

Examenopdracht EWDJ

3e examenperiode 22-23

Bouw een Spring Boot applicatie die toelaat om dieren in een asiel te beheren.

Rollen

Er zijn twee rollen in de applicatie: admin en user. Een admin kan dieren toevoegen en verwijderen. Een user kan dieren reserveren.



Eerste scherm

Op het eerste scherm zien we een overzicht van de dieren in het asiel. We zien de naam van het dier, ras, geboortedatum, of het dier reeds gereserveerd is.

Je voorziet een hyperlink om naar de detailpagina van het dier te gaan.

Een admin ziet ook een knop of link om dieren toe te voegen, waarmee men naar een nieuwe pagina navigeert.

De dieren zijn gesorteerd op ras, daarna op geboortedatum oplopend.

Je voorziet een link om te filteren en enkel een bepaald ras te zien.

Detailpagina dier

Op de detailpagina van het dier ziet een gewone gebruiker alle details van het dier:

- Naam
- Optioneel: afbeelding van het dier
- Ras
- Geslacht
- Leeftijd
- Medische kosten (steeds met 2 cijfers na de komma)
- Kan met jonge kinderen
- Kan met oudere kinderen
- Kan met katten



- Kan met honden
- Geschikt als binnenkat
- Reeds gereserveerd
- Per verblijfplaats: hokcode1, hokcode2 en hoknaam. Een dier kan meerdere verblijfplaatsen hebben, maar een verblijfplaats kan maar door één dier tegelijk in gebruik zijn.

Er staat ook een knop: reserveren/vrijgeven, afhankelijk of deze gebruiker het dier reeds gereserveerd heeft. Als je hierop klikt, keer je terug naar de overzichtspagina, en komt er op deze overzichtspagina een bericht: “[naam van dier] werd gereserveerd” (of vrijgegeven, natuurlijk).

Wanneer iemand anders het dier reeds gereserveerd heeft, staat deze knop er niet, maar staat er: “reeds gereserveerd”.

Een admin kan echter het dier toch vrijgeven (bijvoorbeeld na telefonisch contact, of omdat het dier na een tijd nog niet werd opgehaald).

Elke user heeft een maximum aantal reservaties: een basisgebruiker kan slechts één dier reserveren. Een “asielfan” kan twee dieren reserveren.

Dieren toevoegen

Deze pagina is enkel beschikbaar voor de admin rol. Er staat een link op het overzichtsscherm die enkel zichtbaar is voor admins.

Op deze pagina zien we een formulier:

- Naam
- Optioneel: afbeelding van het dier
- identificatiecode
- Ras
- Geslacht
- Geboortedatum
- Medische kosten
- Kan met jonge kinderen
- Kan met oudere kinderen
- Kan met katten
- Kan met honden

- Geschikt als binnenkat
- Per verblijfplaats: hokcode1, hokcode2 en hoknaam. Een dier kan meerdere verblijfplaatsen hebben, maar een verblijfplaats kan maar door één dier tegelijk in gebruik zijn.

Validatie:

- Naam mag niet leeg zijn, mag geen cijfers of speciale tekens bevatten
- Identificatiecode mag niet leeg zijn, moet bestaan uit 10 karakters, de eerste 3 karakters moet uit letters bestaan, de volgende 7 uit cijfers. De 2 laatste getallen is de rest van de deling van de eerste 5 cijfers door 3. De code moet ook uniek zijn per dier.
- Ras mag leeg zijn, mag geen cijfers of speciale tekens bevatten
- Geslacht: mannetje of vrouwtje
- Geboortedatum mag leeg zijn, maar dan kies je tussen: jong, volwassen, senior (kies zelf hoe je dit op de detailpagina voor de user toont).
- De verschillende “kan met” zijn checkboxen, staan default uit
- Verblijfplaats: hokcode1 en hokcode2 moeten gehele getallen zijn die liggen tussen 50 en 300 (grenzen inbegrepen). Het hokcode2 moet minstens 50 groter zijn dan hokcode1. Hoknaam mag enkel uit letters bestaan.

Gereserveerde dieren

Op deze pagina toon je de dieren die reeds gereserveerd zijn.

In het geval je een gewone user bent, zie je je eigen gereserveerde dier(en).

Als admin zie je alle gereserveerde dieren.

Resource bundle(s)

Je dient enkel vertalingen te voorzien in ofwel het Nederlands of het Engels. Alle foutboodschappen komen uit de resource bundle(s). Er staan ook geen vaste getallen in de resource bundles.

Titel (let op html:head:title!) en alle tekst van een scherm naar keuze (dus geen error scherm, maar een van de schermen hierboven).

Logout

Op elk scherm staat er een knop om uit te loggen, samen met informatie over de huidige rol.

Deelopdrachten

1. Maak de Spring Boot applicatie volgens de gevraagde functionaliteiten.
2. Het is een Java opdracht. Je mag Javascript gebruiken op voorwaarde dat je applicatie ook werkt zonder Javascript.
3. Validatie: gebruik bestaande annotaties van jakarta.validation en validator klasse(n). Optioneel: eigen annotatie(s).
4. Voeg JPA toe: dieren, reservaties, locaties, ... worden gepersisteerd in een MySQL databank. Gebruik hiervoor CrudRepository en CommandLineRunner (om de databank te populeren).
5. Voeg Security toe zoals gevraagd:
 - Wanneer we niet ingelogd zijn, komen we op een login pagina.
 - Paswoorden slaan we op aan de hand van bcrypt.
6. Voeg unit tests toe voor alle controllers (@Controller en @RestController).
7. Implementeer een REST webservice, en voeg een klasse toe waarin je deze services opvraagt (Reactive Web Client). Voorzie volgende services:
 - De overeenkomstige dieren van een bepaald geslacht worden teruggegeven.
 - De dieren van een bepaald ras worden teruggegeven.
 - Je kan een lijst opvragen van nog niet gereserveerde dieren.

Algemeen

De lectoren zijn tijdens de vakantie niet beschikbaar voor feedback/hulp. In juni kunnen er wel nog vragen gesteld worden over de inhoud van de opdracht, niet de technische uitwerking. Bij twijfel: maak het niet nodeloos complex, maar zorg ervoor dat je wel aan de geest van de opdracht voldoet.

Maak je eigen klassen en interface(s). Er werden geen printscreens in de opdracht meegegeven omdat je zelf de layout van de schermen mag kiezen.

Maak gebruik van Thymeleaf en CSS.

Einde van de opdracht

- Maak een zip bestand van het volledige project.
- Hernoem dit bestand naar:
 - 'Klasgroep_Naam_Voornaam.zip'
 - Vb: 'A2_Willem_Wever'
- Laad dit zip bestand op bij de opdrachten op Chamilo ten **laatste zaterdag 2 september voor middernacht**.