

Хмарні технології. Лаборторна робота №1

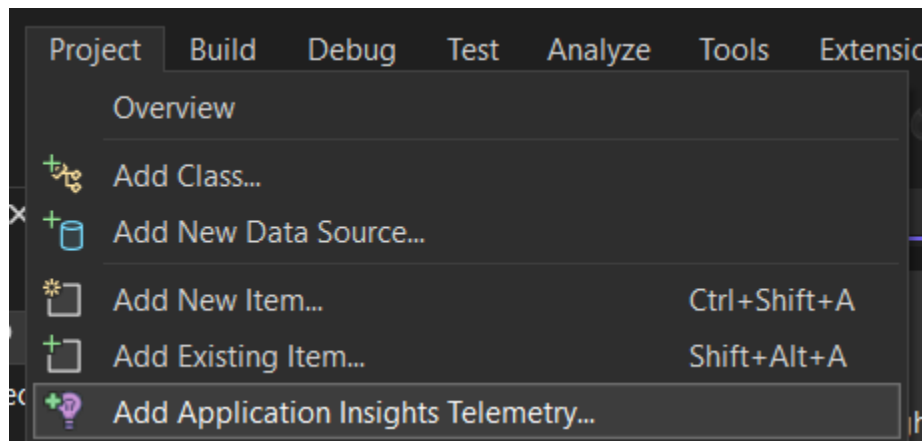
Використання хмарних технологій для оперативного моніторингу стану інформаційної системи з використанням Application Insights

Виконала: студентка групи МІТ-31, Панченко Владислава

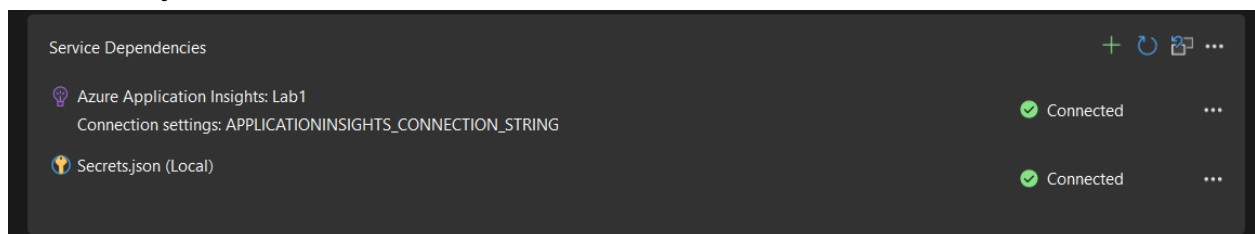
Завдання для виконання:

1. Створити тестовий веб додаток за шаблоном ASP.NET Core MVC.
2. Впровадити у веб додаток систему телеметрії Application Insights.
3. Виконати запити (в тому числі й хибні) до веб додатку та відстежити їх через засоби моніторингу та аналітики у Visual Studio та у хмарі.

Впровадження системи телеметрії Application Insights у веб додаток за шаблоном ASP.NET Core MVC. Project - Add Application Insights Telemetry...



Результат:



Код за замовчуванням у Program.cs:

```

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.AddApplicationInsightsTelemetry();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

```

Починаючи з версії .NET 6, файл Startup.cs не створється автоматично, оскільки його функціонал вже додано до файлу Program.cs. Тобто, код за замовчуванням не має автоматично доданого посилання, яке потрібно було би видалити в старішій версії. Також, телеметрія Application Insights автоматично додана до сервісів додатку (рядок 5: builder.Services.AddApplicationInsightsTelemetry();).

Виконання коректних запитів HTTP 200 OK: після запуску додатку, відкрити у браузері `http://localhost:(вказаний у launchSettings.json порт)/` або `http://localhost:(вказаний у launchSettings.json порт)/Home/Index`.
Результат виконання запиту:

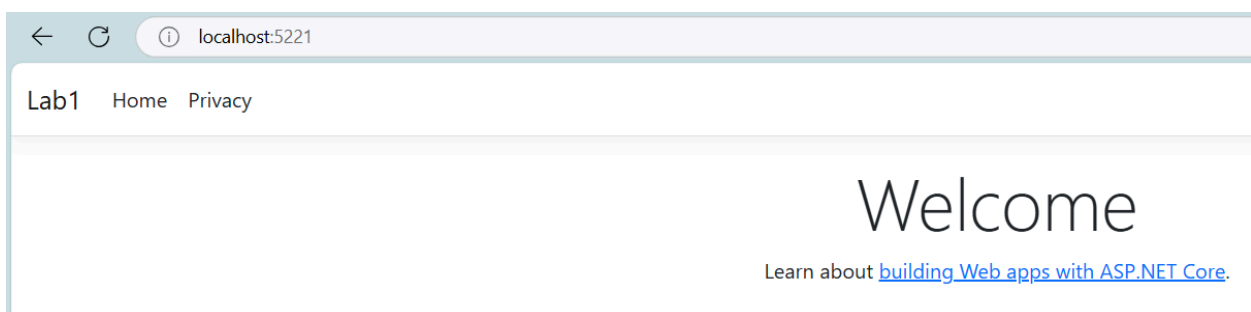
02/03/2025, 14:05:38 - REQUEST

GET Home/Index

URL: `http://localhost:5221/`

Response code: 200

Response time: 364.7 ms



Виконання хибних запитів:

