

Лабораторна робота №1

Bcmyn do ASP.NET Core

Мета: ознайомитися з основними принципами роботи .NET, навчитися налаштовувати середовище розробки та встановлювати необхідні компоненти, набути навичок створення рішень та проектів різних типів, набути навичок обробки запитів з використанням middleware.

Хід роботи

Завдання 1. Встановлення інтегрованого середовища розробки (IDE) та необхідних компонентів

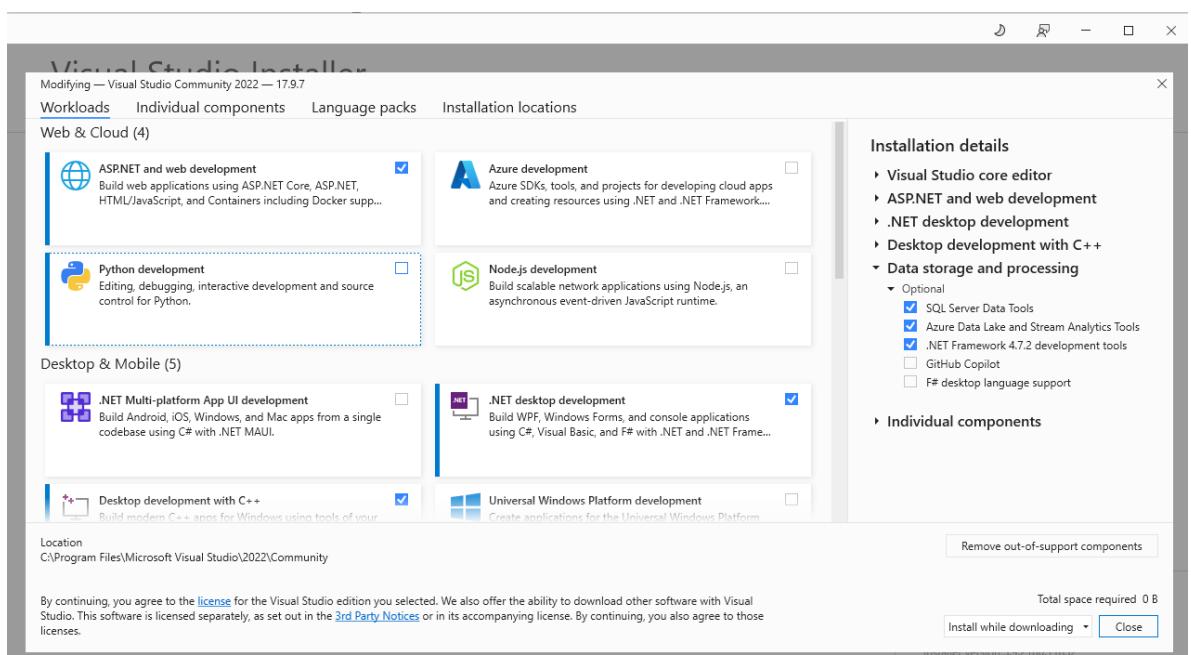


Рис. 1.1. Встановлення інтегрованого середовища розробки

Завдання 2. Створення проектів

Частина 1.

1. Створіть проект для консольного додатку з назвою `ConsoleToWeb` з використанням `dotnet CLI`
 2. Перетворіть створений консольний додаток на веб-додаток

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Трофімчук М.О.			Літ.	Арк.	Аркушів
Перевір.		Українець М.О.					1
Реценз.					Звіт з лабораторної роботи №1		
Н. Контр.					ФІКТ, гр. ІПЗ-22-2		
Зав.каф.							

3. Опишіть виконані кроки у звіті

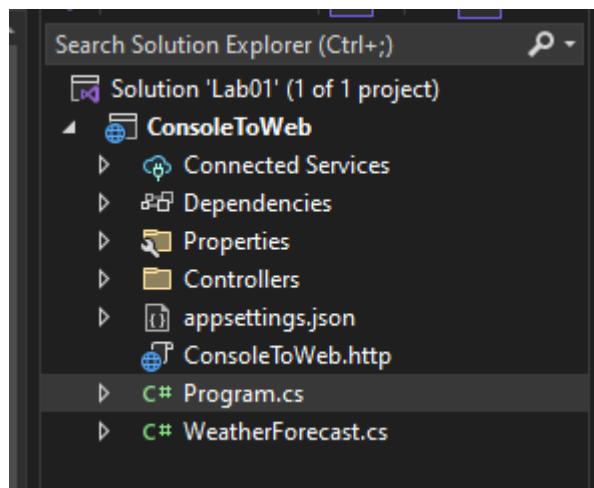


Рис. 2.1.1. Створений веб-додаток

Частина 2.

