

## ЛАБОРАТОРНА РОБОТА № 3

### Початок роботи з ASP.NET Core 5. Middleware

**Мета:** ознайомитися з файловими та нефайловими провайдерами конфігурації в ASP.NET, принципами побудови власних провайдерів логування, визначенням маршрутів, навчитися зберігати дані у сесіях.

#### Хід роботи:

**Завдання 2:** Створити у Visual Studio ASP.NET Core Empty.

Назва рішення: **AspDotNetLabs**

Назва проекту: **AspDotNetLab3**

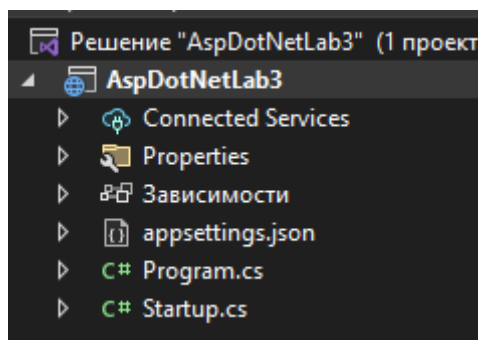


Рис. 2.0.0 Оглядач рішень з відкритим проектом

**Частина 1.** Створити власний json-файл конфігурації з назвою conf.json. У файлі конфігурації розмістити наступні налаштування:

- Title - назва, яка буде виводитися при зверненні до будь-якого маршруту додатку
- LogFile - шлях (назва) текстового файла для запису логів.

					ДУ «Житомирська політехніка».21.121.08.000 – Лр2			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи  ФІКТ Гр. ІПЗ-20-2[2]			
Розроб.	Палій І.В.							
Перевір.	Чижмотря О.В.							
Керівник								
Н. контр.								
Зав. каф.								
					Літ.	Арк.	Аркушів	
						1	12	

