

EF Core introductie

1 Oefening - Opzet EF Core

LEERDOELEN

1. Je kan de benodigde NuGet packages installeren om met EF Core te kunnen werken.
2. Je kan structuur aanbrengen in een project om data gerelateerde objecten in onder te brengen.
3. Je kan POCO classes aanmaken voor het domein.

OEFENING

We breiden het project uit de vorige les uit zodat we er met EF Core in kunnen werken.

1. Open het ASP.NET Core Web API project "MyGameStore" van Oefenbundel 1 van de vorige les.
2. Installeer de nodige packages om met EF Core voor SQL Server te kunnen werken.
3. Maak de map aan om data gerelateerde objecten in onder te brengen.
4. Maak de map aan om de POCO's in onder te brengen.

2 Oefening - Entiteitsklassen definiëren

LEERDOELEN

1. Je kan POCO's voor entiteiten aanmaken op een gepaste locatie in het project.
2. Je kan POCO's voor entiteiten implementeren inclusief navigatie properties.

OEFENING

We voegen de nodige POCO's voor entiteiten toe aan het project uit de vorige les.

1. Maak in het project 'MyGameStore' onderstaande POCO's aan op de juiste locatie of pas bestaande POCO's aan.
 - a. "Store"
 - i. Id (integer)
 - ii. Name (string)
 - iii. Street (string)
 - iv. Number (string)
 - v. Addition (string)
 - vi. Zipcode (string)
 - vii. City (string)

- viii. IsFranchiseStore (bool)
- b. "Person"
 - i. Id (integer)
 - ii. FirstName (string)
 - iii. LastName (string)
 - iv. Gender (int)
 - v. Email (string)

2. Voorzie de nodige properties die de relatie weergeven tussen Store en Person waarbij een persoon in 1 winkel werkt en binnen 1 winkel meerdere personen kunnen werken.

3 Oefening - De DbContext class implementeren

LEERDOELEN

1. Je kan een DbContext class aanmaken op een gepaste locatie in het project.
2. Je kan een DbContext class implementeren.

OEFENING

We breiden het project uit met de DbContext class en de bijhorende DbSet's.

1. Maak de DbContext klasse aan. Zorg ervoor dat deze klasse:
 - a. Een gepaste naam krijgt.
 - b. Op de gepaste locatie aangemaakt wordt in het project.
 - c. Zodanig wordt geïmplementeerd dat beide entiteiten (Store en Person) benaderd kunnen worden.

4 Oefening - De DbContext instellen voor DI + connection string

LEERDOELEN

1. Je kan het ASP framework opzetten zodat het klaar is om DbContext objecten aan te maken en te injecteren via Dependency Injection.
2. Je kan de connection string op de juiste plaats in het project onderbrengen en ophalen waar nodig vanuit de toepassing.

OEFENING

We breiden de main methode van ons ASP project uit zodat het framework weet hoe DbContext objecten moet aanmaken en waar de configuratie info (connection string) moet

worden opgehaald.

1. Pas de **main** aan:
 - a. Voeg de connection string toe in het Appsettings json bestand:
"Server=(localdb)\\mssqllocaldb;Database=MyGameStoreDB".
 - b. Registreer je zelf gemaakte DbContext klasse voor DI (net na de AddControllers aanroep).
 - c. Haal hierbij ook de connectionstring uit het Appsetting bestand.
 - d.

5 Oefening - Een migration aanmaken en uitvoeren

LEERDOELEN

1. Je kan een migration voor een DbContext aanmaken.
2. Je kan een SQL script genereren voor een migration.
3. Je kan een migration uitvoeren.

OEFENING

We maken een eerste migration aan en maken de database aan op basis hiervan.

1. Maak een migration aan voor je DbContext klasse.
2. Genereer een SQL-script voor deze migration.
3. Voer de migration ook uit zodat de database automatisch wordt aangemaakt.
4. Controleer het resultaat m.b.v. de SQL Server Management Studio.

6 Oefening - Basis CRUD operaties

LEERDOELEN

1. Je kan de basis CRUD operaties uitwerken met EF Core.

OEFENING

We passen de CRUD operaties aan zodat deze gebruik maken van de DbContext i.p.v. de lijst in de controller.

1. Voorzie de CRUD operaties in je API voor beide entiteiten.
2. Je voorziet dus per entiteit:
 - a. Create.
 - b. Read, omvat het ophalen op basis van Id en het ophalen van alles.

- c. Update.
 - d. Delete.
- 3. Test de werking uit m.b.v. Swagger en controleer de resultaten m.b.v. SQL Server Management Studio.