Logging

Last updated by | Vincent Boeijen | 6 feb 2024 at 21:28 CET

Logging

General

Logging is the process of recording information about the execution and behavior of a software application. Adding logging to an applications is a must to resolve production issues quickly and efficiently. It will also help a lot while debugging your application. Logging can help you with the following benefits:

- It can help you debug and troubleshoot problems by providing detailed and relevant information about the errors and exceptions that occur in your code.
- It can help you monitor and measure the performance and health of your application by providing metrics and indicators such as response time, memory usage, CPU load, etc.
- It can help you audit and track the activities and events that happen in your application by providing a chronological and traceable record of the actions and changes that affect your data and resources.
- It can help you comply with the regulations and standards that apply to your application by providing evidence and documentation of your application's functionality and security.

✋ DO use Serilog as Logging Framework

Why? Serilog provides diagnostic logging to files the console and elsewhere.

✋ DO implement a structured data format for logging

Why? It is easier to interact with them and can be more easily analyzed by a machine.

✋ DO use the LoggerMessage attribute

Why? This will make the logging faster because .NET will generate highly optimized code.

```
internal partial class Example
{
    private readonly ILogger<Example> logger;

    public Example(ILogger<Example> logger)
    {
        this.logger = logger;
    }

    public void Example()
    {
        LogSomeMessage(this.logger, "some value");
    }

    [LoggerMessage(1, LogLevel.Information, "Some log message {someValue}")]
    public static partial void LogSomeMessage(ILogger logger, string someValue);
}
```