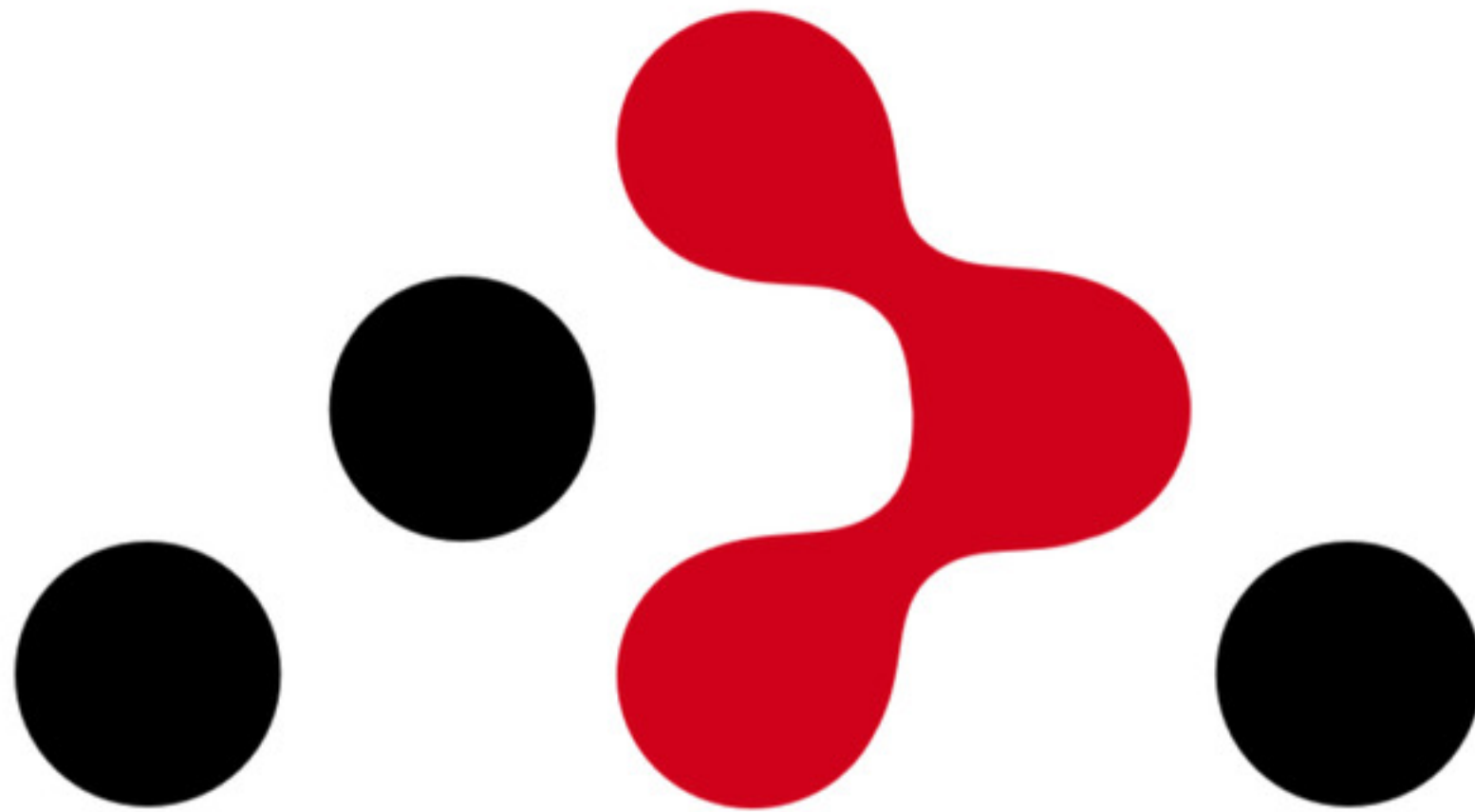


React Router Introduction



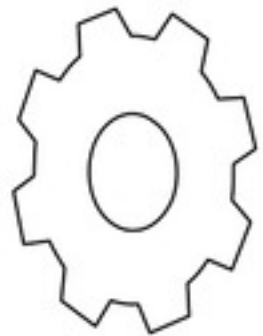
Routing in a Single Page Application with React Router

Navbar Home About Contact

Home

© 2022 Company, Inc

Home About Contact



We are going to cover everything you need to know to get started with React Router.

- Setting up the project
- What is routing?
- Setting up the router
- Rendering routes
- Using links to switch pages
- Final Thoughts
- Next Steps
- Assignments and resources

Setting up the project

To be able to follow along, you will need to create a new React app by running the following command in your terminal:

```
npx create-react-app router-demo
```

Then, if you're ready to go, let's start by answering an important question: what is routing?



What is routing?



Routing is the capacity to show different pages to the user. That means the user can move between different parts of an application by entering a URL or clicking on an element.

As you may already know, by default, React comes without routing. And to enable it in our project, we need to add a library named [react-router-dom](#).

To install it, you will have to run the following command in your terminal:

```
npm install react-router-dom
```

Now, we've successfully installed our router, let's start using it in the next section 😊

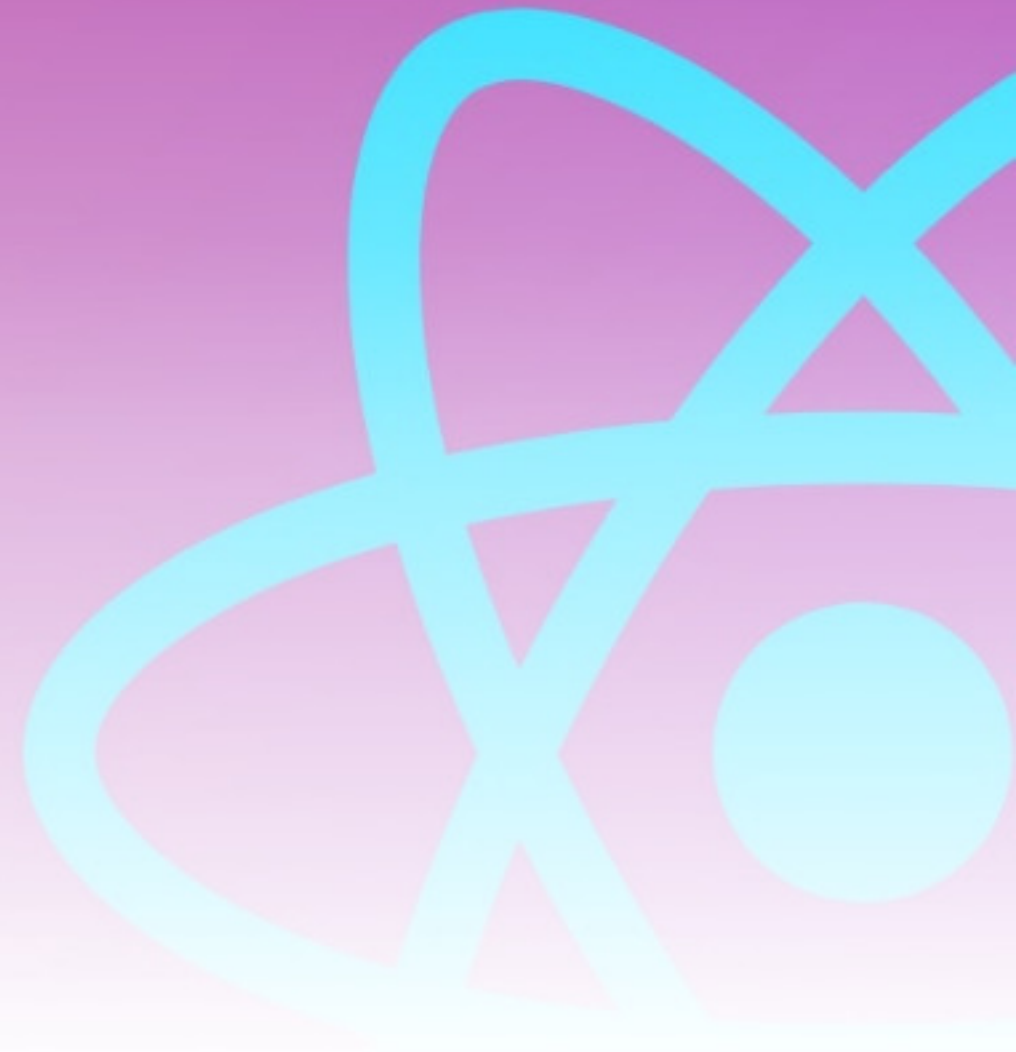


REACT



ROUTER

Video




React Router V6 Tutorial - Routes, Redirecting, UseNavigate, UseParams...




Setting up the router



To enable routing in our React app, we first need to import BrowserRouter from react-router-dom.




In the index.js file, enter the following:



```
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6 import { BrowserRouter } from "react-router-dom";
7 const root = ReactDOM.createRoot(document.getElementById("root"));
8 root.render(
9   <React.StrictMode>
10     <BrowserRouter>
11       <App />
12     </BrowserRouter>
13   </React.StrictMode>
14 );
15
16 // If you want to start measuring performance in your app, pass a function
17 // to log results (for example: reportWebVitals(console.log))
18 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
19 reportWebVitals();
20
```

This should hold everything in our app where routing is needed. That means, if we need routing in our entire app, we must wrap our App component with BrowserRouter.



