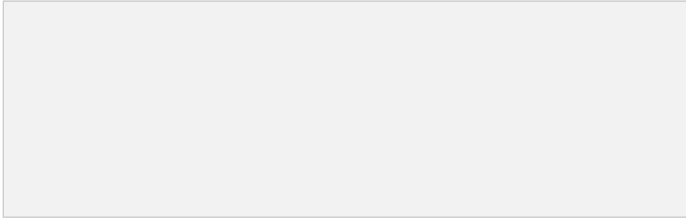


Logging in ASP.NET Core

1. Logging API.....	2
2. Logging Providers.....	4
3. Log Levels.....	5
4. Nlog.....	6

1. Logging API

- Logging plays important role in every application. When we encounter with errors, Logging helps us for solving them. Logs can be stored in text file, database or other sources.
- Microsoft provides logging API as an extension in the wrapper
- **Microsoft.Extensions.Logging** includes the necessary classes and interfaces for logging. The most important are the ILogger, ILoggerFactory, ILoggerProvider interfaces and the LoggerFactory class.
- The following figure shows the relationship between logging classes :



1. ILoggerFactory

- The ILoggerFactory is the factory interface for creating an appropriate ILogger type instance and also for adding the ILoggerProvider instance.
- The Logging API includes the built-in LoggerFactory class that implements the ILoggerFactory interface.
- We can use it to add an instance of type ILoggerProvider and to retrieve the ILogger instance for the specified category.

2. ILoggerProvider

- The ILoggerProvider manages and creates an appropriate logger, specified by the logging category.

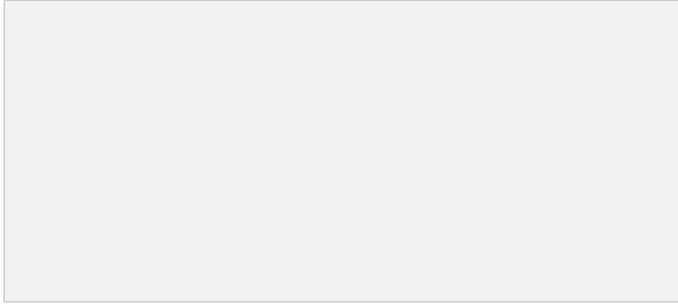
3. ILogger

- The ILogger interface includes methods for logging to the underlying storage. There are many extension methods which make logging easy.

4. Logging Providers

A logging provider displays or stores logs to a particular medium such as a console, a debugging event, an event log, a trace listener, and others.

5. Console Logging Provider



- As you can see in the above figure, the `ConsoleLogger` implements `ILogger`, while the `ConsoleLoggingProvider` implements `ILoggingProvider`. The `ConsoleLoggerExtensions` class includes extension method `AddConsole()`, which adds a console logger to the `LoggerFactory`.

2. Logging Providers

A logging provider displays or stores logs to a particular medium such as a console, a debugging event, an event log, a trace listener, and others.

The following table lists important logging providers :

Logging Provider's NuGet Package	Output Target
Microsoft.Extensions.Logging.Console	Console
Microsoft.Extensions.Logging.AzureAppServices	Azure App Services 'Diagnostics logs' and 'Log stream' features
Microsoft.Extensions.Logging.Debug	Debugger Monitor
Microsoft.Extensions.Logging.EventLog	Windows Event Log
Microsoft.Extensions.Logging.EventSource	EventSource/EventListener
Microsoft.Extensions.Logging.TraceSource	Trace Listener

3. Log Levels

Log levels indicate the importance or severity of log messages. Built-in log providers include extension methods to indicate log levels.

Log Level	Severity	Extension Method	Description
Trace	0	LogTrace()	Logs messages only for tracing purposes for the developers.
Debug	1	LogDebug()	Logs messages for short-term debugging purposes.
Information	2	LogInformation()	Logs messages for the flow of the application.
Warning	3	LogWarning()	Logs messages for abnormal or unexpected events in the application flow.
Error	4	LogError()	Logs error messages.
Critical	5	LogCritical()	Logs failures messages that require immediate attention.