**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 №6**

**Дисциплина:** BackEnd-разработка

**Тема:** Конфигурация веб-приложение 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:**

Освоить процесс конфигурирования веб-приложений на платформе ASP.NET Core для эффективного управления параметрами приложения.

**Ход работы:**

Целью лабораторной работы было изучение **механизма конфигурации веб-приложений на платформе ASP.NET Core**. Это необходимо для централизованного управления параметрами приложения, поддержания разных сред (разработка, тестирование, продакшн) и гибкого изменения настроек без перекомпиляции кода.

Были созданы отдельные JSON-файлы для каждой среды:

* appsettings.Development.json — для локальной разработки,
* appsettings.Test.json — для тестирования,
* appsettings.Production.json — для продакшн-сервера.

Главный файл appsettings.json содержит базовые настройки, которые переопределяются в зависимости от текущей среды выполнения.

1. Приложение получает настройки из:

* JSON-файлов (appsettings.json и среды),
* переменных окружения, что позволяет быстро менять конфигурацию без правки файлов,
* при необходимости можно подключить и командную строку или конфигурацию веб-сервера.

Создан класс AppSettings, в который привязывается конфигурация через IOptions<T>.  
Контроллер ConfigController возвращает текущие настройки приложения для проверки.

Приложение было запущено в разных средах (Development, Test, Production) с помощью переменной среды ASPNETCORE\_ENVIRONMENT.  
SwaggerUI позволяет легко проверить, что приложение подхватывает правильные параметры конфигурации для каждой среды. Например при запуске в среде Development поле DebugMode имеет значение true, а в Production — false.

**Скриншоты работы приложения:**

**Запуск в среде Development:**

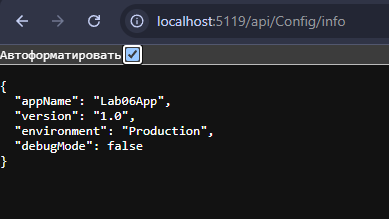
GET /Config/info:

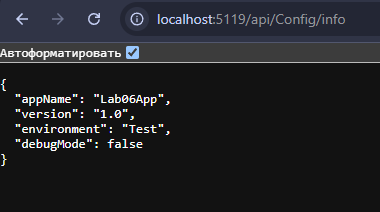
****

(Вывод соответствует настройкам из appsettings.Development.json)

**Запуск в среде Production:**

GET /Config/info:

  
(Аналогично вывод соответсвует настройкам Production. При этом Swagger недоступен т.к. не подключался в коде при этой среде)

**Запуск в среде Test:**GET /Config/info:  
****

(Вывод соответствует настройка среды Test, Swagger также недоступен)

**Листинг:**

**Program.cs:**

|  |
| --- |
| using Lab06.Models;  var builder = WebApplication.CreateBuilder(args);  // *Настройка конфигурации для разных сред*  builder.Configuration      .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)      .AddJsonFile($"appsettings.{builder.Environment.EnvironmentName}.json", optional: true)      .AddEnvironmentVariables(); // *поддержка переменных среды*  // *Привязка конфигурации к классу*  builder.Services.Configure<AppSettings>(builder.Configuration.GetSection("AppSettings"));  // *Добавляем контроллеры и Swagger*  builder.Services.AddControllers();  builder.Services.AddEndpointsApiExplorer();  builder.Services.AddSwaggerGen();  var app = builder.Build();  if (app.Environment.IsDevelopment())  {      app.UseSwagger();      app.UseSwaggerUI();  }  app.UseAuthorization();  app.MapControllers();  app.Run(); |

**Controllers/ConfigController.cs:**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using Microsoft.Extensions.Options;  using Lab06.Models;  namespace Lab06.Controllers;  [ApiController]  [Route("api/[controller]")]  public class ConfigController : ControllerBase  {      private readonly AppSettings \_settings;      public ConfigController(IOptions<AppSettings> options)      {          \_settings = options.Value;      }      [HttpGet("info")]      public IActionResult GetConfig()      {          return Ok(new          {              \_settings.AppName,              \_settings.Version,              \_settings.Environment,              \_settings.DebugMode          });      }  } |

**Models/AppSettings.cs:**

|  |
| --- |
| namespace Lab06.Models;  public class AppSettings  {      public string AppName { get; set; } = string.Empty;      public string Version { get; set; } = string.Empty;      public string Environment { get; set; } = string.Empty;      public bool DebugMode { get; set; } = false;  } |

**appsettings.Development.json:**

|  |
| --- |
| {      "AppSettings": {          "Environment": "Development",          "DebugMode": true      }  } |

**appsettings.Test.json:**

|  |
| --- |
| {      "AppSettings": {          "Environment": "Test",          "DebugMode": false      }  } |

**appsettings.Production.json:**

|  |
| --- |
| {      "AppSettings": {          "Environment": "Production",          "DebugMode": false      }  } |

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