

Logging with ASP.NET Core



Damien Bowden Microsoft MVP

<https://damienbod.com>

[@damien_bod](https://twitter.com/damien_bod)

DIGICOMP



Why Logging?

...explain logging in ASP.NET Core & .NET Core

...how to add third party loggers

What Microsoft provides

Abstraction layer for logging in .NET Core

Wrapper for your favourite logger

Decouples your logging provider from your application

Startup.cs

0 references | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public void Configure(IApplicationBuilder app, ILoggerFactory loggerFactory)
{
    loggerFactory.AddConsole(Configuration.GetSection("Logging"));
    loggerFactory.AddDebug();

    app.UseStaticFiles();
    app.UseMvc();
}
```

In your Class

```
private readonly ILogger _logger;
```

0 references | damienbod, 1 day ago | 1 author, 4 changes | 0 exceptions

```
public ValuesController(ILogger<ValuesController> logger)
{
    _logger = logger;
    _logger.LogTrace("constructor ValuesController");
}
```

In your Class

0 references | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public ValuesController(ILoggerFactory loggerFactory)
{
    :   _logger = loggerFactory.CreateLogger("superCoolLogger");
    :   _logger.LogTrace("constructor ValuesController");
}
```

Log Levels

Trace = 0

Debug = 1

Information = 2

Warning = 3

Error = 4

Critical = 5

Default Logging Providers

Console

Debug

EventSource

EventLog

TraceSource

Azure App Service

Third Party Logging, Structured Logging



Serilog



NLog



elmah io



Third Party Logging, Structured Logging

Seq

Elasticsearch (ELK)

SQL databases

elmah.io

Syslog

ColoredConsole

Files

Example: NLog

0 references | damienbod, 4 days ago | 1 author, 1 change | 0 exceptions

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();
    services.AddMvc();
}
```

0 references | damienbod, 1 day ago | 1 author, 4 changes | 0 exceptions

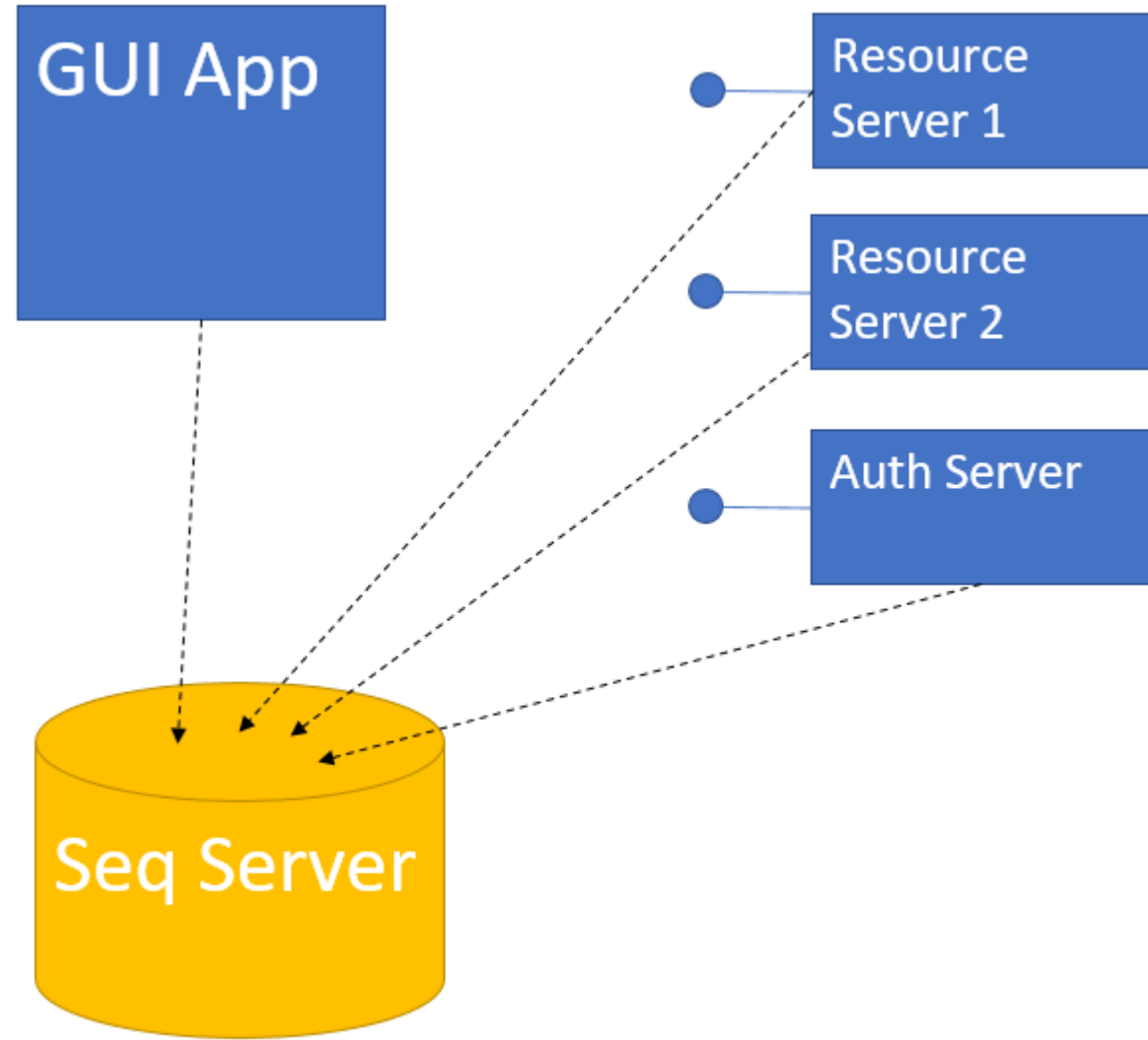
```
public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)
{
    loggerFactory.AddNLog();
    app.AddNLogWeb();

    LogManager.Configuration.Variables["connectionString"] = Configuration.GetConnectionString("NLogDb");
    LogManager.Configuration.Variables["configDir"] = "C:\\\\git\\\\dotnetday_AspNetCoreLogging\\\\examples\\\\Logs";
    LogManager.ConfigurationReloaded += updateConfig;

    app.UseStaticFiles();
    app.UseMvc();
}

<targets async="true">
    <target xsi:type="File" name="al
```

App Demo



ASP.NET Core Serilog

0 references | damienbod, 1 day ago | 1 author, 3 changes

```
public Startup(IHostingEnvironment env)
{
    MyLoggingLevelSwitch = new LoggingLevelSwitch();
    MyLoggingLevelSwitch.MinimumLevel = LogEventLevel.Verbose;

    Log.Logger = new LoggerConfiguration()
        .MinimumLevel.ControlledBy(MyLoggingLevelSwitch)
        .Enrich.WithProperty("App", "AspNetCoreResourceServer")
        .Enrich.FromLogContext()
        .WriteTo.Seq("http://localhost:5341")
        .WriteTo.RollingFile("../..../Logs/AspNetCoreResourceServer")
        .CreateLogger();
}
```

0 references | damienbod, 1 day ago | 1 author, 2 changes

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    // Add Serilog to the logging pipeline
    loggerFactory.AddSerilog();
}
```

- Serilog
- Serilog.Extensions.Logging
- Serilog.Sinks.Seq

Structured Logging

“Logs are worthless if you can’t use them”

```
var userName = HttpContext.User.FindFirst("username")?.Value;  
_logger.LogInformation("User requested all data {UserName}", userName);
```

Plan your logging architecture
and
review the logs / log levels

Resources and Links

- Examples <https://github.com/damienbod/AspNetCoreLogging>
- Microsoft <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/logging>
- Microsoft <https://github.com/aspnet/Logging/wiki/Guidelines>
- Microsoft <https://github.com/aspnet/Logging>
- Seq <http://blog.getseq.net/>
- Serilog <https://serilog.net/>
- Serilog <https://nblumhardt.com/2014/10/dynamically-changing-the-serilog-level/>
- NLog <http://nlog-project.org/>
- NLog <https://damienbod.com/2016/08/17/asp-net-core-logging-with-nlog-and-microsoft-sql-server/>
- NLog <https://tpodolak.com/blog/2017/02/18/asp-net-core-tracking-flow-requests-nlog/>
- ELK <https://www.elastic.co/webinars/introduction-elk-stack>
- elmah.io <https://elmah.io/>
- Structured Logging <https://messagetemplates.org/>
- Syslog <https://github.com/mguinness/syslog-framework-logging>
- PerfView <https://www.microsoft.com/en-us/download/details.aspx?id=28567>

Thank you