**Oefening EF Deel 2: HondenRescue**



**1. Start**  
  
Vertrek vanuit de H05 HondenRescue Startbestand. Deze oefening bevat onderdelen van EF D2 en een aantal extra’s die later in de OPO MVC Project mogelijks nuttig kunnen zijn.

**2. Models**

**2A Model bijwerken**

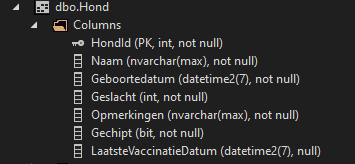
Er is reeds een model voorzien van Hond, er zijn echter kleine aanpassingen nodig:

{NA} = NULLS ALLOWED {NNA} = NULLS NOT ALLOWED = verplicht  
**Voeg volgende attributen/properties toe:**  
  
-LaatsteVaccinatieDatum: datetime {NA}  
-Gechipt: bool {NNA}   
-FotoNaam: string {NA}  
  
W**ijzig volgende attributen/properties die niet null mogen zijn indien nog nodig:**  
-Naam  
-Geboortedatum  
  
Wijzig volgende **attributen/properties die null moeten zijn:**  
-Opmerkingen

**2B (Database) migrations**

Het model is nu gewijzigd. Je voert deze wijzingen toe door een migratie te maken en deze te comitten naar de database. Gebruik als naam: MIG1

Controle resultaat:

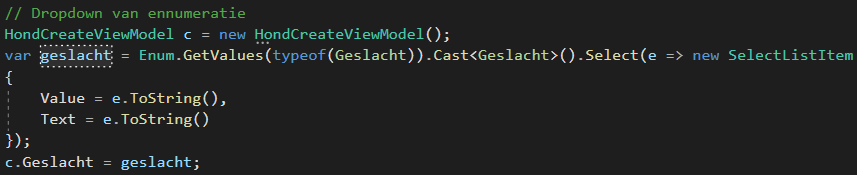
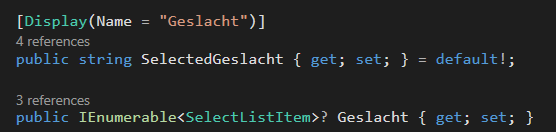
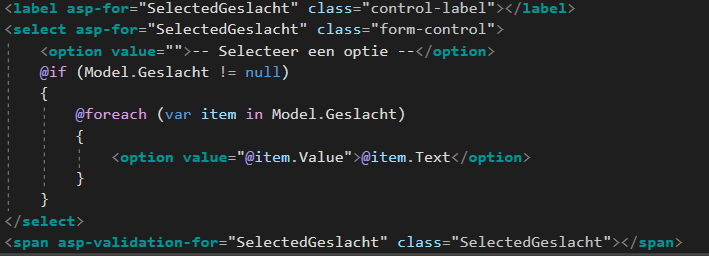


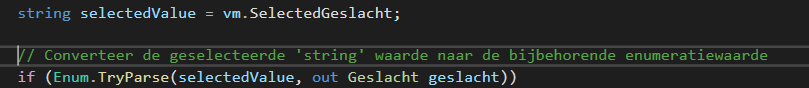
**3. Nieuwe hond toevoegen**

**Create-operatie, viewModels**

Voorzie een Create-operatie in de controller. Deze werkt samen met de View(models) en models. Maak dus ook een nieuw viewmodel aan: CreateHondViewModel

**Dropdown in create-acties**  
Maak een dropdown voor het geslacht, gebruik volgende code voor de enumeratie Geslacht af te loopen:

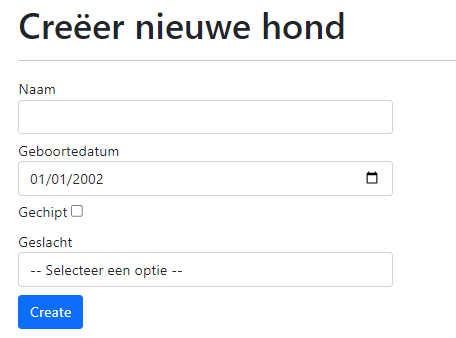
**Create (GET-actie)**  
Deze code cast en parst de enumeratie Geslacht naar string values.  
ViewModel  
  
View  


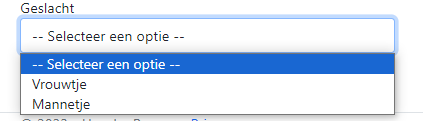
**Create (POST-actie)**  
  
In het formulier komt een string terug binnen, maar geslacht is een enumeratie, dus moeten we terug omzetten naar enumeratie.  


Alternatief enum: Indien je opteert om toch rechtstreeks string te gebruiken (bv. Moeilijkheidsgraad), pas je o.a. model aan voor geslacht naar string en bouw je een dropdown gebaseerd op string geslacht vanuit het model.

Vervolledig nu de view van Create.cshtml.

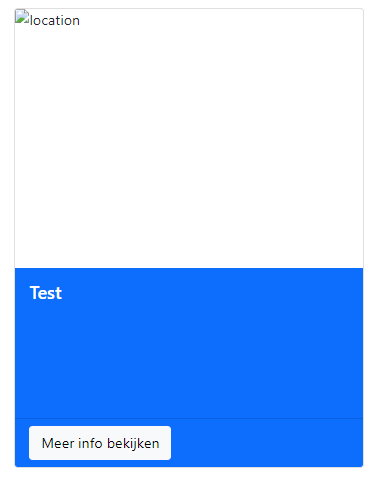
Browser view resultaat:



Dropdown:  


Voorzie ook validaties zoals (eerder) geleerd voor elk veld in het formulier.

