UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Basic .NET – part 1

Trainer: Larisa Nagy

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

.NET ECOSYSTEM

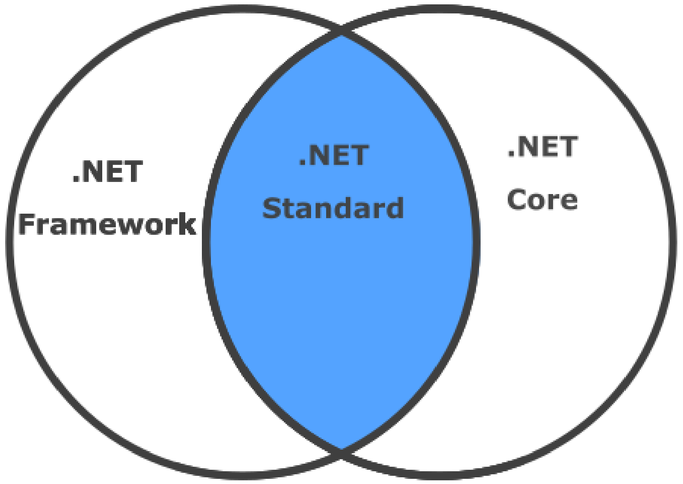
NET Standard vs .NET Framework vs .NET Core

- .NET is a free, open-source development platform for building many kinds of apps, such as: Web apps, web APIs, microservices, mobile apps, desktop apps, etc.
- .NET apps are also supported cross platform, meaning it is compatible with many operating systems, including Windows, macOS, Linux

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

. NET Standard vs .NET Framework vs .NET Core



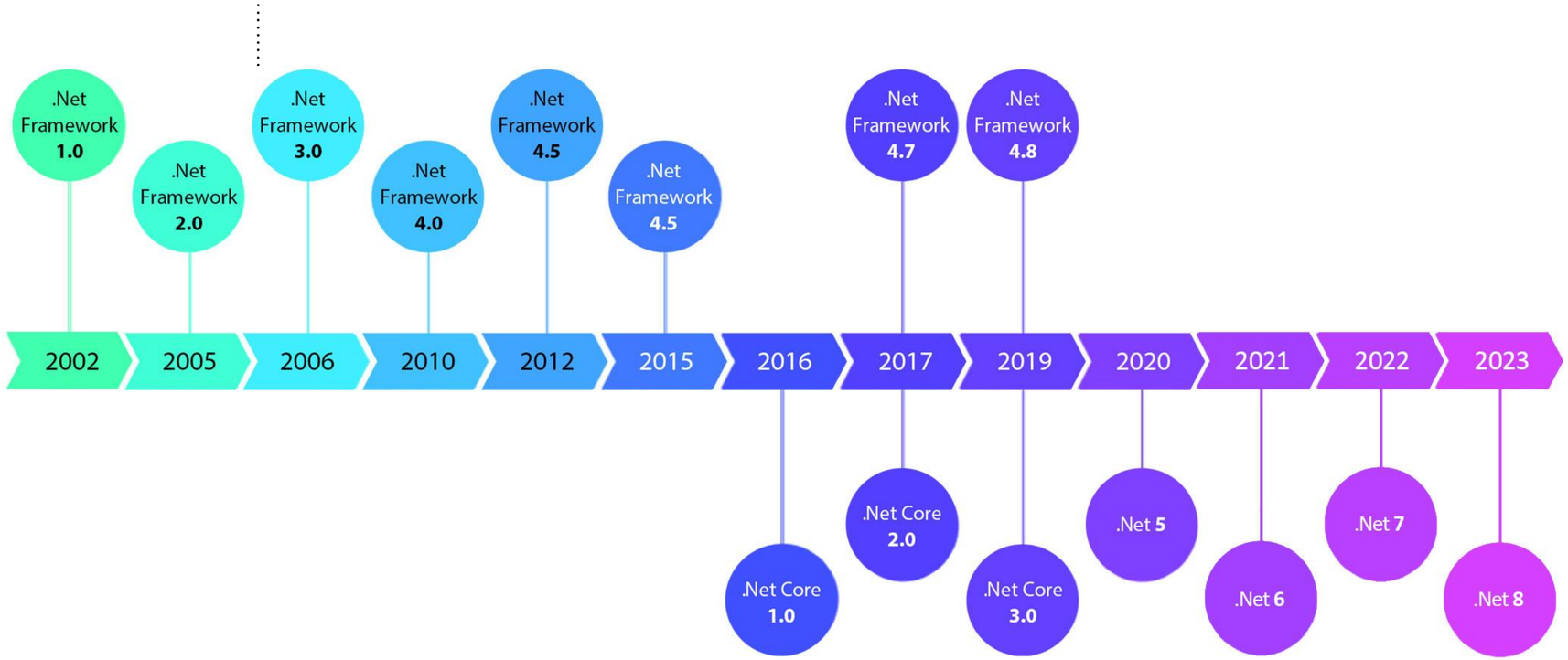
- The .NET Standard is a formal specification of .NET APIs that are intended to be available on all .NET implementations.
- The motivation behind the .NET Standard is establishing greater uniformity in the .NET ecosystem
- Mixing all three of these project types in one solution in Visual Studio it's possible

