**包**

微软老大提供了Microsoft.Extensions.Logging包，我们所需的日志功能

**涉及概念**

ILoggerFactory：日志工厂，用于生成记录器（ILogger）

ILogger：记录器，用于记录日志

ILoggerProvider：日志提供者，携带记录器

**记录日志**

1. 从ILoggerFactory创建ILogger

我们可以使用ILoggerFactory创建ILogger来记录日志

var app = builder.Build();

// 全局记录日志

app.Use(async (httpContent, next) =>

{

    var loggerFactory = httpContent.RequestServices.GetService<ILoggerFactory>();

    var logger = loggerFactory?.CreateLogger("Icewx");

    logger?.LogInformation($"Method: {httpContent.Request.Method} | Url: {httpContent.Request.Path}{httpContent.Request.QueryString}");

    await next();

});

1. 使用依赖注入获取ILogger

注：使用依赖注入获取的ILogger必须指定泛型，如ILogger<WxController>

[Route("[controller]")]

[ApiController]

public class WxController : ControllerBase

{

    private readonly ILogger \_logger;

    // 注入 logger

    public WxController(ILogger<WxController> logger) {

        \_logger = logger;

    }

    [HttpPost]

    public WxOutput Post(WxInput input) {

        // 记录日志

        \_logger.LogInformation($"Url: /Wx | body: {System.Text.Json.JsonSerializer.Serialize(input)}");

        return new WxOutput();

    }

}

注：Asp.net core 默认的日志是debug提供者，如果想把日志记录到文件，请往下看Log4net

**Log4net 与 Asp.net core**

微软提供的ILoggerProvider不是很好用，多数的企业一般使用Log4net，如下示例

1. 安装包：Microsoft.Extensions.Logging.Log4Net.AspNetCore
2. 新建log4配置文件log4net.xml

<?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>

    </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>

1. 添加log4提供者

var builder = WebApplication.CreateBuilder(args);

....

// 添加log4net提供者

builder.Logging.AddLog4Net("log4net.xml");