Skip to main content

Microsoft.Extensions.Logging by Microsoft

Details

Info

> **INFO**
>
> Author: Microsoft
>
> NuGet: https://www.nuget.org/packages/Microsoft.Extensions.Logging/
>
> You can find more details at https://learn.microsoft.com/en-us/dotnet/core/extensions/logger-message-generator-generators/
>
> Source : https://github.com/dotnet/runtime

About

> **NOTE**
>
> Logging defined and compiled

How to use

Example (source csproj, source files)

CSharp Project | Program.cs | LogDemo.cs

This is the CSharp Project that references Microsoft.Extensions.Logging

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net7.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
    <PropertyGroup>
        <EmitCompilerGeneratedFiles>true</EmitCompilerGeneratedFiles>

<CompilerGeneratedFilesOutputPath>$(BaseIntermediateOutputPath)\GeneratedX</CompilerG
    </PropertyGroup>
    <ItemGroup>
        <PackageReference Include="Microsoft.Extensions.Logging.Console" Version="7.0
    </ItemGroup>
</Project>
```

This is the use of Microsoft.Extensions.Logging in Program.cs

```
using System.Text.Json;
using Microsoft.Extensions.Logging;
```

This is the use of Microsoft.Extensions.Logging in LogDemo.cs

```
using Microsoft.Extensions.Logging;

public partial class LoggingSample
{
    private readonly ILogger _logger;

    public LoggingSample(ILogger logger)
    {
        _logger = logger;
    }

    [LoggerMessage(
        EventId = 20,
        Level = LogLevel.Critical,
        Message = "Value is {value:E}")]
    public static partial void UsingFormatSpecifier(
        ILogger logger, double value);

    [LoggerMessage(
        EventId = 9,
        Level = LogLevel.Trace,
        Message = "Fixed message",
        EventName = "CustomEventName")]
    public partial void LogWithCustomEventName();

    [LoggerMessage(
        EventId = 10,
        Message = "Welcome to {city} {province}!")]
    public partial void LogWithDynamicLogLevel(
        string city, LogLevel level, string province);

    public void  TestLogging()
    {
        LogWithCustomEventName();
```

Generated Files

Those are taken from $(BaseIntermediateOutputPath)\GeneratedX

LoggerMessage.g.cs

```
// <auto-generated/>
#nullable enable

    partial class LoggingSample
    {
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.Extensions
        private static readonly global::System.Action<global::Microsoft.Extensions.Log
global::System.Exception?> __UsingFormatSpecifierCallback =

global::Microsoft.Extensions.Logging.LoggerMessage.Define<global::System.Double>(globa
new global::Microsoft.Extensions.Logging.EventId(20, nameof(UsingFormatSpecifier)), "
global::Microsoft.Extensions.Logging.LogDefineOptions() { SkipEnabledCheck = true });

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.Extensions
        public static partial void UsingFormatSpecifier(global::Microsoft.Extensions.
        {
            if (logger.IsEnabled(global::Microsoft.Extensions.Logging.LogLevel.Critic
            {
                __UsingFormatSpecifierCallback(logger, value, null);
            }
        }
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.Extensions
        private static readonly global::System.Action<global::Microsoft.Extensions.Log
__LogWithCustomEventNameCallback =
            global::Microsoft.Extensions.Logging.LoggerMessage.Define(global::Microso
global::Microsoft.Extensions.Logging.EventId(9, "CustomEventName"), "Fixed message",
global::Microsoft.Extensions.Logging.LogDefineOptions() { SkipEnabledCheck = true });
```

Usefull

Download Example

Download Example Microsoft.Extensions.Logging

Download PDF

Download PDF Microsoft.Extensions.Logging

Share this page

https://ignatandrei.github.io/RSCG_Examples/v2/docs/Microsoft.Extensions.Logging