

Opgave: Opsæt din egen Express-server

I denne opgave skal du sætte en **Express-server** op, der kommunikerer med en simpel frontend, eller en frontend du har i forvejen. Målet er at lære, hvordan backend (server) og frontend kan arbejde sammen ved hjælp af et REST API.

Trin 1: Forbered projektet

1. Opret en ny mappe til dit projekt, f.eks. express-server-opgave.
2. Åbn terminalen, naviger til mappen, og initialiser et nyt Node.js-projekt:

```
npm init -y
```

3. Installer **Express**:

```
npm install express
```

Trin 2: Opsæt serveren

1. Opret en fil med navnet server.js.
2. Brug følgende skabelon til at opsætte serveren:

```
3. const express = require("express");
4. const app = express();
5.
6. app.use(express.json());
7.
8. let products = [
9.   { id: 1, navn: "Product A", pris: 100 },
10.  { id: 2, navn: "Product B", pris: 200 },
11. ];
12.
13. app.get("/products", (req, res) => {
14.   res.json(products);
15. });
16.
17. app.post("/products", (req, res) => {
18.   const newProduct = req.body;
19.   newProduct.id = products.length + 1;
20.   products.push(newProduct);
21.   res.status(201).json(newProduct);
22. });
23.
24. app.listen(3000, () => {
25.   console.log("Serveren kører på http://localhost:3000");
26. });
27.
```

Trin 3: Start serveren

1. Åbn terminalen i projektmappen, og kørs følgende kommando for at starte serveren:

```
node server.js
```

2. Du bør se beskeden:

Serveren kører på <http://localhost:3000>

Trin 4: Test serveren

1. GET endpoint:

- Åbn en browser, og naviger til <http://localhost:3000/products>.
- Du bør se en liste med produkter som JSON

2. POST endpoint:

- Åbn postman og send en POST-forespørgsel til <http://localhost:3000/products>.
- Forespørgslen skal have følgende JSON i kroppen:

```
{ "navn": "Product C", "pris": 300 }
```

- Serveren skal svare med det tilføjede produkt:

```
{ "id": 3, "navn": "Product C", "pris": 300 }
```

Trin 4: Byg en simpel frontend (React), eller start én du allerede har.

1. Opret en Products.jsx
2. Lav en fetch-funktion der sender en GET-request til din express-server (<http://localhost:3000/products>)
3. Udskriv produkterne på siden.

Virker det ikke?

```
✖ Access to fetch at 'http://localhost:3000/products' from origin 'http://localhost:5173' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.
✖ ▶ GET http://localhost:3000/products net::ERR_FAILED BackofficeItems.jsx:254 47
200 (OK)
```

Denne fejl skyldes, at din server ikke tillader cross-origin requests som standard. Dette er en sikkerhedsforanstaltning i browsere, der kaldes **CORS (Cross-Origin Resource Sharing)**.

Når din frontend og backend kører på forskellige domæner eller porte (f.eks. frontend på `http://localhost:5173` og backend på `http://localhost:3000`), skal backend eksplicit tillade sådanne requests.

For at løse dette problem, skal du konfigurere din Express-server til at tillade CORS. Dette kan nemt gøres ved hjælp af et populært middleware kaldet **CORS**.

1. Installer **CORS-middleware**

I din terminal, i projektmappen, kørs følgende kommando:

```
npm install cors
```

2. Tilføj **CORS-middleware** i din Express-server:

Åbn `server.js`, og tilføj følgende linje i toppen af filen (efter `const app = express();`):

```
const cors = require('cors');  
app.use(cors());
```

3. Genstart din server:

Når du har tilføjet CORS-middleware, skal du stoppe din server og starte den igen:

```
node server.js
```