Zero to Hero! Logging in .NET

Log Levels

Trace = 0 - Super detailed, disabled by default
Debug = 1 - Helpful only for active debugging
Information = 2 - Tracking general app flow
Warning = 3 - Abnormal or unexpected events
Error = 4 - Flow of application stopped due to failure
Critical = 5 - Unrecoverable error or crash
None = 6 - Specifies you want no logs

ILogger built into all programs

Use a LoggerFactory to initialize a logger for the application i.e. Console logger

Structured Logging! also called Semantic Logging
Allows us to capture context/scope instead of just a simple String

Log Category! Generally use class name

SetMinimumLevel allows you to choose which messages you will see.

EventId - Custom field you can set to help group logs.

You can add a host to a console app to mimic API behavior and call host.Services.GetRequiredService<ILogger> to get access to logger appsettings.json by default has logger configuration in it.

Log Filters allow you to customize which logs make it through based on custom attributes per provider.

You can use providers to target specific logging targets like Application Insights

currency ↓

Log Message Template formatting: ${Total:C}

You can use logger scopes to do custom things with a using statement for a chunk of code

You can create an IDisposable object using a Stopwatch to time operations

You can create a BackgroundService to change log levels dynamically while the application is running

Serilog logging Librery

- Serilog Provider equivalent is called Sinks.
- There are a ton more sinks than there are Ilogger Providers
- Serilog has a Static Log.Logger entry
- Serilog provides an enricher to add extra data to each log.
- @ Symbol for parameters gives json for parameter
- Can use operations to get times logged and you can use thresholds to log warnings/errors for operations taking too long
- You can mask Certain properties with [LogMasked] or remove it with [NotLogged] attribute
- You Can log async in WriteTo.Async()
- You can use LoggerMessage.Define to define message types for more performant logging
- Logging Source Generators allow highest performance