**Fluent Validation**

**Probleem**

We willen ervoor zorgen dat wanneer een gebruiker van de API een formulier (het aanmaken van een werknemer bijvoorbeeld) invult, dit geen foutieve of lege elementen heeft zoals hieronder geïllustreerd. Verder willen we natuurlijk ook dat de gebruiker weet waar hij precies verkeerd is gegaan.

A picture containing background pattern

Description automatically generated

**Oplossing**

Om dit te realiseren is er gebruik gemaakt van FluentValidation.

Om te beginnen moet de package eerst geïnstalleerd worden via de NuGet package manager console van Visual Studio.



Wij hebben in ons project gebruik gemaakt van CQRS (Command Query Responsibility Segregation) dus de voorbeelden die hier worden gebruikt, zijn vooral voor projecten die werken aan de hand van het CQRS *pattern*.

Vooraleer we validators maken, gaan we eerst de verdere setup in orde brengen. Plaats *services.AddMvc();* in de configureServices() methode van het Startup bestand.

Text

Description automatically generated

Het voorbeeld hieronder maakt gebruik van het CreateEmployeeDTO object dat gebruik maakt van de CreateEmployeeDTOValidator. In dit specifieke voorbeeld zorgen we ervoor dat wanneer het FIrstName veld leeg is of meer dan 255 karakters bevat, er een errorbericht komt (zoals in de onderste afbeelding wordt getoond, is het errorbericht hier “Voornaam mag niet leeg of langer dan 255 karakters zijn.”). Men kan aan de hand van validators veel meer doen dan wat hier wordt getoond. Als de lezer complexere zaken wilt valideren stel ik voor dat u naar de documentatie van FluentValidation gaat.



Text

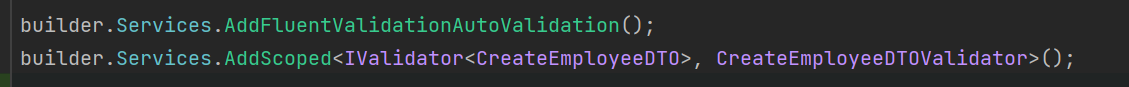
Description automatically generated

Verder zetten we in onze CreateEmployeeCommandHandler klasse volgende code voor de validators. Het onderlijnde is relevant voor de validators.  


Vooraleer we onze validators kunnen toevoegen aan de views, plaatsen we deze code nog in de configureServices() methode van het Startup bestand.



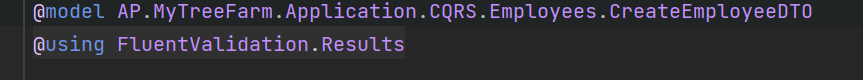
Verder zetten we deze lijnen code in het Pogram.cs bestand van het dashboard.



Deze code plaatsen we in de EmployeeController bij de webapi. Deze code zorgt ervoor dat als er errors zijn, deze worden opgevangen en doorgestuurd naar de view.



Nu kunnen we eindelijk de validator in de view plaatsen. Vanboven zetten we deze lijnen code.



Ook hier is de onderlijnde code relevant voor de validator. Dit zorgt ervoor dat het errorbericht wel degelijk wordt getoond aan de gebruiker.



Wanneer we nu proberen een werknemer aan te maken zonder het voornaam (of FirstName) veld in te vullen, krijgen we het toepasselijke errorbericht.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated