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

<configuration>

<log4net>

<root>

<level value="ALL" />

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

<appender name="console" type="log4net.Appender.ManagedColoredConsoleAppender">

<mapping>

<level value="INFO" />

<forecolor value="GREEN" />

</mapping>

<mapping>

<level value="WARN" />

<forecolor value="Yellow" />

</mapping>

<mapping>

<level value="ERROR" />

<forecolor value="Red" />

</mapping>

<mapping>

<level value="DEBUG" />

<forecolor value="Blue" />

</mapping>

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

<conversionPattern value="%date %level %logger.%method[%line] - %message%newline" />

</layout>

</appender>

<appender name="file" type="log4net.Appender.RollingFileAppender">

<file value="main.log" />

<appendToFile value="true" />

<rollingStyle value="Size" />

<maxSizeRollBackups value="5" />

<maximumFileSize value="25MB" />

<staticLogFileName value="true" />

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

<conversionpattern value="%level %message %date\n">

</conversionpattern>

</layout>

</appender>

</log4net>

</configuration>

using log4net;

using log4net.Config;

using System.Reflection;

//using Log4NetSample.LogUtility;

namespace CourseApi

{

public class Program

{

public static void Main(string[] args)

{

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

var logRepository = LogManager.GetRepository(Assembly.GetEntryAssembly());

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

builder.Services.AddControllers();

builder.Services.AddSwaggerGen();

var app = builder.Build();

var \_log4net = log4net.LogManager.GetLogger(typeof(Program));

\_log4net.Info("Hello Logging World");

// Configure the HTTP request pipeline.

app.UseAuthorization();

app.UseSwagger();

app.UseSwaggerUI();

app.MapControllers();

app.Run();

}

}

}

[HttpGet]

public List<Student> Get()

{

var \_log4net = log4net.LogManager.GetLogger(typeof(StudentController));

\_log4net.Error(list.Count);

return list;

}

<https://www.c-sharpcorner.com/article/how-to-configure-log4net-in-net-7-api/>

<https://code-maze.com/csharp-log4net-appenders-introduction/#:~:text=Log4net%20appenders%20are%20the%20components,%2C%20files%2C%20databases%2C%20etc>.