Godkendelsesopgave 3 – Programmering og udvikling af små systemer samt databaser

Opgave 1: Lav en rest api, som holder styr på Jens Hansen's besætning

// Del A:
Ved brug af express.js opret en server, som lytter på port 8080, og som console logger beskeden: "Server lytter på port 8080":

Før jeg starter, benytter jeg mig af NPM til at installere pakken "express" der gør det muligt for mig at hoste den lokale server med porten 8080. Når den er installeret, kan jeg skrive følgende kode:

```
1   // Del A
2
3   const express = require('express');
4
5   const app = express ();
6   app.use(express.json());
7
8   const PORT = 8080;
9
10  app.listen(PORT, () => {
11      console.log(`Server lytter på port ${PORT}`)
12  });
```

I terminalen:

```
simonhj@Simons-MacBook-Pro-2 Opgave3 % node opg3.js
Server lytter på port 8080
```

// Del B:
Lav en var kaldet besætning med det følgende data, som beskriver, hvor mange dyr Jens Hansen har på sin bondegård.

I koden på linje 17, definerer jeg variablen ”besaetning” som et array der indeholder fire objekter:

```
15  // Del B
16
17  var besaetning = [
18      { id: 0, dyr: `koeer`, antal: 50 },
19      { id: 1, dyr: `Hunde`, antal: 1 },
20      { id: 2, dyr: `Grise`, antal: 100 },
21      { id: 3, dyr: `Faar`, antal: 20 },
22  ];
23
24  console.log(besaetning);
```

I terminalen:

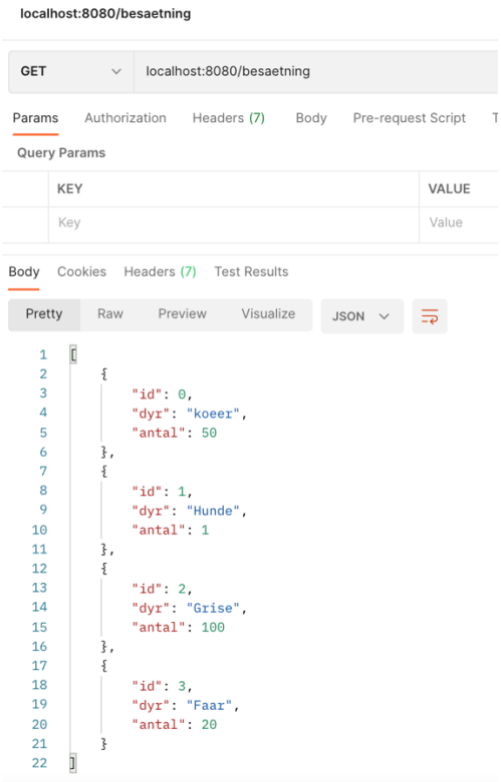
```
simonhj@Simons-MacBook-Pro-2 Opgave3 % node opg3.js
[
  { id: 0, dyr: 'koeer', antal: 50 },
  { id: 1, dyr: 'Hunde', antal: 1 },
  { id: 2, dyr: 'Grise', antal: 100 },
  { id: 3, dyr: 'Faar', antal: 20 }
]
Server lytter på port 8080
```

//Del C
Lav nu et get request med følgende sti '/returner_hele_besætningen/' som sende et json objekt tilbage med hele Jens Hansens besætning.

Bruger app.get til at definere get requested:

```
26  // Del C
27
28  app.get(`/besaetning`, (req, res, next) => {
29      res.send(besaetning);
30  });
31
```

Derefter bruger jeg postman til at tjekke mit get request:



//Del D
Nogle gange ønsker Jens Hansen dog ikke at få hele sin besætning, men bare antallet for en enkelt kategori.

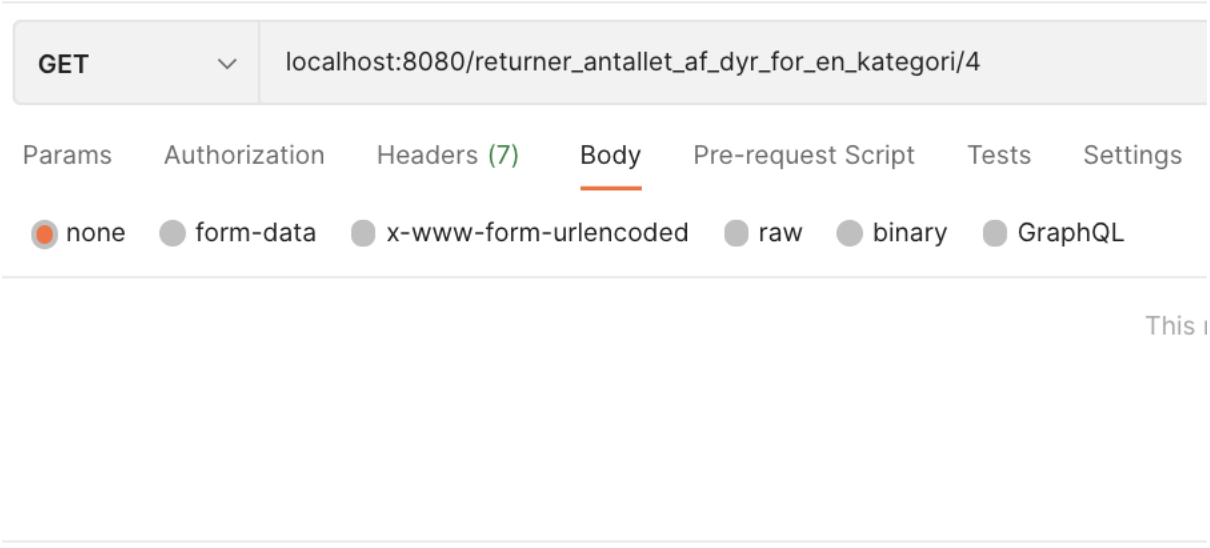
● Lav derfor et get request som returnerer antallet af dyr givet et get request med følgende sti 'returner_antallet_af_dyr_for_en_kategori/:kategori'.

Her definerer jeg ligesom i forrige opgave et get request,

```
32  // Del D
33
34  app.get(`/returner_antallet_af_dyr_for_en_kategori`,(req, res, next) => {
35      const kategori = req.params.kategori;
36      const antal = besaetning [kategori]
37          if (!antal){
38              res.status(400).send(`Kategorien du leder efter, findes ikke.`);
39              return
40          }
41          res.status(200).send(antal)
42  });
```

● Sørg for at tjekke at kategorien eksisterer og hvis den ikke gør at sende den korrekte fejlmeddelelse

Her prøver jeg at hente en udefineret kategori, og får nedenstående fejlkode:



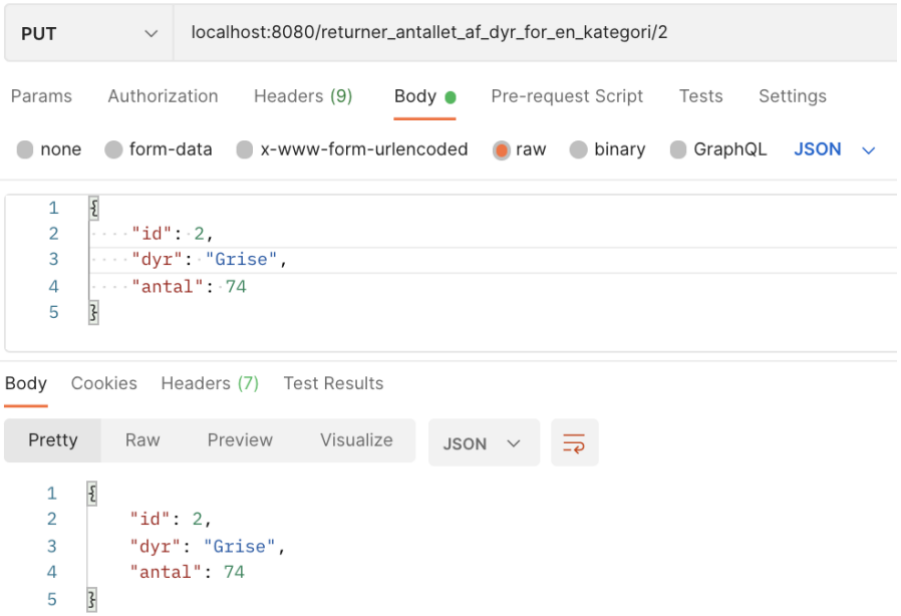
// Del E
Det sker fra tid til anden, at Jens Hansen's besætning ændrer sig. Han får enten flere eller færre dyr. Det vil han selvfølgelig godt kunne holde styr på.

● Lav derfor et put request, som kan opdatere hans besætning og returnere den opdaterede værdi.