1. Створіть ASP.NET WebAPI проект без авторизації з назвою WebFromCli з використанням dotnet CLI

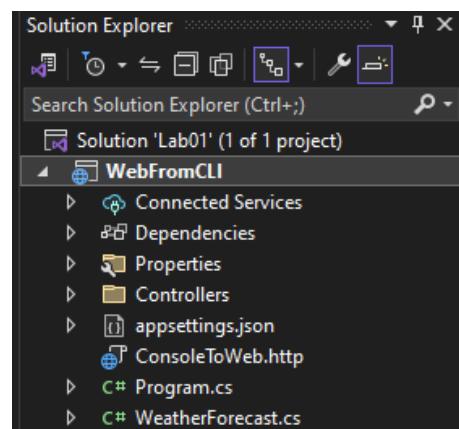


Рис. 2.2.1. Створений ASP.NET WebAPI додаток

2. Реалізуйте GET-ендпоїнт “/who”, який повертає у відповідь ваше ім’я та прізвище

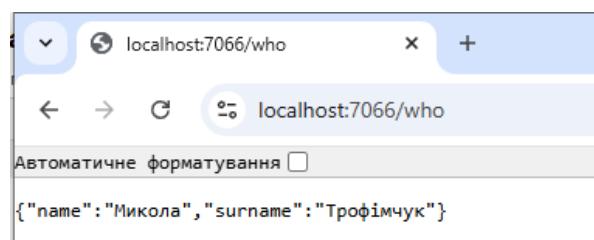


Рис. 2.2.2. GET-ендпоїнт “/who”, який повертає у відповідь ім’я та прізвище

Змн.	Арк.	№ докум.	Підпис	Дата

3. Реалізуйте GET-ендпоїнт “/time”, який повертає у відповідь поточний час на сервері

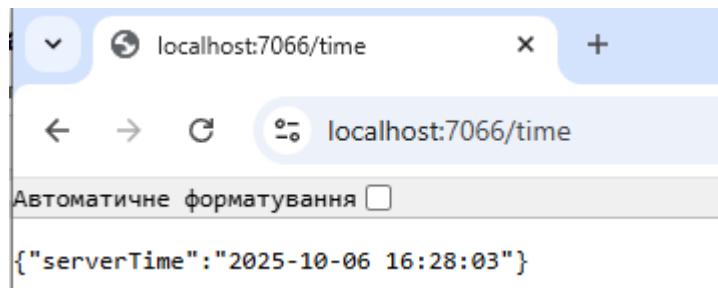


Рис. 2.2.3. GET-ендпоїнт “/time”, який повертає у відповідь поточний час на сервері

4. Наведіть лістинг реалізованих обробників у звіті

Лістинг програми:

```
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.Hosting;

var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

// --- GET /who ---
app.MapGet("/who", () => new
{
    Name = "Микола",
    Surname = "Трофімчук"
});

// --- GET /time ---
app.MapGet("/time", () => new
{
    ServerTime = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss")
});

app.Run();
```

Частина 3.

- Створіть ASP.NET MVC проект будь-яким зручним способом.
- Реалізуйте контролер з назвою LabController
- В створеному контролері реалізуйте обробник /info, який повертає View з даними про номер лабораторної роботи, тему, мету та ім’я та прізвище виконавця в табличному вигляді
- Дані для відображення передати з контролера

Лістинг програми:

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
						3

LabController:

```
using Microsoft.AspNetCore.Mvc;

namespace MvcLabProject.Controllers
{
    public class LabController : Controller
    {
        public IActionResult Info()
        {
            var labData = new
            {
                Number = 1,
                Topic = "Вступ до ASP.NET Core",
                Purpose = "ознайомитися з основними принципами роботи .NET, навчитися налаштовувати середовище розробки та встановлювати необхідні компоненти, " +
                "набути навичок створення рішень та проектів різних типів, набути навичок обробки запитів з використанням middleware",
                Author = "Микола Трофімчук ІПЗ-22-2"
            };

            // Передаємо дані через ViewData
            ViewData["Number"] = labData.Number;
            ViewData["Topic"] = labData.Topic;
            ViewData["Purpose"] = labData.Purpose;
            ViewData["Author"] = labData.Author;

            return View();
        }
    }
}
```

View/Lab/Info:

```
@{
    ViewData["Title"] = "Lab Info";
}

<h2 style="text-align:center;">Інформація про лабораторну роботу</h2>

<table border="1" cellpadding="10" cellspacing="0" style="margin:auto; border-collapse: collapse;">
    <tr>
        <th>Номер лабораторної</th>
        <td>@ViewData["Number"]</td>
    </tr>
    <tr>
        <th>Тема</th>
        <td>@ViewData["Topic"]</td>
    </tr>
    <tr>
        <th>Мета</th>
        <td>@ViewData["Purpose"]</td>
    </tr>
    <tr>
        <th>Виконавець</th>
        <td>@ViewData["Author"]</td>
    </tr>
</table>
```

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
						4

Результат роботи програми:

Мій сайт

Головна | Lab Info

Інформація про лабораторну роботу

Номер лабораторної	1
Тема	Вступ до ASP.NET Core
Мета	ознайомитися з основними принципами роботи .NET, навчитися налаштовувати середовище розробки та встановлювати необхідні компоненти, набути навичок створення рішення та проектів різних типів, набути навичок обробки запитів з використанням middleware
Виконавець	Микола Трофімчук ПЗ-22-2

© 2025 - MyApp

Рис. 2.3.1. Результат роботи програми

Завдання 3. Робота з middleware

1. Ознайомтеся з поняттям middleware в ASP.NET
2. Створіть ASP.NET WebAPI проект з назвою MiddlewareSandbox
3. Реалізуйте наступні middleware
 - a. Реалізуйте middleware, яке рахує кількість запитів до сервера і повертає це число у відповіді, наприклад: The amount of processed requests is X.
 - b. Створіть middleware, яке аналізує параметри рядка запиту. Якщо в рядку запиту (query string) присутній параметр “custom”, то повертати у відповідь “You’ve hit a custom middleware!”, інакше пропускати запит далі. Приклад запиту зображенено на рисунку 3.1.

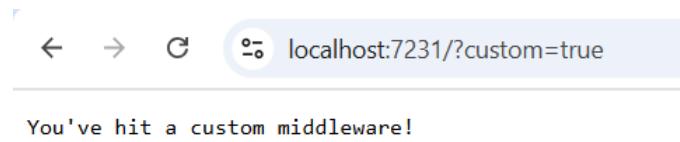


Рис. 3.1. Приклад виконання запиту

- c. Створіть middleware, яке логуватиме у консоль інформацію про всі запити. В логах відображати метод запиту (GET, POST тощо) та

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
5						

його шлях (/user, /product тощо). Для перевірки POST та інших методів запиту можна скористатися Postman або іншою подібною утилітою.

с. Реалізуйте middleware, яке перевіряє наявність API-ключа у заголовках запиту. Для цього слід Перевіряти, чи є в запиті заголовок X-API-KEY. Якщо ключ неправильний або відсутній, повернати 403 Forbidden, не передавати запит для виконання далі. Якщо ключ в заголовку запиту співпадає з тим, який заданий на сервері, передавати запит далі. Для додавання відповідного запиту можна скористатися Postman або подібними утилітами.

Лістинг програми:

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Hosting;
using System.Threading.Tasks;

var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

// --- Middleware c: Логування всіх запитів ---
app.Use(async (context, next) =>
{
    Console.WriteLine($"[LOG] Method: {context.Request.Method}, Path: {context.Request.Path}");
    await next();
});

// --- Middleware b: Перевірка параметра query string "custom" ---
app.Use(async (context, next) =>
{
    if (context.Request.Query.ContainsKey("custom"))
    {
        await context.Response.WriteAsync("You've hit a custom middleware!");
    }
    else
    {
        await next();
    }
});

// --- Middleware d: Перевірка API ключа ---
const string apiKey = "12345"; // Ключ на сервері
app.Use(async (context, next) =>
{
    if (!context.Request.Headers.TryGetValue("X-API-KEY", out var extractedApiKey) || extractedApiKey != apiKey)
    {
        context.Response.StatusCode = 403; // Forbidden
        await context.Response.WriteAsync("Forbidden: invalid or missing API key.");
        return;
    }
    await next();
});
```

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
						6

```

// --- Middleware a: Лічильник запитів ---
int requestCount = 0;
app.Use(async (context, next) =>
{
    requestCount++;
    await next();
});

// --- Простий GET ендпоїнт для тесту ---
app.MapGet("/test", async context =>
{
    await context.Response.WriteAsync($"The amount of processed requests is {requestCount}");
});

app.Run();

```

Результат роботи програми:

The screenshot shows a Postman test interface. The URL is https://localhost:7230/test. The Headers tab is selected, showing an X-API-KEY header with the value 12345. The Body tab shows the response: "The amount of processed requests is 3". The status bar at the bottom indicates a 200 OK response.

Рис. 3.1.1. middleware, яке рахує кількість запитів до сервера і повертає
це число у відповіді

The screenshot shows a Postman test interface. The URL is https://localhost:7230/test. The Headers tab is selected, showing a Key header with the value Value. The Body tab shows the response: "Forbidden: invalid or missing API key.". The status bar at the bottom indicates a 403 Forbidden response.

Рис. 3.1.2. middleware, яке перевіряє наявність API-ключа у заголовках
запиту

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
						7

Рис. 3.1.3. middleware, яке аналізує параметри рядка запиту

```
D:\4 курс\ASP Net\Lab01\MiddlewareSandbox\bin\Debug\net8.0\MiddlewareSandbox.exe
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7230
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5293
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\4 курс\ASP Net\Lab01\MiddlewareSandbox
[LOG] Method: GET, Path: /swagger
[LOG] Method: GET, Path: /favicon.ico
[LOG] Method: GET, Path: /test
```

Рис. 3.1.4. middleware, яке логує у консоль інформацію про всі запити

Висновок: в ході виконання лабораторної роботи було ознайомлено з основними принципами роботи .NET, навчено налаштовувати середовище розробки та встановлювати необхідні компоненти, набуто навичок створення рішень та проектів різних типів, набуто навичок обробки запитів з використанням middleware. Роботу виконано в повному обсязі.

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Пр.1	Арк.
						8