

Eksamen - UIN25_Gruppe35

Githubbruker JoachimLC tilsvaret Joachim Lundsgaard Christiansen på canvas

Github: <https://github.com/JoachimLC/Eksamen-interaktive-nettsteder>

Prosess

I dette dokumentet har jeg dokumentert hvor jeg møtte på vanskeligheter, eller hvor jeg var usikker på hvordan jeg skulle gå frem for å oppnå fungerende kode. Her beskrives hvilke kilder som ble benyttet, samt prompts til kunstig intelligens som ble brukt for å løse spesifikke problemer.

Kode som ikke er spesifikt dokumentert her, ble utviklet enten gjennom egen utprøving eller ved å bruke tilgjengelig materiale fra Webtricks og emnet. Det er mulig at jeg noen steder har benyttet metodikk som ikke samsvarer helt med kursets retningslinjer, ettersom jeg plukket opp noen teknikker fra SPA-emnet med Marius forrige semester.

Jeg har ikke deltatt i alle forelesninger, og er derfor ikke 100 % sikker på hva som var ønsket fremgangsmåte for all kode. Likevel har jeg gjort mitt beste for å følge stilen som er brukt i de kildene og ressursene som er lenket til på Canvas og Webtricks.

Som rapporten viser, har jeg i hovedsak benyttet ChatGPT for å få hjelp med syntaks og for å rette opp i kode der jeg har stått fast. Jeg har aktivt forsøkt å bruke denne eksamenen som en del av eksamenslesingen ved å prøve å løse mest mulig fra minnet, uten å søke opp løsninger umiddelbart.

Målet har vært å styrke min forståelse av pensum ved å jobbe gjennom problemer på egen hånd først, og deretter bruke ChatGPT som et verktøy for å justere syntaks eller finne alternative måter å strukturere koden på når jeg ikke kommer på flere ting å prøve.

Målsetning og arbeidsmengde

Jeg har siktet meg inn på karakter B og mener å ha oppfylt alle minstekrav opp til og inkludert karakter B. Jeg er alene på gruppen, og har gjennomført mesteparten av arbeidet over en ukes tid. Siden jeg kun er en person på gruppen, har jeg utelukkende benyttet main branch til koding.

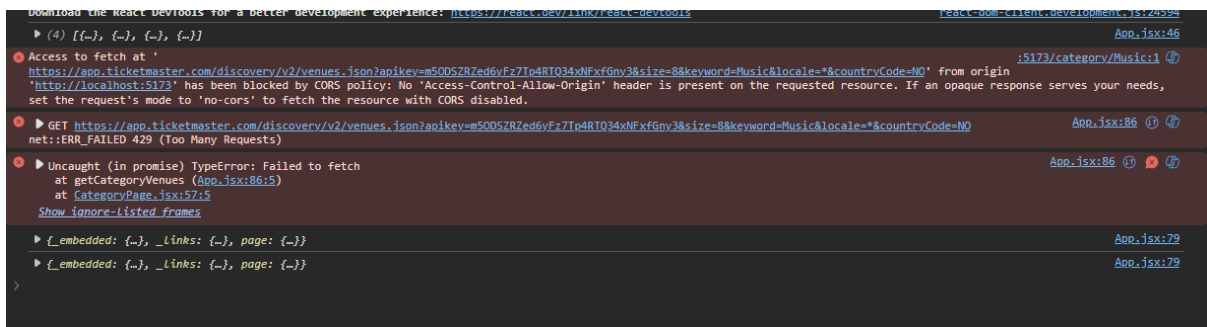
Under følger rapport vedrørende eksterne kilder og bruk av prompt til ChatGpt. Det er kommentert i koden, og deretter forklart rundt prosessen i nærmere detalj her.

Bemerkning: Problemer med too many requests

Jeg har satt opp funksjonalitet for å hente data fra alle endepunktene i API-et, inkludert festivaler, storbyevents og kategoriserte events med filtrering. Under utvikling og testing har jeg imidlertid støtt på et problem som oppstår når jeg refresher siden flere ganger på kort tid. Da får jeg av og til opp feilmeldinger som indikerer at jeg har sendt for mange forespørsler til API-et, noe som resulterer i feilkoden Too Many Requests.

Denne feilen tolker jeg som en indikasjon på at API-et har en rate limit som begrenser antall forespørsler innenfor et gitt tidsintervall. Siden fungerer som forventet dersom jeg venter litt og refresher etter en liten stund. Jeg har forsøkt forskjellige implementasjoner av fetch, inkludert løsninger utover det som ble vist i kurset, men oppførselen forblir den samme.