- 1) Неіснуючий маршрут: вказати у браузері <http://localhost:5221/NotFoundPage> (5221 - вказаний у launchSettings.json порт, NotFoundPage - неіснуюча сторінка).
Результат: помилка 404 (Not Found).

02/03/2025, 14:16:10 - REQUEST

GET /NotFoundPage

URL: http://localhost:5221/NotFoundPage

Response code: 404

Response time: 3.1 ms

localhost:5221/NotFoundPage



Не вдалося знайти цю сторінку localhost

Веб-сторінку для веб-адреси не знайдено: <http://localhost:5221/NotFoundPage>

HTTP ERROR 404

- 2) Виклик помилки HTTP 500 (Internal Server Error). Додамо у HomeController.cs наступний метод:

```
public IActionResult ThrowError()
{
    try
    {
        throw new Exception("Тестова помилка!");
    }
    catch (Exception ex)
    {
        return View("Error", ex);
    }
}
```

Запускаємо додаток. Переходимо по посиланню
http://localhost:5221/Home/ThrowError та отримуємо:

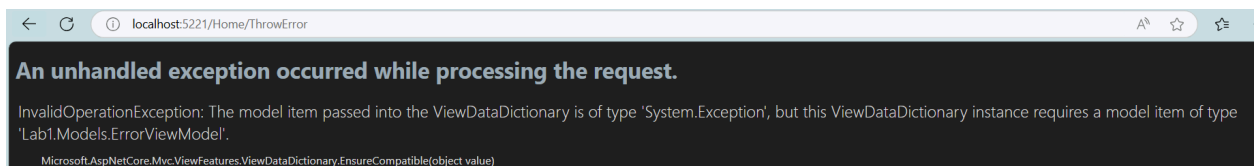
02/03/2025, 14:25:21 - REQUEST

GET Home/ThrowError

URL: http://localhost:5221/Home/ThrowError

Response code: 500

Response time: 211.0 ms



- 3) HTTP 400 – Неправильний запит. Додамо наступний метод:

```
public IActionResult BadRequestTest(string name)
{
    if (string.IsNullOrEmpty(name))
    {
        return BadRequest("Ім'я є обов'язковим!");
    }
    return Content($"Привіт, {name}");
}
```

Запускаємо додаток, переходимо до
http://localhost:5221/Home/BadRequestTest. Отримуємо

неправильний

запит:

02/03/2025, 14:35:39 - REQUEST

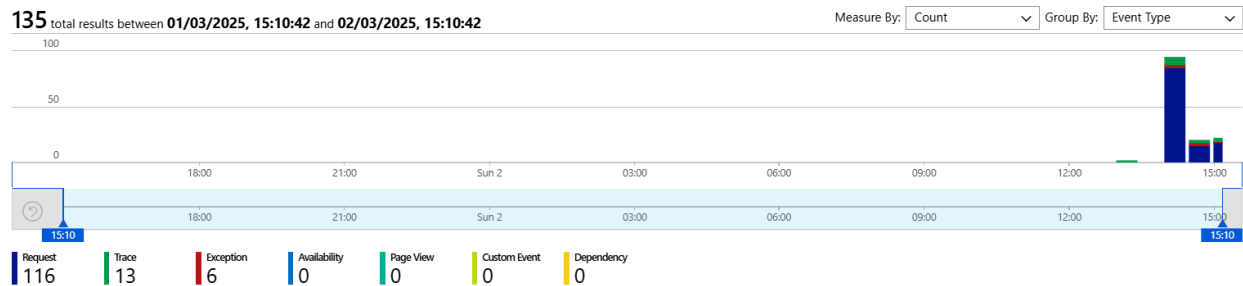
GET Home/BadRequestTest

URL: http://localhost:5221/Home/BadRequestTest

Response code: 400

Response time: 84.9 ms

Досліджуємо результати телеметрії:



Висновок: під час виконання цієї лабораторної роботи навчилися відстежувати виконання запитів, в тому числі і хибних, за допомогою Application Insights.