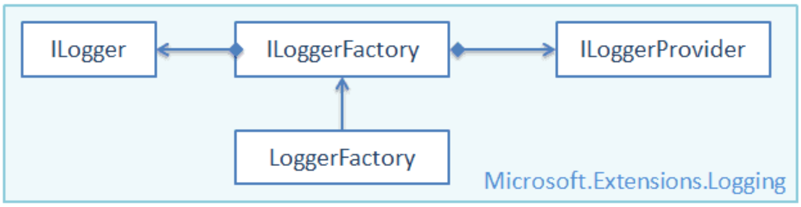
5. Logging in ASP .NET Core

5.1 Logging API

- 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 :



- ILoggerFactory :
  - The ILoggerFactory is the factory interface for creating an appropriate ILogger type instance and also for adding the ILoggerProvider instance.
  - Syntax :

```
public interface ILoggerFactory : IDisposable
{
        ILogger CreateLogger(string categoryName);
        void AddProvider(ILoggerProvider provider);
}
```

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

```
public interface ILoggerProvider : IDisposable
{
        ILogger CreateLogger(string categoryName);
}
```

- ILogger :
  - The ILogger interface includes methods for logging to the underlying storage.

- Syntax :

```
public interface ILogger
{
        void Log<TState>(LogLevel logLevel, EventId eventId,
        TState state, Exception exception, Func<TState, Exception,
        string> formatter);
        bool IsEnabled(LogLevel logLevel);
        IDisposable BeginScope<TState>(TState state);
}
```

5.2 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. Microsoft provides various logging providers as NuGet packages.
- The following table lists important logging providers :

Logging Provider's NuGet Package	**Output Target**
Microsoft.Extensions.Logging.Console	Console
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

5.1 Log Level

- The following table lists the LogLevel values, the convenience Log{LogLevel} extension method, and the suggested usage :

LogLevel	Value	Method	Description
Trace	0	LogTrace	Contain the most detailed messages. These messages may contain sensitive app data. These messages are disabled by default and should not be enabled in production.
Debug	1	LogDebug	For debugging and development. Use with caution in production due to the high volume.
Information	2	LogInformation	Tracks the general flow of the app. May have long-term value.
Warning	3	LogWarning	For abnormal or unexpected events. Typically includes errors or conditions that don't cause the app to fail.
Error	4	LogError	For errors and exceptions that cannot be handled. These messages indicate a failure in the current operation or request, not an app-wide failure.
Critical	5	LogCritical	For failures that require immediate attention. Examples: data loss scenarios, out of disk space.
None	6		Specifies that a logging category shouldn't write messages.