På denne bakgrunnen antar jeg at problemet skyldes en begrensning i API-et, og ikke nødvendigvis en feil i min egen implementasjon. Alt skal kunne lastes inn som forventet, både festivaler, storbyevents og kategoriserte events, men noen ganger må jeg vente litt og refreshe siden et par ganger før dataene lastes inn.



```
Download the React DevTools for a better development experience: https://react.dev/learn/react-devtools
React Code Editor: View component
App.jsx:46
▶ (4) [{"_": "..."}, {"_": "..."}, {"_": "..."}]
Access to fetch at 'https://app.ticketmaster.com/discovery/v2/venues.json?apikey=M50DSZRZed6yFz7Tp4RT034xNFxfGny3&size=8&keyword=Music&locale=*%&countryCode=NO' 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.
App.jsx:86
▶ GET https://app.ticketmaster.com/discovery/v2/venues.json?apikey=M50DSZRZed6yFz7Tp4RT034xNFxfGny3&size=8&keyword=Music&locale=*%&countryCode=NO net::ERR_FAILED 429 (Too Many Requests)
App.jsx:86
Uncaught (in promise) TypeError: Failed to fetch
    at getCategoryVenues (App.jsx:86:5)
    at CategoryPage.jsx:57:5
Show ignore-listed frames
App.jsx:79
  { "_embedded": { "...", "_links": { "...", "page": { "..."} } }
App.jsx:79
  { "_embedded": { "...", "_links": { "...", "page": { "..."} } }
```

Karakter E minstrekrav

Dashboard - forms

Jeg husket ikke nøyaktig bruk av forms i javascript, så dette måtte jeg sjekke opp. Jeg benyttet kilden: [<form> – React](#) for innloggingsskjema på dashboard

App.jsx - getFestivalEvents

For å hente ut festivaler fra API-et forsøkte jeg først å bruke ulike parametere for å få alle fire festivalene i samme respons. Det mest spesifikke kallet jeg kom frem til var:

<https://app.ticketmaster.com/discovery/v2/attractions.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&domain=ticketmaster.no&size=200&classificationName=festival>

Tanken var å bruke classificationName=festival slik findings hadde på subtype

```
▶ (4) [{...}, {...}, {...}, {...}] ⓘ
  ▶ 0:
    ▶ classifications: Array(1)
      ▶ 0:
        family: false
        ▶ genre: {id: 'KnvZfZ7vAe6', name: 'Undefined'}
        primary: true
        ▶ segment: {id: 'KZFzniwnSyZfZ7v7nJ', name: 'Music'}
        ▶ subGenre: {id: 'KZazBEonSMnZfZ7v6JI', name: 'Undefined'}
        ▶ subType: {id: 'KZFzBErXgnZfZ7vA6E', name: 'Festival'}
        ▶ type: {id: 'KZAyXgnZfZ7v7lt', name: 'Event Style'}
        ▶ [[Prototype]]: Object
        length: 1
      ▶ [[Prototype]]: Array(0)
    ▶ externallinks: {spotify: Array(1), instagram: Array(1), homepage: Array(1)}
    id: "K8vZ917K7fV"
    ▶ images: (11) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]
    locale: "en-us"
    name: "Findings Festival"
```

men NEON og Skeikampen hadde undefined på disse

Jeg forsøkte filtrere på domain=ticketmaster.no, men skeikampenfestivalen ligger under .com domain i api responsen, så jeg kan ikke filtrere domain uten å uten å ekskludere noen festivaler.

Jeg fikk altså bare til å hente alle festivalene dersom jeg la inn keyword for hver festival, men da måtte jeg hente de en og en. Etter googling fant jeg ingen løsning, så jeg ville undersøke om det finnes en bedre løsning enn å kalle på dem en og en.

Derfor spurte jeg chatGpt om dette

Hei, jeg benytter ticketmaster discovery api til å hente ut attractions, og skal hente ut disse 4 Findings, Neon, Skeikampenfestivalen, og Tons of Rock. Hvordan kan jeg tilpasse api spørringen til å få ut alle disse 4 i et api kall? Her er koden min per nå

```
function App() {
  const [events, setEvents] = useState([])
  const getEvents = async () => {
    const response = await fetch(
      'https://app.ticketmaster.com/discovery/v2/attractions.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&domain=ticketmaster.no&size=200&classificationName=festival'
    );
    const data = await response.json();
    const evenList = data._embedded
    setEvents(evenList)
  }
  useEffect(() => {
    getEvents()
  }, [])
}
```

For å hente spesifikt ut bare Findings, Neon, Skeikampenfestivalen og Tons of Rock fra Ticketmaster Discovery API, må du filtrere på en måte som matcher bare disse.

Dessverre tilbyr ikke Ticketmaster Discovery API en måte å søke på flere spesifikke attraction-navn i én spørring direkte, men du har noen alternativer:

Alternativ 1: Bruk `keyword` og filter i frontend

Du kan søke bredt og så filtrere resultatene i frontend:

```
js
const festivalNames = ['Findings', 'Neon', 'Skeikampenfestivalen', 'Tons of Rock'];

const getEvents = async () => {
  const response = await fetch(
    'https://app.ticketmaster.com/discovery/v2/attractions.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&domain=ticketmaster.no&size=200&classificationName=festival'
  );
  const data = await response.json();
  const allAttractions = data._embedded?.attractions || [];

  // Filtrer på navnene
  const filtered = allAttractions.filter(attraction =>
    festivalNames.includes(attraction.name)
  );

  setEvents(filtered);
};
```

Alternativ 2: Lag 4 separate kall med `keyword`

```
js
const getEvents = async () => {
  const festivals = ['Findings', 'Neon', 'Skeikampenfestivalen', 'Tons of Rock'];
  const fetches = festivals.map(name =>
    fetch(`https://app.ticketmaster.com/discovery/v2/attractions.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&domain=ticketmaster.no&size=200&classificationName=festival&keyword=${name}`)
      .then(res => res.json())
  );

  const results = await Promise.all(fetches);
  const all = results.flatMap(r => r._embedded?.attractions || []);
  setEvents(all);
};
```