A router alone doesn't do much. So let's add a route in the next section.





Rendering routes



- Router: The router that keeps the UI in sync with the URL
- Link: Renders a navigation link
- Route: Renders a UI component depending on the URL



To render routes, we have to import the Route and Routes component from the router package.

In your App.js file, add the following code:

```
1 import './App.css';
2 import { Routes, Route } from 'react-router-dom';
3 import About from './components/Pages/About/About';
4 import Contact from './components/Pages/Contact/Contact';
5 import Home from './components/Pages/Home/Home';
6 import Navbar from './components/Navbar/Navbar';
7 import Footer from './components/Footer/Footer';
8
9 function App() {
10   return (
11     <div className="App">
12       <Navbar />
13       <main style={{ height: "80vh" }}>
14         <Routes>
15           <Route path="/about" element={ <About /> } />
16           <Route path="/contact" element={ <Contact /> } />
17           <Route path="/" element={ <Home /> } />
18           <Route path="*" element={ <div>404</div> } />
19         </Routes>
20       </main>
21       <Footer />
22     </div>
23   );
24 }
25
26 export default App;
27
```





Rendering routes



Then, add it where we want to render the content. The Route component has several properties. But here, we just need path and element.

path: the path of the route. Here, we use / to define the path of the home page.

element: will display the content whenever the route is reached.

To get the full power of React Router, we need to have multiple pages and links to play with. We already have pages (components if you want, too), so now let's add some links so we can switch between pages.





Using links to switch pages



To add links to our project, we will use the React Router again.

But first let's add bootstrap to our project [Get started with Bootstrap CDN links](#)

In your public/index.html file, add the following code:



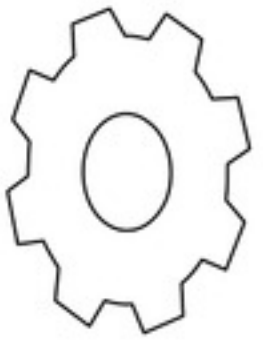
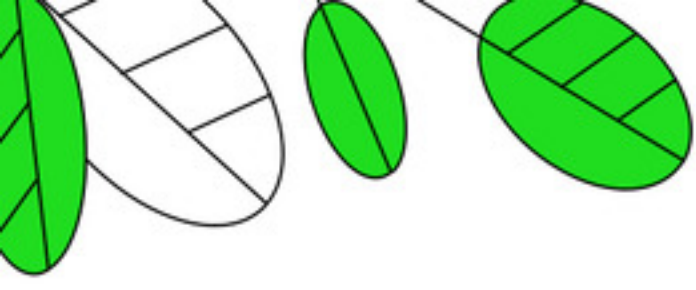
```
<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
  rel="stylesheet"
  integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+0rBnVFB�6D0itfPri4tjFHxaWutUpFmBp4vmVor"
  crossorigin="anonymous"
/>
```

```
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
  crossorigin="anonymous"
></script>
```



