# **Demo App – HondenAsiel**

## **Databank**

**Tabel HondenAsiel\_Hond:**

-) ID

-) Naam

-) Leeftijd

-) Foto (URL of Path File)

-) Ras (Foreign Key naar genormaliseerde Ras Ref tabel)

-) Kleur (Foreign Key naar genormaliseerde Kleur Ref tabel)

-) Geslacht (Foreign Key naar genormaliseerde Geslacht Ref tabel)

-) Heeft\_Stamboom (boolean)

-) Omschrijving\_Karakter

-) Asiel\_ID (Foreign key naar Asiel tabel)

**Tabel HondenAsiel\_Asiel**

-) ID

-) Naam

-) Adres

-) Eigenaar

**Tabel HondenAsiel\_Ref\_Ras**

-) ID

-) Code

-) Omschrijving

**Tabel HondenAsiel\_Ref\_Kleur**

-) ID

-) Code

-) Omschrijving

**Tabel HondenAsiel\_Ref\_Geslacht**

-) ID

-) Code

-) Omschrijving

## **Commands**

* LaadFotoOpCommand (waar de URL of het Path aan meegegeven wordt)
* VoegHondToeCommand (waar een HondToevoegenDTO aan meegegeven wordt)
  + Geeft een HondDTO met de ID van de toegevoegde hond terug

## **Queries**

* GetHondenQuery (waar een SearchCrteriaDTO aan meegegeven wordt)

## De SearchCrteriaDTO bevat ID, Naam, Ras, Kleur en Geslacht

## Geeft een lijst van HondDTO’s terug

## **Repositories**

* HondRepository met:
  + GetHondById method (waar een ID aan meegegeven wordt) 🡪 Geeft hond entity terug en gooit exception bij ID <=0
  + GetHonden method (waar een SearchCrteriaDTO aan meegegeven wordt) 🡪 geeft een lijst van HondEntities terug
  + VoegHondToe (waar een HondToevoegenDTO aan meegegeven wordt) 🡪 geeft HondEntity met ID terug

## **Unit testen**

* Werken met .Net Core In Memory db
* GetHondenQuery\_geeft\_correcte\_aantal
  + meedere testcase op basis van verschillende searchcriteria)
  + Maakt gebruik van TestCaseDataSource en expected value
* LaadFotoOpCommand\_Foto\_Uploaded
* GetHondById\_Ongeldige\_ID\_Throws\_Exception
  + Test op Repo die exception moet opwerpen
* VoegHondToeCommand\_hond\_toegevoegd