

# Logging EF Core Activity and SQL



**Julie Lerman**

EF Core Expert and Software Coach

@julielerman | thedatafarm.com



# Module Overview



**Learn how EF Core extends .NET logging**

**Adding EF Core logging to the app**

**Many ways to filter what's captured and  
format how it's output**

**Output to various targets**



# Adding Logging to EF Core's Workflow



**EF Core's logging is an  
extension of  
.NET Logging APIs**



# EF Core Logging Capabilities



## EF Core captures:

- SQL
- ChangeTracker activity
- Interaction with database
- Database transactions

## EF Core specific configurations:

- `EnableDetailedErrors`, `EnableSensitiveData`
- Filter based on message type  
(e.g., Database messages)
- Even more detailed filtering (see docs)



We'll focus on adding  
logging to our console app,  
then apply these lessons  
later to an ASP.NET Core  
app.



# Simple Logging



**Minimal API**  
**No extra NuGet packages**



# DbContextOptionsBuilder.LogTo Method

```
protected override void OnConfiguring  
(DbContextOptionsBuilder optionsBuilder)  
{  
    optionsBuilder  
        .UseSqlServer(someconnectionstring)  
        .LogTo(target);  
}
```



# | Filtering Log Output with EF Core Message Categories and LogLevel



# EF Core Message Categories

Source: EF Core Docs

## Message categories

Every log message is assigned to a named hierarchical logger category. The categories are:

Category	Messages
Microsoft.EntityFrameworkCore	All EF Core messages
Microsoft.EntityFrameworkCore.Database	All database interactions
Microsoft.EntityFrameworkCore.Database.Connection	Uses of a database connection
Microsoft.EntityFrameworkCore.Database.Command	Uses of a database command
Microsoft.EntityFrameworkCore.Database.Transaction	Uses of a database transaction
Microsoft.EntityFrameworkCore.Update	Saving entities, excluding database interactions
Microsoft.EntityFrameworkCore.Model	All model and metadata interactions
Microsoft.EntityFrameworkCore.Model.Validation	Model validation
Microsoft.EntityFrameworkCore.Query	Queries, excluding database interactions
Microsoft.EntityFrameworkCore.Infrastructure	General events, such as context creation
Microsoft.EntityFrameworkCore.Scaffolding	Database reverse engineering
Microsoft.EntityFrameworkCore.Migrations	Migrations
Microsoft.EntityFrameworkCore.ChangeTracking	Change tracking interactions



# .NET Logging API Configurations

## Output targets

e.g., **console, file**

## Log Levels

e.g., **critical, debug**



# Configuring LogLevel via .NET Logging API

## LogLevel enums

Debug (default)

Error

Critical

Information

Trace

Warning

None



# Additional Filtering Capabilities



# Even More Logging Features



**Formatting**



**Detailed query error information**



**Filter on event types**



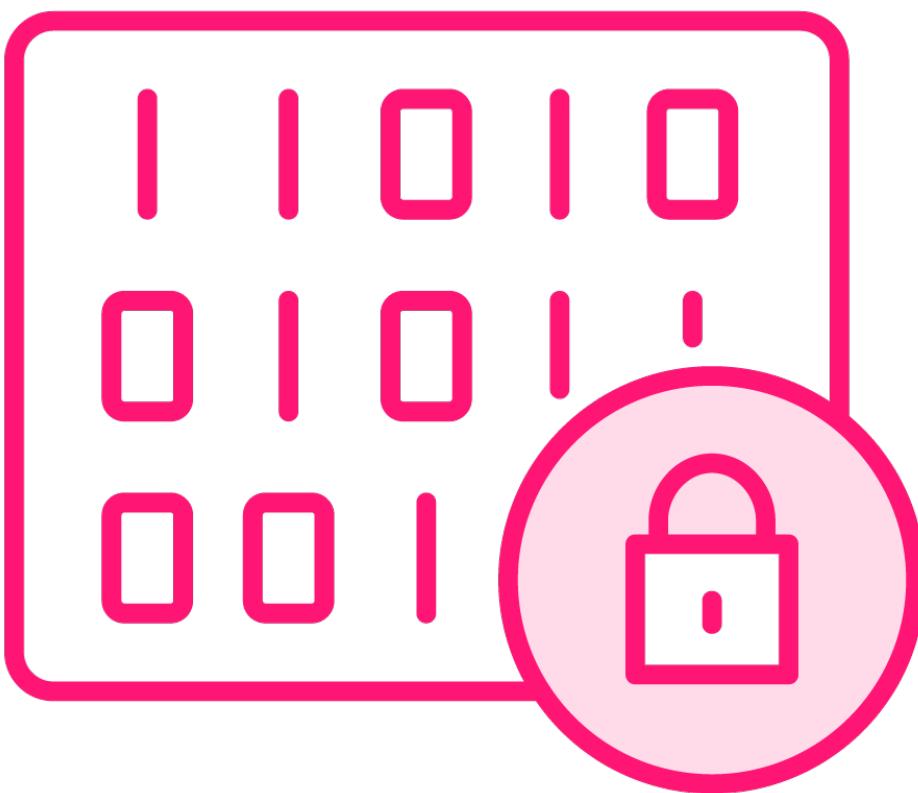
**What information goes in a log message**