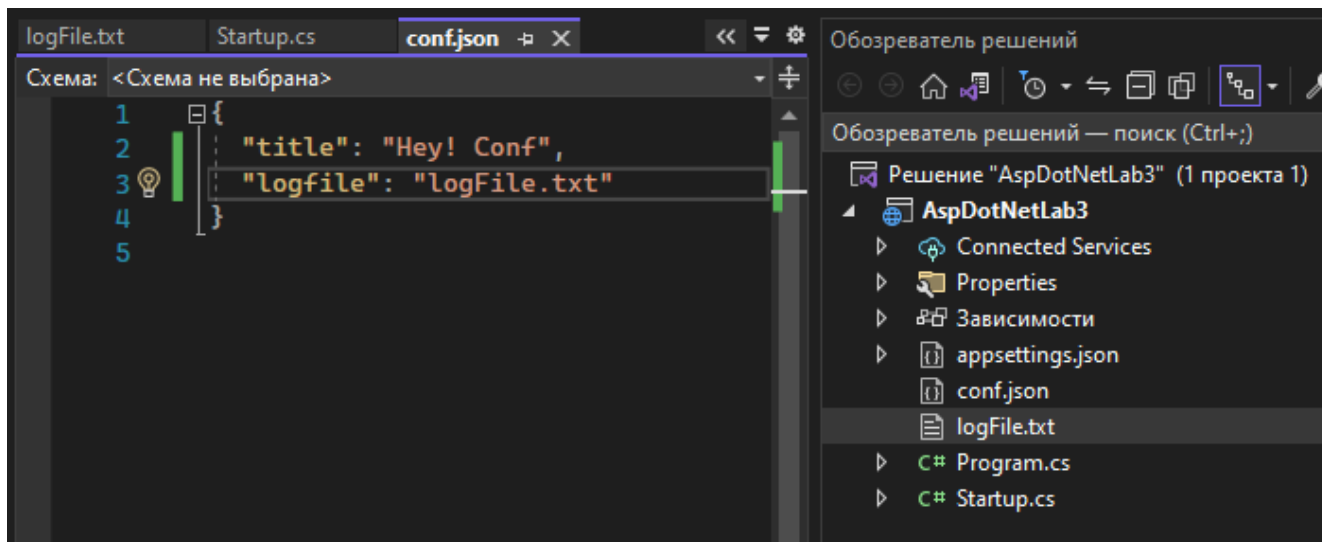


Рис. 2.1.1 Створений власний json-файл конфігурації з назвою conf.json

**Частина 2:** За допомогою MapRoute визначити маршрути та їх обробники для того, щоб працювали наступні маршрути:

Шаблони маршруту	Приклади
/Controller/Action/Id, де Id - необов'язковий параметр	/Site/Index /En/Site/Index /Uk/Site/Index
/Lang/Controller/Action/Id, де Id - необов'язковий параметр	/Book/Add /En/Book/Add /Uk/Book/Add
	/En/User/Delete/3 /Uk/User/Delete/3 /User/Delete/3

Рис. 2.2.1. Приклад шаблонів маршрутів, які мають оброблятися

```
var routeBuilder = new RouteBuilder(app);
routeBuilder.MapRoute("/{controller}/{action}/{id:int?}",
async context => {
    context.Response.ContentType = "text/html; charset=utf-8";
    await context.Response.WriteAsync($"Route \"{context.Request.Path}\" is valid");
});
routeBuilder.MapRoute("/{lang}/{controller}/{action}/{id:int?}",
async context => {
    context.Response.ContentType = "text/html; charset=utf-8";
    await context.Response.WriteAsync($"Route \"{context.Request.Path}\" is valid");
});
app.UseRouter(routeBuilder.Build());
```

Результат виконання:

		Палій І.В.			ДУ «Житомирська політехніка».21.121.08.000 – Лр2	Арк.
		Чижмоторя О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

localhost:7260/Site/Index  
Route /Site/Index is available

localhost:7260/En/Site/Index  
Route /En/Site/Index is available

localhost:7260/Uk/Site/Index  
Route /Uk/Site/Index is available

localhost:7260/Book/Add  
Route /Book/Add is available

localhost:7260/En/Book/Add  
Route /En/Book/Add is available

localhost:7260/En/User/Delete/3  
Route /En/User/Delete/3 is available

localhost:7260/User/Delete/3  
Route /User/Delete/3 is available

Рис. 2.2.2- 2.2.7. Результат обробки маршрутів

**Частина 3:** Визначити маршрути та реалізувати вказані дії при переході за ними:

		Палій І.В.			ДУ «Житомирська політехніка».21.121.08.000 – Лр2	Арк.
		Чижевська О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Маршрут	Дія
/Session/Add/Параметр/Значення	Додає у сесію значення заданого параметра
/Session/View/Параметр/Значення	Виводить значення заданого параметра з сесії. Якщо такого параметра немає, то вивести відповідне повідомлення
/Cookie/Add/Параметр/Значення	Додає у куки значення заданого параметра
/Cookie/View/Параметр/Значення	Виводить значення заданого параметра з куки. Якщо такого параметра немає, то вивести відповідне повідомлення

### Лістинг коду ConfigureServices:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDistributedMemoryCache();
    services.AddSession(options =>
    {
        options.IdleTimeout = TimeSpan.FromMinutes(30);
        options.Cookie.HttpOnly = true;
        options.Cookie.IsEssential = true;
    });
}
```

### Лістинг коду Startup.cs:

```
        routeBuilder.MapRoute("/session/add/{key}/{value}",
async context =>
{
    var key = $"{context.Request.Path}".Split("/")[3];
    var value = $"{context.Request.Path}".Split("/")[4];
    context.Session.SetString(key, value);
    await context.Response.WriteAsync($"Key with value added to the
session");
});
routeBuilder.MapRoute("/session/view/{key}",
async context =>
{
    var key = $"{context.Request.Path}".Split("/")[3];
    var value = context.Session.GetString(key);
    if (value != null)
    {
        await context.Response.WriteAsync($"<b>{key}</b> (key) =
{value}{value}");
        return;
    }
    await context.Response.WriteAsync($"No value was found for this key in
the session");
});

