LINQ & EF Core

Lesson 05 : Logging
&
Calling Stored Procedure

Capgemini

Lesson Objectives

- In this lesson we will cover the following
  - Understanding Logging in EF Core
  - Invoking Stored Procedure using Raw SQL Command

Logging in .NET Core

- In EF Core, We need to log the SQL and change tracking information for debugging purpose.
- .NET Core apps have built-in mechanisms for creating & managing various loggers.
- Three Logging API is three players
  - LoggerProvider
  - Logger
  - LoggerFactory.

Logging in .NET Core

- Logger writes the log entities to output.
- The output can be a console, database a text file etc.
- It must implement the interface ILogger
- The LoggerProvider must implement the interface ILoggerProvider. T
- he responsibility of the LoggerProvider to create the Logger.
- There are many built-in LoggerProviders available for us to use.
- Below package used to implement Logging in EFCore application

```
Install-Package Microsoft.Extensions.Logging.Console
```

Logging in .NET Core

Logger Providers	Description
Microsoft.Extensions.Logging.Console	A simple console logger
Microsoft.Extensions.Logging.AzureAppServices	Supports Azure App Services 'Diagnostics logs' & 'Log stream' features.
Microsoft.Extensions.Logging.Debug	Logs to a debugger monitor using System.Diagnostics.Debug.WriteLine()
Microsoft.Extensions.Logging.EventLog	Logs to Windows Event Log
Microsoft.Extensions.Logging.EventSource	upports EventSource/EventListener
Microsoft.Extensions.Logging.TraceSource	Logs to a trace listener using System.Diagnostics.TraceSource.TraceEvent()

Raw SQL Queries in EF Core

- Entity Framework Core provides mechanisms for executing raw SQL queries directly against the database.
- Entity Framework Core has a powerful method known as **FromSql()** which is used to Execute **Raw SQL Queries** including **Parameterized Queries**.
- Entity Framework Core provides the DbSet.FromSql() method to execute raw SQL queries for the underlying database and get the results as entity objects.
- The .FromSql() method resides in the **Microsoft.EntityFrameworkCore** namespace.

Raw SQL Queries in EF Core

- Example:

```
var context = new SchoolContext();
var students = context.Student.FromSql("Select * from Student where Standard = 10").
```

- Parameterized Queries with .FromSql method:

```
string name = "Tony";
var students1 = context.Student.FromSql($"Select * from Student where Name = '{name}'
```

- LINQ Operators with .FromSql method:

```
var students = context.Student.FromSql("Select * from Student").OrderBy(x => x.Name).ToList();
```

Execute Stored Procedure in EF Core

- SQL Stored Procedures can also be easily executed using the .FromSQL() & .ExecuteCommand() methods in Entity Framework Core.
- The result from Stored Procedure is returned back to the code.

```
var context = new SchoolContext();

var students = context.Students.FromSql("GetStudents 'Bill'").ToList();
```

ExecuteCommand() in EF Core

- The ExecuteSqlCommand() method is used to execute database commands as a string.
- It returns an integer for the number of rows was affected through the specified command.

```
var context = new SchoolContext();
var rowsAffected = context.Database.ExecuteSqlCommand("Update Students set Name = 'Donald Trump' where Id = 5");
```

```
var affectedRows = context.Database.ExecuteSqlCommand("usp_CreateAuthor @AuthorName, @Email",
    new SqlParameter("@AuthorName", "author"),
    new SqlParameter("@Email", "email"));
```

```
var affectedRows = context.Database.ExecuteSqlCommand
    ("usp_CreateAuthor @AuthorName = {0}, @Email= {1}",
    "author", "email");
```

Demo

- Demo of Calling Stored Procedure in EF Core

Lab

- Implementing Stored Procedure in EF Core

Summary

- In this lesson we have learnt the following topic
  - Understanding Logging in EF Core
  - Invoking Stored Procedure using Raw SQL Command