NOTTINGHAM TRENT UNIVERSITY

# SOFT20181

## Internet Application Programming

## Working With Data in Razor Pages Using Entity Framework

**Week 19**

**Dr Temi, Mar 2022**

# Learning Outcomes – Week 19

NOTTINGHAM TRENT UNIVERSITY

## Adding Entity Framework Core

- Introducing Entity Framework
- Development approaches
- Adding Entity Framework to your App

## Creating, Querying and Updating Data

- Adding Records to your Database
- Querying Data with EF Core
- Updating / Deleting Data with EF Core

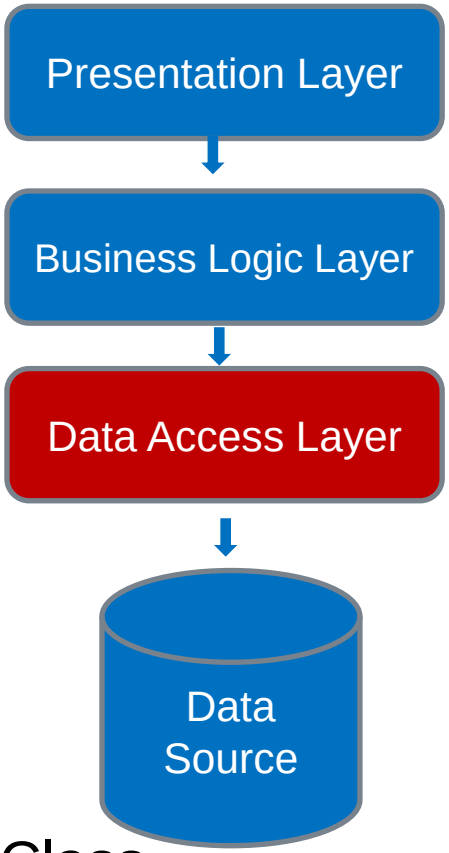
Microsoft

ASP.NET

NOTTINGHAM TRENT UNIVERSITY

# Introduction to Entity Framework Core

- Entity Framework (EF) Core enables access to and manipulation of a database in an object oriented way.
  - It is an object relational mapper.
  - Manages communication between an app and a database
  - It provides data access over the DbContext Class

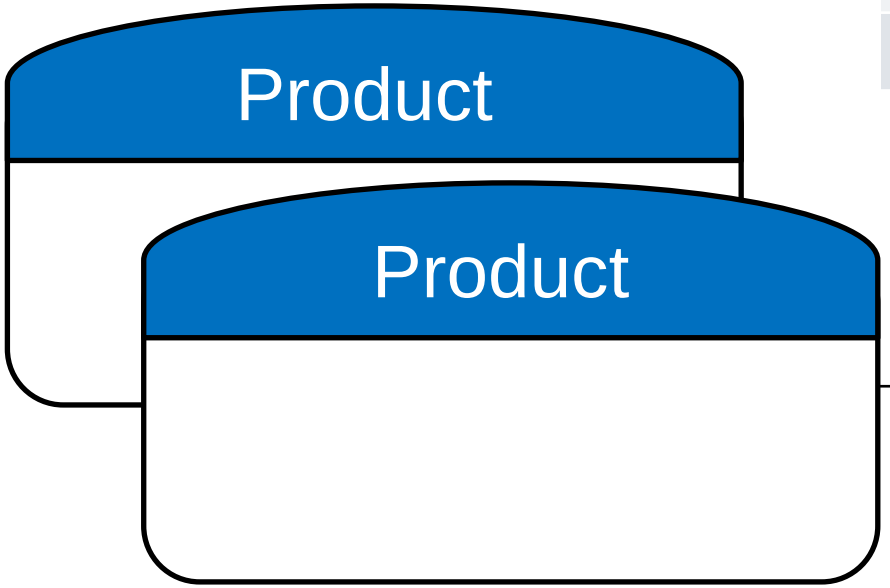


- Entity Framework maps classes and objects to database tables and rows

| Product X |
| --- |
| int id = 1 |
| string Name = “Brown Coat” |
| string Description =“Brown...” |

**Product**

| Id | Name | Description | Price |
| --- | --- | --- | --- |
| 1 | Brown Coat | Abc | 12 |
| 2 | Black Coat | Xyz | 21 |



# Entity Framework Development Approaches

- Database First
  - Generate Data models (entities) from an existing database
- **Code First**
  - Build data models (entities) and then create the database from data model classes.
- Model first
  - Creates data models (entities) using a visual designer built into visual studio and then generate entities.

# Adding EF Core to an Application involves

1. Installing the EF Core NuGet packages
2. Building a data model – Model Class (Entities classes)
   - Defining the applications DbContext
3. Registering the application's DbContext with ASP.NET Core Dependency Injection Container
4. Setting up Database connection string in appsettings.json file
5. Creating a Migration to describe the data model (creating the database)
6. Applying the Migration to the database (updating the database schema)

# 1 Installing EF Core NuGet Packages

- If using SQL Server Database

- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Design
- Microsoft.EntityFrameworkCore.SqlServer.Design
- Microsoft.EntityFrameworkCore.Tools

# 2 Building a Data Model

- Define EF Core Entity Classes
  - Create a new folder within Pages Folder named Models
  - Create a new class within Models to represent the containers for individual rows of data from the database

```
public class Product
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
}
```

# Registering the Application's DbContext

- Identifies the data model classes to EF Core
- Used to access the data in the database
  - Create a new class file in the Models folder called AppDataContext with the following code

```
namespace ApplicationName.Models
{
    public class AppDataContext : DbContext
    {
        public AppDataContext(DbContextOptions< AppDataContext > options) :
        base(options) { }

        public DbSet<Product> Products { get; set; }
    }
}
```

3

# Registering the Application's DbContext with ASP.NET Core Dependency Injection Container

- Within ConfigureServices method of Startup.cs.

```
public void ConfigureServices(IServiceCollection services)
   {
        services.AddDbContext<AppDataContext>

    (options=>options.UseSqlServer(Configuration.GetConnectionString
    ("Default")));

...

}
```

# 4 Setting up your database connection string

- Define connection string information in appsettings.json file
  - For local SQL Server Database

```
"ConnectionStrings": { "Default":
    "Data Source=(localdb)\\MSSQLLocalDB;
    Initial Catalog=DatabaseName;
    integrated security=true" }
}
```

# 5 Create Database using EF Core Migration

- Open ***Tools -> NuGet Package Manager > Package Manager Console*** and run the following command in the package manager console:

  - add-migration migrationName

    - A migrations folder is now added to your application Solution Explorer

# 6 Applying the migration to the database

- To commit changes to the database run the following command in the package manager console
  - update-database
    - you can view the database created from Sql Server Object Explorer. Open ***View -> Sql Server Object Explorer.***

Microsoft

ASP.NET

NOTTINGHAM TRENT UNIVERSITY


# Storing and Retrieving Data Using Entity Framework Core

- Create the Business Logic Layer to handle
  - Seeding the database if it is empty
  - Querying the database for information
    - Using Language-Integrated Query (LINQ) extension methods
      - ToList(), FirstOrDefault(), Count(), Find(), Last(), etc.
  - Creating new database information
    - Using Form entries
      - Add(), SaveChanges()

- Modifying existing information on the database
  - The DbContext class provides and Update() for updating individual entities
    - An EntityState can also be explicitly modified as

```
_DbContext.Entry(Products).State = EntityState.Modified;
```

- Delete information from the database
  - `_DbContext.Remove(DbContext.Products.Find(Id));`

# Create / update database Information using a Form

```
<form method="post" asp-page="Form">
    <input type="text" asp-for="@Model.products.Id" readonly="readonly" />
    <input type="text" asp-for="@Model.products.Name" />
    <input type="text" asp-for="@Model.products.Description" />
            ...
    <input type="submit" value="Save" />
</form>
```

```
public IActionResult OnPost()
        {
            DbContext.Product.Add(products)
            DbContext.SaveChanges();
            return RedirectToPage("Index");
        }
```

# Summary

NOTTINGHAM TRENT UNIVERSITY

- EF Core is an object-relational mapper (ORM) that lets you interact with a database by manipulating standard data model classes, called entities in your application.
  - EF builds a model based on the entity classes in your application
  - You add EF Core to your app by adding a NuGet database provider package.
  - EF uses migration to track changes to your data model classes
  - The Add() method of the DbContext is used to insert data into a database
  - The SaveChanges() method execute the Add method and save changes to the database
  -

# Further Reading

NOTTINGHAM TRENT UNIVERSITY

- Complex Data Model in ASP.NET

- Working with Files in C#

  https://www.w3schools.com/cs/cs_files.asp

- Add a Model to Razor Page

  https://docs.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/model?view=aspnetcore-5.0&tabs=visual-studio

- Entity Framework 6 Code First Approach

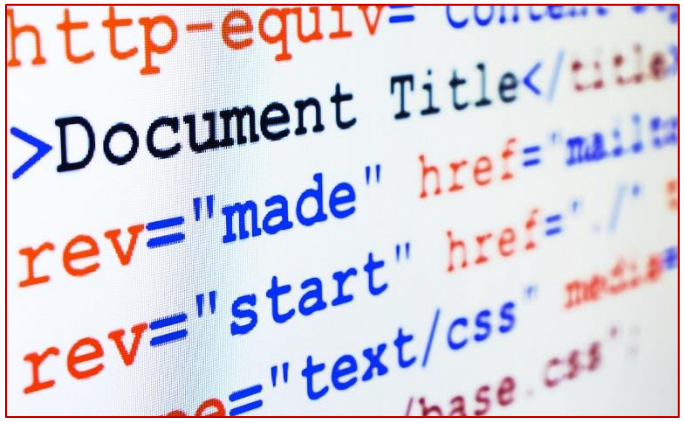
  https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx

# List of Common Abbreviations and Acronyms

NOTTINGHAM TRENT UNIVERSITY

- ASP – Active Server Pages
- DI – Dependency Injection
- EF – Entity Framework
- URL – Uniform Resource Locator
- VS – Visual Studio

# Practical Worksheet Available on the Module Learning Room



Any Questions?

# Thanks