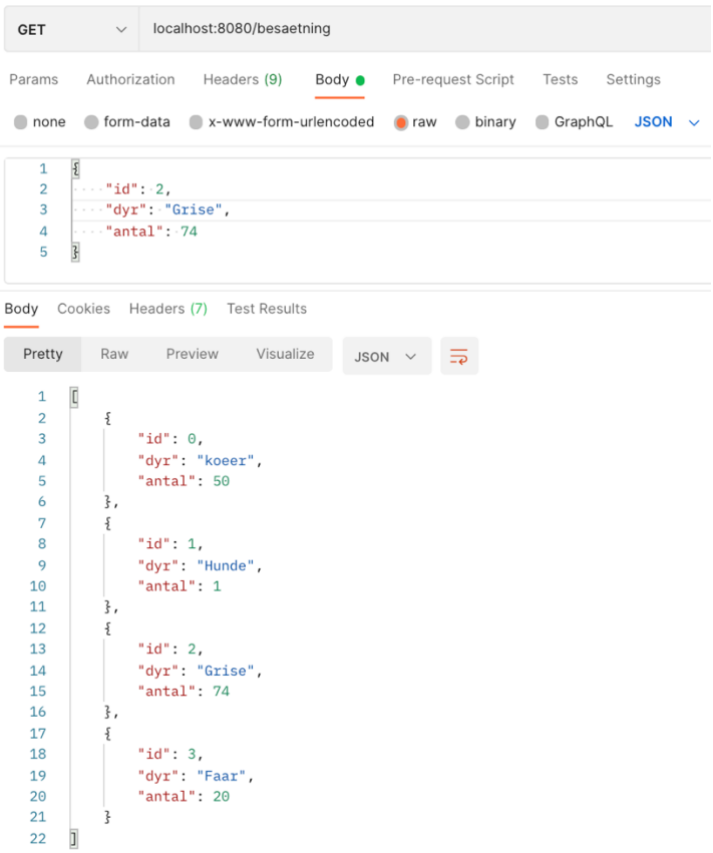
Jeg opretter her et put request der er i stand til at ændre i besætningen vha. en række array metoder.

```
// Del E

app.put(`/returner_antallet_af_dyr_for_en_kategori/:kategori`, (req, res, next) => {
    let found = besaetning.find(function (item) {
        return item.id === parseInt(req.params.kategori);
    });

    if (!found) {
        res.sendStatus(404);
    }
    let updated = {
        id: found.id,
        dyr: req.body.dyr,
        antal: req.body.antal
    };

    let targetIndex = besaetning.indexOf(found);

    besaetning.splice(targetIndex, 1, updated);

    res.status(200).json(updated);

});
```

Jeg bruger nu postman for at ændre og tjekke i besætningen:



Her går jeg manuelt ind og redefinerer kategori to, eller grisene, jeg ænder i antallet, det gøres i JSON format. Når jeg så bruger get, kan jeg se at den har opdateret antallet af grise:



// Del F

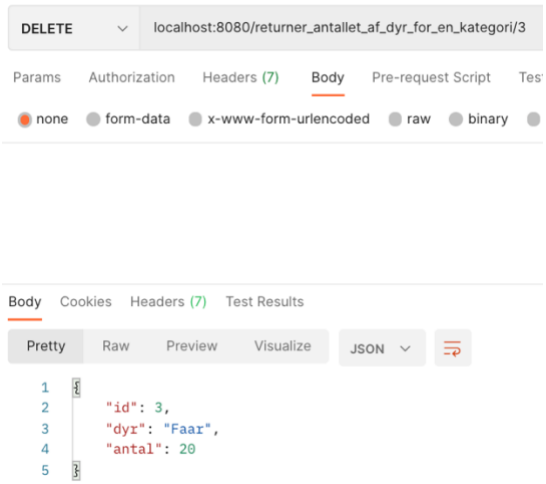
Nogle gange vælger Jens Hansen at stoppe med at have nogle bestemte dyr og så vil han også gerne kunne fjerne dem fra hans besætning.

● Lav derfor et delete endpoint som kan slette de dyr, Jens Hansen ikke længere vil have i sin besætning.

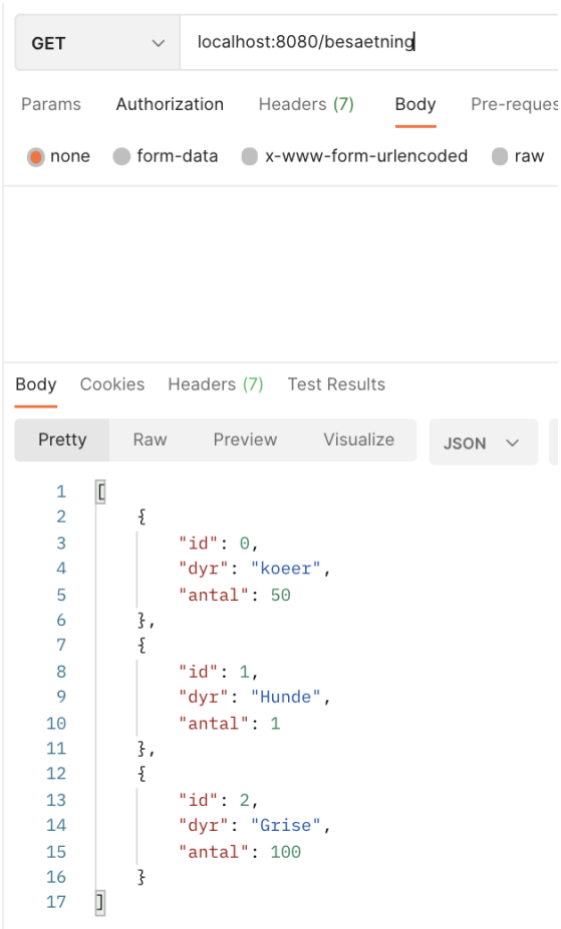
Benytter mig af app.delete til at oprette mit delete request, her benytter jeg mig igen af array metoder til at kunne slette hele objekter fra mit oprindelige array ”besaetning”.

```
68  // Del F
69
70  app.delete(`/returner_antallet_af_dyr_for_en_kategori/:kategori`, (req, res, next) => {
71      let found = besaetning.find(function (item) {
72          return item.id === parseInt(req.params.kategori);
73      });
74
75      if (found) {
76          let targetIndex = besaetning.indexOf(found);
77          besaetning.splice(targetIndex, 1);
78      }
79      res.status(200).json(found);
80  });
81
```

