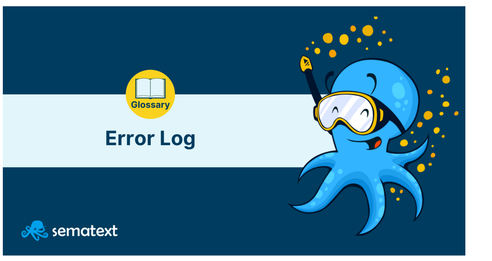
在.net中通过Microsoft.Extensions.Logging将日志保存到数据库方法（以mysql为例）

作者：微信公众号：【架构师老卢】 11-14 10:20 138



概述： 在.NET中，Microsoft.Extensions.Logging是一个灵活的日志库，它允许你将日志信息记录到各种不同的目标，包括数据库。在这个示例中，我将详细介绍如何使用Microsoft.Extensions.Logging将日志保存到MySQL数据库。我们将使用Entity Framework Core来与MySQL数据库进行交互。

在.NET中，Microsoft.Extensions.Logging是一个灵活的日志库，它允许你将日志信息记录到各种不同的目标，包括数据库。在这个示例中，我将详细介绍如何使用Microsoft.Extensions.Logging将日志保存到MySQL数据库。我们将使用Entity Framework Core来与MySQL数据库进行交互。

步骤一：创建.NET Core项目

首先，我们需要创建一个.NET Core项目。你可以使用Visual Studio、Visual Studio Code或者命令行工具来创建项目。在创建项目时，确保选择一个合适的项目类型，比如控制台应用程序或Web应用程序，以便测试和演示日志记录到MySQL数据库的功能。

步骤二：安装必要的NuGet包

为了能够将日志记录到MySQL数据库，我们需要安装一些必要的NuGet包。打开项目的.csproj文件，添加以下包引用：

```
<ItemGroup>
    <PackageReference Include="Microsoft.Extensions.Logging" Version="5.0.0" />
    <PackageReference Include="Microsoft.Extensions.Logging.Abstractions" Version="5.0.0" />
    <PackageReference Include="Microsoft.Extensions.Logging.Console" Version="5.0.0" />
    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="5.0.0" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="5.0.0" />
    <PackageReference Include="MySql.EntityFrameworkCore" Version="5.0.5" />
</ItemGroup>
```

这些包包括Microsoft.Extensions.Logging用于日志记录，Microsoft.EntityFrameworkCore和MySql.EntityFrameworkCore用于与MySQL数据库进行交互。

运行以下命令以还原项目中的NuGet包：

```
dotnet restore
```

步骤三：配置日志记录

在项目的Program.cs文件中，配置Logger和数据库上下文。以下是示例代码：

```
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Configuration;

class Program
{
    static void Main()
    {
        var serviceProvider = new ServiceCollection()
            .AddLogging(builder =>
            {
                builder.AddConsole(); // 输出到控制台
                builder.AddMySqlDatabase("Server=localhost;Database=mydatabase;User=root;Password=mypassword;");
            })
            .BuildServiceProvider();

        var logger = serviceProvider.GetRequiredService<ILogger<Program>>();

        logger.LogInformation("这是一条信息日志");
        logger.LogWarning("这是一条警告日志");
        logger.LogError("这是一条错误日志");
    }
}
```

在上述代码中，我们首先创建了一个ServiceCollection，然后配置了Logger以将日志输出到控制台和MySQL数据库。在AddMySqlDatabase方法中，我们传递了MySQL数据库的连接字符串。你需要将其替换为你自己的数据库连接信息。

步骤四：创建数据库上下文

我们需要创建一个数据库上下文，以便Entity Framework Core知道如何与MySQL数据库进行交互。创建一个名为AppDbContext的类，继承自DbContext，并添加一个DbSet来表示日志表。以下是示例代码：

```
using Microsoft.EntityFrameworkCore;

public class AppDbContext : DbContext
{
    public DbSet<LogEntry> LogEntries { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseMySql("Server=localhost;Database=mydatabase;User=root;Password=mypassword;");
    }
}
```

