

## Context

1. Skapa ett nytt react-projekt med typescript
2. Öppna projektet i Visual Studio Code
3. För att kunna göra en övning med context behöver vi göra några komponenter som ligger inuti varandra. Detta betyder att komponenterna kommer att vara barn till den komponent som den ligger inuti. Det spelar inte så stor roll vad ni kallar komponenterna men det är viktigt att ni håller reda på vilka komponenter som är barn till vilken komponent. Skapa nu tre stycken komponenter där komponent 1 ligger i app. Komponent 2 är barn till komponent 1. Komponent 3 är barn till komponent 2.
4. I varje komponent, skriv någon enkel text så att ni kan säkerställa att komponenten visas på skärmen när allt är klart från övning 3 ovan.

5. Poängen med att använda sig av context är att göra viss information tillgänglig för alla komponenter i en applikation eller för alla komponenter inom en viss gren i applikationen. Att använda context bör göras väldigt sparsamt då det försvårar för utvecklare att återanvända komponenter då det är knutna till ett context.

Teorin bakom context är att göra ett objekt tillgängligt högt uppe i en struktur med komponenter och då kan alla komponenter som är barn till den komponenten med context använda sig av informationen i contextet.

Skapa nu ett kontext i app-komponenten som innehåller en text som värde.

Detta görs genom att skapa en ny variabel som kan heta MyContext (förmodligen vill du byta namn på denna variabel så att den speglar vad den faktiskt har för innehåll). Denna variabel kommer att vara resultatet av funktionen createContext(). createContext i sin tur tar in en parameter som skall agera som startvärde för contextet. Lite som med useState().

Du bör lägga denna variabel ovanför komponenten App i app.tsx.

6. Nu behöver vi använda variabeln myContext från förra övningen. Denna variabel skall användas för att skapa en wrapper runt allt innehåll i app-komponenten. Detta gör att informationen i contextet också finns tillgängligt för samtliga barn i app-komponenten.

Detta görs genom att använda följande syntax:

```
<MyContext.Provider value="En godtycklig text här">
  ...
</MyContext.Provider>
```

7. Nu finns det ett objekt MyContext som är spritt till de komponenter som ligger inuti app-komponenten. Glöm nu inte att exportera er variabel MyContext. Detta för att vi skall kunna importera den i någon av våra andra komponenter.
8. Gå nu till den komponent som är längst ner i trädet som du skapade i Övning 3 ovan. Nu skall vi använda oss av informationen i contextet som skapades i App-komponenten. För att göra detta behöver vi importera MyContext.

Efter importen behöver vi använda oss av hooken `useContext()`. Denna hook ger oss det värde som skrevs in i providern tidigare. Lagra resultatet av hooken i en `const`, då vi inte vill kunna ändra detta värde.

```
const contextValue = useContext(MyContext);
```

Nu kan du använda variabeln `contextValue` var som helst i din komponent.

9. Kontrollera att allting ser ut att fungera så här långt genom att skriva ut värdet av `context` någonstans i din kontroller.
10. För att göra exemplet lite roligare skall vi nu kombinera våra kunskaper om state och context för att hämta filmerna som vi tidigare har arbetat med.

I app-komponenten behöver vi nu ändra så att startvärdet i vårt `createContext` blir någonting annat än en string. I vårt fall behöver vi sätta startvärdet till en lista av filmobjekt, `IMovie[]`. På samma sätt som vi tidigare gjorde för att sätta startvärde i state.

Skapa ett objekt som är av typen `IMovie[]` och har värdet av en tom array. Använd detta objekt som startvärde för ert context.

11. Skapa upp ett nytt stateobjekt som skall innehålla filmerna (kalla det för `movies`), använd samma startvärde som för contextet i övning 10 ovan.
12. Använd dina tidigare kunskaper inom hooks för att skriva en `fetch` som uppdaterar ditt state genom `setMovies`-funktionen.
13. Uppdatera nu `MyContext.Provider`s value så att det är komponentens stateobjekt `movies`.
14. Gå till din sista komponent i kedjan och uppdatera så att den nu använder sig av det nya contextobjektet för att skriva ut informationen från filmerna.