

A la création d'un projet :

- aller dans nuget (shop des dépendances) → télécharger :
  - entityFrameworkCore (prendre le + en haut)
  - entityFrameworkCore.sqlserver
  - entityFrameworkCoredesign
- créer les classes modèles ( ex : Book.cs )

```
using System.ComponentModel.DataAnnotations;
using Microsoft.EntityFrameworkCore;

namespace CinemaAPI.Models;

public class Movie
{
    [Key] public int Id { get; set; }
    [MaxLength(255), Required] public string? Title { get; set; }
    [MaxLength(255), Required] public string? Description { get;
set; }
    [Required] public DateOnly ReleaseDate { get; set; }
    [Required] public int Duration { get; set; }
    [Required, Precision(16,2)] public decimal Budget { get; set; }
    [MaxLength(64)] public string? PosterHash { get; set; }

    public Studio? Studio { get; set; }
    [Required] public int StudioId { get; set; }
    public Language? Language { get; set; }
    [Required] public int LanguageId { get; set; }

    public List<MovieCategory>? MovieCategories { get; set; }
    public List<MovieCountry>? MovieCountries { get; set; }
    public List<MovieDirector>? MovieDirectors { get; set; }
    public List<MovieActor>? MovieActor { get; set; }
}
```

- créer une classe de context ( ex LibraryContext.cs )

```
using CinemaAPI.Models;
using Microsoft.EntityFrameworkCore;

namespace CinemaAPI;

public class CinemaContext(DbContextOptions<CinemaContext>
options) : DbContext(options)
{
    protected override void OnConfiguring(DbContextOptionsBuilder
options)
    {
        options.UseSqlServer(
            "Server=romaric-thibault.fr;" +
            "Database=CinemaAPI_Romain;" +
            "User Id=romaina;" +
            "Password=romaina;" +
            "TrustServerCertificate=true;"
        );
    }

    public DbSet<Actor> Actors { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<Country> Countries { get; set; }
    public DbSet<Language> Languages { get; set; }
    public DbSet<Movie> Movies { get; set; }
    public DbSet<MovieActor> MovieActors { get; set; }
    public DbSet<MovieCategory> MovieCategories { get; set; }
    public DbSet<MovieCountry> MovieCountries { get; set; }
    public DbSet<MovieRealisator> MovieRealisators { get; set; }
    public DbSet<Person> Persons { get; set; }
    public DbSet<Director> Directors { get; set; }
    public DbSet<Studio> Studios { get; set; }
}
```

- créer les migrations :  
***dotnet ef migrations add NomMigration --project NomProject***  
***dotnet ef database update --project Library***
- créer les classes DTO :

Post :

```
using System.ComponentModel.DataAnnotations;

namespace CinemaAPI.DTO.Movie;

public class PostMovieDto
{
    [Required] public string? Title { get; set; }
    [Required] public string? Description { get; set; }
}

[Required] public DateOnly ReleaseDate { get;
set; }
[Required] public int Duration { get; set; }
[Required] public decimal Budget { get; set; }
[Required] public int? StudioId { get; set; }
[Required] public string? LanguageId { get; set;
}
}
```

Put :

```
using System.ComponentModel.DataAnnotations;

namespace CinemaAPI.DTO.Movie;

public class PutMovieDto
{
    [Required] public string? Title { get; set; }
    [Required] public string? Description { get; set;
}
}

[Required] public DateOnly ReleaseDate { get;
set; }
[Required] public int Duration { get; set; }
[Required] public decimal Budget { get; set; }
[Required] public int? StudioId { get; set; }
[Required] public string? LanguageId { get; set;
}
}
```

```
}
```

## Get :

```
using CinemaAPI.DTO.Actor;  
using CinemaAPI.DTO.Category;  
using CinemaAPI.DTO.Director;  
using CinemaAPI.DTO.Studio;  
  
namespace CinemaAPI.DTO.Movie;  
  
public class GetMovieDto  
{  
    public int Id { get; set; }  
    public string? Title { get; set; }  
    public string? Description { get; set; }  
    public DateOnly ReleaseDate { get; set; }  
    public int Duration { get; set; }  
    public decimal Budget { get; set; }  
    public GetStudioDto? Studio { get; set; }  
    public string? Language { get; set; }  
    public List<GetActorDto>? Actors { get; set; }  
    public List<GetDirectorDto>? Directors { get; set; }  
    public List<GetCategoryDto>? Categories { get; set; }  
    public List<string>? Countries { get; set; }  
}
```

- créer l'appli ( ex : program.cs )

## Créer migration :

```
dotnet ef migrations add NomMigration --project GourmetGuild
```

```
dotnet ef database update --project GourmetGuild
```

remplir auto une class : Alt + Entrée

Double jointure :

```
Recipe recipe = await myContext.Recipe
    .Include(x => x.Tags)
    .Include(x => x.Contains)
        .ThenInclude(x => x.Ingredient)
    .Include(x => x.Contains)
        .ThenInclude(x => x.Ingredient)
    .SingleOrDefault();
```

Context :

```
public DbSet<Actor> Actors { get; set; }
public DbSet<Category> Categories { get; set; }
public DbSet<Country> Countries { get; set; }
public DbSet<Language> Languages { get; set; }
public DbSet<Movie> Movies { get; set; }
public DbSet<MovieActor> MovieActors { get; set; }
public DbSet<MovieCategory> MovieCategories { get; set; }
public DbSet<MovieCountry> MovieCountries { get; set; }
public DbSet<MovieRealisator> MovieRealisators { get; set; }
public DbSet<Person> Persons { get; set; }
public DbSet<Realisator> Realisators { get; set; }
public DbSet<Studio> Studios { get; set; }
}
```