Log4Net:

1. Intsall  Microsoft.Extensions.Logging.Log4Net.AspNetCore. – from nuget pacjkage manager
2. Goto starup file (add namespace using Microsoft.Extensions.Logging;))and in configure () add the below coding

public void Configure(IApplicationBuilder app, IWebHostEnvironment env, ILoggerFactory loggerFactory)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

loggerFactory.AddLog4Net();

app.UseHttpsRedirection();

1. Add new item to your project add web configuration file with name log4net.config

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<log4net>

<root>

<level value="ALL" />

<appender-ref ref="RollingFile" />

</root>

<appender name="RollingFile" type="log4net.Appender.FileAppender">

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%-5p %d{hh:mm:ss} %message%newline" />

</layout>

</appender>

</log4net>

</configuration>

The above in yellow is file tag value where our log file path is present

1. In program.cs enter the following coding

public static void Main(string[] args)

{

var log4netRepository = log4net.LogManager.GetRepository(System.Reflection.Assembly.GetEntryAssembly());

log4net.Config.XmlConfigurator.Configure(log4netRepository, new FileInfo("log4net.config"));

CreateHostBuilder(args).Build().Run();

}

Namespace to be added:

1. using System.IO;
2. using Microsoft.Extensions.Logging;
3. using System.Reflection;
4. Now in controller setup \_log4net object
5. This object will access all log4net methods in controller
6. Here we have called \_log4net.Info() in Index()

using Log4netCore\_demo.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace Log4netCore\_demo.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

static readonly log4net.ILog \_log4net = log4net.LogManager.GetLogger(typeof(HomeController));

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

\_log4net.Info("Hello logging world log4net!");

return View();

}

public IActionResult Index1()

{

\_log4net.Info("Hello It's me!");

return View();

}

1. Now run the application() and see the log in D;\example.log
2. https://www.webtrainingroom.com/aspnetcore/logging
3. <https://www.c-sharpcorner.com/blogs/how-to-use-log4net-in-asp-net-core-application>
4. <https://jakubwajs.wordpress.com/2019/11/28/logging-with-log4net-in-net-core-3-0-console-app/> - log4net in console application
5. <https://blog.elmah.io/log4net-tutorial-the-complete-guide-for-beginners-and-pros/> - complete tutorial