Jeg tester nu med postman:



Vi kan se på billedet til højre at ”Faar” objektet nu helt er slettet fra arrayet.



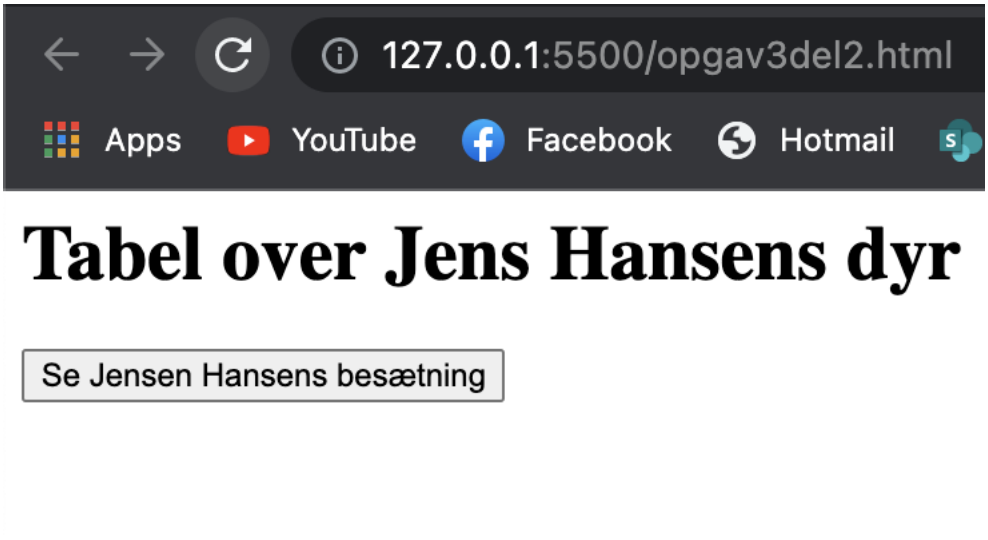
Opgave 2: Lav en frontend til så Jensen Hansen kan se sin besætning ved et klik

//Del A:
Lav en simpel html side, hvor Jensen Hansen kan se hele sin besætning i tabel form, når han trykker på en knap som siger: Se besætning (brug jeres løsnings fra del C)

Jeg opretter nu et nyt html dokument i visual studio, hvor jeg her opretter en simpel hjemmeside med noget inline javascript, jeg opretter to funktioner: showTable og hideTable. Vi gør sådan at ved "onload" eller som udgangspunkt, at tabellen er skjult (hideTable) og ved klik på knappen at tabellen er vist (showTable). Koden kan ses i nedenstående billede:

```
<html>
<h1> Tabel over Jens Hansens dyr </h1>
<script>
    function showTable() {
        document.getElementById(`table`).style.visibility = `visible`;
    }

    function hideTable() {
        document.getElementById(`table`).style.visibility = `hidden`;
    }
</script>

<body onload="javascrip:hideTable()">
    <input type="button" onClick="javascript:showTable();" value="Se Jensen Hansens besætning">
    <table id="table" border=1>
        <tr>
            <th>Kategori</th>
            <th>Antal</th>
        </tr>
        <tr>
            <td>Køer</td>
            <td>50</td>
        </tr>
        <tr>
            <td>Hunde</td>
            <td>1</td>
        </tr>
        <tr>
            <td>Grise</td>
            <td>100</td>
        </tr>
        <tr>
            <td>Får</td>
            <td>20</td>
        </tr>

    </table>
</body>

</html>
```

Onload:



Onclick: