

React Routing

Rolando Gonzalez 2022

Innhold

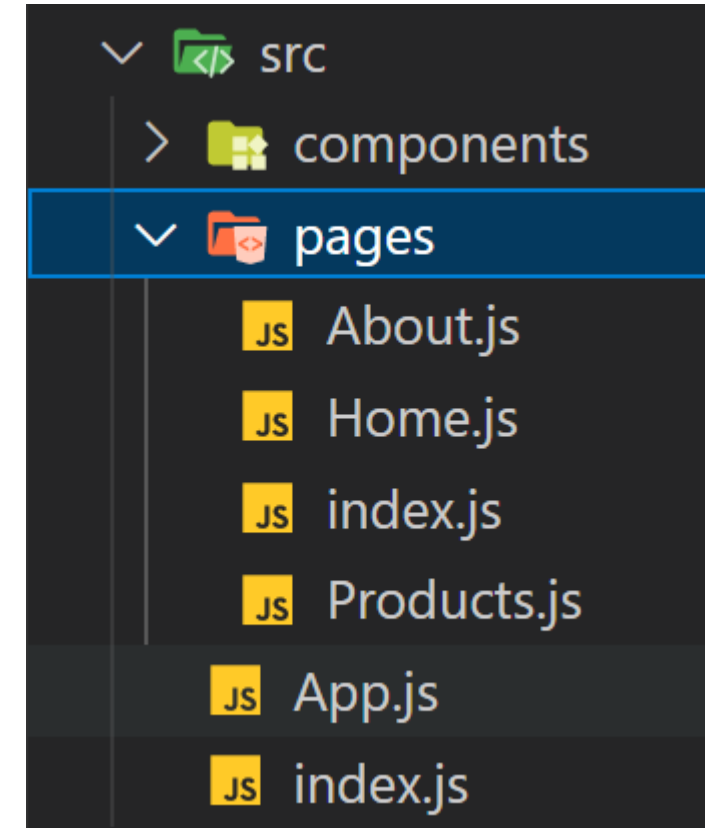
- Hva er React Routing?
- Oppsett av page-mappe med page-komponenter
- Hvordan bruke React Routing?

Hva er React Routing?

- React Routing er et navigeringssystem i React.
- Med React Routing «navigerer» man ikke mellom nettsider, men mellom komponenter. Mer spesifikt bare bytter man ut komponenter istedenfor å måtte bytte ut hele nettsiden.
- Det overnevnte resulterer i at siden aldri reloads, men bare bytter ut innhold, noe som resulterer i raskere og mer sømløshet.

Hvordan bruke React Routing?: Oppsett

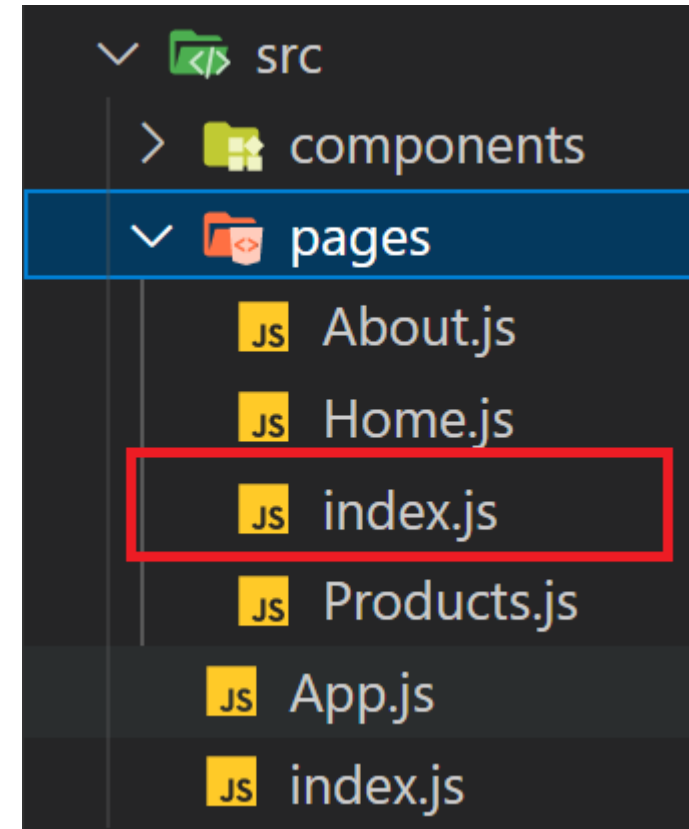
- Det første man kan gjøre er å tenke seg hvilke «nettsider» man trenger i nettstedet.
- Man oppretter en mappe som heter pages (evt. views) som inneholder komponenter som representerer hovedinnhold.
- Hvis disse sidene er veldig enkle kan det være at de inneholder all informasjonen de trenger, men page-komponenter er ideelt sett bare hub-komponenter for komponenter fra components-mappen.