Det virket dermed å kun være mulig gjennom 4 separate kall. Jeg justerte festivalnavnene i arrayen for festivals til å inkludere ordet festival, slik at det returnerte færre responser. Koden min for å hente ut de spesifikke festival eventene fra APllet ble følgende (funksjonen i ferdig applikasjon heter getFestivalEvents):

```
const getEvents = async () => {
  const festivals = ['Findings festival', 'Neon festival',
'Skeikampenfestivalen', 'Tons of Rock'];
  const fetches = festivals.map(name =>

fetch(`https://app.ticketmaster.com/discovery/v2/attractions.json?apikey=m50DS
ZRZed6yFz7Tp4RTQ34xNFxfGny3&keyword=${encodeURIComponent(name)}`)
    .then(res => res.json())
  );

  const results = await Promise.all(fetches);
  const all = results.flatMap(r => r._embedded?.attractions || []);
  setEvents(all);
};
```

Denen koden henter ut festivalene ved å gjør et kall med hvert festivalnavn som keyword. Den venter på alle kallene gjennomføres, og når alle kallene er gjort henter den ut attractions fra hver festival, som ligger under ._embedded med flatmap.

Dette returnerte fortsatt med noen ekstra festivaler, så jeg måtte få ut kun de spesifiserte festivalnavnene, så da måtte jeg legge til filtrering med javascript some før jeg brukte setEvents. jeg brukte w3schools som referanse på hvordan bruke disse;

[JavaScript Array filter\(\) Method](#)
[JavaScript Array some\(\) Method](#)

og endte opp med:

```
const filtered = all.filter(attr =>
  festivals.some(name =>
    attr.name === name
  )
);

setEvents(filtered);
```

karakter D minstekrav

Url syntaks for å lenke til teaterstykker kategorien

Jeg valgte å benytte slug på kategorisiden (CategoryPage.jsx) til å hente relevante data i api kallene mine gjennom classifications. Derfor måtte jeg lage mine slugs slik at de matchet med tilsvarende classifications i APIet. For musikk og sport gikk dette fint, music og sports, men teater ligger under classification Arts & Theater. Utfordringen er mellomrommene jeg da ville måtte benytte i sluggen.

Jeg tenkte at jeg bare kunne benytte % som mellomrom slik jeg har gjort i url parametere før, men dette fungerte ikke. Så jeg spurte chatgpt om løsningen min.

would this work

```
<li><Link to={/category/Arts%&%Theater}>Arts & Theater</Link></li>
```

No, that will **not work** as expected. The URL will interpret the `%&%` as a literal string, and it won't be properly encoded for URL parameters.

✓ Solution:

You need to encode the parameter properly using `encodeURIComponent` or manually replace spaces and special characters.

✓ Option 1: Manually Encode:

jsx

Copy

Edit

```
<li><Link to={`/${category}/Arts%20%26%20Theater`}>Arts & Theater</Link></li>
```

- `%20` represents a space.
- `%26` represents the `&` symbol.

✓ Option 2: Using `encodeURIComponent` :

jsx

Copy

Edit

```
const category = encodeURIComponent("Arts & Theater");
<li><Link to={`/${category}/${category}`}>Arts & Theater</Link></li>
```

This will automatically convert the string to `Arts%20%26%20Theater` .

Jeg likte ingen av alternativene jeg fikk av chatgptt, så jeg valgte heller å legge inn en liten quickfix med å benytte en ekstra variabel for å håndtere dette spesfikke tilfellet, da det ellers fungerte som ønsket.

```
const { slug } = useParams();

let newVariableForSlug = slug;

if (newVariableForSlug === "Arts%%Theater") {

  newVariableForSlug = "Arts & Theater";

}
```

karakter C krav

Eventcard - hjelp til syntaks

Jeg skulle tilpasse eventcard.jsx slik at jeg kunne benytte denne til visning av data også fra storyby-eventer. Tilpassningen besto av å inkludere visning av navn for city når dette var relevant, men jeg ønsket fortsatt å kunne benytte eventcard.jsx til både festivaldata og de 10 arrangementer fra en storby. Festivaldata holder på attraksjoner, og har ikke city som variabel. For å kunne legge til visning av city, men fortsatt benytte eventcard.jsx til festival, visste jeg av erfaring at det måtte noen ? tegn med i koden. Jeg prøvde meg frem til dette:

```
<p>{event._embedded?.venues[0]?.city?}</p>
```

Dette fungerte ikke, men jeg mente det skulle være ganske nærme, så jeg henvendte meg til chatgpt med kodelinjen og prompten "what is wrong in this syntax?"

```
<p>{event._embedded?.venues[0]?.city?}</p>
```

what is wrong in this syntax?

The issue with your syntax is in the placement of the optional chaining (?) operator and the way you're trying to render the data in JSX/React.

What's wrong:

jsx

Copy

Edit

```
<p>{event._embedded?.venues[0]?.city?}</p>
```

The trailing question mark (?) after `.city` is incorrect.

Correct syntax:

jsx

Copy

Edit

```
<p>{event._embedded?.venues?.[0]?.city}</p>
```

Explanation:

- Optional chaining (? .) can be used after properties or array indices, but never placed directly after the property name like a trailing question mark.
- When using optional chaining with arrays, ensure you place the question mark (?) before the array index as well: `venues?.[0]`.