Als dit succesvol lukt, zal de hond worden toegevoegd, echter de afbeelding werkt **nog niet**, omdat je nog geen upload hebt voorzien. **Hiervoor verwijzen we naar de verplichte uitbreiding om dit recht te zetten.**



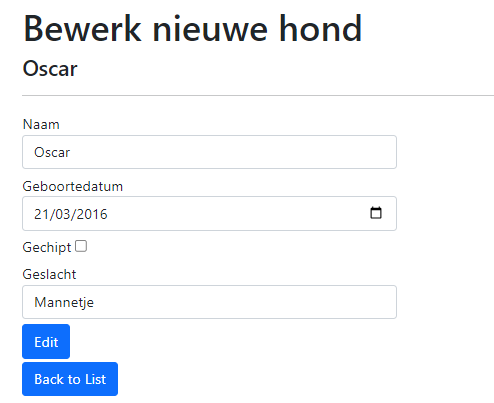
**4. Bestaande hond aanpassen**

Tip: Edit vergelijkbaar met Create..

Indien de hond niet bestaat op basis van de PK, dan toon je een notFound error.  
Maak dus een nieuw viewmodel aan: EditHondViewModel  
  
Voorzie een Update-operatie in de controller. Deze werkt samen met de View(models) en models.

**Dropdown in edit-acties**  
De dropdown is gelijkaardig aan die van Create, maar er is dit keer al een geslacht bekend.

Voorzie net zoals in de Create de inlading van de enum, dit keer is wel selectedGeslacht rechtstreeks al bekend vanuit de database. Geef dit mee aan het viewModel. In de database zal het geslacht als 1 of 2 staan genoteerd. We moeten dit casten naar de enum en dan naar de string.  


Browser view resultaat:  


Voorzie ook validaties zoals eerder geleerd voor (zie create)

**5. Bestaande hond verwijderen**

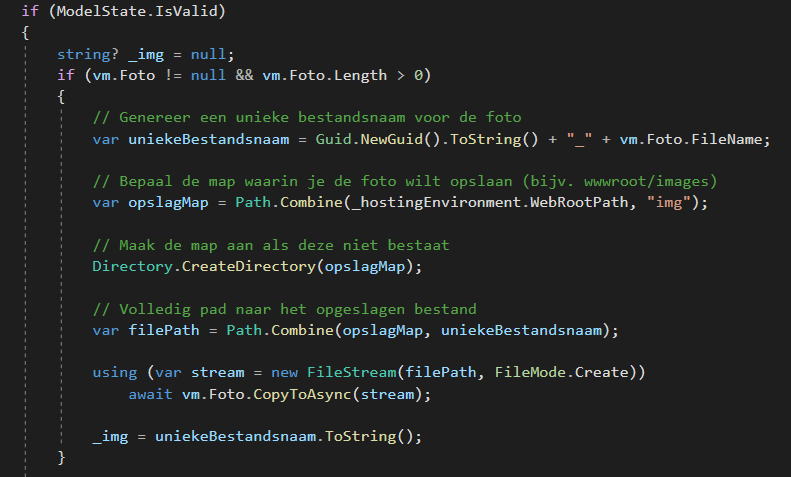
Je voorziet maar één methode voor delete. Dit is geen POST, maar een GET. Deze voert rechtstreeks de delete-operatie uit.

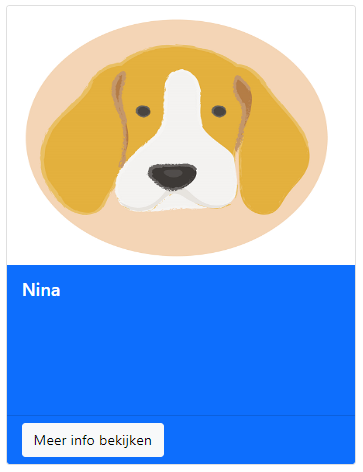
Indien de hond niet bestaat op basis van de PK, dan toon je een return notFound() 404-error.   
In het andere geval **verwijder je het record met die PK** én stuur je deze naar de methode Index.  
  
Niet gevonden PK:126:  


**6. Uitbreiding**

**Uitbreiding deel 1 (verplicht te maken)**

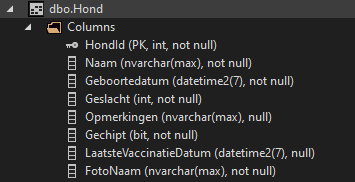
Tijdens het toevoegen van een nieuwe hond, voorzie je een **upload veld** voor de hond zijn/haar afbeelding. Je zal dus het model en de algemene view ook moeten bijwerken. Je zorgt dat deze upload ook in de database zal worden opgeslagen.   
  
Injecteer volgende bovenaan de constructor Hond en pas DI toe.  


De code om up te loaden:  


Wijzig de NA naar NNA van property FotoNaam.  
  



Tijdens het bewerken voorzie je geen upload afbeelding mogelijkheid.

**De database ziet er uiteindelijk zo uit:**  


**Uitbreiding deel 2:**  
-Zorg ervoor dat je maar één viewModel hebt voor CU (Create/Update).  
-Delete:  
Maak dus een nieuw viewmodel aan: VerwijderHondViewModel  
Voorzie een Delete-operatie in de controller. Deze werkt samen met de View(models) en models.  
  
Uitbreiding: De reeds aanwezige eerste hond (PK=1) vanuit het script kan nooit verwijderd worden, zorg hiervoor. Gebeurt dit toch via de url/submit dan toon je een return BadRequest();