

Teorihandbok - React

Emelie Ek
UXF22

1. Ramverket React (en översikt, vad som är unikt för ramverket, fördelar, ev nackdelar, osv)

React är ett open source bibliotek, som är byggt på javascript. Det innebär att vem som helst kan komma åt koden och man kan återanvända, ändra och publicera sina egna lösningar utan några problem. Det används oftast för att bygga ett användargränssnitt och gärna ensidiga applikationer/webbplatser. Vilket innebär att vissa delar i applikationen/webbplatser ändras utan att sidan laddas om.

Med hjälp av React kan du effektivisera och återanvända din kod med hjälp av att man skapar komponenter vilket också ger en tydlig struktur på arbetet.

Komponenterna man skapar kan innehålla både utseendet (gränssnittet) och önskad funktionalitet. Med hjälp av komponenter kan man förenkla processen om någonting behöver ändras. Du hittar lätt i vilken komponent och var du vill göra din ändring eftersom du har en tydlig överblick.

En stor nackdel med React är att sökmotorer har svårt att indexera och läsa av innehållet, men det har kommit flera ramverk som bygger på React så man kan rendera React på servern, exempelvis Next.js. En annan nackdel är att React kan påverka prestandan för klienten/browsern negativt om det inte är bra optimerat, det kan bli väldigt mycket kod som klienten ska rendera.

2. Vad innebär Rendering och Virtual DOM?

När användaren går in på en webbplats så läses all HTML, CSS och Javascript in. För att webbplatsen ska få ett användargränssnitt behöver koden omvandlas till det som användaren ser på skärmen, det kallas rendering.

Virtual DOM är när DOMen uppdateras i realtid med hjälp av javascript, eller i det här fallet React. Javascripten uppdaterar HTML strukturen vid behov och det behövs inte en sidhämtning från servern utan allting sker på klient-sidan.

3. Vad är JSX? Vad används det till?

JSX är Reacts struktur för HTML och rekommenderas att användas till React. Men man kan använda det till andra ramverk också. I koden ser det ut som att man skriver normal HTML struktur men React kompilerar ut det i native Javascript. Man använder det för komponenternas struktur, alltså själva sidans struktur. Man kan skriva vanliga HTML taggar, men man kan också importera andra JSX komponenter som man har skrivit.

4. Vad är ett undantag inom programmering? Eng. Exception

Man kan säga att ett undantag inom programmering är när något man tänkt sig ska ske på sin webbplats inte sker och det blir ett avbrott så att koden inte kan fortsätta köras. Vilket då resulterar i att ett error uppstår kollar man i konsolen i inspektorn så kan man se vad som gått fel och kan då förhoppningsvis enkelt lösa problemet, oftast så syns inget användargränssnitt i skärmen. Då får man antingen upp ett felmeddelande eller så är hela sidan helt vit.

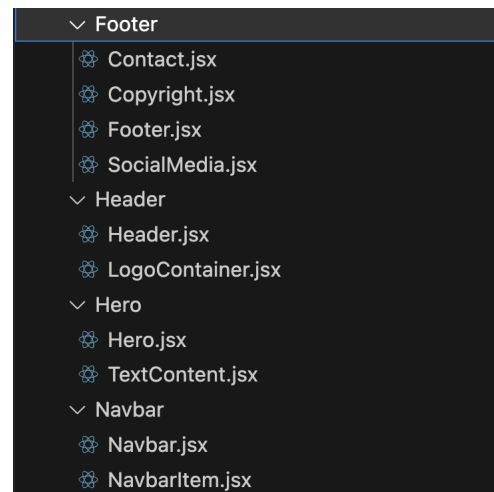
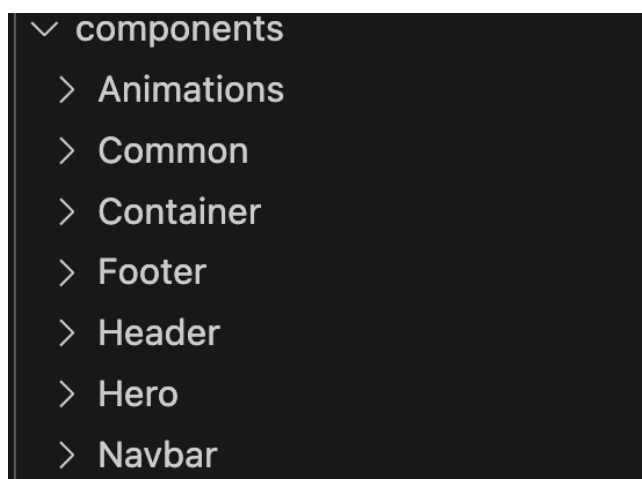
5. Vad innebär autentisering inom webbapplikationer? Vad används det till?

Autentisering inom webbapplikationer är att en användare ska kunna identifiera sig för att vara den hen är genom till exempel genom ett inloggningsformulär innan den får tillgång till en specifik webbapplikation. På så sätt kan vi som står bakom webbapplikationen verifiera att det är rätt användare som vill ha åtkomst till webbapplikationen. Man vill inte att obehöriga personer ska komma åt känslig information om webbapplikationen, eller om man till exempel använder ett inloggningsformulär att man vet att rätt person kommer åt sitt konto.

Sammanfattning av projekt

I min portfölj har jag använt mig av många olika komponenter som jag har döpt efter vart de ligger någonstans för att lätt ska kunna hitta mina komponenter och de delar jag skulle vilja ändra i. Mina komponenter är uppdelade i olika mappar som vi ser på bilden. Då vet jag vilken del av min sida jag vill ändra eller lägga till någonting på så kan jag enkelt hitta den delen och sedan inuti den mappen har jag flera olika JSX filer som då är uppdelade i olika "block" i huvudkomponenten. I mina JSX filer så har jag skrivit all styling och all funktionalitet om ett specifikt element ska ha någon typ av funktionalitet.

Som vi ser på bilden här bredvid så har jag döpt alla olika mina olika delar i en footer/header/hero/navbar till logiska namn så att jag på ett enkelt sätt kan gå in och justera de delar jag vill ändra på.



I mappen Footer har jag en JSX-fil som heter Social media och i den importerar jag data från en JSON fil som jag har skrivit själv. Med hjälp av KEYS kan jag plocka ut det innehåll från JSON-filen som jag vill plocka ut och skicka ut till användaren.

Givetvis också för att inte behöver återupprepa min kod i mina komponenter.

Eftersom det är flera olika väljer jag att använda mig av en map funktion som gör att listan med objekt i JSON filen körs igenom och ökar med en för varje gång. Så att varje ikon/text/knapp får olika innehåll tack vare den datan jag har importerat.

När jag har skapat en komponent så vill man returnera ett värde som antingen kan vara data från JSON fil, HTML-element eller tidigare skapade komponenter som man vill återanvända.

```

const SocialMedia = () => {
  return (
    <SocialMediaInner>
      {icons.map(({ id, href, name, className }, i) =>
        <StyleSocialMedia key={id} href={href} className={className}> {name}
          { (i + 1) !== icons.length && (
            <DecoLine className="line" />
          )}
        </StyleSocialMedia>
      )}
    </SocialMediaInner>
  );
}

export default SocialMedia;

```

När jag bygger mina webbapplikationer med React använder jag alltid mig utav localhost och testar design och funktionaliteten i inspektorn i webbläsare, även kallat Virtual DOM. Jag har inte gjort några ändringar i min kodfil förrän jag ser att det blir som jag faktiskt vill. När jag sedan är säker på att det fungerar som jag vill lägger jag in koden i min kod-fil och sparar det. Då renderas min fil om på nytt och hamnar i den vanliga DOMen och uppdaterar den virtuella DOMen. Vilket underlättar utvecklingen av webbapplikationer något enormt då allting sker så snabbt och smidigt.