log4net日志是比较常用的第三方日志，.net core中整合了log4net

包：Microsoft.Extensions.Logging.Log4Net.AspNetCore

示例：

项目结构：



Program.cs：

static void Main (string[] args) {

    // 新建日志工厂

    ILoggerFactory loggerFactory = new LoggerFactory ();

    // 添加日志提供者，使用扩展方法 AddLog4Net 或 AddProvider 添加日志提供者

    loggerFactory.AddLog4Net ("log4net.xml");

// loggerFactory.AddProvider(new Log4NetProvider("log4net.xml"));

    // 生成 Logger

ILogger logger = loggerFactory.CreateLogger (nameof (Program));

    // 写入日志消息

logger.LogInformation ("这是一条log4net日志消息");

    Console.WriteLine ("Hello World!");

}

在log4net可以提供log4net的日志配置文件，这里我们提供了log4net.xml配置文件

Log4net.xml（记得设置文件始终复制）：

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

<log4net>

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

<file value="App\_Data/Logs/Logs.txt" />

<appendToFile value="true" />

<rollingStyle value="Size" />

<maxSizeRollBackups value="10" />

<maximumFileSize value="10000KB" />

<staticLogFileName value="true" />

<!-- 日志记录格式 -->

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

<conversionPattern value=" 【时间】:%d%n 【级别】:%p%n 【类名】:%c%n 【线程ID】: %thread %n 【日志内容】:%m%n 【日记详细】：%exception %n---------------------------------------------------------------------------------------------------------------%n"/>

</layout>

<!-- 级别过滤 -->

<filter type="log4net.Filter.LevelRangeFilter">

<levelMin value="INFO" />

<levelMax value="ERROR" />

</filter>

<!-- 字符串匹配过滤 -->

<!-- <filter type = "log4net.Filter.StringMatchFilter" >

<stringToMatch value = "IELog" />

</filter>

<filter type = "log4net.Filter.DenyAllFilter" /> -->

</appender>

<!-- 处理DEBUG级别以上日志 -->

<root>

<appender-ref ref="RollingFileAppender" />

<level value="DEBUG" />

</root>

<logger name="NHibernate">

<level value="WARN" />

</logger>

</log4net>