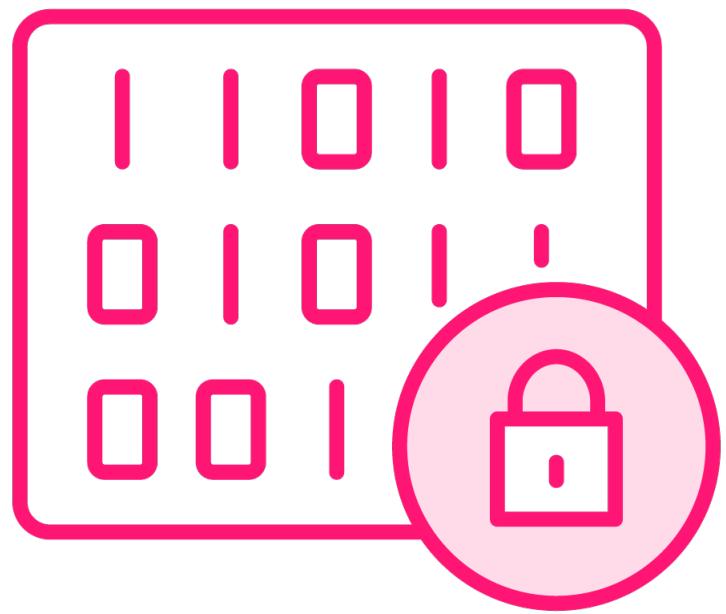
**Show sensitive information e.g., parameters**



# Protecting Your Command Parameters



# Command Parameters in Your Logs



Protected by default



Expose them with explicit code



# Enabling Sensitive Data to Show in Your Logs

```
[__name_0='?' (Size = 4000)]
```

```
[__name_0='Sampson' (Size = 4000)]
```

```
optionsBuilder  
    .UseSqlServer(connectionString)  
    .LogTo(Console.WriteLine)  
    .EnableSensitiveDataLogging();
```

◀ Parameters hidden by default

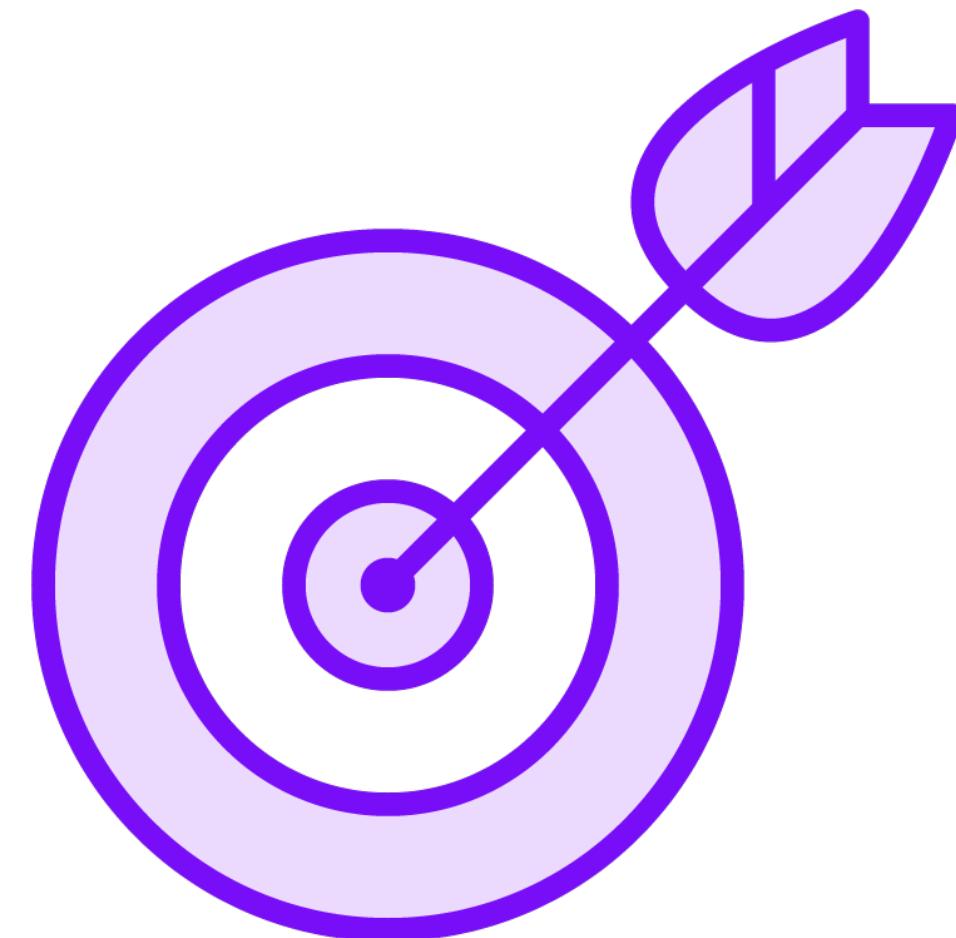
◀ Parameters exposed ...

