

# Лабораторна робота №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 рішень та проектів різних типів, набути навичок обробки запитів з використанням middleware.

Хід роботи:

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

Завдання 2. Створення проектів

## Частина 1.

- Створіть проект для консольного додатку з назвою ConsoleToWeb з використанням [dotnet CLI](#)
- Перетворіть створений консольний додаток на веб-додаток
- Опишіть виконані кроки у звіті

### Кроки виконання:

- Створено консольний додаток за допомогою dotnet new console -n ConsoleToWeb.
- Перевірено роботу через dotnet run.
- В ConsoleToWeb.csproj змінено Sdk="Microsoft.NET.Sdk" на Sdk="Microsoft.NET.Sdk.Web".
- У Program.cs додано WebApplication та ендпоїнт /.
- Запущено веб-додаток через dotnet run і перевірено у браузері.

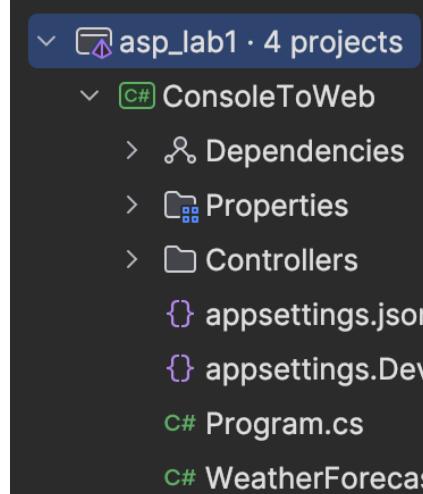


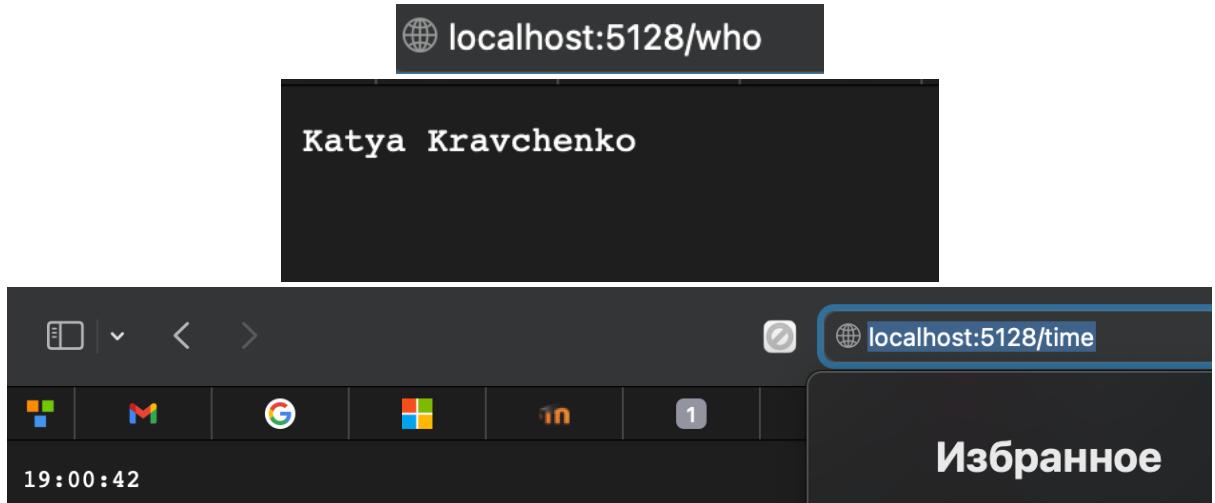
Рис. 1.1. Створено консольний додаток

## Частина 2.

- Створіть ASP.NET WebAPI проект без авторизації з назвою WebFromCli з використанням [dotnet CLI](#)

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Пр.1		
Розроб.		Кравченко К.М.				Lіт.	Арк.
Перевір.		Українець М.О.					Аркушів
Реценз.							
Н. Контр.							
Зав.каф.							
Звіт з лабораторної роботи №1						1	
ФІКТ, гр. ІПЗ-22-2							

- Реалізуйте GET-ендпоїнт “/who”, який повертаємо у відповідь ваше ім’я та прізвище
- Реалізуйте GET-ендпоїнт “/time”, який повертаємо у відповідь поточний час на сервері
- Наведіть лістинг реалізованих обробників у звіті



*Рис. 1.2. – 1.4. Реалізовані GET-ендпоїнти*

```
app.MapGet("/who", () => "Katya Kravchenko");

app.MapGet("/time", () => DateTime.Now.ToString("HH:mm:ss"));
```

### Частина 3.

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

#### LabController.cs:

```
using Microsoft.AspNetCore.Mvc;
namespace MvcLab1Project.Controllers;

public class LabController : Controller
{
    public IActionResult Info()
    {
        // Дані, які передамо у View
        ViewData["LabNumber"] = "Лабораторна робота №1";
        ViewData["Topic"] = "Вступ до ASP.NET Core";
        ViewData["Goal"] = "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.";
```

