

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

<log4net>

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

<lockingModel type="log4net.Appender.FileAppender+MinimalLock"/>

<file value="D:\Log4NetTest\" />

<datePattern value="yyyy-MM-dd.'txt'"/>

<staticLogFileName value="false"/>

<appendToFile value="false"/>

<rollingStyle value="Date"/>

<maxSizeRollBackups value="100"/>

<maximumFileSize value="15MB"/>

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

<conversionPattern value="%date [%thread] %-5level App %newline %message %newline %newline"/>

</layout>

</appender>

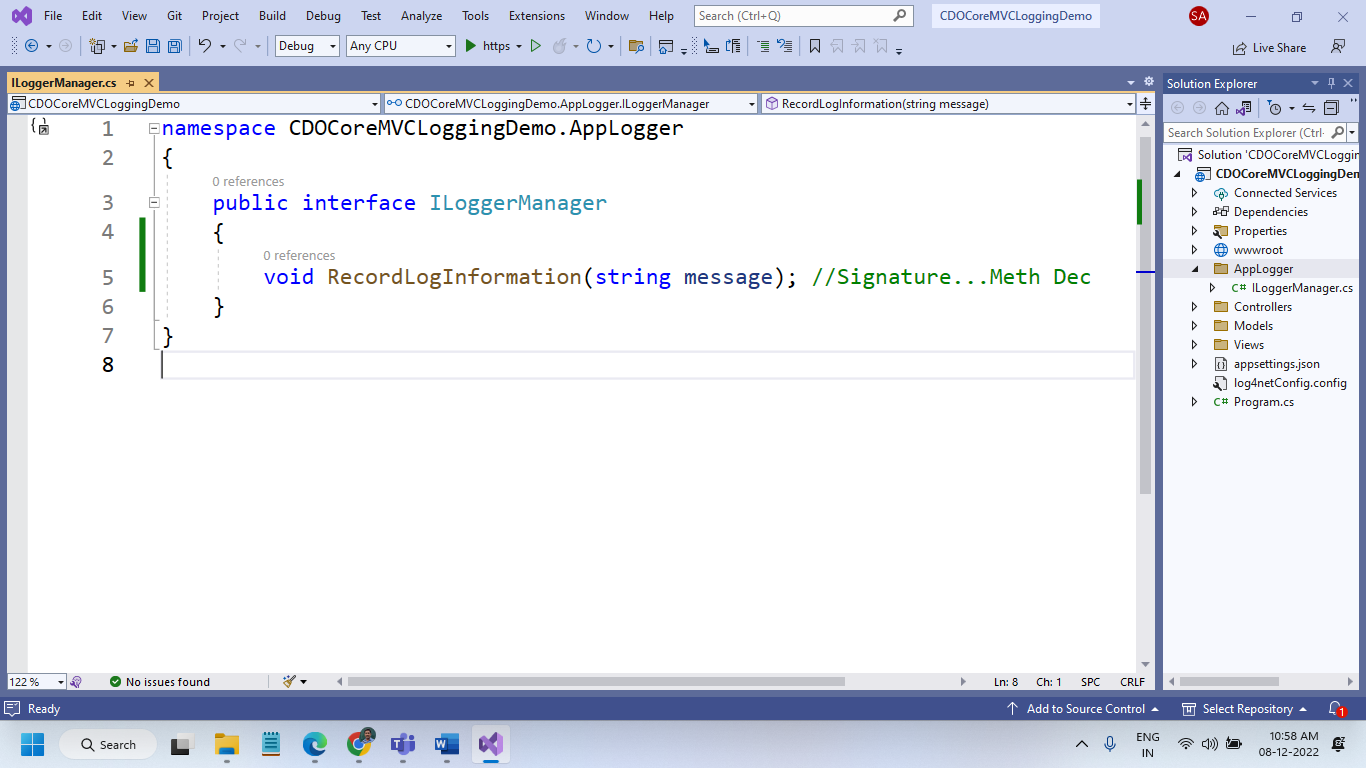
<root>

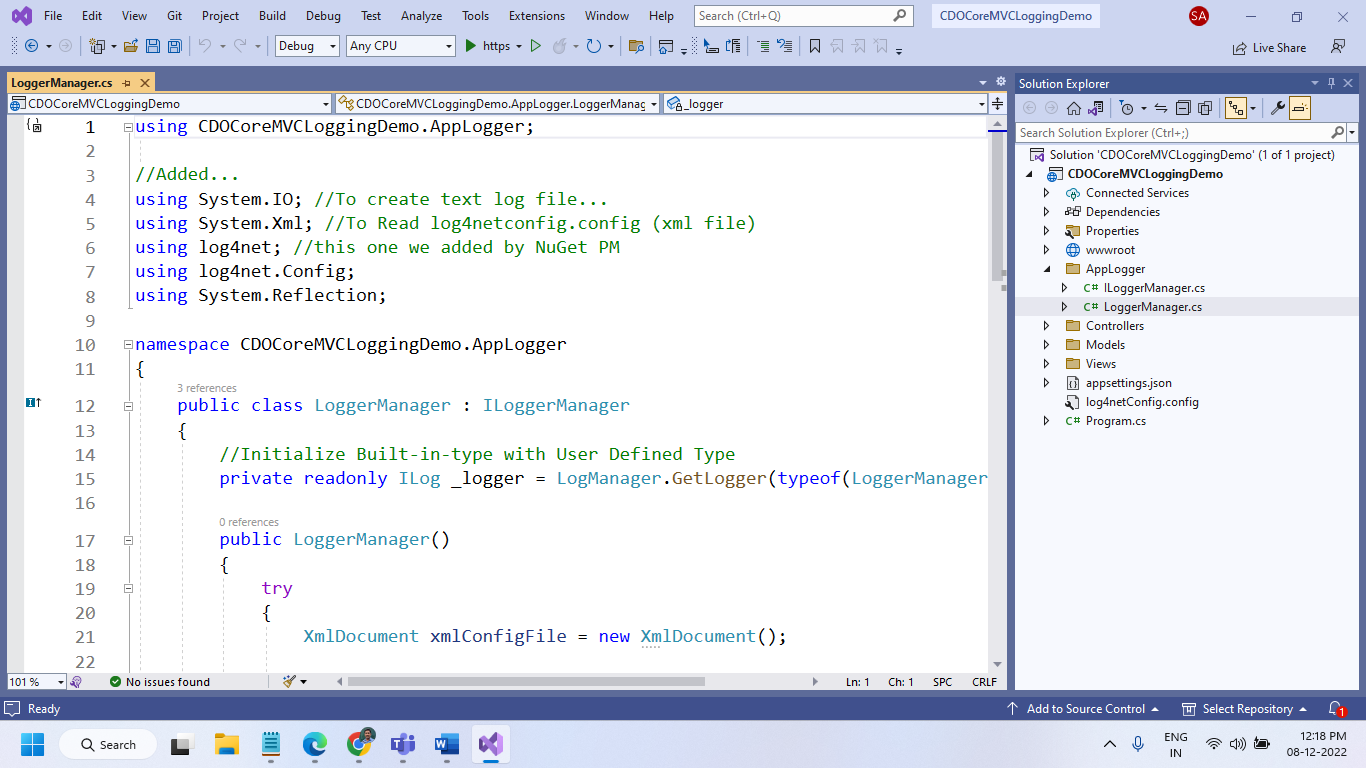
<level value="INFO"/>

<appender-ref ref="RollingLogFileAppender"/>

</root>

</log4net>





using CDOCoreMVCLoggingDemo.AppLogger;

//Added...

using System.IO; //To create text log file...