◀ ...using this OptionsBuilder method

# Specifying the Target of the Log Output

# Configuring Targets via .NET Logging API

Console  
File via Streamwriter  
Debug window  
External logger APIs



# LogTo Target Examples

```
optionsBuilder  
.LogTo(Console.WriteLine)
```

```
private StreamWriter _writer  
= new StreamWriter  
( "EFCoreLog.txt", append: true );
```

```
optionsBuilder  
.LogTo(_writer.WriteLine)
```

```
optionsBuilder  
.LogTo(log=>Debug.WriteLine(log));
```

◀ Delegate to `Console.WriteLine`

◀ Delegate to `StreamWriter.WriteLine`  
*Be sure to dispose StreamWriter to save file*

◀ Lambda expression for `Debug.WriteLine`



# Encoding and Logging to the Console

Some encodings, e.g. code page 1256 for Arabic, won't display in the VS console. If SQL contains one of these encodings, here's a tweak to know.



# Console Won't Output Some Encodings

```
Entity Framework Core 6.0.1 initialized 'PeopleContext'
info: Microsoft.EntityFrameworkCore.Database.Command[101]
      RelationalEventId.CommandExecuted
      Executed DbCommand (25ms) [Parameters=@p0='Julie' @p1='ئەلھىنە ئەلھىنە ئەلھىنە' (Size = 4000)], CommandType='Text', CommandTimeout='30'
      SET NOCOUNT ON;
      INSERT INTO [People] ([FirstName], [LastName])
      VALUES (@p0, @p1);
      SELECT [Id]
      FROM [People]
      WHERE @@ROWCOUNT = 1 AND [Id] = scope_identity();
```

Default encoding won't display  
special characters

```
Entity Framework Core 6.0.1 initialized 'PeopleContext'
info: Microsoft.EntityFrameworkCore.Database.Command[101]
      RelationalEventId.CommandExecuted
      Executed DbCommand (25ms) [Parameters=@p0='Julie' @p1='ئەلھىنە ئەلھىنە ئەلھىنە' (Size = 4000)], CommandType='Text', CommandTimeout='30'
      SET NOCOUNT ON;
      INSERT INTO [People] ([FirstName], [LastName])
      VALUES (@p0, @p1);
      SELECT [Id]
      FROM [People]
      WHERE @@ROWCOUNT = 1 AND [Id] = scope_identity();
```

Override encoding displays the  
special characters

[github.com/julielman/EFCoreEncodingDemo](https://github.com/julielman/EFCoreEncodingDemo)



## Review



**Logs are useful for debugging and learning**

**EF Core logs provide more than just SQL**

**You'll probably want to use Simple Logging API (LogTo)**

**EF Core logging extends .NET logging APIs**

**Explicitly configure sensitive data in parameters to show in logs**

**Many ways to customize and filter EF Core's logging output**



**Up Next:**

# **Interacting with Related Data**

---



# Resources

**Entity Framework Core on GitHub:** [github.com/dotnet/efcore](https://github.com/dotnet/efcore)

**EF Core Simple Logging Documentation:** [bit.ly/LoggingDocs](https://bit.ly/LoggingDocs)

**Logging SQL and Change-Tracking Events in EF Core,  
MSDN Mag, Oct 2019:** [msdn.microsoft.com/magazine/mt830355](https://msdn.microsoft.com/magazine/mt830355)

**Repository with Console Encoding help:**  
[github.com/julielman/EFCoreEncodingDemo](https://github.com/julielman/EFCoreEncodingDemo)

