

Shop

Vi skal i denne opgave lave en indkøbsapplikation. Det sker i en række skridt, hvor noget er præcist beskrevet, andet kun er antydnet, og noget skal man tænke en del til.

1. Lav en express server (f.eks. shopapp.js). Lav alle de nødvendige require, app.set mm. I første omgang er session ikke nødvendig.
2. Lav en app.get('/index' ...), der rammer en index.pug. Vi bruger en pug template, ellers er det svært at få parametre med.
3. Test, at siden kan nås.
4. Der skal være en liste over produkter, man kan købe, så der skal måske være en tabel over dem. Det kan nok betale sig at lave en variabel med produkter i shopapp.js-filen. Listen overføres til index.pug, og listen skal skrives ud her vha. en løkke
5. Vi skal i de næste punkter, for hvert produkt, lave en knap, der laver en post til serveren, dvs. en onclick, der kalder en fetch, evt. indirekte vha. post.metoden. Dette post skal medsende en id på det produkt, man har klikket på. Så produkterne skal altså have en id.
 - a. Lav knappen i pug-siden vha. button(onclick='clickfunction("'" + product.id + "'')
 - b. Selve scriptet laves i html-siden, da pug ikke genererer en .js fil. Her er lidt hjælp til det:
 - c. **script.**

```
    async function post(url, objekt) {
      const respons = await fetch(url, {
        method: "POST",
        body: JSON.stringify(objekt),
        headers: {
          'Content-Type': 'application/json'
        }
      });
      if (respons.status !== 201) // Created
        throw new Error(respons.status);
      return await respons.text();
    }
```

- ```
let clickfunction = async (id) => {
 try {

```
- d. Se i eksemplet hvordan post fungerer i index.js og i app.js. Man skal bare tilbage til index-siden igen
6. Dette post skal ramme en matchende app.post på serveren, så den skal også laves.
    - a. app.post på serveren: lav den f.eks. med url'en /buy, så klientsiden skal altså ramme http://localhost:8080/buy  
i første omgang gælder det bare om at fange post'et fra browseren, så lav evt. bare en console.log.
    - b. app.post på serveren: nu skal vi til at gemme id på produktet. Sæt request.session.<variabelnavn> = id. Fang id vha const { id } = request.body;
  7. Du får nok en fejl, at request.session er undefined. Lav den nødvendige session-opsætning. Så længe request.session er undefined er det ikke korrekt sat op

8. Nu skulle vi have fået gemt den id, man klikkede på, i session. Så skal vi have den skrevet ud nedenunder listen af produkter.
  - a. Id findes fra request.session.<variabelnavn> i app.get('/index' metoden
  - b. Lad id blive overført til index.pug sammen med listen af produkter.
  - c. Udskriv id under listen af produkter
9. Tilret, så det ikke er en enkelt id, der sendes med session, men derimod en liste af id'er (eller en liste af produkter, det gør nok ikke den store forskel), en indkøbskurv.- Så hver gang, der klikkes på et produkt, tilføjes id til listen.
10. Udskriv listen af produkter, man har valgt (indkøbskurven), i stedet for det enkelte id. Der skal kodes lidt, for produkterne skal findes ud fra id i listen over alle produkter
11. Valgfri. Udvid med login.