using System.Xml; //To Read log4netconfig.config (xml file)

using log4net; //this one we added by NuGet PM

using log4net.Config;

using System.Reflection;

namespace CDOCoreMVCLoggingDemo.AppLogger

{

public class LoggerManager : ILoggerManager

{

//Initialize Built-in-type with User Defined Type

private readonly ILog \_logger = LogManager.GetLogger(typeof(LoggerManager));

public LoggerManager()

{

try

{

XmlDocument xmlConfigFile = new XmlDocument();

using (var fs = File.OpenRead("log4netConfig.config"))

{

//Loading the configuration file

xmlConfigFile.Load(fs);

//Creating LogManager repository

var repo = LogManager.CreateRepository(Assembly.GetEntryAssembly(), typeof(LogManager));

//Creating Configurator

XmlConfigurator.Configure(repo, xmlConfigFile["log4net"]);

//Leaving Very first message

\_logger.Info("Log System Initialized");

}

}

catch (Exception ex)

{

\_logger.Error("Error", ex);

}

}

public void RecordLogInformation(string message)

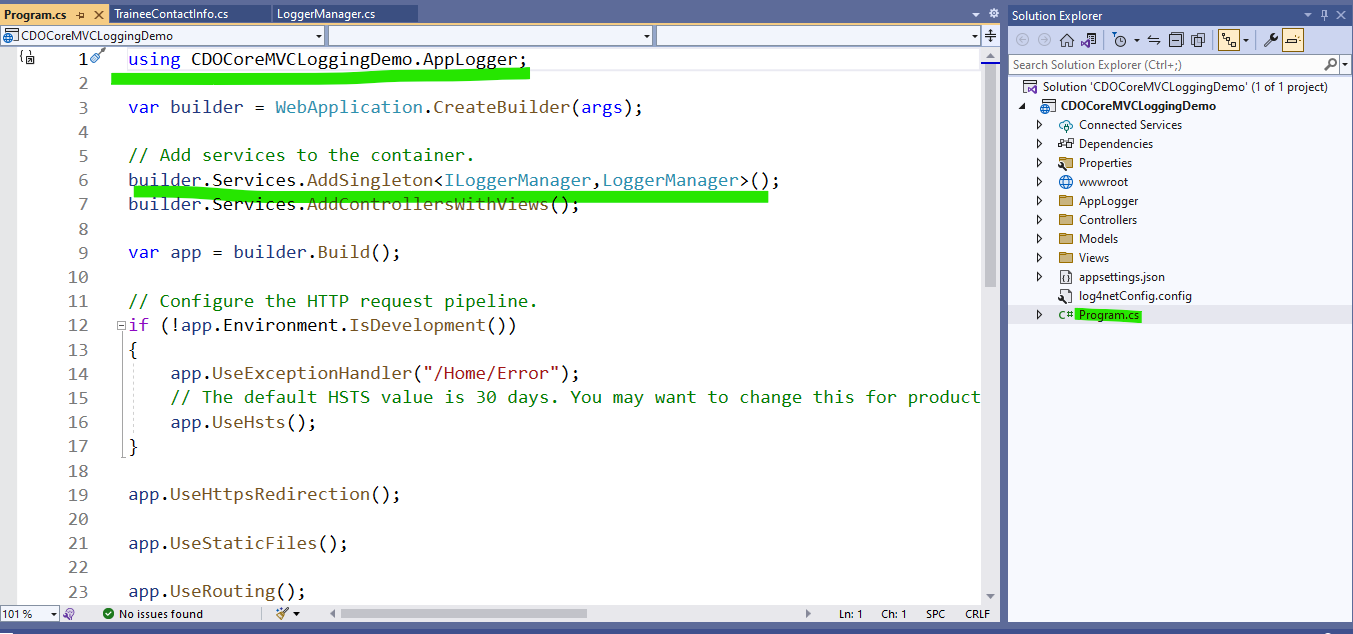
{

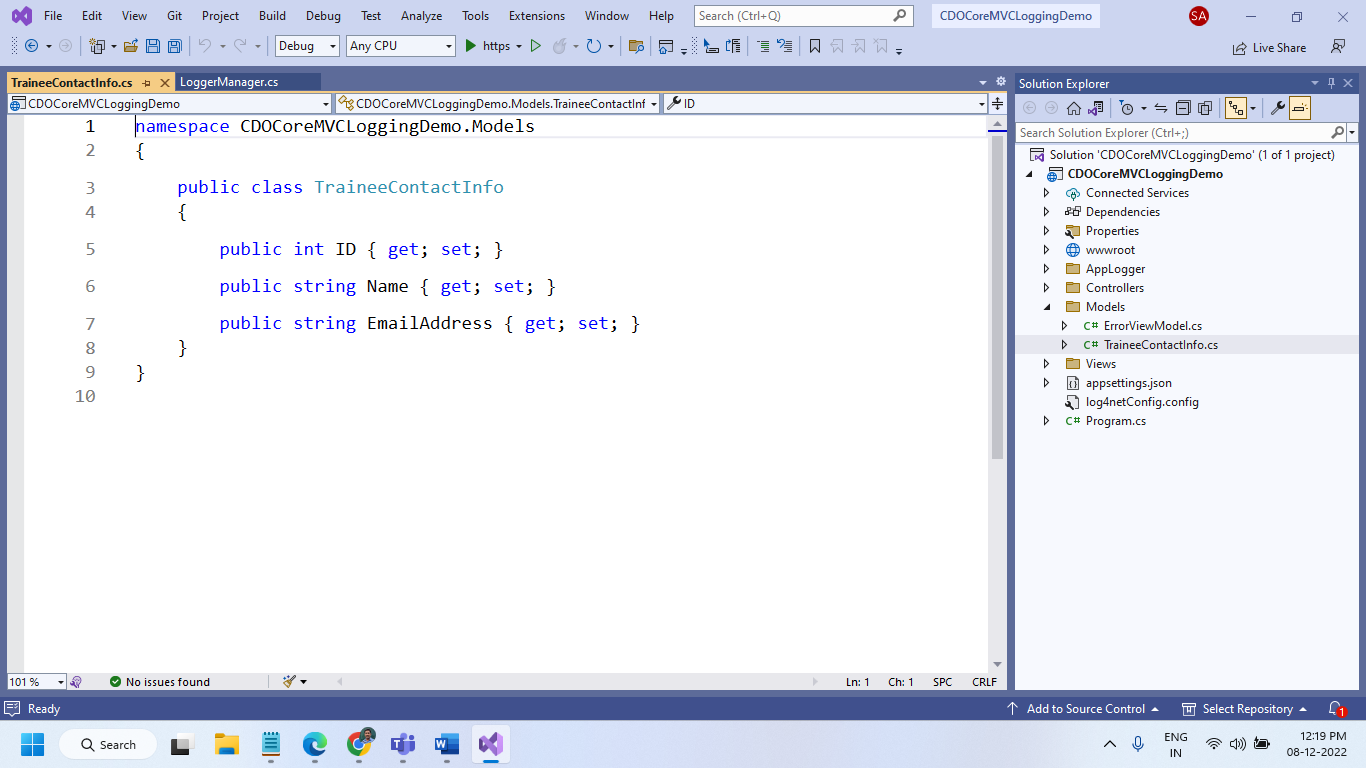
\_logger.Info(message);

