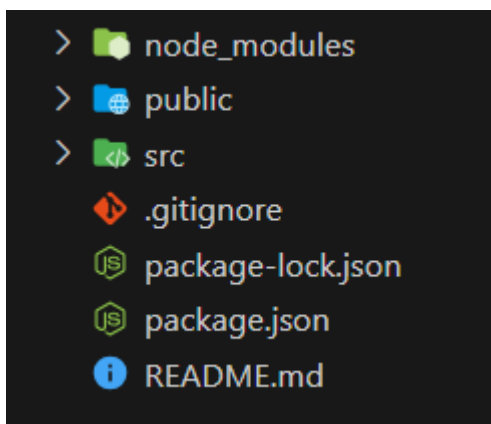


Här är en guide som förhoppningsvis kommer att vara till hjälp när det gäller att få upp React-koden på Webshare.

- Metod 1 kan man följa om man redan har börjat arbeta i en Node.js miljö, medan Metod 2 är för de som planerar att ha all sin React-kod (JSX) i en HTML-fil.
- Om man vill arbeta och få en verklig känsla för React rekommenderar jag att arbeta i en Node.js miljö och följa Metod 1. Detta är för att det blir smidigare med att dela upp sin kod i olika komponenter som skrivs i olika filer. I Metod 2 skrivs det mesta i en enda fil.
- Jag vill också påpeka att det finns två sätt att skapa sina React-komponenter, antingen genom klass komponenter eller funktionella komponenter. Det senare sättet, alltså Functional Components (på engelska), är det nyare sättet som jag rekommenderar att ni lär er.

Metod 1:

Det första sättet kan man följa om man har valt att jobba i en Node.js miljö och har filer som kan vara strukturerade på följande sätt:



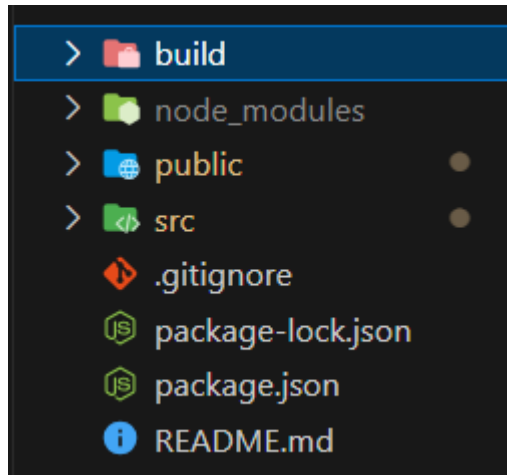
Som ni förmodligen redan vet är ovan den strukturen man brukar få efter när man har skapat en ny React-app genom att slå in "*npx create-react-app my-react-app*" i terminalen/konsolen. I "src" mappen kan man skapa React komponenter och i "public" kan man hitta bland annat en HTML-fil (index.html).

Om ni känner er osäkra på hur man kommer igång med React i en Node.js-miljö, kan ni ta en titt på följande länk:

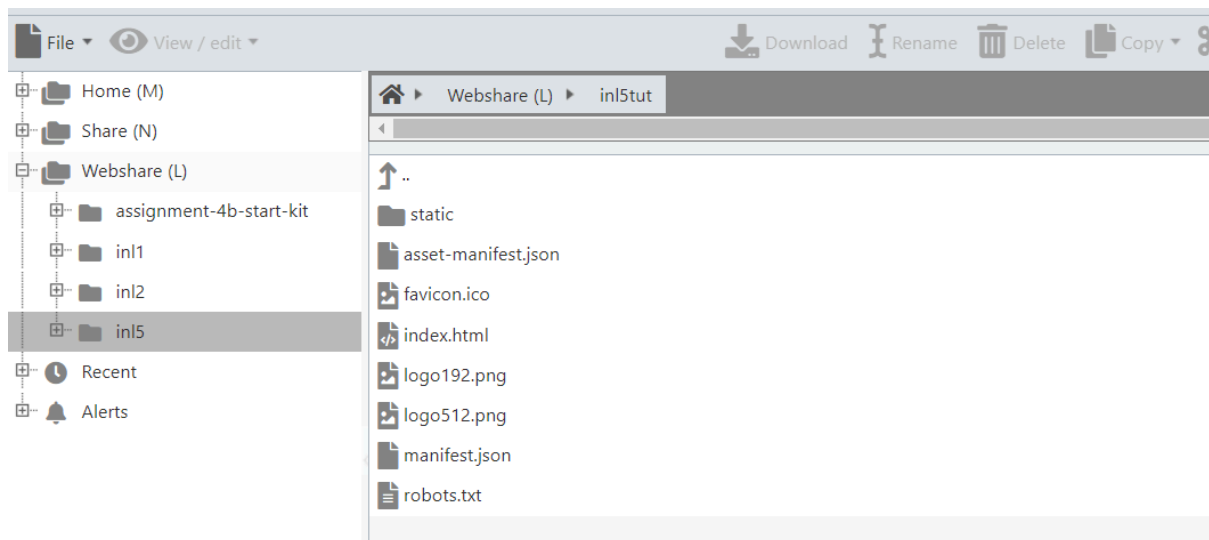
https://coenraets.org/react-js-tutorial-example/#Part_1_Installing_Reactjs_Setting_up_a_React_Project