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Пр.1	Арк.

```

        ViewData["Student"] = "Катерина Кравченко";

        return View();
    }
}

```

### Views/Lab файл Info.cshtml:

```

@{
    ViewData["Title"] = "Інформація про лабораторну";
}

<h2>@ViewData["Title"]</h2>



| Поле                      | Значення               |
|---------------------------|------------------------|
| Номер лабораторної роботи | @ViewData["LabNumber"] |
| Тема                      | @ViewData["Topic"]     |
| Мета                      | @ViewData["Goal"]      |
| Виконавець                | @ViewData["Student"]   |


```

MvcLab1Project Home Privacy

### Інформація про лабораторну

Поле	Значення
Номер лабораторної роботи	Лабораторна робота №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 рішень та проектів різних типів, набути навичок обробки запитів з використанням middleware.
Виконавець	Катерина Кравченко

Рис. 1.5. Результатом роботи обробника /info

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.16.000 – Пр.1

### 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.
  - c. Створіть middleware, яке логуватиме у консоль інформацію про всі запити. В логах відображати метод запиту (GET, POST тощо) та його шлях (/user, /product тощо). Для перевірки POST та інших методів запиту можна скористатися Postman або іншою подібною утилітою.
  - c. Реалізуйте middleware, яке перевіряє наявність API-ключа у заголовках запиту. Для цього слід Перевірити, 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.

#### Program.cs:

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at
// https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

int requestCount = 0;
app.Use(async (context, next) =>
{
```

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Пр.1

```

        requestCount++;
        await next.Invoke();
        await context.Response.WriteAsync($"The amount of processed requests is {requestCount}.");
    });

// Middleware 2: Перевірка параметру 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.Invoke();
    }
});

// Middleware 3: Логування всіх запитів у консоль
app.Use(async (context, next) =>
{
    Console.WriteLine($"{DateTime.Now} [{context.Request.Method}{context.Request.Path}]");
    await next.Invoke();
});

// Middleware 4: Перевірка X-API-KEY
const string apiKey = "SECRET123";
app.Use(async (context, next) =>
{
    if (!context.Request.Headers.TryGetValue("X-API-KEY", out var key) || key != apiKey)
    {
        context.Response.StatusCode = 403;
        await context.Response.WriteAsync("Forbidden");
        return;
    }
    await next.Invoke();
});

app.MapGet("/", () => "Hello from MiddlewareSandbox!");

