

AD – SportStore (deel 1)

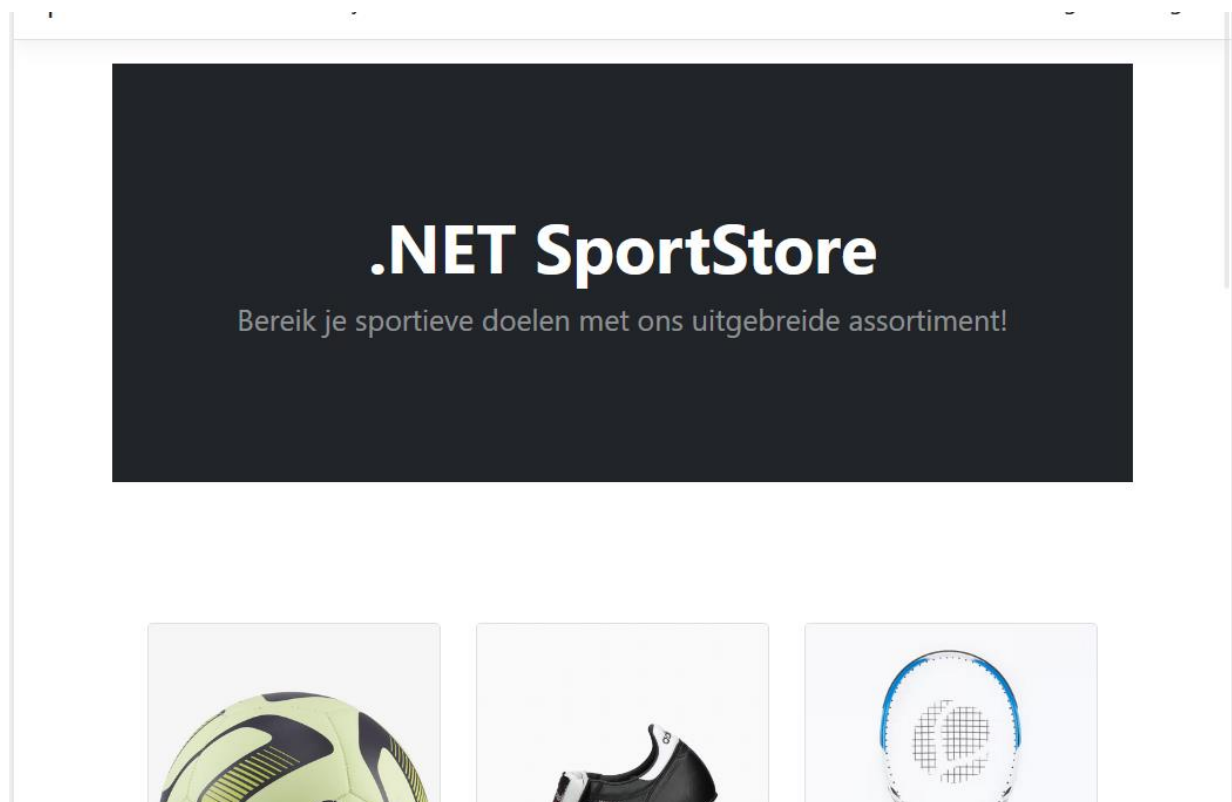
Doelstellingen van dit labo

- Entity Framework installeren in een nieuw project
- Entiteiten met eenvoudige relaties mappen in Entity Framework
- Een DbContext toevoegen en configureren
- Een connection string kunnen opstellen en configureren
- Gebruik maken van de Fluent API voor het configureren van entity's
- Bij het genereren van een tabel deze automatisch van de nodige data voorzien (= seeding)
- Eenvoudige CRUD-operaties uitvoeren
- Het concept van Migraties correct toepassen

Opdracht SportStore (deel 1)

Context

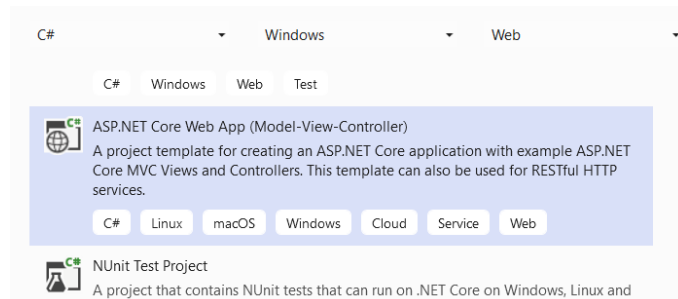
Voor deze opdracht maak je een webshop voor de verkoop van sportartikelen. Via de applicatie moet het mogelijk zijn om de catalogus van de webshop te beheren (producten toevoegen, bekijken, wijzigen en verwijderen). In een later stadium, zal de gebruiker tevens producten aan zijn/haar winkelmandje kunnen toevoegen en verwijderen. Ten slotte zal de applicatie ook voorzien moeten worden van de nodige authenticatie- en autorisatiemaatregelen.



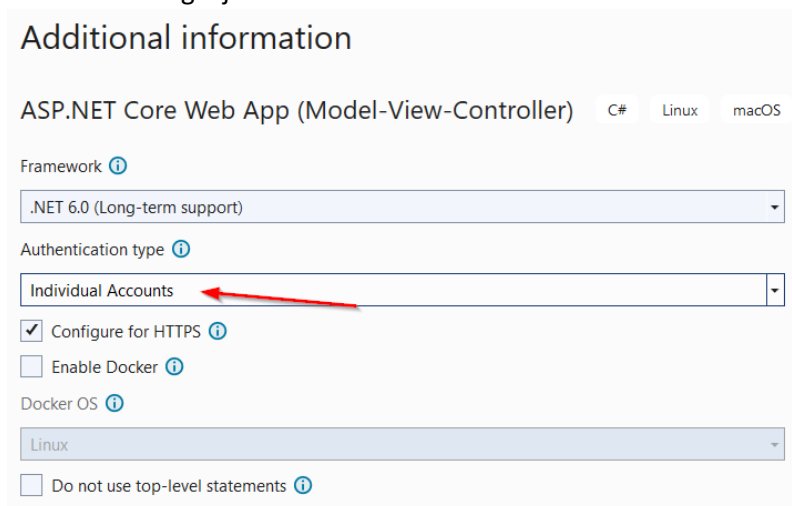
Opdracht

Set-up

1. Maak een nieuw project aan van het type ASP.NET Core Web App (Model-View-Controller):

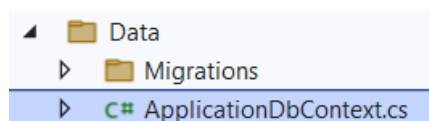


2. Kies *SportStore* als naam voor het nieuwe project
3. In de volgende stap, kies je bij *Authentication Type* de optie *Individual Accounts*. Hierdoor worden de nodige libraries toegevoegd aan de template om later authenticatie en autorisatie mogelijk te maken.



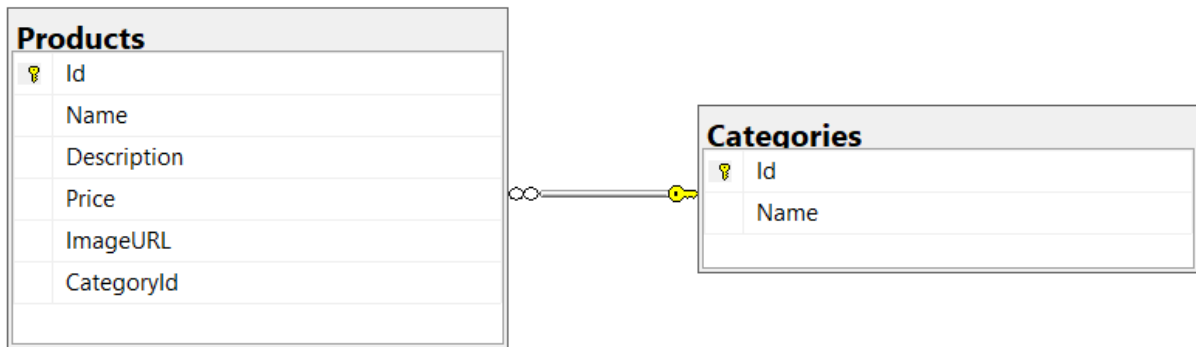
Entity Framework

Je zult zien dat de template reeds enkele basis-zaken bevat voor het werken met Entity Framework. Zo zal er reeds een *DbContext*-klasse aanwezig zijn (in het folder *Data*) en zal er een default *connection-string* toegevoegd zijn aan het *appsettings.json*-bestand van het project.



Taken:

- Pas de connection-string aan, zodat de naam van de database duidelijker wordt
- Maak de Entity-classes aan, zoals hieronder weergegeven:



Opmerkingen:

Products:

- De kolom *Price* kan kommagetallen bevatten

Seeding

Aangezien de categorieën weinig zullen veranderen tijdens de loop van de applicatie, kunnen we gebruik maken van seeding om een aantal categorieën vast in de databank te zetten bij het opstarten van de applicatie.

Zorg ervoor dat bij het opstarten van de applicatie 5 à 10 categorieën ge-insert worden in de databank (bv.: voetbal, tennis, wielrennen, ...).

Producten beheren

Implementeer de functionaliteiten om de catalogus met producten te beheren en te raadplegen.

Producten toevoegen

Mak een pagina met een formulier waarmee de gebruiker een nieuw product kan toevoegen aan de catalogus van de webshop. Baseer je hiervoor op onderstaand voorbeeld:

Name

Description

Price

Category

Image

Bladeren...

Geen bestand...selecteerd.

