**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**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-разработка

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

**Выполнил: студент группы 231-332**

**Чесноков Александр Евгеньевич**

(Фамилия И.О. обучающегося)

**Дата, подпись** 01.10.2025 

(Дата)(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.

**Ход работы:**

Целью данной работы было изучение различных способов формирования ответов в веб-приложениях на платформе ASP.NET Core. В процессе выполнения практического занятия были реализованы и протестированы разные типы ответов — HTML, JSON, текстовые данные, файлы и коды состояния HTTP.

В рамках задания был создан новый проект ASP.NET Core Web API и добавлен контроллер ResponseController, демонстрирующий различные подходы к формированию ответа. Для возвращения HTML использовался метод Content(), в который передавалась HTML-разметка и MIME-тип "text/html". Такой способ удобен для простых HTML-страниц или динамически генерируемого содержимого без отдельного шаблонизатора.

Для формирования JSON-ответа использовался метод Ok(), который возвращает объект, автоматически сериализуемый в формат JSON. Этот способ является стандартным при построении REST API, где обмен данными между клиентом и сервером происходит в JSON-формате.

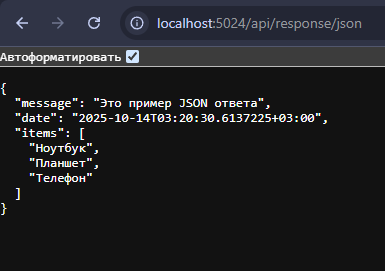
Также был реализован метод GetText(), возвращающий обычный текст с MIME-типом "text/plain", что часто используется при отладке, логировании или для простых ответов без форматирования.  
Метод GetFile() возвращает файл с диска через метод File(), указывая тип содержимого и имя файла. Это позволяет выдавать пользователю документы, изображения или другие ресурсы напрямую из контроллера.  
Дополнительно был реализован метод GetStatus(), возвращающий только код состояния 204 No Content, что полезно при успешном выполнении операции без необходимости отправлять данные.

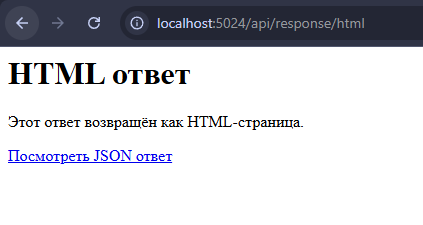
В ходе тестирования все методы корректно возвращали соответствующие типы ответов: при обращении к маршрутам /api/response/html, /api/response/json, /api/response/text, /api/response/file и /api/response/status в браузере и Postman можно было наблюдать корректную выдачу разных типов данных.

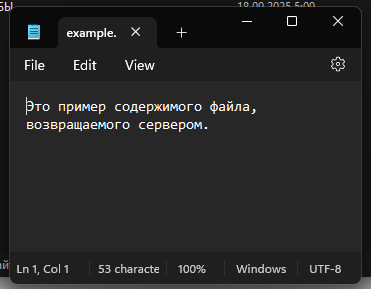
Таким образом, в работе были изучены и реализованы все основные способы формирования ответов в ASP.NET Core: HTML, JSON, текст, файлы и статусы. Был проведён сравнительный анализ их применимости:

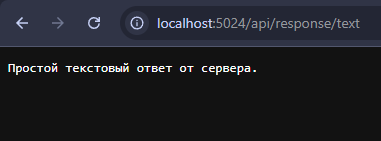
* HTML удобен для простых веб-страниц и визуального вывода;
* JSON — основной формат обмена данными в API;
* текст — для отладочных или вспомогательных ответов;
* файлы — для передачи документов и ресурсов;
* коды статусов — для обозначения результата запроса.

**Скриншоты работы приложения:**

****

****

****

****

**Листинг:**

**Program.cs:**

|  |
| --- |
| var builder = WebApplication.CreateBuilder(args);  builder.Services.AddControllers();  var app = builder.Build();  app.MapControllers();  app.Run(); |

**Controllers/ResponseController.cs:**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  namespace Lab10.Controllers;  /// *Контроллер демонстрирует различные типы ответов в ASP.NET Core*  [ApiController]  [Route("api/[controller]")]  public class ResponseController : ControllerBase  {      //*Пример возвращения HTML-страницы*      [HttpGet("html")]      public IActionResult GetHtml()      {          var html = @"              <html>                  <head><title>Пример HTML ответа</title></head>                  <body>                      <h1>HTML ответ</h1>                      <p>Этот ответ возвращён как HTML-страница.</p>                      <a href='/api/response/json'>Посмотреть JSON ответ</a>                  </body>              </html>";          return Content(html, "text/html; charset=utf-8");      }      //*Пример возвращения JSON-объекта*      [HttpGet("json")]      public IActionResult GetJson()      {          var data = new          {              Message = "Это пример JSON ответа",              Date = DateTime.Now,              Items = new[] { "Ноутбук", "Планшет", "Телефон" }          };          return Ok(data);      }      //*Пример возвращения текстового ответа*      [HttpGet("text")]      public IActionResult GetText()      {          return Content("Простой текстовый ответ от сервера.");      }      //*Пример возвращения файла*      [HttpGet("file")]      public IActionResult GetFile()      {          var filePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "example.txt");          // *Создаём файл, если он отсутствует*          if (!System.IO.File.Exists(filePath))          {              Directory.CreateDirectory(Path.GetDirectoryName(filePath)!);              System.IO.File.WriteAllText(filePath, "Это пример содержимого файла, возвращаемого сервером.");          }          var fileBytes = System.IO.File.ReadAllBytes(filePath);          return File(fileBytes, "text/plain", "example.txt");      }      //*Пример возвращения статуса без тела*      [HttpGet("status")]      public IActionResult GetStatus()      {          return NoContent(); // *Возвращает статус 204 без тела ответа*      }  } |

**Ссылка на репозиторий:** <https://github.com/avoryyy/backend_mospoly>