Hooks + hämta och visa data

Jens Palmqvist

Hooks i React

- Hooks är funktioner man kan använda för utökad funktionalitet i React
- Exempel på några hooks:
  - useState
  - useEffect
  - useMemo
  - useContext

Hämta och visa data

- Vi kommer nu titta närmare på *useState* och *useEffect* som vi behöver för att kunna hämta och visa data.
- För att hämta data använder vi funktionen *fetch* som finns inbyggd i webbläsaren

```
const response = await fetch("https://www.api.com/data");
const data = await response.json();
console.log(data);
```

useState

- När vi väl har hämtat datan vill vi visa den i vår komponent
- Med hjälp av *useState* kan vi skapa en variabel som vi kan använda för att visa datan

```
const [count, setCount] = useState(0);
```

- **count** är variabeln som vi kan använda för att visa datan
- **setCount** är funktionen vi använder för att uppdatera datan
- **0** är värdet som count får från början

useEffect

- Kod i en komponent körs varje gång komponenten renderas vilket sker när vi uppdaterar statet
- För att förhindra en oändlig loop av api-anrop måste vi lägga fetch-koden i useEffect

```
useEffect(() => {
  const fetchData = async () => {
    const response = await fetch("https://www.api.com/data");
    const data = await response.json();
    setCount(data.count);
  };
  fetchData();
}, []);
```

useEffect - Dependency array

```
useEffect(() => {
  //kod som körs när komponenten renderas
, []);
```

- useEffect tar emot en funktion med kod
- Den har även en *dependency array* som tar emot variabler
- Koden i funktionen körs varje gång en variabel i *dependency array* uppdateras
- Om arrayen är tom körs koden endast när komponenten renderas första gången

Komplett exempel ladda och visa data

```
import React, { useState, useEffect } from "react";

const App = () => {
  const [data, setData] = useState(0);

  useEffect(() => {
    const fetchData = async () => {
      const response = await fetch("https://www.api.com/data");
      const data = await response.json();
      setData(data);
    };
    fetchData();
  }, []);

  return (
    <div>
      <p>{data}</p>
    </div>
  );
};
```

Övning: Kattfakta

- Genom att anropa 'https://catfact.ninja/fact' får vi tillbaka en slumpmässig kattfakta i json-format
- Skriv en react-komponent som hämtar en kattfakta och visar den på sidan

Tänk på att: beroende på om api:et svarar med ett object {} eller en array [] så måste du hantera datan olika i din komponent

```
const [data, setData] = useState({});
```

```
const [data, setData] = useState([]);
```

Facit: Kattfakta

```
import React, { useState, useEffect } from "react";

const App = () => {
  const [data, setData] = useState({});

  useEffect(() => {
    const fetchData = async () => {
      const response = await fetch("https://catfact.ninja/fact");
      const data = await response.json();
      setData(data);
    };
    fetchData();
  }, []);

  return (
    <div>
      <p>{data.fact || ""}</p>
    </div>
  );
};
```