Bewaren

Validatie:

- Alle velden zijn verplicht

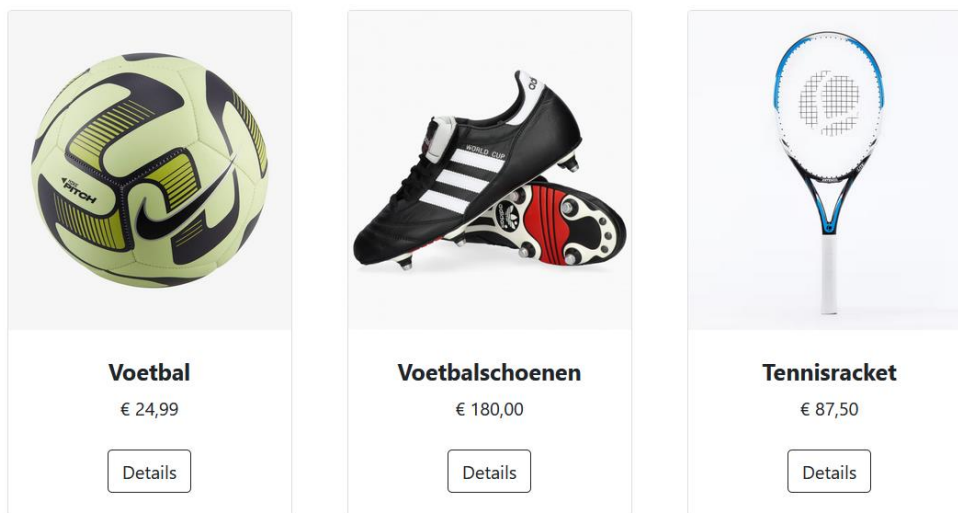
Tips:

- De categorie kan geselecteerd worden via een dropdown-lijst met alle categorieën
- Maak een ViewModel-klasse met properties voor alle formulievelden en properties voor het meegeven van de lijst met categorieën
- Baseer je op de volgende [tutorial](#) om de afbeelding van het product te uploaden.

Opgelet: bewaar de geüploade afbeeldingen in de *wwwroot*-folder van het project (dus **niet** in de databank). Genereer een **willekeurige bestandsnaam** voor de afbeelding en bewaar deze in de ImageURL-property van de Product-entity. Vergeet ook niet om het attribuut `enctype="multipart/form-data"` toe te voegen aan je `<form>`-tag!

Catalogus weergeven

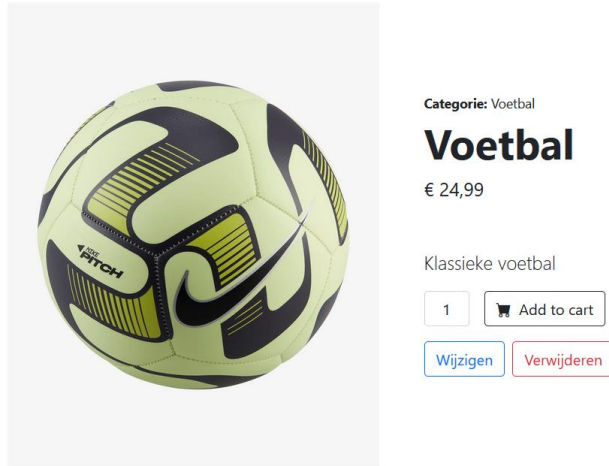
Maak een overzichtsscherm waarin je de catalogus van de webshop weergeeft. Toon voor elk product de afbeelding, de naam en de prijs. Voorzie ook een link om naar de Details-pagina van het product te gaan en om het product te editeren (zie verder). Baseer je hiervoor op onderstaand voorbeeld:



Tip: je kunt hiervoor gebruikmaken van de Card-component van Bootstrap.

Details van product weergeven

Wanneer de gebruiker in het overzichtsscherf op de Details-link klikt bij een product, wordt de detail-pagina van het bijhorende product weergegeven. In dit detail-scherf worden alle gegevens van het geselecteerde product weergegeven. Probeer onderstaand voorbeeld zo goed mogelijk na te bouwen:



Product editeren

Wanneer de gebruiker op de *Edit*-knop klikt, wordt een pagina weergegeven waarmee de gebruiker de gegevens van het product kan wijzigen.

Name

Description

Price

Category

Image

Geen bestand geselecteerd.

Opgelet: het pad van de afbeelding (IFormFile) hoeft je niet opnieuw in te stellen. Dit kan je leeg laten. Enkel wanneer de gebruiker bij het verzenden van het formulier een ander bestand heeft gekozen, kan je het pad van de afbeelding overschrijven met de nieuwe locatie.

Nadat de wijzigingen doorgevoerd werden, redirect je de gebruiker naar de details-pagina van dit product.

Product verwijderen

Zorg ervoor dat wanneer de gebruiker op de knop “Verwijderen” klikt, het product onmiddellijk uit de database verwijderd wordt. Je hoeft geen bevestigingsscherf te tonen.

Afwerking

- Breid de navigatiebalk uit met een link naar de catalogus-pagina en de pagina om een nieuw product aan te maken
- Verwijder overbodige controllers en verwijzingen in het _Layout.cshtml-bestand

SportStore Catalog Add Product