}

}

}





namespace CDOCoreMVCLoggingDemo.Models

{

public class TraineeContactInfo

{

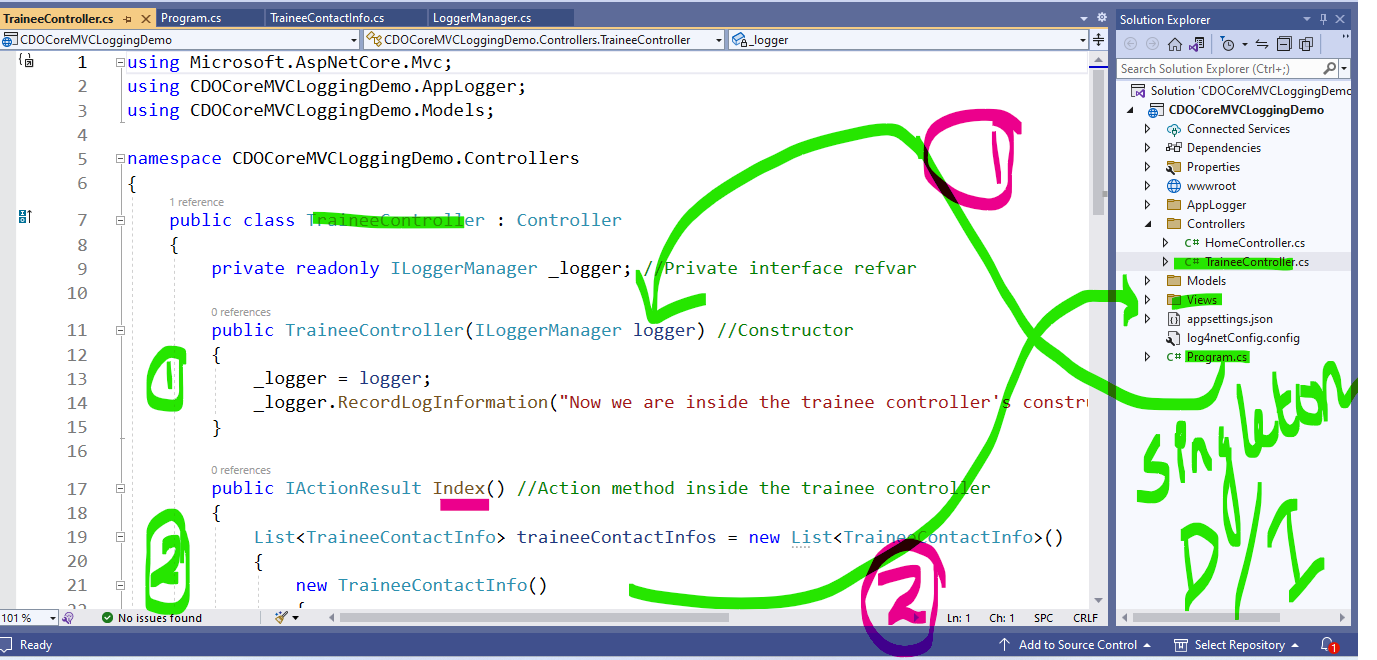
public int ID { get; set; }

public string Name { get; set; }

public string EmailAddress { get; set; }

}

}



using Microsoft.AspNetCore.Mvc;

using CDOCoreMVCLoggingDemo.AppLogger;

using CDOCoreMVCLoggingDemo.Models;

namespace CDOCoreMVCLoggingDemo.Controllers

{

public class TraineeController : Controller

{

private readonly ILoggerManager \_logger; //Private interface refvar

public TraineeController(ILoggerManager logger) //Constructor

{

\_logger = logger;

\_logger.RecordLogInformation("Now we are inside the trainee controller's constructor"); //2nd message

}

public IActionResult Index() //Action method inside the trainee controller

{

List<TraineeContactInfo> traineeContactInfos = new List<TraineeContactInfo>()

{

new TraineeContactInfo()

{

ID=1001,

Name = "Megala",

EmailAddress = "megala@gmail.com"

},

new TraineeContactInfo()

{

ID=1002,

Name = "Sheik",

EmailAddress = "sheik@gmail.com"

},

new TraineeContactInfo()

{

ID=1003,

Name = "Aryan",

EmailAddress = "aryan@gmail.com"

},

new TraineeContactInfo()

{

ID=1004,

Name = "Kusha",

EmailAddress = "Kushal@gmail.com"

}

};

\_logger.RecordLogInformation($"{traineeContactInfos.Count} records created. Passed to view for rendering");

return View(traineeContactInfos);

}

}

}

