

# Effective Logging in ASP.NET Core

---

QUICK-START FOR LOGGING IN ASP.NET CORE



**Erik Dahl**

PRINCIPAL ARCHITECT

@dahlsailrunner [www.knowyourtoolset.com](http://www.knowyourtoolset.com)



# What Makes Logging “Effective”?

Good, consistent  
Information

Keeps application  
code clean

Easily consumable

Improves  
“fixability”

Enables deeper  
understanding

Adds data for  
prioritization



# Course Overview



**Get some entries created**

**Understand the framework**

**Automate/centralize where possible**

**Get complete information**

**Enable consumption**



# Module Overview



## What's “in the box”?

- WebHost.CreateDefaultBuilder
- ILogger interface and extension methods

## Demos, demos, demos!

- Default settings
- Add entry via ILogger
- Write to files via Serilog
- Using ILogger in other classes/assemblies



```
WebHost.CreateDefaultBuilder(args)
```

```
    .ConfigureLogging((hostingContext, logging) =>
    {
        logging.AddConfiguration(
            hostingContext.Configuration.GetSection("Logging"));
        logging.AddConsole();
        logging.AddDebug();
        logging.AddEventSourceLogger();
    })
```

## Template Code Includes Logger

**Following providers included by default and reads “Logging” config:**

- Console
- Debug
- EventSource



```
Log(ILogger, LogLevel, String, Object[])
```

```
...
```

```
LogCritical(ILogger, Exception, String, Object[])
```

```
LogCritical(ILogger, String, Object[])
```

```
LogCritical(ILogger, EventId, String, Object[])
```

```
...
```

## Extension Methods Used to Create Entries

Lots of “levels” supported and many overloads within the levels

`ILogger<T>` is accessed via dependency injection



# Demo



## Review application and code

- <https://github.com/dahlsailrunner/aspnetcore-effective-logging>

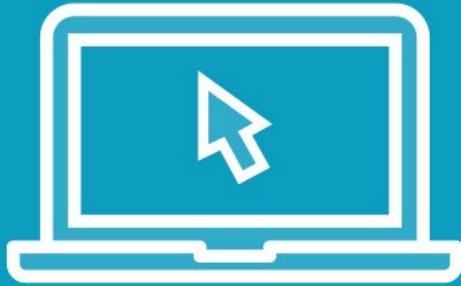
## Run application – check out entries in Debug output

## Add an entry using ILogger<T>

## Tweak appSettings.json



# Demo



**Update application to use Serilog**

- Write to files

**Adjust appSettings.json with different levels**

**Add log entries from a class library**





# Summary



Defined effective logging

Looked at default settings

Added simple log entries

Updated to use Serilog and write to files

Wrote an entry from a class library

Next up: Going deeper with  
`Microsoft.Extensions.Logging`