app.Run();

```

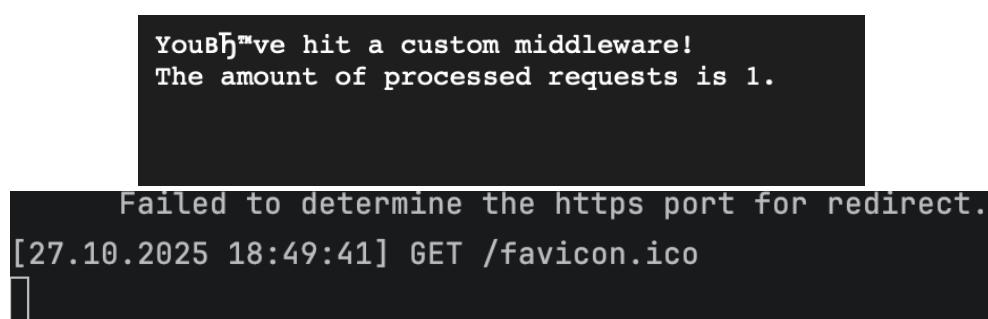
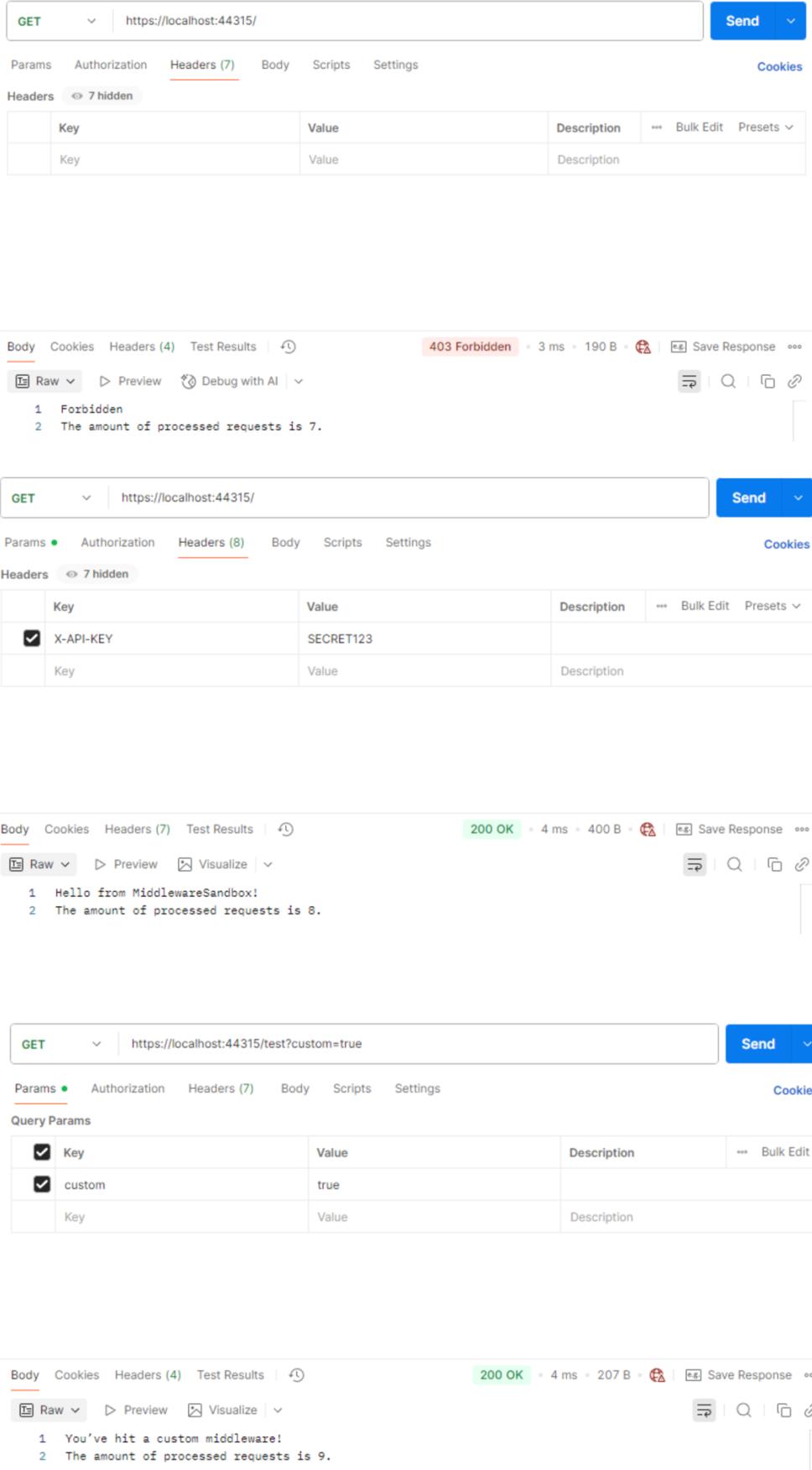


Рис. 3.1.-3.2. Виконання запиту

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Пр.1

**Postman:**



The screenshot shows the Postman application interface with four requests:

- Request 1:** GET https://localhost:44315/. Headers tab selected. Response: 403 Forbidden. Body: 1. Forbidden  
2. The amount of processed requests is 7.
- Request 2:** GET https://localhost:44315/. Headers tab selected. Response: 200 OK. Body: 1. Hello from MiddlewareSandbox!  
2. The amount of processed requests is 8.
- Request 3:** GET https://localhost:44315/test?custom=true. Query Params tab selected. Response: 200 OK. Body: 1. You've hit a custom middleware!  
2. The amount of processed requests is 9.
- Request 4:** GET https://localhost:44315/test?custom=true. Query Params tab selected. Response: 200 OK. Body: 1. You've hit a custom middleware!  
2. The amount of processed requests is 9.

*Рис. 3.3. – 3.5 Перевірка запитів у Postman*

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Пр.1	Арк.
------	------	----------	--------	------	--	------

### **Пояснення**

1. **Middleware 1** – рахує всі запити і додає у відповідь The amount of processed requests is X.
2. **Middleware 2** – перевіряє ?custom= у URL, якщо є – повертає повідомлення, інакше пропускає далі.
3. **Middleware 3** – логування всіх запитів у консоль: метод та шлях.
4. **Middleware 4** – перевірка заголовка X-API-KEY; якщо ключ неправильний або відсутній → 403 Forbidden.

**Висновок:** У ході лабораторної роботи я ознайомилася з основними принципами роботи платформи .NET, навчилася налаштовувати середовище розробки та встановлювати необхідні компоненти для створення проектів. Було опановано створення різних типів проектів: консольного додатку, веб-додатку, ASP.NET WebAPI та MVC. Також отримано практичні навички обробки запитів з використанням middleware, включно з підрахунком запитів, логуванням, перевіркою параметрів рядка запиту та контролем доступу через API-ключ. Лабораторна робота дозволила закріпити теоретичні знання і застосувати їх на практиці для створення функціональних веб-додатків.

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Пр.1