Global Logging: ASP.NET Core

Erik Dahl
PRINCIPAL ARCHITECT

@dahlsailrunner www.knowyourtoolset.com

Overview

Covering three modules in one!

ToDo applications: MVC and (Secure) API

Log entry considerations

- Types
- Information

Demo: Apply global logging to both

Types of Log Entries: ASP.NET Web Apps

Usage

Track business functions

Instrument code for each item

Performance

Track every request

Plug into the pipeline (filter)

Error

Use custom middleware

Create an “Error” page

Careful with JavaScript!

Diagnostic

Put where you need it

Enable / Disable by config

Information to Log : ASP.NET Web Apps

Where from?

Product = ToDo

Layer = App Type

Location = Page/Action

Hostname = Server

Who is in play?

User = Logged in

Customer = N/A

What else?

Time spent

QueryString

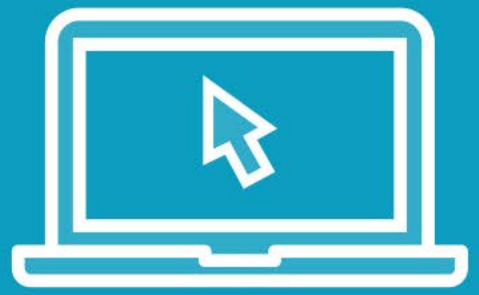
URL

Browser ?

Session ?

Cookie ?

Demo



Review the MVC and API apps

Create utility methods vs. UseSerilog()

- Create the logging methods
- Get data for the log entry
- Attributes, Filters

Add usage and diagnostics

Apply performance globally with a filter

Add custom exception handling middleware

Summary

Created utility methods for use in any ASP.NET Core website

- Even .NET Core console apps!

Log usage and diagnostic information

Global performance tracking via filter

Shielded exceptions with custom object and containing error ID

Up Next:
Sinks, log review, and analysis!