треков, API, возвращающее данные о треках в формате JSON, возможность скачивания аудиофайлов.

Модель Track

В первую очередь реализуем простую модель музыкальной композиции, содержащей основные свойства: уникальный идентификатор (`Id`), название композиции, автора, путь аудиофайлу и год издательства композиции:

```

1 namespace Laba_10.Models
2 {
3     Ссылка: 8 | 0 изменений | 0 авторов, 0 изменений
4     public class Track
5     {
6         Ссылка: 7 | 0 изменений | 0 авторов, 0 изменений
7         public int Id { get; set; }
8         Ссылка: 7 | 0 изменений | 0 авторов, 0 изменений
9         public string Title { get; set; } = string.Empty;
10        Ссылка: 6 | 0 изменений | 0 авторов, 0 изменений
11        public string Artist { get; set; } = string.Empty;
12        Ссылка: 6 | 0 изменений | 0 авторов, 0 изменений
13        public string FilePath { get; set; } = string.Empty;
14        Ссылка: 5 | 0 изменений | 0 авторов, 0 изменений
15        public int Year { get; set; }
16    }
17 }

```

Рисунок 1. Листинг модели Track.

В отдельном классе MusicDb реализуем некий аналог базы данных в виде небольшого статического списка из 5-и песен:

```

1 using Laba_10.Models;
2 namespace Laba_10.Data
3 {
4     Ссылка: 1 | 0 изменений | 0 авторов, 0 изменений
5     public static class MusicDb
6     {
7         public static List<Track> Tracks = new List<Track>
8         {
9             new Track
10            {
11                Id = 1,
12                Title = "Что ни день, то новость",
13                Artist = "Никита Настяк",
14                FilePath = "music/track_1.mp3",
15                Year = 2021
16            },
17            new Track
18            {
19                Id = 2,
20                Title = "Хомо Саспенс",
21                Artist = "DEEP-EX-SENSE",
22                FilePath = "music/track_2.mp3",
23                Year = 2018
24            },
25            new Track
26            {
27                Id = 3,
28                Title = "From the D 2 the LBC",
29                Artist = "Eminem, Snoop Dogg",
30                FilePath = "music/track_3.mp3",
31                Year = 2022
32            },
33            new Track
34            {
35                Id = 4,
36                Title = "Субстрат",
37                Artist = "Лжедмитрий IV",
38                FilePath = "music/track_4.mp3",
39                Year = 2021
40            },
41            new Track
42            {
43                Id = 5,
44                Title = "Клубок",
45                Artist = "verch.fate",
46                FilePath = "music/track_5.mp3",
47                Year = 2024
48            }
49        };
50    }
51 }

```

Рисунок 2. Листинг класса MusicDb.

Реализация музыкального сервиса

Для демонстрации работы различных типов ответов был создан контроллер `MusicController`, содержащий три GET-метода: генерация HTML-страницы со списком треков, API, возвращающее данные о треках в формате JSON, а также возможность скачивания аудиофайлов.

GET-запрос, возвращающий HTML-страницу

Для реализации такого метода сформируем строку, в которую запишем основу HTML-страницы, объявив в ней открывающей тег неупорядоченного списка. Затем с помощью цикла `foreach` пройдемся по каждому элементу списка песен и из каждого трека составим элемент списка и добавим к нашей строке. В конце добавим все закрывающие теги и вернем результат с помощью метода `Content`, в который вторым аргументом передадим тип отображаемого контента:

Листинг 1. Метод `GetTracksHtml`.

```
[HttpGet("html")]
public IActionResult GetTracksHtml()
{
    string html = "<html><head><meta charset=\"utf-8\"></head><body><h1>Music
List</h1><ul>";
    foreach (var track in _tracks)
    {
        html += $"<li>{track.Title} - {track.Artist}. <a href='/api/music/down-
load/{track.Id}'>Download</a></li>";
    }

    html += "</ul></body></html>";

    return Content(html, "text/html");
}
```

Результатом вызова такого метода будет являться простая HTML-страница:

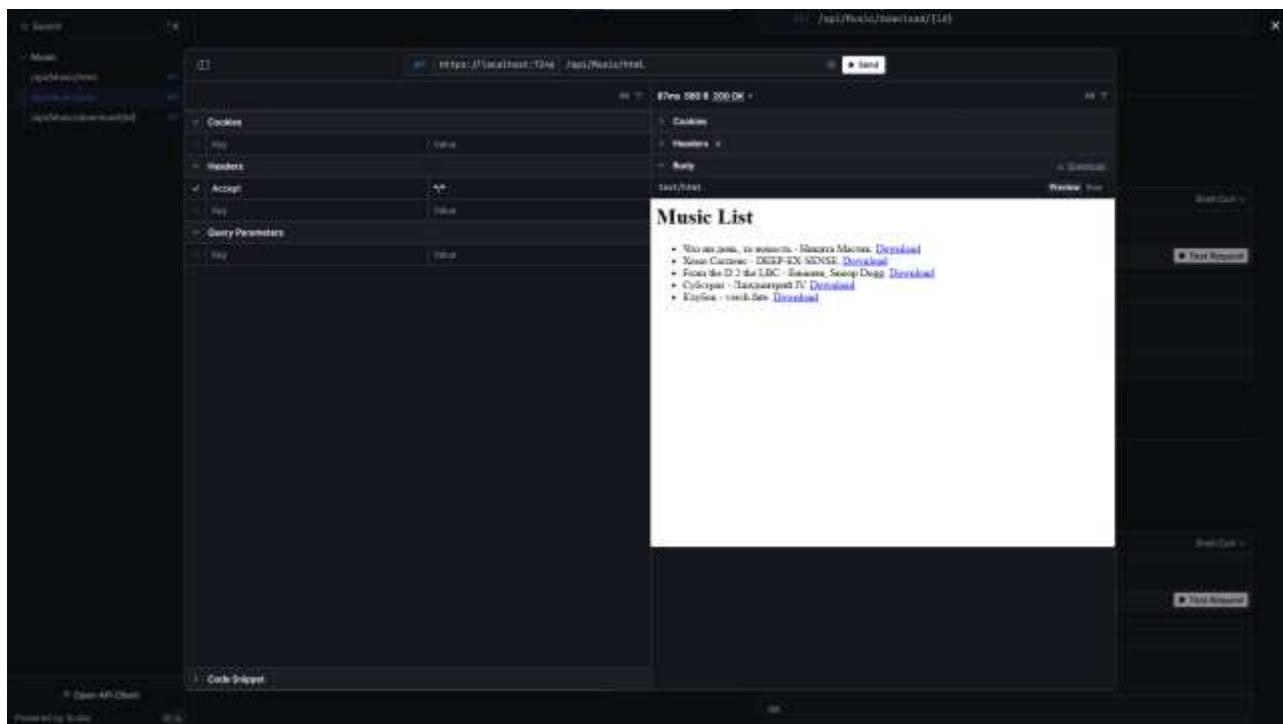


Рисунок 3. Вывод HTML-страницы.

GET-запрос, возвращающий данные в формате JSON

Для того, чтобы вернуть содержимое списка треков в формате JSON, необходимо обратиться к встроенной библиотеке `System.Text.Json`, вызвать метод `JsonSerializer.Serialize` и провести сериализацию данных в формат JSON, после чего с помощью уже знакомого метода `Content` вернуть данные, но изменив тип контента:

Листинг 2. Метод `GetTracksJson`.

```
[HttpGet("json")]
public IActionResult GetTracksJson()
{
    string jsonData = JsonSerializer.Serialize(_tracks);
    return Content(jsonData, "application/json");
}
```

Результатом вызова этого запроса в `Scalar` мы получим массив объектов, в которых хранятся параметры, указанные в модели `Track`:

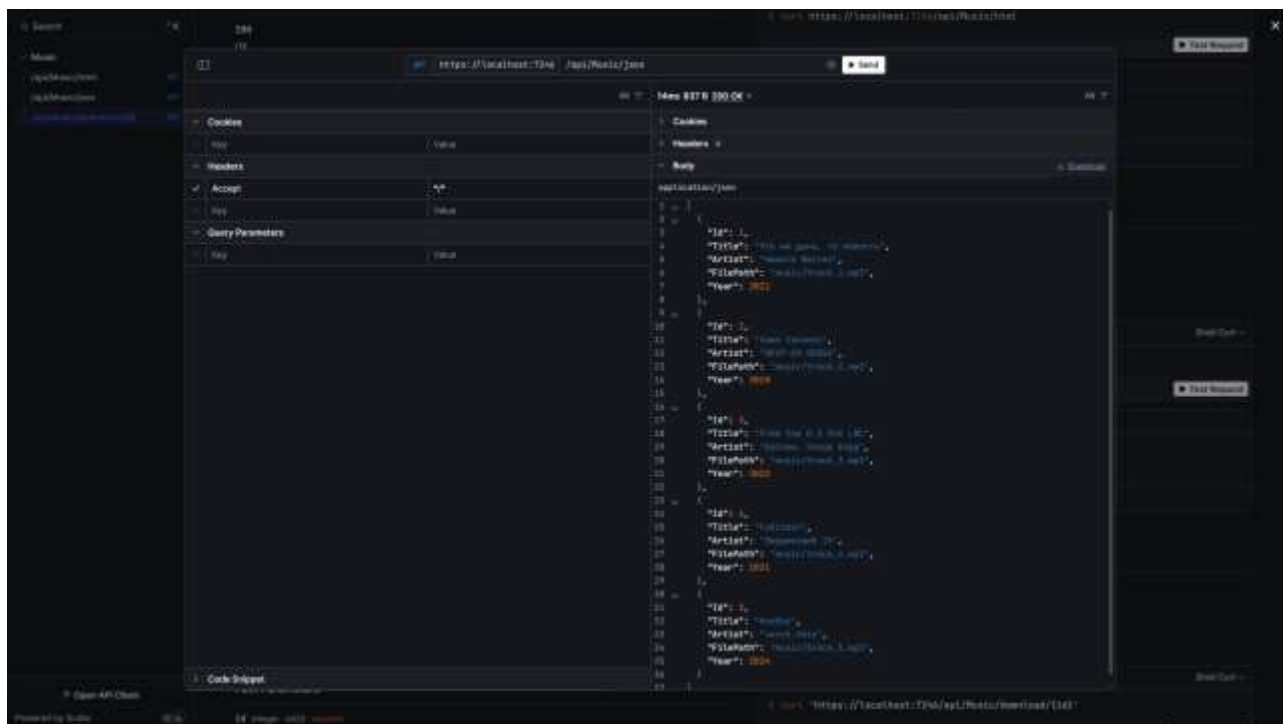


Рисунок 4. Результат вызова — данные в формате JSON.

GET-запрос, возвращающий аудиофайл

GET-запрос, возвращаемым типом которого является файл, можно смело называть скачивание файлов. Поскольку скачивать мы хотим песни по одному, нам необходимо передавать методу уникальный идентификатор (Id) песни. Затем нужно обязательно проверить, существует ли в нашей импровизированной базе данных песня с таким Id. В случае, если в списке песни с запрашиваемым Id не оказывается, возвращаем клиенту ошибку 404 (Not Found). В противном же случае определяем путь до нашей песни и возвращаем в этот раз уже с помощью метода `PhysicalFile`, которому передаем три параметра: путь к файлу, тип возвращаемого контента и название песни:

Листинг 3. Метод `DownloadTrack`.

```
[HttpGet("download/{id}")]
public IActionResult DownloadTrack([FromRoute] int id)
{
    var track = _tracks.FirstOrDefault(t => t.Id == id);

    if (track is null)
        return NotFound();

    var filePath = Path.Combine(Directory.GetCurrentDirectory(), track.FilePath);
```

```
return PhysicalFile(filePath, "audio/mpeg", track.Title + ".mp3");
}
```

Таким образом при обращении к этому методу, перейдя по URL-адресу эндпоинта, запускается скачивание нужной песни:

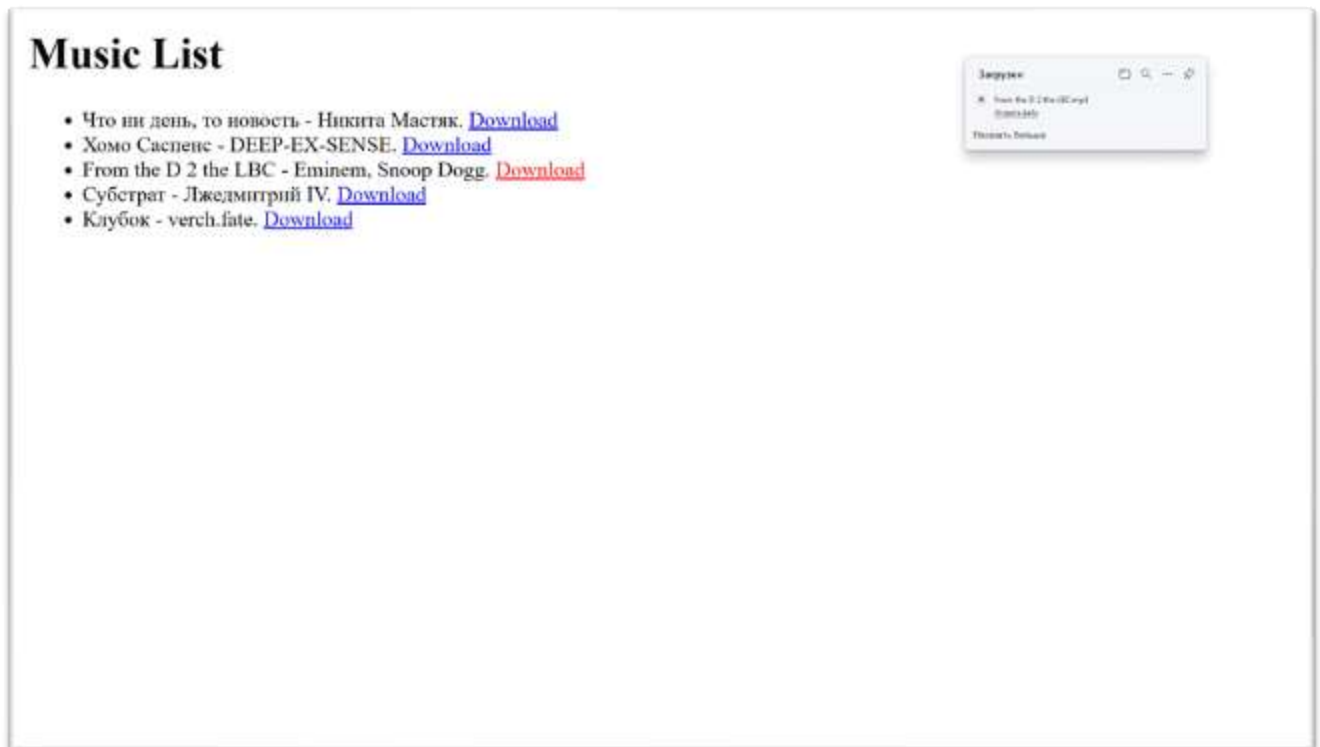


Рисунок 5. Вызов метода запускает скачивание файла.

Приложение

Листинг 4. Контроллер MusicController.

```
using Laba_10.Models;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace Laba_10.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class MusicController : ControllerBase
    {
        private readonly List<Track> _tracks;

        public MusicController()
        {
            _tracks = Data.MusicDb.Tracks;
        }

        [HttpGet("html")]
    }
```

```

        public IActionResult GetTracksHtml()
        {
            string html = $"<html><head><meta charset=\"utf-8\"></head><body><h1>Music List</h1><ul>";
            foreach (var track in _tracks)
            {
                html += $"<li>{track.Title} - {track.Artist}. <a href='/api/music/download/{track.Id}'>Download</a></li>";
            }

            html += "</ul></body></html>";

            return Content(html, "text/html");
        }

        [HttpGet("json")]
        public IActionResult GetTracksJson()
        {
            string jsonData = JsonSerializer.Serialize(_tracks);
            return Content(jsonData, "application/json");
        }

        [HttpGet("download/{id}")]
        public IActionResult DownloadTrack([FromRoute] int id)
        {
            var track = _tracks.FirstOrDefault(t => t.Id == id);

            if (track is null)
                return NotFound();

            var filePath = Path.Combine(Directory.GetCurrentDirectory(), track.FilePath);

            return PhysicalFile(filePath, "audio/mpeg", track.Title + ".mp3");
        }
    }
}

```

Листинг 5. Модель Track.

```

namespace Laba_10.Models
{
    public class Track
    {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
        public string Artist { get; set; } = string.Empty;
        public string FilePath { get; set; } = string.Empty;
        public int Year { get; set; }
    }
}

```



```
}  
}
```

Листинг 6. Хранилище MusicDb.

```
using Laba_10.Models;  
namespace Laba_10.Data  
{  
    public static class MusicDb  
    {  
        public static List<Track> Tracks = new List<Track>  
        {  
            new Track  
            {  
                Id = 1,  
                Title = "Что ни день, то новость",  
                Artist = "Никита Мастяк",  
                FilePath = "music/track_1.mp3",  
                Year = 2021  
            },  
            new Track  
            {  
                Id = 2,  
                Title = "Хомо Саспенс",  
                Artist = "DEEP-EX-SENSE",  
                FilePath = "music/track_2.mp3",  
                Year = 2018  
            },  
            new Track  
            {  
                Id = 3,  
                Title = "From the D 2 the LBC",  
                Artist = "Eminem, Snoop Dogg",  
                FilePath = "music/track_3.mp3",  
                Year = 2022  
            },  
            new Track  
            {  
                Id = 4,  
                Title = "Субстрат",  
                Artist = "Лжедмитрий IV",  
                FilePath = "music/track_4.mp3",  
                Year = 2021  
            },  
            new Track  
            {  
                Id = 5,  
                Title = "Клубок",  
                Artist = "verch.fate",
```

```
        FilePath = "music/track_5.mp3",  
        Year = 2024  
    },  
};  
}  
}
```