



ENTITY FRAMEWORK CORE code first – API MVC

Visual Studio 2022

Le Sujet

Pour une future application de streaming, nous avons besoin d'une API proposant un catalogue de films.

Le Modèle de données

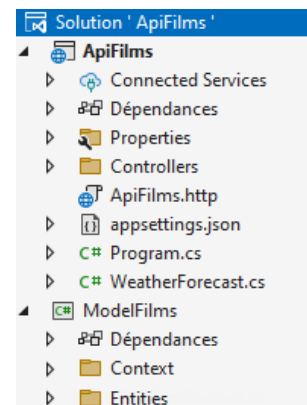
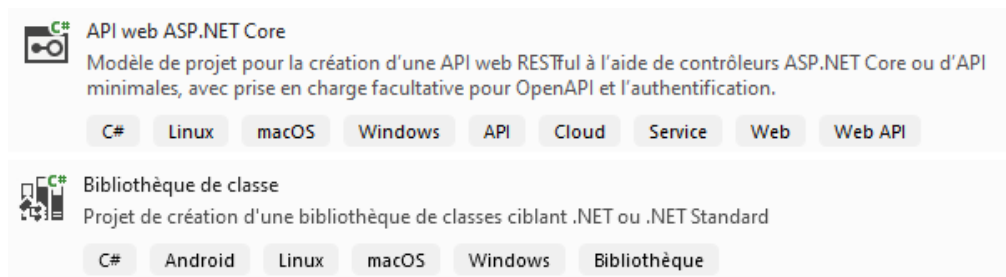
La modélisation UML du diagramme de classes est un bon point de départ :



Gestion du catalogue de films

La solution

Créer une solution avec 2 projets :



Implémentation du modèle de données

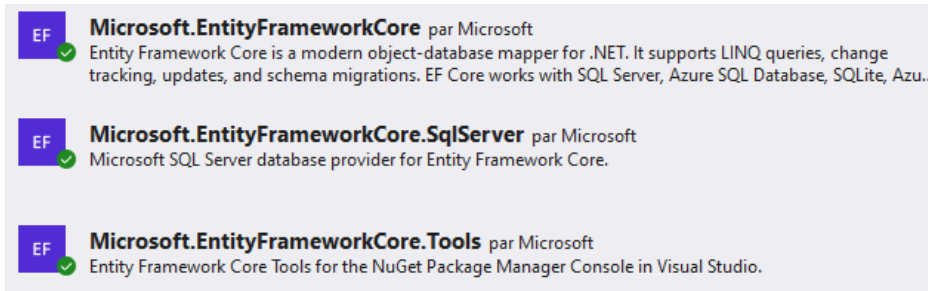
Implémenter le modèle en utilisant les annotations, sans oublier les relations.

Créer le contexte de données pour lier le modèle aux entités :

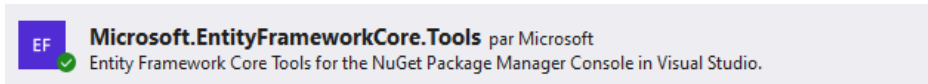
```
public class FilmsDbContext : DbContext
{
    public virtual DbSet<Genre> Genres { get; set; } = null!;
    public virtual DbSet<Film> Films { get; set; } = null!;
    public virtual DbSet<Personne> Personnes { get; set; } = null!;
    public virtual DbSet<Realisateur> Realisateurs { get; set; } = null!;
    public virtual DbSet<Acteur> Acteurs { get; set; } = null!;
    public FilmsDbContext(DbContextOptions<FilmsDbContext> options)
        : base(options)
    {
    }
}
```

Les packages NuGet à ajouter

- Bibliothèque du modèle :



- API :



La chaîne de connexion

Dans le projet AP (le seul exécutable), paramétrer la chaîne de connexion dans *appsettings.json* :

```
"ConnectionStrings": {  
  "Database": "Server=(localdb)\\mssqllocaldb;Database=Films;Trusted_Connection=True;"  
}
```

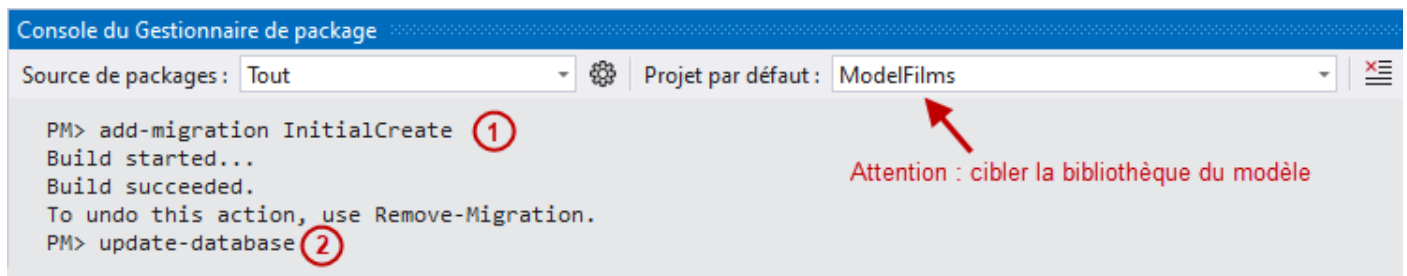
Et configurer le *DbContext* en tant que service injectable dans *program.cs* :

```
var builder = WebApplication.CreateBuilder(args);  
  
var mainConnectionString = builder.Configuration.GetConnectionString("Database") ??  
    throw new Exception("Connection string is missing");  
builder.Services.AddDbContext<FilmsDbContext>(opt =>  
    opt.UseSqlServer(mainConnectionString));
```

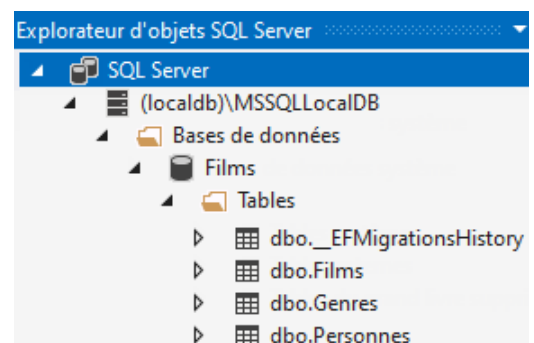
Ne pas oublier de générer la solution avant de poursuivre ...

Les migrations pour créer la base de données

Dans la fenêtre de Console du Gestionnaire de package :



Vérifier dans l'Explorateur d'objets SQL Server que la base est bien créée :



Créer un contrôleur d'API

Dans le projet d'API, sur le dossier *Controllers*, ajouter un contrôleur :

Ajouter un nouvel élément généré automatiquement



×

Ajouter Contrôleur d'API avec actions, utilisant Entity Framework

Classe de modèle	Film (ModelFilms.Entities)
Classe DbContext class	FilmsDbContext (ModelFilms.Context) +
Fournisseur de bases de données	Configuré à partir du DbContext sélectionné
Nom du contrôleur	FilmsController

AjouterAnnuler

Exécuter pour vérifier le Swagger :