Example (correct):

jsx

Copy

Edit

```
// Safely access city without runtime error  
<p>{event._embedded?.venues?.[0]?.city}</p>
```

This corrected syntax ensures safe access to nested data without causing runtime errors if any intermediate property is missing.

Så da viste det seg at jeg hadde tenkt riktig, bortsett fra at [0] skulle stå for seg selv, og et ekstra ? for mye på slutten. Jeg oppdaterte kode linjen.

Men nå støtte jeg på dette problemet - failed to fetch - ved å legge inn visning av city greide den ikke hente data for storbyarrangementer lenger. Jeg prøvde å fjerne og legge tilbake linjen igjen et par ganger, og det var denne endringen som forårsaket problemet.

Jeg henvendte meg til chatgpt igjen

Why would adding this line give a `TypeError: Failed to fetch`

The error you're encountering—`TypeError: Failed to fetch`—is usually unrelated to the syntax of optional chaining in React/JSX itself. Instead, it's typically related to making API calls or attempting network requests in your application.

Adding this line alone:

```
jsx
<p>{event._embedded?.venues?.[0]?.city}</p>
```

would not cause a `Failed to fetch` error. This specific error message usually occurs due to:

1. Network Requests (Fetch API calls)

This error often arises when you're fetching data with `fetch()`:

- **Incorrect URL or Endpoint:**

```
js
fetch('https://incorrect-api.com/data') // URL is wrong or inaccessible
```

- **CORS issues (Cross-Origin Resource Sharing):**

```
plaintext
Access to fetch at 'https://api.example.com' has been blocked by CORS policy
```

- **Server down or unreachable:**
The API you are calling might be down or returning no response at all.
- **Invalid protocol or malformed URL:**

```
js
fetch('http://example.com'); // incorrect protocol
```

2. Connection Issues

- Loss of network connection or unstable internet.
- Firewall or browser settings blocking the fetch request.

Dette hjalp meg ikke frem til årsaken til problemet. Chatgpt virket altså ikke å kunne hjelpe, men etter å ha sett litt mer på dataene som blir returnert fra apiet i konsollen, så jeg at city: oslo er et objekt med id:name, så jeg må hente ut enda ett ledd for å få innholdet til name.

koden ble da:

```
<p>{event._embedded?.venues?.[0]?.city?.name}</p>
```

Denne koden fungerer, så da kunne jeg benytte samme mønster for å legge til den ekstra informasjonen.

```
export default function EventCard ({event}) {  
  return (  
    <>  
      <li>  
        <h3>{event.name}</h3>  
        <img src={event.images[0].url} ></img>  
        <p>{event._embedded?.venues?.[0]?.city?.name}</p>  
        <p>{event._embedded?.venues?.[0]?.country?.name}</p>  
        <p>{event.dates?.start?.dateTime}</p>  
        <Link to={` /event/${event.id}`}>Gå til event</Link>  
      </li>  
    </>  
  );  
}
```

Jeg ville nå legge inn muligheten for at man skulle kunne vise en lenke for å gå til eventet dersom dette var spesifisert. Jeg husket fra SPA emnet med Marius at jeg kunne gjøre dette med en boolean som prop, så jeg la inn en clickable prop. Men husket ikke hvordan jeg benytter conditional i react, så jeg spurte chatgpt.

How can i do a conditional check on the boolean clickable to check if the code should display the link "gå til event"

```
import { Link } from "react-router-dom";

export default function EventCard ({event, clickable}) {
  return (
    <>
      <li>
        <h3>{event.name}</h3>
        <img src={event.images[0].url} ></img>
        <p>{event._embedded?.venues?.[0]?.city?.name}
      </p>
        <p>{event._embedded?.venues?.
[0]?.country?.name}</p>
        <p>{event.dates?.start?.dateTime}</p>
        <Link to={"/event/${event.id}}>Gå til
event</Link>
      </li>
    </>
  );
}
```

You can conditionally render the link using a simple ternary or logical (`&&`) operator in JSX.

Here's the simplest and most common way:

✓ With logical `&&` :

jsx

Copy

Edit

```
{clickable && <Link to={` /event/${event.id}`} >Gå til event</Link>}
```

✚ Full updated component example:

jsx

Copy

Edit

```
import { Link } from "react-router-dom";

export default function EventCard({ event, clickable }) {
  return (
    <>
      <li>
        <h3>{event.name}</h3>
        <img src={event.images[0].url} alt={event.name}></img>
        <p>{event._embedded?.venues?.[0]?.city?.name}</p>
        <p>{event._embedded?.venues?.[0]?.country?.name}</p>
        <p>{event.dates?.start?.dateTime}</p>

        {clickable && (
          <Link to={` /event/${event.id}`} >Gå til event</Link>
        )}
      </li>
    </>
  );
}
```

Jeg la det inn i koden og justerte bruken av denne komponentent så den sendte med `clickable=true` for festivalkortene, men ikke i de andre.

Jeg støtte på en annen ting jeg i eventcard når jeg skulle liste oppe spillesteder / venues med eventcard. Venues har ikke alltid bilder, så jeg måtte ha en mulighet for eventcard til å kun prøve å vise bilder dersom de eksisterer. Jeg benyttet denne kilden til å se hvordan benytte inline conditionals i javascript med `&&` sammen med det jeg lærte fra forrige chatgpt prompt:

[Conditional Rendering – React](#)

slik ble bilderendering i eventcard da

```
{event?.images?.[0]?.url && (  
  <img src={event.images[0].url} />  
)}
```

Knapper med onclick funksjon og conditional rendering

Jeg hadde klart å drille getBigCityEvents-funksjonen ned til Home.jsx, slik at jeg kunne bruke den til å oppdatere hvilken by som hentes (cityChosen).