Efter att ni har gjort klart uppgiften och känner er redo att publicera projektet på Webshare, är det dags att bygga er React-app. För att göra det kan ni slå in följande kommando i terminalen: *"npm run build"*

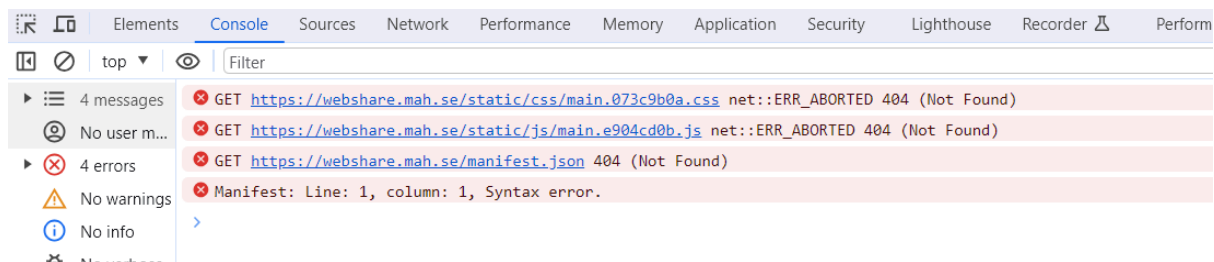
När ni har gjort det bör ni få en ny mapp som heter "build", vilken innehåller en produktionsklar version av er applikation:



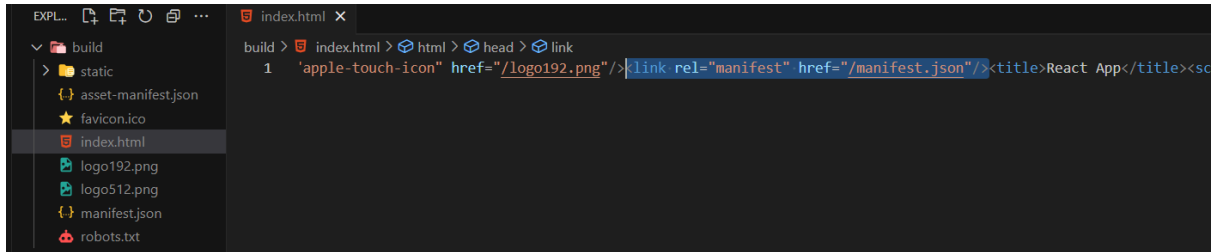
Det jag själv brukar göra nu är att dra in alla filer från "build"-mappen och släppa dem i en annan mapp (till exempel "inl5") på Webshare.



Det kan hända att ni stöter på följande errors när ni går in på uppgiftens Webshare länk:

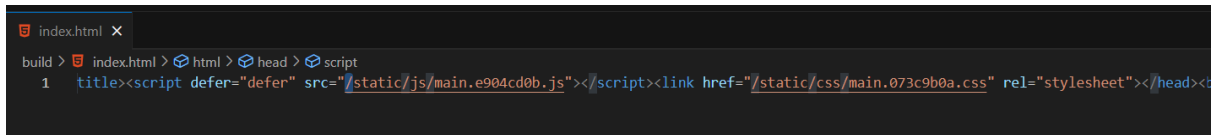


Ett sätt för att lösa det är att gå tillbaka till kodeditor och sedan in i "index.html" filen i "build" mappen. Det första som man vill göra i denna fil är att ta bort länken till "manifest" (det markerade).



Sedan vill man också ta bort det första "/" tecknet framför "static" i både "js" och "css". Se bilderna nedan.

Före:



Efter:



Nu vill man spara ändringarna och sedan gå tillbaka till sin Webshare. Ersätt den befintliga "index.html"-filen med den nya. Efter det kan man besöka uppgiftens Webshare-länk och förhoppningsvis se att alla fel har försvunnit.

OBS! Dessa fel är vad jag själv får när jag laddar upp en React-app på Webshare. Det kan dock hända att ni stöter på andra typer av fel på grund av olika React-/Node-versioner. Se därför till att först själva söka efter lösningar på problemen innan ni ber om hjälp, om det skulle behövas.

Metod 2:

Jag lärde mig nyligen denna metod från studenter som använt den i sina inlämningar. Eftersom inlämningsuppgift 5 egentligen inte är särskilt stor, går det lika bra att följa den metoden också.

Tanken är att skapa en HTML-fil och inkludera tre externa skript som möjliggör att skriva React-kod. Dessa är:

```
<script src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
<script src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"></script>
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

Så här skulle alltså startstrukturen för din HTML-fil kunna se ut:

```
1  <!doctype html>
2  <html>
3    <head>
4      <title>Inl5</title>
5      <script src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
6      <script src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"></script>
7      <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
8      <link rel="stylesheet" href="style.css" />
9      <meta charset="utf-8">
10   </head>
11   <body>
12     <div id="root"></div>
13
14     <script type="text/babel">
15       const App = () => {
16         return (
17           <div id="main">
18             <h1>Hello World!</h1>
19           </div>
20         );
21       };
22
23       ReactDOM.render(<App />, document.getElementById('root'));
24     </script>
25   </body>
26 </html>
```

Under script-taggen på rad 14 kan man skriva sin React-kod. Huvudkomponenten i React brukar kallas "App". Inuti denna komponent kan man lägga till innehåll, till exempel andra komponenter och HTML-taggar, som bygger upp hela sidan. Dessa vill man då skriva inom sin div med id:t "main" (rad 17).