Kommentar til forrige slide – index.js

- I forrige slide kunne man se at det i pages-mappen er en fil som heter index.js.
- Dette er en fil for å gjøre det enklere/ryddigere å importere alle komponentene fra pages-mappen.

```
Js index.js  X  
  
src > pages > Js index.js  
1  import About from "./About";  
2  import Home from "./Home";  
3  import Products from "./Products";  
4  
5  export { About, Home, Products };
```



BrowserRouter – del 1

- BrowserRouter er hovedkomponenten i Routing-systemet.
- Man importerer BrowserRouter, Routes og Route fra react-router-dom, og også page-komponentene.
- Man har BrowserRouter rundt Routes som er rundt Routes.
- Route-komponentene angir urlene som skal aktivere de forskjellige page-komponentene

```
JS App.js  X
src > JS App.js > ...
1  import { BrowserRouter, Routes, Route } from 'react-router-dom';
2  import { Home, About, Products } from './pages/';
3
4  function App() {
5    return (
6      <div>
7        <BrowserRouter>
8          <Routes>
9            <Route path="/" element={<Home/>}></Route>
10           <Route path="about" element={<About/>}></Route>
11           <Route path="products" element={<Products/>}></Route>
12         </Routes>
13       </BrowserRouter>
14     </div>
15   );
16 }
17
18 export default App;
```

BrowserRouter – del 2 – <Link>

- Neste viktige moment er hvordan bruker skal kunne navigere seg mellom «sidene» (les: page-komponentene).
- Vi inkluderer importering av Link
- Link-komponenten resulterer i lenker (men ikke vanlige HTML-lenker) som brukeren kan klikke på
- Hver Link referer til url angitt i Route path.

```
import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
```

```
<BrowserRouter>  
  <nav>  
    <ul>  
      <li><Link to="/">Hjem</Link></li>  
      <li><Link to="/about">Om oss</Link></li>  
      <li><Link to="/products">Produkter</Link></li>  
    </ul>  
  </nav>  
  <Routes>  
    <Route path="/" element={<Home/>}></Route>  
    <Route path="about" element={<About/>}></Route>  
    <Route path="products" element={<Products/>}></Route>  
  </Routes>  
</BrowserRouter>
```

Merknad til forrige slider

- Merk at de forrige slidene viser all Route-koden i App.js, men det vanlige vil eksempelvis at man har en egen komponent for navigasjonen.
- Man kan også vurdere om hele BrowserRouter bør være i en egen komponent som bare hentes inn i App.js

Side ikke funnet: *

- For å håndtere url'er som evt. ikke eksisterer i routing-systemet kan man bruke path satt til asterix.
- Den kan referere til en komponent som viser informasjon om at siden ikke finnes. Gjerne også en lenke til hjem.

```
<Routes>  
  <Route path="/" element={<HomePage/>}></Route>  
  <Route path='about' element={<AboutPage/>}></Route>  
  <Route path="*" element={<NotFoundPage/>}></Route>  
</Routes>
```

References

- ReactTraining (2019): react-router-dom, <https://reacttraining.com/react-router/web/example/basic>, [Last accessed 03.10.19]