routeBuilder.MapRoute("/cookie/add/{key}/{value}",
async context =>
{
    var key = $"{context.Request.Path}".Split("/")[3];
    var value = $"{context.Request.Path}".Split("/")[4];
    context.Response.Cookies.Append(key, value);
    await context.Response.WriteAsync($"Key with value added to the
cookies");
});
```

		Палій І.В.			ДУ «Житомирська політехніка».21.121.08.000 – Пр2	Арк.
		Чижмоторя О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    );

    routeBuilder.MapRoute("/cookie/view/{key}",
    async context =>
    {
        var key = $"{context.Request.Path}.Split("/") [3];
        if (context.Request.Cookies.ContainsKey(key))
        {
            var value = context.Request.Cookies[key];
            await context.Response.WriteAsync($"<b>{key}</b> (key) = {value}
(value)");
            return;
        }
        await context.Response.WriteAsync($"No value was found for this key");
    });
app.UseRouter(routeBuilder.Build());

```

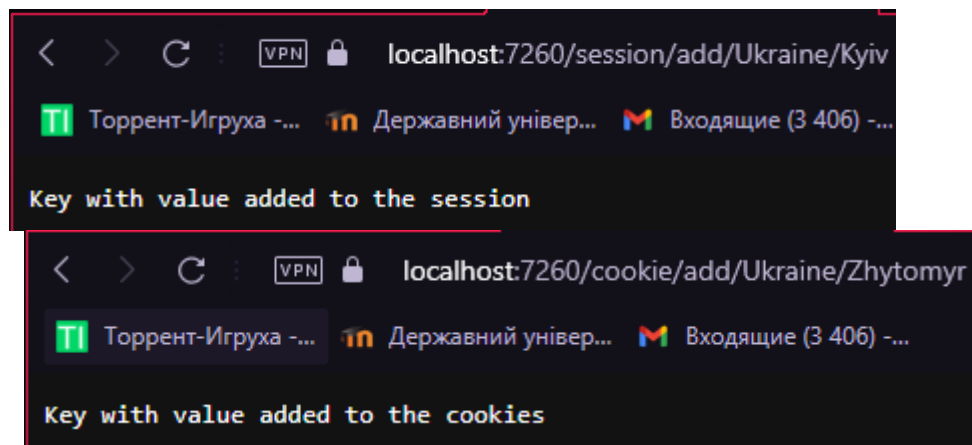


Рис. 2.3.1- 2.3.2. Додавання значення до сесії та куки.

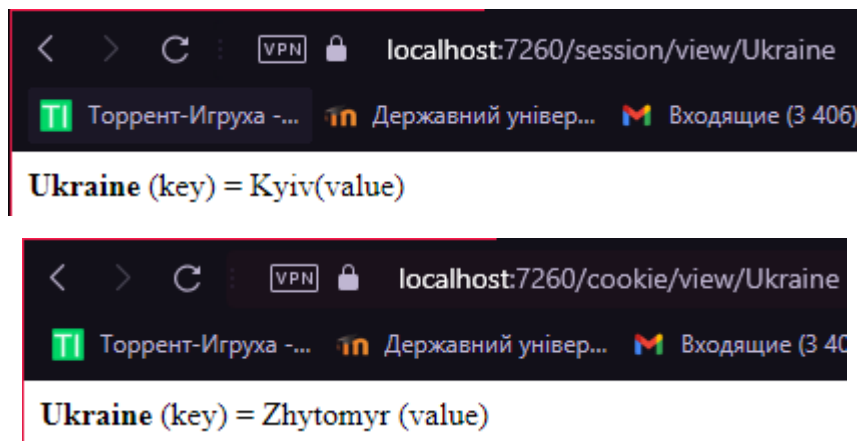
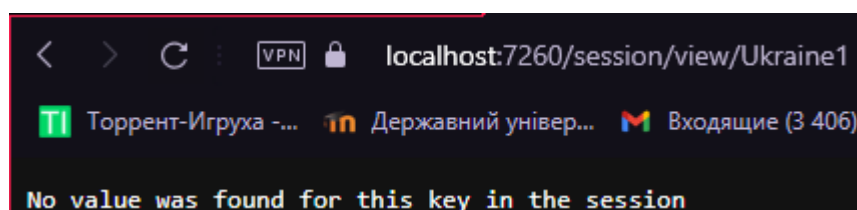


Рис. 2.3.3- 2.3.4.Діставання значення з сесії та куки.