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

.NET - Release history

- **NET 5 is the first major release of .NET Core following 3.1**
- Dropping "Core" from the name to emphasize that this is the main implementation of .NET going forward



Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

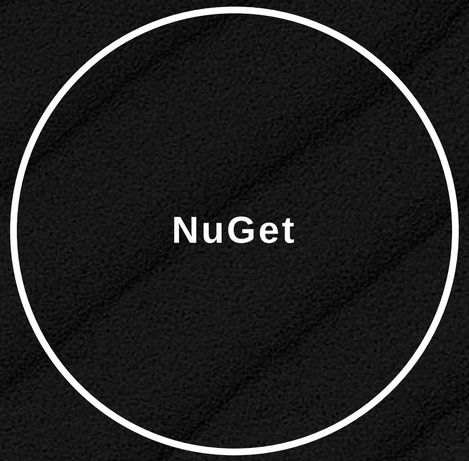
UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com



For .NET the Microsoft-supported mechanism for sharing code is **NuGet**, which defines how packages for .NET are created, hosted, and consumed, and provides the tools for each of those roles.

- A *NuGet package* contains reusable code that other developers have made available to you for use in your projects.
- You can install a NuGet package in a Microsoft Visual Studio project by using:

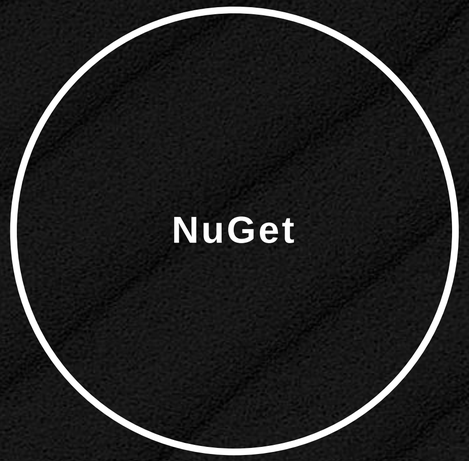
✓ the NuGet Package Manager

✓ the Package Manager Console

✓ the .NET CLI.

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com



- You refer to installed packages in code with a using <namespace> directive, where <namespace> is often the package name. You can then use the package's API in your project.

- There are ~330,397 packages available

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

CODE FIRST vs. DB FIRST:

- Code first is when you start with coding the objects then find a way to represent them in the database.

Example: If it's agile, code first seems more appropriate.

- Database first means that you start with modeling the data in the database, then make code to use it.

Example: Database first if working on an existing project. When you have more logic in the database, for whatever reasons, database first makes more sense.