Jeg har brukt onClick mange ganger før, men denne gangen slet jeg med å huske hvordan man sender med en parameter i onClick-funksjonen. Jeg brukte denne dokumentasjonen for å friske opp hukommelsen på hvordan man kan sende med parametere i onClick:

[React onClick Event Handling \(With Examples\) - Upmostlly](#)

Jeg ønsket å sette opp favorittknapper på kategorisidene (CategoryPage.jsx). Jeg brukte samme onClick-oppsett for knappene som i Home.jsx, men her kunne jeg ikke bare bruke state setter-funksjonen direkte. Årsaken var at jeg kun trengte å legge til én ekstra favoritt i arrayen, ikke overskrive hele state med én ny favoritt.

Her benyttet jeg denne dokumentasjonen for hvordan legge og fjerne et objekt i staten, som jeg benyttet til addToFavorites og removeFromFavorites:

[Updating Arrays in State – React](#)

Jeg hadde en tanke om å benytte conditional rendering for knappene, slik jeg gjorde det for bilder i EventCard.jsx. Jeg forsøkte å etterligne mønstret, og jeg kom meg så langt, men skjønte at det ble noe feil med syntaks, så jeg spurte chatgpt om å hjelpe. Her var koden min før prompten.

```
<h2>Attraksjoner</h2>
```

```
<ul>  
  {categoryAttractions.map(attraction => (  
    <>  
      <EventCard key={attraction.id} event={attraction}  
clickable={false} />  
    </>  
  )}
```

```
        {attraction.in(favorites) (
          <button onClick={() =>
addToFavorites(attraction)}>♥♥</button>
        ) : <button onClick={() =>
removeFromFavorites(attraction)}>♥♥</button>}
      </>
    )}]
  </ul>
```

```

<h2>Attraksjoner</h2>
<ul>
  {categoryAttractions.map(attraction => (
    <>
    <EventCard key={attraction.id} event={attraction} clickable=
    {false} />

    {attraction.in(favorites) (
    <button onClick={() => addToFavorites(attraction)}> ❤️
  </button>
    ): <button onClick={() => removeFromFavorites(attraction)}>
  ❤️ </button>}
    </>
  )}}
</ul>

```

is it not possible to use attraction.in(favorites) like this?

No, the syntax `attraction.in(favorites)` is not valid in JavaScript or TypeScript. The `.in()` method doesn't exist.

If you want to check if `attraction` is in the `favorites` array, you can use the `.includes()` method or `.some()` method.

Here's a corrected version using `.some()`:

```

jsx
<h2>Attraksjoner</h2>
<ul>
  {categoryAttractions.map((attraction) => (
    <li key={attraction.id}>
      <EventCard event={attraction} clickable={false} />

      {favorites.some((fav) => fav.id === attraction.id) ? (
        <button onClick={() => removeFromFavorites(attraction)}> ❤️ </button>
      ) : (
        <button onClick={() => addToFavorites(attraction)}> ❤️ </button>
      )}
    </li>
  )}}
</ul>

```

In this version, `.some()` checks if any item in `favorites` has the same `id` as the current `attraction`. You could also use `.find()` or `.includes()`, but `.some()` is more appropriate for boolean checks.

Would you like me to explain why `.includes()` wouldn't work here without modification?

📄 🗑️ 🗑️ 🗑️ 🗑️ 🗑️ 🗑️

Jeg ønsket å benytte noe tilsvarende en if else funksjon, men chatgpt foreslo å benytte ? istedet, men jeg ønsket å benytte min allerede eksisterende kode - så jeg forsøkte å legge over til å benytte .include istedetfor some i min kode.

Deretter fikset jeg et svart hjerte på ene knappen, og sette addFromFavorites og removeFromFavorites på riktige knapper, så fungerte det som jeg ønsket. Da så koden min slik ut:

```
<h2>Attraksjoner</h2>
<ul>
  {categoryAttractions.map(attraction => (
    <>
      <EventCard key={attraction.id} event={attraction}
clickable={false} />

      {favorites.includes(attraction) ? (
        <button onClick={() =>
removeFromFavorites(attraction)}>♥</button>
        ) : (<button onClick={() =>
addToFavorites(attraction)}>♥</button>)}
    </>
  )}}
</ul>
```

Etter dette var gjort fant jeg ut at denne løsningen med å sammenligne hele attraction objekter ikke ville fungere

Når komponenten laster inn nye objekter, så vil det ikke lenger være noen match med det som finnes i staten. Dette tror jeg blir fordi nye objekter som lastes inn fra API-et er ikke helt identiske med de som allerede ligger i state, selv om de representerer samme data.

Derfor kom jeg frem til at jeg må sjekke for Id istedet for. Jeg spurte chatgpt om hvilken funksjon jeg kunne bruke for å sjekke etter om id matchet istedet for hele objektet.


```

<h2>Attraksjoner</h2>
<ul>
  {categoryAttractions.map(attraction => (
    <>
      <EventCard key={attraction.id} event={attraction} clickable=
      {false} />

      {favorites.includes(attraction) ? (
        <button onClick={() => removeFromFavorites(attraction)}>
♥ </button>
      ) : (<button onClick={() => addToFavorites(attraction)}> ♥
</button>)}
    </>
  )}
</ul>

```

hvordan kan jeg sjekke om id til attraksjon, istedet for hele objektet, matcher med objekter i favoritter?

Du kan endre `includes(attraction)` til å sjekke om `attraction.id` finnes i en liste av `id`-er fra `favorites`. Dette kan gjøres ved å bruke `some()` eller ved å mappe `favorites` til en liste av `id`-er.

Her er en enkel løsning med `some()`:

```

jsx
Copy Edit

<h2>Attraksjoner</h2>
<ul>
  {categoryAttractions.map(attraction => (
    <li key={attraction.id}>
      <EventCard event={attraction} clickable={false} />

      {favorites.some(fav => fav.id === attraction.id) ? (
        <button onClick={() => removeFromFavorites(attraction)}> ♥ </button>
      ) : (
        <button onClick={() => addToFavorites(attraction)}> ♥ </button>
      )}
    </li>
  )}
</ul>

```

Her bruker vi `some()` for å sjekke om noen av objektene i `favorites`-listen har samme `id` som `attraction.id`. Dette er mer effektivt enn `includes()` fordi `includes()` sammenligner hele objekter, ikke spesifikke felt.

📄 🌟 🗨️ 🔊 🔗 ↻ 40

løsningen var altså å benytte `some` istedet for `includes`, da får vi sjekket deler av objektene (for å sammenligne med id) og ikke bare hele objektet som en enhet.

Jeg tok da med meg denne kodelinjen fra chatgpt videre i koden:
`{favorites.some((fav) => fav.id === attraction.id) ?`

Den ferdige koden i prosjektet endte til slutt opp i CategoryCardsContainer.jsx

Filtrering av dato, land og byer

Jeg benyttet dokumentasjonene til [<select> – React](#) når jeg skulle implementere filtrering med dato, land og byer på categoryPage.jsx

Jeg benyttet Ticketmaster Discovery API sitt attraction search, event search, og venue search for å hente relevante data. Attractions search tar ikke imot dato, country eller city som filtre, venue search filtreres bare på country, mens event search kan filtreres på alle 3.

Når jeg skulle implementere filtrering med dato satte jeg meg litt fast. Jeg kom meg så langt som å legge til state for dato, input for date picker, og props på riktig plass, slik at når jeg trykket på filter så sendte jeg kall til apiet på

https://app.ticketmaster.com/discovery/v2/events.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&classificationName=Music&locale=*&countryCode=NO&city=Oslo&startDate=2025-05-22&keyword=

men jeg mistenkte at datoformateringen jeg sendte med fra html date picker - 2025-05-22 - ikke var helt riktig, og jeg synes det var vanskelig å finne ut av hva APIet ønsket av formattering. Jeg forsøkte å teste med ulike datoformat til apiet på [The Ticketmaster Developer Portal](#), men fikk bare Error unknown: error.

Så jeg henvendte meg til chatgpt for riktig formattering.

im currently quering the ticketmaster api with this date format

```
https://app.ticketmaster.com/discovery/v2/events.json?  
apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&classificationName  
=Music&locale=* &countryCode=NO&city=Oslo&startDateTime=202  
5-05-22&keyword=
```

is that correct?



The `startDateTime` parameter in the Ticketmaster API expects a **datetime in ISO 8601 format**, which includes both the date and the time, followed by a 'Z' to indicate UTC.

The correct format should look like this:

```
2025-05-22T00:00:00Z
```

So your URL should be:

```
bash                                                                    Copy Edit  
  
https://app.ticketmaster.com/discovery/v2/events.json?apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&cla
```

If you want to search for events starting from a specific time on that date, you can adjust the time accordingly. For example:

- `2025-05-22T12:00:00Z` would search for events starting from noon UTC on May 22nd, 2025.

Does that clarify it?



Jeg spurte deretter om hvordan jeg kan formatere datoen min før jeg benytter den i api kallet mitt.

Per nå benyttet jeg denne koden til å sette dato for filtrering i APIet direkte fra html date picker:

```
<input  
  type="date"  
  value={selectedDate}  
  onChange={(e) => setSelectedDate(e.target.value)}  
/>
```

```
<input
  type="date"
  value={selectedDate}
  onChange={(e) => setSelectedDate(e.target.value)}
/>
```

how would you convert a date from the html datepicker to iso

the date from the input is 2025-02-02

You can convert the date from the HTML date picker to ISO format using the `Date` object and its `toISOString()` method. Here's how you can do it:

```
jsx
const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  const selectedDate = e.target.value;
  const isoDate = new Date(selectedDate).toISOString();
  console.log(isoDate); // Outputs in ISO format, e.g., "2025-02-02T00:00:00.000Z"
  setSelectedDate(selectedDate);
};
```

If you want to only keep the date part without the time (YYYY-MM-DD), you can use:

```
jsx
const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  const selectedDate = e.target.value;
  const isoDate = new Date(selectedDate).toISOString().split('T')[0];
  console
```