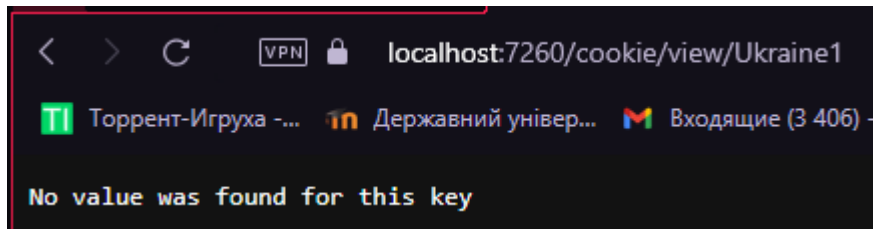


Рис. 2.3.5- 2.3.6. Спроба діставання неіснуючого значення з сесії та куки.

**Частина 4:** Створити власний файловий провайдер логування, який буде записувати інформацію про дату, час, IP-адресу користувача та шлях запиту до сервера в текстовий файл, назва якого задана в конфігурації в параметрі LogFile. При кожному запиті має додавати один рядок до файла.

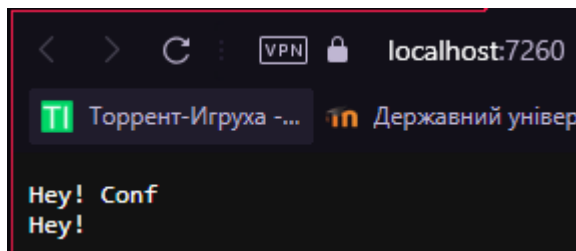


Рис. 2.4.1. Результат виконання програми (відображення при кожному маршруті привітання, яке знаходиться в конф. файлі)

logFile.txt	Program.cs	Startup.cs*	conf.json	AspDotNetLab3: Огляд
1	13.07.2022 18:23:14 / ::1 HTTP/2 GET			
2	13.07.2022 18:30:12 / ::1 HTTP/2 GET			
3	13.07.2022 18:31:49 /cookie/view/Ukraine1 ::1 HTTP/2 GET			
4	13.07.2022 18:32:35 /session/add/Ukraine/Kyiv ::1 HTTP/2 GET			
5	13.07.2022 18:33:08 /cookie/add/Ukraine/Zhytomyr ::1 HTTP/2 GET			
6	13.07.2022 18:34:08 /cookie/view/Ukraine ::1 HTTP/2 GET			
7	13.07.2022 18:34:46 /session/view/Ukraine1 ::1 HTTP/2 GET			
8	13.07.2022 18:35:52 /cookie/view/Ukraine1 ::1 HTTP/2 GET			

Рис. 2.4.2. Вміст файлу log.txt

### Завдання 3.

Запустити виконану роботу у репозиторій на GitHub за назвою AspDotNetLab1 і відкрити доступ для викладачів:

- [morozov@ztu.edu.ua](mailto:morozov@ztu.edu.ua)
- [4ov.ztu@gmail.com](mailto:4ov.ztu@gmail.com)

Посилання на репозиторій:

<https://github.com/IvanPaliy/AspDotNetLab3.git>

**Висновки:** в ході виконання лабораторної роботи я ознайомився з файловими та нефайловими провайдерами конфігурації в ASP.NET, принципами побудови

		Палій І.В.			ДУ «Житомирська політехніка».21.121.08.000 – Пр2	Арк.
		Чижмоторя О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

власних провайдерів логування, визначенням маршрутів, навчитися зберігати дані у сесіях.

		Палій І.В.			ДУ «Житомирська політехніка».21.121.08.000 – Лр2	Арк.
		Чижмотря О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		