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

CODE FIRST vs. DB FIRST:

- If you are starting a new project, with a new database,

  => use **code first**

- If you are starting a new project, but there is an existing database you want to talk to

  => use **database first**

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

What is Entity Framework Core ?

Entity Framework (EF) Core is a lightweight, extensible, open source and cross-platform version of the popular Entity Framework data access technology.

=> Microsoft's official data access technology for .NET development

EF Core = **ORM** (Object Relational Mapper)

- Enables .NET developers to work with a database using .NET objects
- EF Core supports many database (SQLServer, MySQL, PostgreSQL)

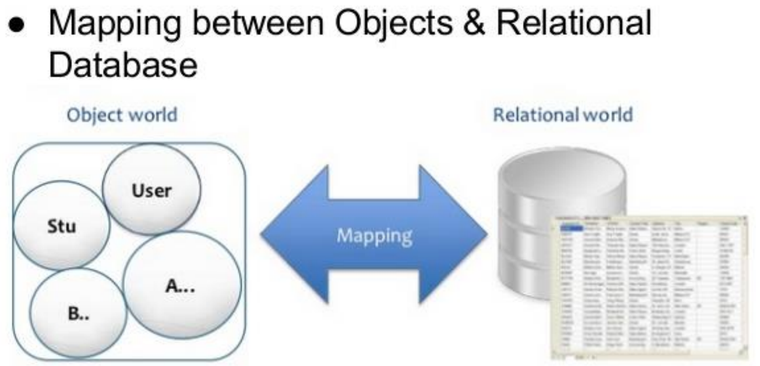
Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

What is ORM ?

ORM (Object Relational Mapper)

Object-relational mapping is a technique that enables developers to work with data in object-oriented way by performing the work required to map between objects defined in an application's programming language and data stored in relational database



Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

1. STEP 1 – INSTALL **NUGET PACKAGES** WITH CORRECT **VERSION**

- ✓ Microsoft Entity Framework Core
- ✓ Microsoft Entity Framework Core SQL Server (**Why ?**)
- ✓ Microsoft Entity Framework Core Tools

! When we have to install some NuGet packages we have to decide also about the **version** that you have to install because when it comes to EF Core and .Net Core we need to use the exact same versions.

=> In our Asp .Net api we use .Net 6 => we will use also for EF Core the same version

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

2. STEP 2 - USE **MS SQL SERVER**

- Install SQL SERVER - dev: https://www.microsoft.com/en-gb/sql-server/sql-server-downloads
- Install SSMS: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020
nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

3. STEP 3 – ADD **CONNECTION STRING**

- We need to find the **location** from our Database (DB), because EF Core need to know exactly where the DB is located and which credentials it should use to connect to the database

  => we use something that is called **connection string**

! Each app needs a connection string to connect to a database

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

3. STEP 3 – **CONNECTION STRING**

What is a **connection string ?**

Is nothing else than an instruction to the ORM (EF Core) about how it should reach the server where the DB resides and how the DB is called.

where do we add this connection string?

=> we will see at the demo

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

What is Db Context ?

Db Context = the code representation of the database itself

= “a photo of your database”

- Set the tables from our DB:

In DataContext class we can specify what our tables should be (employees table, books table, products …)

=> we do this via some properties that are of type DbSet (DbSet = generic class)

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

What is Db Context ?

Db Context = the code representation of the database itself

= “a photo of your database”

✓ We need to register our Db Context in our application

How ?

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS

• WHAT IS A MIGRATION ?

○ In real world projects, data models change as features get implemented: new entities or properties are added and removed, and database schemas need to be changed accordingly to be kept in sync with the application

=>The migrations feature in EF Core **provides a way** to incrementally **update the database** schema to keep it in sync with the application's data model

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS

• WHAT IS A MIGRATION ?

✓ When a **data model change** is introduced, the developer uses EF Core tools to **add a corresponding migration** describing the updates necessary to keep the database schema in sync. EF core compares the current model against a snapshot of the old model to determine the differences, and generates migration source files

✓ Once a new migration has been generated, it can be applied to a database in various ways. EF Core records all applied migrations in a special **history table**, allowing it to know which migrations have been applied and which haven't.

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS

- WHAT IS A MIGRATION ?

✓ we need to perform an instruction and tell EF this is what I want you to add in the database

!!! Only **structural things** about the database: new table, new column, change data type of a column

!!! Inserting in database will be done via code

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

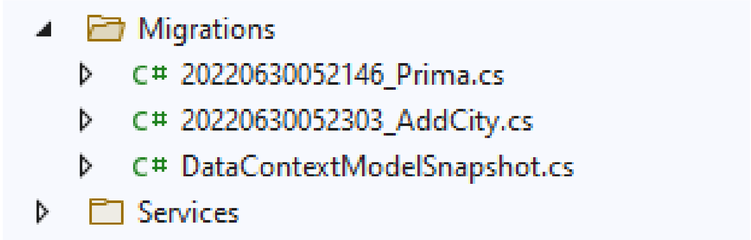
GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS: COMMANDS

1. ADD YOUR FIRST MIGRATION with package manager console from Visual Studio: a**dd-migration Name**

EX: add-migration Initial

=> EF Core create Migrations folder and add specific class for our migration



nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

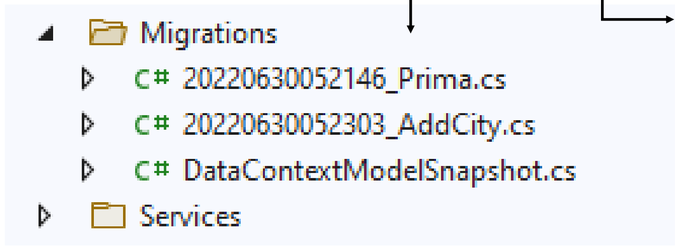
GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS:

1. ADD YOUR FIRST MIGRATION with package manager console from Visual Studio: **ADD-MIGRATION NAME**

What contains the first class?

For example:



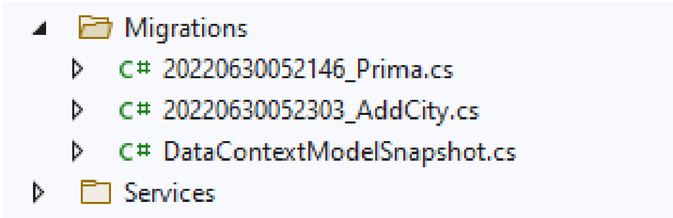

```
1 reference
public partial class Prima : Migration
{
    0 references
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.CreateTable(
            name: "Employees",
            columns: table => new
            {
                Id = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Age = table.Column<int>(type: "int", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_Employees", x => x.Id);
            });

        migrationBuilder.CreateTable(
            name: "TodoItems",
            columns: table => new
            {
                Id = table.Column<long>(type: "bigint", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                FirstName = table.Column<string>(type: "nvarchar(max)", nullable: true),
                IsComplete = table.Column<bool>(type: "bit", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_TodoItems", x => x.Id);
            });
```

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS:

What contains the first class?

For example:




```
1 reference
public partial class Prima : Migration
{
    0 references
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.CreateTable(
            name: "Employees",
            columns: table => new
            {
                Id = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Age = table.Column<int>(type: "int", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_Employees", x => x.Id);
            });

        migrationBuilder.CreateTable(
            name: "TodoItems",
            columns: table => new
            {
                Id = table.Column<long>(type: "bigint", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                FirstName = table.Column<string>(type: "nvarchar(max)", nullable: true),
                IsComplete = table.Column<bool>(type: "bit", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_TodoItems", x => x.Id);
            });
```

1. It contains a definition of what changes will be applied to the database when we want to update or to execute this migration on the DB itself.

The following informations: table name, columns, some constraints

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS

✓ we need to perform an instruction and tell EF this is what I want you to add in the database

!!! When you change anything in your domain model

=> you need to run a migration

(if we add a new property in our model => add a new migration)

UNIUNEA EUROPEANĂ

Instrumente Structurale 2014-2020

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS: COMMANDS

2. CREATE YOUR DATABASE AND SCHEMA

- At this point you can have EF **create your database** and **create your schema** from the migration. This can be done via the following:

Update database from package manager console from Visual Studio: **update-database**

EX: update-database

=> That's all there is to it - your application is ready to run on your new database, and you didn't need to write a single line of SQL.

=> cool ☺

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ **MIGRATIONS:**

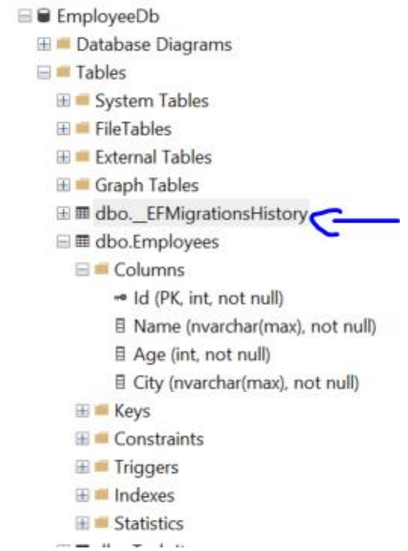
➢ After the 2 commands:

- **add-migration Name**

! For name use something specific and clear: Initial, AddNewColumn, etc.

- **update-database**

=> a new table is created in the database: **dbo__EFMigrationsHistory**



! That's how EF Core keeps track about what migrations has been already updated to the Database and which not

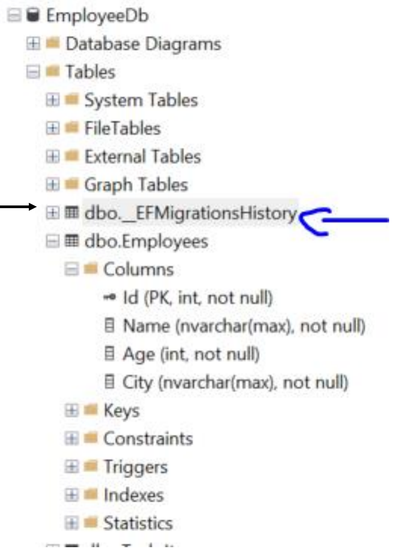
Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS:

- **update-database**

=> a new table is created in the database: **dbo__EFMigrationsHistory**



! Whenever we run a migration
=> EF Core will look at this table and check (using the MigrationId) if this migration has already been performed.

! If the migration has already been updated (migration is part of this table) then it won't be updated again

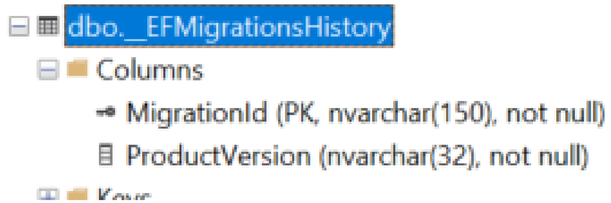
nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

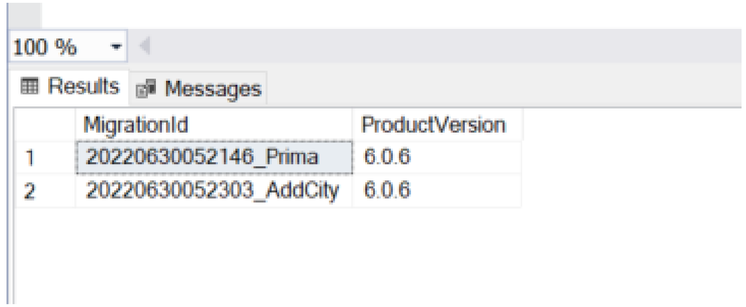
Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

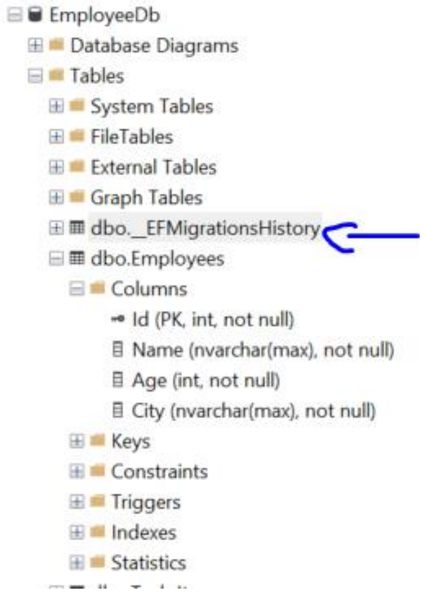
GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS:

What does this table (**dbo__EFMigrationsHistory**) contain?







Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

GETTING STARTED WITH ENTITY FRAMEWORK CORE

➢ MIGRATIONS: COMMANDS

3. REMOVE MIGRATION

- Sometimes you add a migration and realize you need to make additional changes to your EF Core model before applying it.

To remove the last migration, use this command:

Remove-Migration

➢ Resetting all migrations:

In some **extreme cases**, it may be necessary to remove all migrations and start over. This can be easily done by deleting your migrations folder and dropping your database;

! At that point you can create a new initial migration, which will contain your entire current schema.

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UNIUNEA EUROPEANĂ
Instrumente Structurale 2014-2020

Agenda

1. .NET STANDARD VS .NET FRAMEWORK VS .NET CORE
2. NUGET
3. CODE FIRST VS. DATABASE FIRST
4. WHAT IS ENTITY FRAMEWORK CORE
5. GETTING STARTED WITH ENTITY FRAMEWORK CORE - SETUP
6. FIRST APPLICATION USING EF CORE – DEMO
7. LAYER ARCHITECTURE: DTOs and controllers
8. UML BASIC CLASS

PROIECT COFINANTAT DIN FONDUL SOCIAL EUROPEAN PRIN PROGRAMUL OPERAŢIONAL CAPITAL UMAN 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

Demo

Code, code, code ☺

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UML BASIC CLASS

- Class diagrams are a neat way of visualizing the classes in your system before you start coding them up.

  => They’re a static representation of your system structure.

More info:

- https://medium.com/@smagid_allThings/uml-class-diagrams-tutorial-step-by-step-520fd83b300b
- https://medium.com/@smagid_allThings/uml-class-diagram-example-fab6197200e6

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

UML BASIC CLASS

- The **UML** Class diagram is a graphical notation used to construct and visualize object-oriented systems.
- A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's: classes, their attributes, operations (or methods), and the relationships among objects.

More info:

- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

nantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

Useful links

- Download .NET 6: https://dotnet.microsoft.com/en-us/download
- NuGet: https://learn.microsoft.com/en-us/nuget/what-is-nuget
- Install Sql Server - dev: https://www.microsoft.com/en-gb/sql-server/sql-server-downloads
- Install SSMS: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16
- EF Core: https://docs.microsoft.com/en-us/ef/core
- Installing Entity Framework Core: https://docs.microsoft.com/en-us/ef/core/get-started/overview/install
- Entity Framework Core tools reference - Package Manager Console in Visual Studio: https://docs.microsoft.com/en-us/ef/core/cli/powershell
- Migrations: https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=vs
- EF Core Database Providers: https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli
- UML basic class: https://medium.com/@smagid_allThings/uml-class-diagrams-tutorial-step-by-step-520fd83b300b, https://medium.com/@smagid_allThings/uml-class-diagram-example-fab6197200e6, https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com

Thank you!

Proiect cofinantat din Fondul Social European prin Programul Operaţional Capital Uman 2014-2020

Endava Romania SRL
Str. Al. Vaida Voevod Nr. 51
400436 Cluj-Napoca, România
T +40 372 363 282
F +40 264 429 027
endava.com