Asp.NET MVC Core

Trainer : Priyanka R Rangole

Email: priyanka.rangole@sunbeaminfo.com

Entity Framework Core (Introduction)

•EF Core is the modern version of Entity Framework (after EF 6.x).
•Open-source, lightweight, extensible, and cross-platform.
•Works with **.NET Core** and **.NET 4.5+** applications.
•Provides automated data access using ORM (Object Relational Mapping).
•Enhances ADO.NET by allowing developers to work with **C# objects instead of SQL**.

What is EF Core?

•An **ORM framework** for .NET.
•Maps classes ↔ database tables automatically.
•Helps in storing & retrieving data without writing raw SQL.
•Ideal for modern cross-platform applications.
•Supported in Web, Desktop, Cloud, and Mobile applications.

EF Core Version History

EF Core Version	Release Date	Target Framework
10.0	Nov 11, 2025	.NET 10
9.0	Nov 12, 2024	.NET 8
8.0	Nov 14, 2023	.NET 8
7.0	Nov 8, 2022	.NET 6
6.0	Nov 8, 2021	.NET 6
5.0	Nov 10, 2020	.NET Standard 2.1
3.0	Sept 23, 2019	.NET Standard 2.1
2.0	Aug 14, 2017	.NET Standard 2.0
1.0	June 27, 2016	.NET Standard 2.1

EF Core Development Approaches

1. Code-First Approach (recommended)

- You write C# classes first.
- EF Core creates database & tables using **migrations**.
- Best for **Domain-Driven Design (DDD)**.
- Highly customizable.

2. Database-First Approach

- Database exists first.
- EF Core generates classes & DbContext from DB.
- Limited support — **no visual designer, no EDMX**.
- Uses command-line scaffolding.

EF Core Database Providers

Entity Framework Core uses **database providers**—installed via **NuGet packages**.
Common Providers:

- **SQL Server**
- **SQLite**
- **MySQL / MariaDB**
- **Oracle**
- **In-Memory Database** (for testing)
- **PostgreSQL**

- EF Core supports multiple DB systems through providers.
- Each provider is a **separate NuGet package**.
- Always install the provider matching your database.
- Example:
- SQL Server → Microsoft.EntityFrameworkCore.SqlServer
- SQLite → Microsoft.EntityFrameworkCore.Sqlite
- PostgreSQL → Npgsql.EntityFrameworkCore.PostgreSQL
- Without a provider, EF Core **cannot connect to any database**.

Installing Entity Framework Core

•EF Core 8 is used with **.NET 8 / .NET Core 8** applications.
•EF Core is **NOT included** by default – must be installed via **NuGet**.
•Works similar for EF Core 8 (LTS version).

EF Core Base & Provider Packages

•Microsoft.EntityFrameworkCore → Base package (common functionality).
•You **must install a DB provider** based on the database you use.

•Example for SQL Server:
Microsoft.EntityFrameworkCore.SqlServer
•Provider package **automatically installs all dependencies**, so no need to install anything else separately.

Ways to Install EF Core

EF Core can be installed using:
1.NuGet Package Manager (GUI)
2.Package Manager Console (PMC)
3..NET CLI (Command Line)
4.PowerShell Terminal
5.Manually editing .csproj

EF Core Database Providers

Database	NuGet Package
SQL Server	Microsoft.EntityFrameworkCore.SqlServer
SQLite	Microsoft.EntityFrameworkCore.Sqlite
Cosmos DB	Microsoft.EntityFrameworkCore.Cosmos
In-Memory DB	Microsoft.EntityFrameworkCore.InMemory
MySQL	MySql.EntityFrameworkCore
PostgreSQL	Npgsql.EntityFrameworkCore.PostgreSQL

DbCOntext Class

What is DbContext?

- Core class of Entity Framework Core
- Represents a **session with the database**
- Combines **Unit of Work + Repository pattern**
- Manages objects and database communication

Responsibilities of DbContext

DbContext provides:

- Database connection management
- Model & relationship configuration
- Querying the database
- Saving data (CRUD)
- Change tracking
- Caching
- Transactions

Creating a DbContext Class

```
public class SbContext : DbContext{}
```

- Inherit from **DbContext**
- This becomes your **context class**

Adding Entities to DbContext

```
public class SbContext : DbContext
{  public DbSet<Student> Students { get; set; }
   public DbSet<Course> Courses { get; set; }
}
```

- **DbSet<T> converts domain classes → EF Core entities**
- EF will create **Student** and **Grade** tables
- LINQ queries on DbSet translate into SQL

Purpose of DbSet<TEntity>

- Represents a table in the database
- Used for:
  - Querying (LINQ)
  - Adding new rows
  - Updating data
  - Deleting data
- EF tracks changes automatically

Configuring Database Connection

Use OnConfiguring() to configure the DB provider & connection:

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseSqlServer(
        "Server=(localdb)\\mssqllocaldb;Database=SchoolDb;Trusted_Connection=True;");
}
```

- UseSqlServer() tells EF Core to use SQL Server
- EF will **create the database automatically** if missing

Understanding the Connection String

"Server=(localdb)\\mssqllocaldb; Database=SchoolDb; Trusted_Connection=True;"

- **Server** → Local SQL instance
- **Database** → SchoolDb
- **Trusted_Connection=True** → Windows Authentication

(*Connection strings can also be stored in appsettings.json)*