

Lloc web multi-pàgina

Una aplicació React consisteix en un sol fitxer HTML **index.html**. Les vistes es codifiquen com a components en format **JSX**.

Però de vegades hem de crear llocs web de diverses pàgines perquè un lloc web d'una sola pàgina no sempre es pot presentar tota la informació completa.

Nota: - M'agrada afegir un punt a aquest escrit que, de fet, podem integrar React en una aplicació web de diverses pàgines desenvolupant una configuració de Webpack que inclogui components React en pàgines estàtiques generades (és a dir, fitxers HTML).

Ara ens ocupem de les rutes.

Aquí crearem un lloc web amb diverses rutes en lloc de diverses pàgines. Com que React no està dissenyat per desenvolupar llocs web de diverses pàgines. Per tant, hem de crear diverses rutes per gestionar diverses visualitzacions.

A continuació es descriuen les rutes de l'aplicació i la seva visualització corresponent.

| Ruta | Pàgina corresponent |
|----------|---------------------|
| / | Pàgina Home |
| /about | Pàgina About |
| /contact | Pàgina contacte |

Crear una aplicació React

En primer lloc, hem de crear una aplicació de reacció mitjançant l'eina NPX . No us preocupeu per NPX, ja que és una eina que ve amb NPM (Node Package Manager) 5.2+ en endavant.

Per tant, després d'una instal·lació reeixida de Node.js , creeu una aplicació React mitjançant NPX.

```
npx create-react-app react-multi-page-website
```

Aquesta ordre crearà una aplicació react amb el nom del projecte react-multi-page-website .

Ara entreu el directori del projecte i inicieu l'aplicació.

```
cd react-multi-page-website
```

```
npm start o yarn start
```

S'obrirà l'aplicació React que hem creat a la finestra del nostre navegador amb l'adreça `https://localhost:3000`. El port pot variar si hi el 3000 està ocupat.

Instal·leu React-Router-Dom

Per tant, estem construint una aplicació React amb diverses rutes i requereix un paquet `react-router-dom`. Instal·leu-lo amb NPM.

Nota: - Al pas anterior, hem iniciat l'aplicació en una pestanya de terminal. Per tant, obriu una altra pestanya / finestra del terminal per executar l'ordre següent.

```
npm i react-router-dom
```

Ara podem utilitzar el nostre editor de codi favorit per editar el nostre projecte. Personalment recomano Visual Studio Code.

Afegiu Bootstrap CDN

Avui aprendrem a afegir bootstrap a la nostra aplicació React com ho feiem en html i farem servir CDN Bootstrap, o sigui ja optimitzat.

Per tant, obriu el fitxer **index.html** al directori públic i afegiu el CDN d'arrencada a la secció.

```
<link rel = "stylesheet" href =  
"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"  
/>
```

De manera que el fitxer `index.html` complet s'assembla al següent.

```
<!DOCTYPE html>  
  <html lang="en">  
    <head>  
      <meta charset="utf-8" />  
      <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />  
      <meta name="viewport" content="width=device-width, initial-  
scale=1" />  
      <meta name="theme-color" content="#000000" />  
      <meta  
name="description"  
content="Web site created using create-react-app"  
/>  
      <link  
rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.
css"
    />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.
  </noscript>
    <div id="root"></div>
  </body>
</html>
```

Construint els components

En aquest article, estem construint 5 components.

Navigation.jsx i Footer.jsx són habituals per a totes les rutes.

Home.jsx, About.jsx i Contact.jsx són les rutes.

En primer lloc, creeu dins de **src** un directori **components** . Tots els nostres components estaran codificats al seu interior.

1. Navigation.jsx

Aquesta és la secció més alta de la nostra aplicació que tracta de la navegació de les nostres rutes.

Estem utilitzant en lloc de l'etiqueta . En fer clic a cada enllaç, es dirigirà al camí específic descrit.

```
<Link to="/">Home</Link> - La ruta a la pàgina d'inici que és http: //
localhost: 3000
<Link to="/about">About</Link> - Quant a la ruta de la pàgina, és a dir,
http: // localhost: 3000 / about
<Link to="/contact">Contact</Link> - Ruta a la pàgina de contacte que és
http: // localhost: 3000 / contact
```

```
import React from "react";
import { Link, withRouter } from "react-router-dom";

function Navigation(props) {
  return (
    <div className="navigation">
      <nav class="navbar navbar-expand navbar-dark bg-dark">
        <div class="container">
          <Link class="navbar-brand" to="/">
            React Multi-Page Website
          </Link>
```

```

    <div>
      <ul class="navbar-nav ml-auto">
        <li
          class={`nav-item ${
            props.location.pathname === "/" ? "active" : ""
          }`}
        >
          <Link class="nav-link" to="/">
            Home
            <span class="sr-only">(current)</span>
          </Link>
        </li>
        <li
          class={`nav-item ${
            props.location.pathname === "/about" ? "active" : ""
          }`}
        >
          <Link class="nav-link" to="/about">
            About
          </Link>
        </li>
        <li
          class={`nav-item ${
            props.location.pathname === "/contact" ? "active" : ""
          }`}
        >
          <Link class="nav-link" to="/contact">
            Contact
          </Link>
        </li>
      </ul>
    </div>
  </div>
</nav>
</div>
);
}

export default withRouter(Navigation);

```

També he afegit una lògica senzilla en aquest component per activar diferents elements de navegació de canvi de color en a:Hover. A Bootstrap, he afegit una classe activa que mostrarà un color actiu per a aquest enllaç. A React, **props.location.pathname** donarà el camí exacte de la ruta actual.

He combinat aquests dos estils i, si la ruta actual és la mateixa que la ruta, s'afegirà a l'enllaç una classe anomenada **active** .

Nota: - Per obtenir el nom de ruta exacte, el component ha d'estar embolicat dins amb el router . Podeu veure que he exportat el component de navegació embolicat dins amb el router (línia 61).

2. Footer.jsx

Necessitem una secció inferior que sigui el peu de pàgina. Aquest tram també és comú per a totes les rutes.

```
import React from "react";

function Footer() {
  return (
    <div className="footer">
      <footer class="py-5 bg-dark fixed-bottom">
        <div class="container">
          <p class="m-0 text-center text-white">
            Copyright &copy; Your Website 2020
          </p>
        </div>
      </footer>
    </div>
  );
}

export default Footer;
```

3. Home.jsx

Aquesta és la vista de la pàgina d'inici i representada quan la ruta és "/".

```
import React from "react";

function Home() {
  return (
    <div className="home">
      <div class="container">
        <div class="row align-items-center my-5">
          <div class="col-lg-7">
            
          </div>
          <div class="col-lg-5">
            <h1 class="font-weight-light">Home</h1>
            <p>
              Lorem Ipsum home is simply dummy text of the printing and
              typesetting
              industry. Lorem Ipsum has been the industry's standard
              dummy text
              ever since the 1500s, when an unknown printer took a galley
              of
              type and scrambled it to make a type specimen book.
            </p>
          </div>
        </div>
      </div>
    </div>
  );
}
```

```

        </div>
      </div>
    </div>
  );
}

export default Home;

```

3. About.jsx

Aquesta és la vista de la pàgina Quant a i es mostra quan la ruta és "/ about".

```

import React from "react";

function About() {
  return (
    <div className="about">
      <div class="container">
        <div class="row align-items-center my-5">
          <div class="col-lg-7">
            
          </div>
          <div class="col-lg-5">
            <h1 class="font-weight-light">About</h1>
            <p>
              Lorem Ipsum about is simply dummy text of the printing and
              typesetting
              dummy text
              of
              industry. Lorem Ipsum has been the industry's standard
              ever since the 1500s, when an unknown printer took a galley
              type and scrambled it to make a type specimen book.
            </p>
          </div>
        </div>
      </div>
    </div>
  );
}

export default About;

```

3. Contact.jsx

Aquesta és la vista de la pàgina de contacte i es visualitza quan la ruta és "/ contact".

```

import React from "react";

function Contact() {
  return (
    <div className="contact">
      <div class="container">
        <div class="row align-items-center my-5">
          <div class="col-lg-7">
            
          </div>
          <div class="col-lg-5">
            <h1 class="font-weight-light">Contact</h1>
            <p>
              Lorem Ipsum contact is simply dummy text of the printing
              and typesetting
              industry. Lorem Ipsum has been the industry's standard
              dummy text
              ever since the 1500s, when an unknown printer took a galley
              of
              type and scrambled it to make a type specimen book.
            </p>
          </div>
        </div>
      </div>
    </div>
  );
}

export default Contact;

```

Per simplificar la importació de tots els components que hem desenvolupat, exportem dins del directori de components, tots els components formant un únic fitxer **index.js**.

```

export { default as Navigation } from "./Navigation";
export { default as Footer } from "./Footer";
export { default as Home } from "./Home";
export { default as About } from "./About";
export { default as Contact } from "./Contact";

```

Declaració de totes les rutes a App.js

Per tant, després de crear tots els components, els hem d'importar a App.js i canviar les rutes amb cada ruta corresponent.

```
import React from "react";
import { BrowserRouter as Router, Route, Switch } from "react-router-dom";
import { Navigation, Footer, Home, About, Contact } from
"./components";
function App() {
  return (
    <div className="App">
      <Router>
        <Navigation />
        <Switch>
          <Route path="/" exact component={() => <Home />} />
          <Route path="/about" exact component={() => <About />} />
          <Route path="/contact" exact component={() => <Contact />} />
        </Switch>
        <Footer />
      </Router>
    </div>
  );
}

export default App;
```