Den foreslo å benytte `toISOString()` funksjon for å få ISO format, og jeg forsøkte å benytte dette, men dette fungerte ikke, da ticketmaster forventet et format på ISO formatteringen uten millisekunder.

```
const getCategoryEvents = async (category, country, city, searchText, date) => {
  console.log(date)
  let preformattedDate = new Date(date)
  let postformattedDate = preformattedDate.toISOString()
  console.log(postformattedDate)

  fetch(`https://app.ticketmaster.com/discovery/v2/events.json?apikey=m50DSZRZed6yFz7Tp4RTQ34xNFxfGny3&classificationName=${category}&locale=*&countryCode=${c
```

```

country}&city=${city}&startDateTime=${postformattedDate}&keyword=${searchText}`
)

    .then(res => res.json())
    .then(data => {
        console.log(data)
        const events = data._embedded?.events || [];
        setCategoryEvents(events);
    });
};

```

jeg så at eneste forskjellen på dato fra html date picker og ISO formattering ønsket av ticketmaster var millisekundene på slutte. Så da tenkte jeg at det blir lettere å bare legge til nødvendig formattering manuelt.

derfor kom jeg selv frem til denne løsningen på egen hånd. Det er ikke alltid chatGPT kommer med de enkleste løsningene kan man se...

```

const getCategoryEvents = async (category, country, city, searchText, date) =>
{
    let postformattedDate = "";

    if (date) {
        postformattedDate = date.concat("T00:00:00Z");
    }

    fetch(`https://app.ticketmaster.com/discovery/v2/events.json?apikey=m50DSZRZed6yFz7Tp4RTQ34xNFxfGny3&classificationName=${category}&locale=*&countryCode=${country}&city=${city}&startDateTime=${postformattedDate}&keyword=${searchText}`)
    .then(res => res.json())
    .then(data => {
        console.log(data)
        const events = data._embedded?.events || [];
        setCategoryEvents(events);
    });
};

```

karakter B krav

Schemaer til sanity

Jeg benyttet dokumentasjonen til sanity for å tilpasse skjemaene med referanser.

[Reference | Sanity Docs](#)

og jeg benyttet også sanity Docs til hvordan sette opp client og service i prosjektet mitt for Sanity.

I oppgaven står det ikke spesifisert at users skal ha bilde, men så står det senere at hver bruker skal ha profilbilde. Jeg valgte å tolke det slik at sanity skulle hoste profilbildet, og tilpasset skjemaene til å reflektere dette.

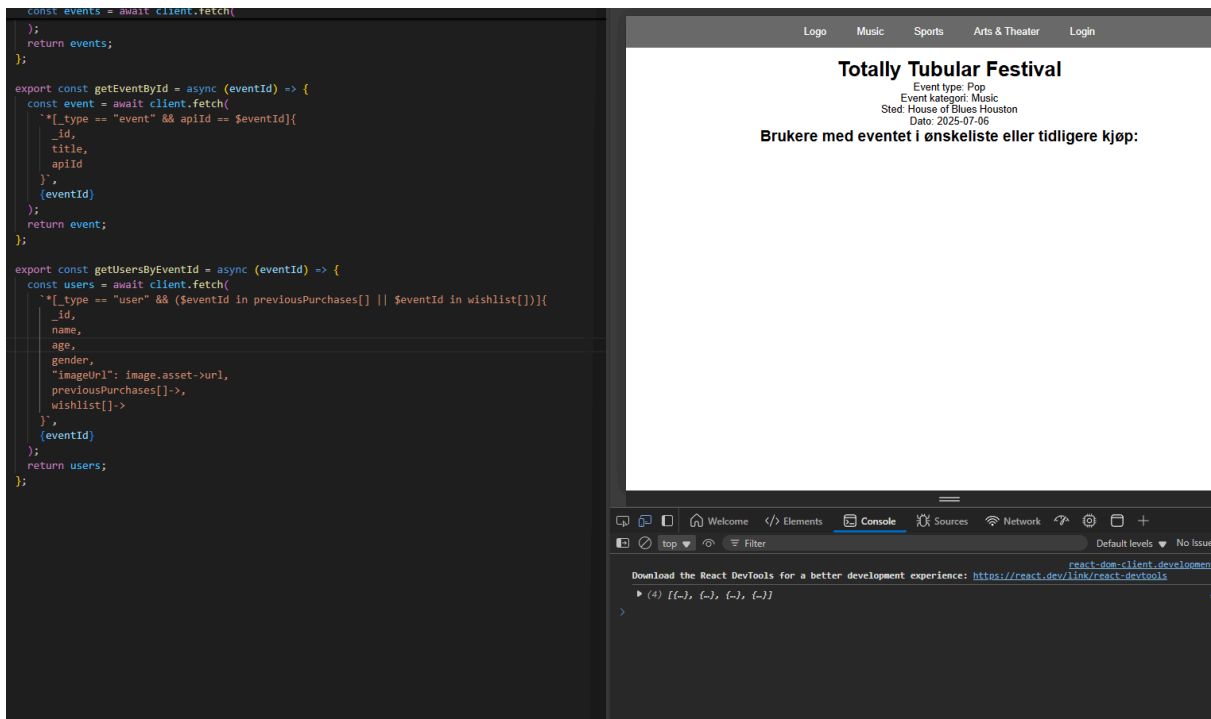
jeg fant i dokumentasjonen hvordan

[Presenting Images - Sanity Image Pipeline | Sanity Docs](#)

