

# Movie Catalog API

Built with ASP.NET Core (.NET 8), Entity Framework Core, SQLite, and Swagger.

Includes DTOs and fully structured source code.

## Overview

This document contains the full source code, setup instructions, and technology stack for a Movie Catalog Web API built with ASP.NET Core (.NET 8), Entity Framework Core, and SQLite persistence. It provides CRUD APIs for Movies and Directors, with proper DTOs to avoid circular JSON references.

## Technology Stack

- .NET 8 Web API
- Entity Framework Core 8
- SQLite (lightweight file-based persistence)
- Swagger (API documentation & testing)
- DTOs (Data Transfer Objects) to avoid circular references

## Setup & Running Instructions

1. Install .NET 8 SDK from <https://dotnet.microsoft.com/download>
2. Create a new folder and copy the code files from this document.
3. Run the following commands in terminal:  
    `dotnet restore`  
    `dotnet run`
4. The API will start at: `http://localhost:5000`
5. Open Swagger UI: `http://localhost:5000/swagger`

SQLite database file (movies.db) will be auto-created in the project folder.

## Source Code – Program.cs

```
using Microsoft.EntityFrameworkCore;
using MovieCatalog.Api.Data;
using MovieCatalog.Api.DTOS;
using MovieCatalog.Api.Models;

var builder = WebApplication.CreateBuilder(args);

// Add DbContext with SQLite
builder.Services.AddDbContext<AppDbContext>(options =>
    options.UseSqlite("Data Source=movies.db"));

// Add controllers and configure JSON to ignore cycles
builder.Services.AddControllers()
    .AddJsonOptions(opt => opt.JsonSerializerOptions.ReferenceHandler =
        System.Text.Json.Serialization.ReferenceHandler.IgnoreCycles);

// Add Swagger
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Apply migrations & create database
using (var scope = app.Services.CreateScope())
{
    var db = scope.ServiceProvider.GetRequiredService<AppDbContext>();
    db.Database.Migrate();
}

// Configure middleware
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

## Source Code – AppDbContext.cs

```
using Microsoft.EntityFrameworkCore;
using MovieCatalog.Api.Models;

namespace MovieCatalog.Api.Data
{
    public class AppDbContext : DbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }

        public DbSet<Movie> Movies { get; set; }
        public DbSet<Director> Directors { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Director>().HasData(
                new Director { Id = 1, Name = "Christopher Nolan", BirthDate = new DateTime(1970,
7, 30) },
                new Director { Id = 2, Name = "Quentin Tarantino", BirthDate = new DateTime(1963,
3, 27) }
            );

            modelBuilder.Entity<Movie>().HasData(
                new Movie { Id = 1, Title = "Inception", Genre = "Sci-Fi", ReleaseDate = new
DateTime(2010, 7, 16), DirectorId = 1 },
                new Movie { Id = 2, Title = "Pulp Fiction", Genre = "Crime", ReleaseDate = new
DateTime(1994, 10, 14), DirectorId = 2 }
            );
        }
    }
}
```

## Source Code – Models/Movie.cs

```
namespace MovieCatalog.Api.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public string? Genre { get; set; }
        public DateTime? ReleaseDate { get; set; }
        public int DirectorId { get; set; }
        public Director Director { get; set; }
    }
}
```

## Source Code – Models/Director.cs

```
namespace MovieCatalog.Api.Models
{
    public class Director
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public DateTime? BirthDate { get; set; }
        public ICollection<Movie> Movies { get; set; }
    }
}
```



## Source Code – DTOs/MovieDto.cs

```
namespace MovieCatalog.Api.DTOs
{
    public class MovieDto
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public string? Genre { get; set; }
        public DateTime? ReleaseDate { get; set; }
        public int DirectorId { get; set; }
    }
}
```

## Source Code – DTOs/DirectorDto.cs

```
namespace MovieCatalog.Api.DTOs
{
    public class DirectorDto
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public DateTime? BirthDate { get; set; }
    }
}
```