在上述代码中，我们定义了一个LogEntries DbSet来表示日志表。我们还在OnConfiguring方法中配置了数据库连接字符串。

步骤五：创建日志实体

我们需要创建一个表示日志的实体类。创建一个名为LogEntry的类，包括一些基本属性，如时间戳、日志级别和消息。以下是示例代码：

```
public class LogEntry
{
    public int Id { get; set; }
    public DateTimeOffset Timestamp { get; set; }
    public LogLevel LogLevel { get; set; }
    public string Message { get; set; }
}
```

在上述代码中，我们定义了Id、Timestamp、LogLevel和Message属性来存储日志信息。

步骤六：编写MySQL日志提供程序

为了将日志记录到MySQL数据库，我们需要编写一个自定义的日志提供程序。创建一个名为MySqlDatabaseLoggerProvider的类，继承自ILoggerProvider，并实现相关方法。以下是示例代码：

```
using System;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.EntityFrameworkCore;

public class MySqlDatabaseLoggerProvider : LoggerProvider
{
    private readonly IServiceProvider _serviceProvider;

    public MySqlDatabaseLoggerProvider(IServiceProvider serviceProvider)
    {
        _serviceProvider = serviceProvider;
    }

    public override ILogger CreateLogger(string categoryName)
    {
        return new MySqlDatabaseLogger(categoryName, _serviceProvider);
    }

    public override void Dispose()
    {
    }
}
```

在上述代码中，我们创建了一个MySqlDatabaseLoggerProvider类，它负责创建MySqlDatabaseLogger实例。

步骤七：编写MySQL日志记录器

创建一个名为MySqlDatabaseLogger的类，继承自ILogger，并实现相关方法。以下是示例代码：

```
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using Microsoft.EntityFrameworkCore;
using System;

public class MySqlDatabaseLogger : ILogger
{
    private readonly string _categoryName;
    private readonly IServiceProvider _serviceProvider;

    public MySqlDatabaseLogger(string categoryName, IServiceProvider serviceProvider)
    {
        _categoryName = categoryName;
        _serviceProvider = serviceProvider;
    }

    public IDisposable BeginScope<TState>(TState state)
    {
        return null;
    }

    public bool IsEnabled(LogLevel logLevel)
    {
        return true;
    }

    public void Log<TState>(LogLevel logLevel, EventId eventId, TState state, Exception exception, Func<TState, Exception, string> formatter)
    {
        if (!IsEnabled(logLevel))
        {
            return;
        }

        var message = formatter(state, exception);
        var timestamp = DateTimeOffset.Now;

        using (var scope = _serviceProvider.CreateScope())
        {
            var dbContext = scope.ServiceProvider.GetRequiredService<AppDbContext>();
            dbContext.LogEntries.Add(new LogEntry
            {
                Timestamp = timestamp,
                LogLevel = logLevel,
                Message = message
            });
            dbContext.SaveChanges();
        }
    }
}
```

在上述代码中，我们创建了一个MySqlDatabaseLogger类，它实现了ILogger接口的方法。在Log方法中，我们将日志消息保存到

MySQL数据库中。

步骤八：注册MySQL日志提供程序

在Program.cs文件中，我们需要注册自定义的MySQL日志提供程序。以下是示例代码：

```
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Configuration;

class Program
{
    static void Main()
    {
        var serviceProvider = new ServiceCollection()
            .AddLogging(builder =>
            {
                builder.AddConsole(); // 输出到控制台
                builder.AddProvider(new MySqlDatabaseLoggerProvider(serviceProvider));
            })
            .BuildServiceProvider();

        var logger = serviceProvider.GetRequiredService<ILogger<Program>>();

        logger.LogInformation("这是一条信息日志");
        logger.LogWarning("这是一条警告日志");
        logger.LogError("这是一条错误日志");
    }
}
```

在上述代码中，我们通过AddProvider方法注册了自定义的MySQL日志提供程序。

步骤九：运行应用程序

现在，你可以运行应用程序，它将记录日志到MySQL数据库中。你可以在数据库中查看日志信息并进行分析。

通过上述步骤，你可以将日志记录到MySQL数据库中，使用了Microsoft.Extensions.Logging、Entity Framework Core和自定义的日志提供程序。这使你能够更灵活地管理日志，并能够轻松地将日志信息保存到数据库中以供进一步分析和监控。