Alt annet i karakterkrav b gikk fint, bortsett fra når jeg på SanityEventDetails.jsx siden skulle forsøke å liste opp brukere med det spesifikke eventet i enten ønskelisten eller tidligere kjøp. Jeg benyttet dokumentasjonen og kom meg litt hvor jeg mente det skulle fungere, men det gjorde det altså ikke det likevel. Koden jeg kom frem til med sanity dokumentasjonen så slik ut:

```
export const getUsersByEventId = async (eventId) => {  
  const users = await client.fetch(  
    `*[_type == "user" && ($eventId in previousPurchases[] || $eventId in  
previousPurchases[])]{  
      _id,  
      name,  
      age,  
      gender,  
      "imageUrl": image.asset->url,  
      previousPurchases[]->,  
      wishlist[]->  
    }`,  
    {eventId}  
  );  
  return users;  
};
```

jeg fikk ingen feilmeldinger i konsollen, men data ble ikke returnert fra sanity.



Så jeg sendte med relevant kode fra service queries og SanityEventDetails til chatgpt, for å se om den visste løsningen.

```
import client from './client';
import imageUrlBuilder from '@sanity/image-url'
```

```
export const getAllUsers = async () => {
  const users = await client.fetch(
    *[_type == "user"]{
      _id,
      name,
      age,
      gender,
      "imageUrl": image.asset->url,
      previousPurchases[]->,
      wishlist[]->
    }
  );
  return users;
};
```

```
export const getAllEvents = async () => {
  const events = await client.fetch(
    *[_type == "event"]{
      _id,
      title,
      apild
    }
  );
  return events;
};
```

```
export const getEventById = async (eventId) => {
  const event = await client.fetch(
    *[_type == "event" && apild == $eventId]{
      _id,
      title,
      apild
    },
    {eventId}
  );
  return event;
};
```



```

import { useParams } from "react-router-dom"
import { useState, useEffect } from 'react';
import { getEventById } from "../sanity/service";
import { getUsersByEventId } from "../sanity/service";

export default function SanityEventDetails() {
  const [id] = useParams()
  const [eventDetails, setEventDetails] = useState()
  const [sanityEvent, setSanityEvent] = useState()
  const [users, setUsers] = useState()

  const getEventById = async () => {
    fetch(https://app.ticketmaster.com/discovery/v2/events/${id}?
apikey=m5ODSZRZed6yFz7Tp4RTQ34xNFxfGny3&locale=*)
      .then(res => res.json())
      .then(data => {
        setEventDetails(data);
      })
  };

  const getAndSetSanityEvent = async () => {
    const data = await getEventById(id);
    setSanityEvent(data[0]);
  };

  const getAndSetUsers = async () => {
    const data = await getUsersByEventId(id);
    setUsers(data);
  };

  useEffect(() => {
    getEventById()
    getAndSetSanityEvent()
    getAndSetUsers()
  }, [id])

```

```

return (
  <>
  <h1>{sanityEvent?.title}</h1>
  <p>Event type: {eventDetails?.classifications?.[0]?.genre?.name}
</p>
  <p>Event kategori: {eventDetails?.classifications?.
[0]?.segment?.name}</p>
  <p>Sted: {eventDetails?._embedded.venues?.[0]?.name}</p>
  <p>Dato: {eventDetails?.dates?.start?.localDate}</p>
  </>
);
};

```

id gotten from useparams is Apild'

im trying to make a query to get users that have the parameter id in their wishlist or previouspurchases

```

export const getUsersByEventId = async (eventId) => {
  const users = await client.fetch(
    *[_type == "user" && ($eventId in previousPurchases[]
|| $eventId in wishlist[])]{
      _id,
      name,
      age,
      gender,
      "imageUrl": image.asset->url,
      previousPurchases[]->,
      wishlist[]->
    },
    {eventId}
  );
  return users;
};

```



Your query logic is close, but the way you're checking for the `eventId` in `previousPurchases` and `wishlist` needs some adjustment.

Since `previousPurchases` and `wishlist` are arrays of references, you need to access the referenced documents' `apiId` field to properly match against the `eventId`.

Here's the corrected query:

✅ Updated `getUsersByEventId` function:

```
javascript Copy Edit

export const getUsersByEventId = async (eventId) => {
  const users = await client.fetch(
    `[_type == "user" && (
      $eventId in previousPurchases[]->apiId || $eventId in wishlist[]->apiId
    )]{
      _id,
      name,
      age,
      gender,
      "imageUrl": image.asset->url,
      previousPurchases[]->{
        _id,
        title,
        apiId
      },
      wishlist[]->{
        _id,
        title,
        apiId
      }
    }`,
    { eventId }
  );
  return users;
};
```

✅ Why This Update?

- **Array Access:** In Sanity, when working with references, the `->` operator is necessary to access the referenced document's fields.

Jeg forsøkte å gjøre delvise endringer basert på svare til chatgpt, og fant at koden min fungerte dersom jeg la til `->apiId` bak hver liste i filtreringen. Så da gikk koden fra:

```
`[_type == "user" && ($eventId in previousPurchases[] || $eventId in
wishlist[]->)]{
```

til:

```
`[_type == "user" && ($eventId in previousPurchases[]->apiId ||
$eventId in wishlist[]->apiId)]{
```

Så det viste seg altså at jeg bare manglet ->apild for at det skulle fungere. Så da valgte jeg å fikse akkurat det, men droppet de andre forslagene til syntaksendringer, da koden min allerede fulgte dokumentasjonen til Sanity.