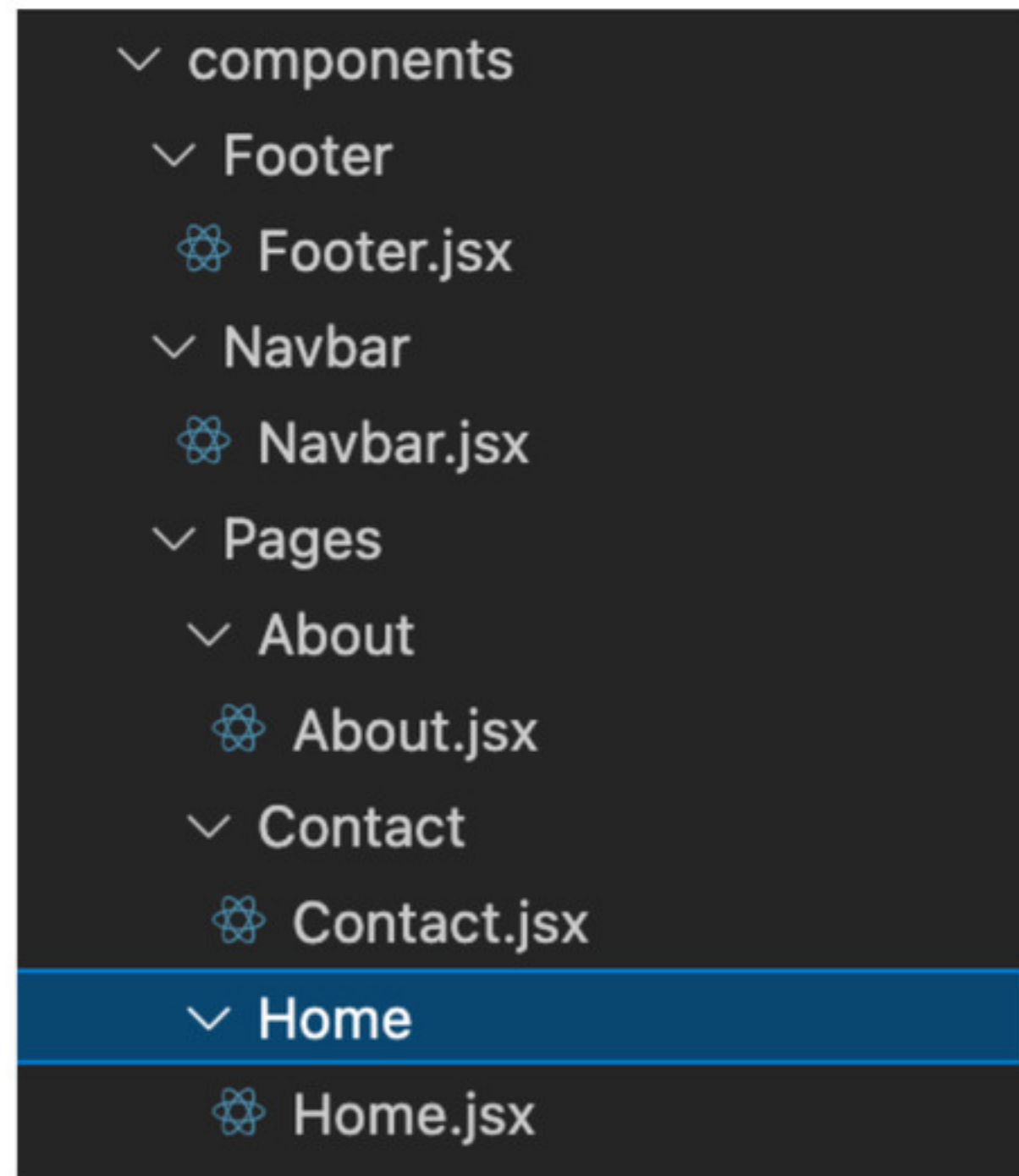
Using links to switch pages

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13    <!--
14      manifest.json provides metadata used when your web app is installed on a
15      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16    -->
17    <link
18      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
19      rel="stylesheet"
20      integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
21      crossorigin="anonymous"
22    />
23
24    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
25    <!--
26      Notice the use of %PUBLIC_URL% in the tags above.
27      It will be replaced with the URL of the 'public' folder during the build.
28      Only files inside the 'public' folder can be referenced from the HTML.
29
30      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
31      work correctly both with client-side routing and a non-root public URL.
32      Learn how to configure a non-root public URL by running 'npm run build'.
33    -->
34    <title>React Routes Demo</title>
35  </head>
36  <body>
37    <noscript>You need to enable JavaScript to run this app.</noscript>
38    <div id="root"></div>
39    <!--
40      This HTML file is a template.
41      If you open it directly in the browser, you will see an empty page.
42
43      You can add webfonts, meta tags, or analytics to this file.
44      The build step will place the bundled scripts into the <body> tag.
45
46      To begin the development, run 'npm start' or 'yarn start'.
47      To create a production bundle, use 'npm run build' or 'yarn build'.
48    -->
49    <script
50      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
51      integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUSLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
52      crossorigin="anonymous"
53    ></script>
54  </body>
55 </html>
56
```

Using links to switch pages

Create components folder and and put the following folders and files.



Using links to switch pages

Let's update the Navbar component In your components/Navbar/Navbar file, add the following code:

[Navbar Components \(Documentation and examples for Bootstrap's powerful, responsive navigation\)](#)

```
1 import React from "react";
2 import { Link } from "react-router-dom";
3
4 export default function Navbar() {
5   return (
6     <nav className="navbar navbar-expand-lg bg-light">
7       <div className="container-fluid">
8         <Link className="navbar-brand" to="/">
9           Navbar
10        </Link>
11        <div className="collapse navbar-collapse" id="navbarNavAltMarkup">
12          <div className="navbar-nav">
13            <Link className="nav-link" to="/">
14              Home
15            </Link>
16            <Link className="nav-link" to="/about">
17              About
18            </Link>
19            <Link className="nav-link" to="/contact">
20              Contact
21            </Link>
22          </div>
23        </div>
24      </div>
25    </nav>
26  );
27 }
```



Using links to switch pages



Let's add the pages



In your components/Pages/About file, add the following code:

```
1 import React from "react";
2
3 export default function About() {
4   return <h1>About</h1>;
5 }
6
```



In your components/Pages/Contact file, add the following code:

```
1 import React from "react";
2
3 export default function Contact() {
4   return <h1>Contact</h1>;
5 }
6
```

In your components/Pages/Home file, add the following code:

```
1 import React from "react";
2
3 export default function Home() {
4   return <h1>Home</h1>;
5 }
6
```



Using links to switch pages

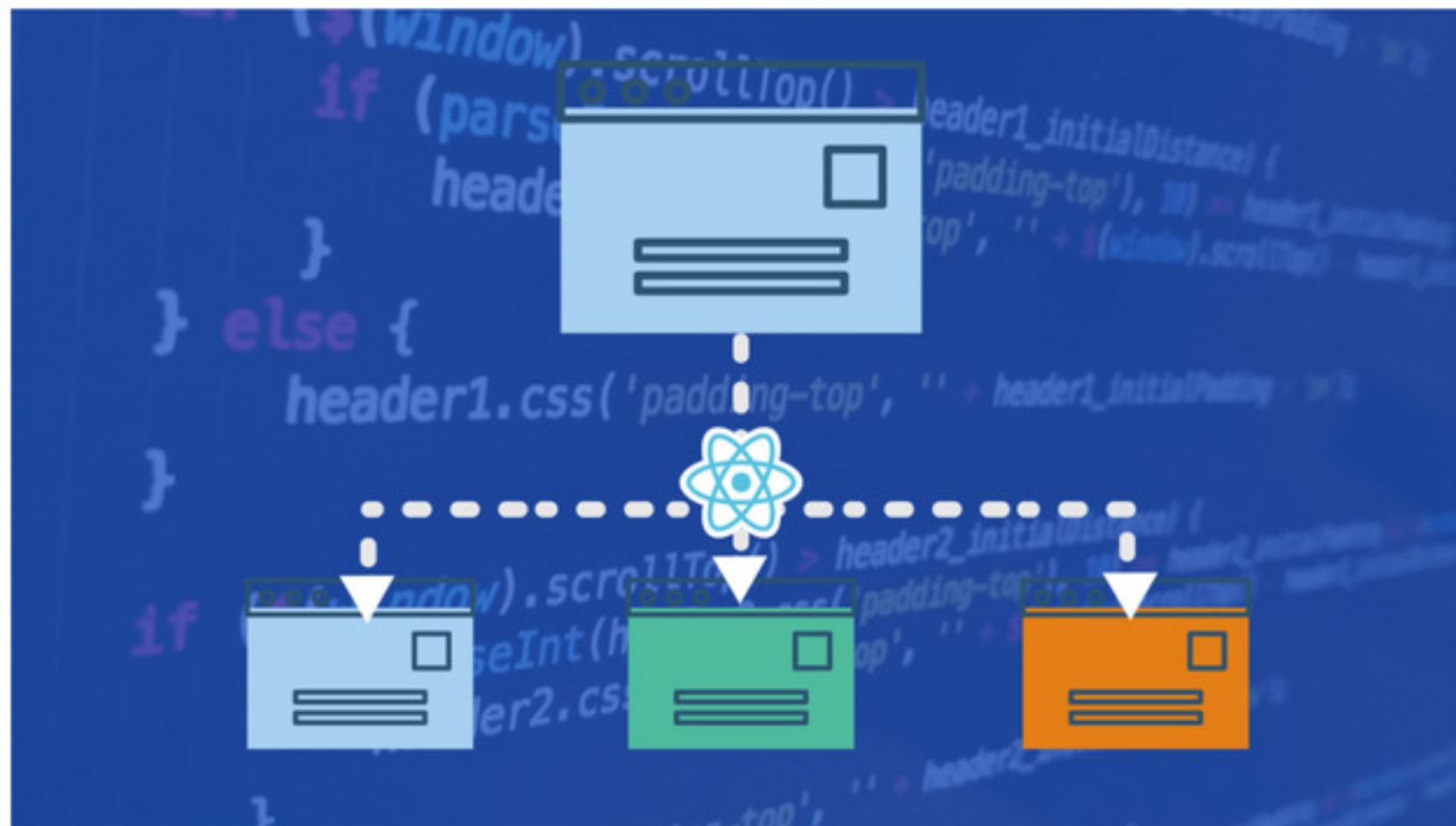
Let's update the Footer component

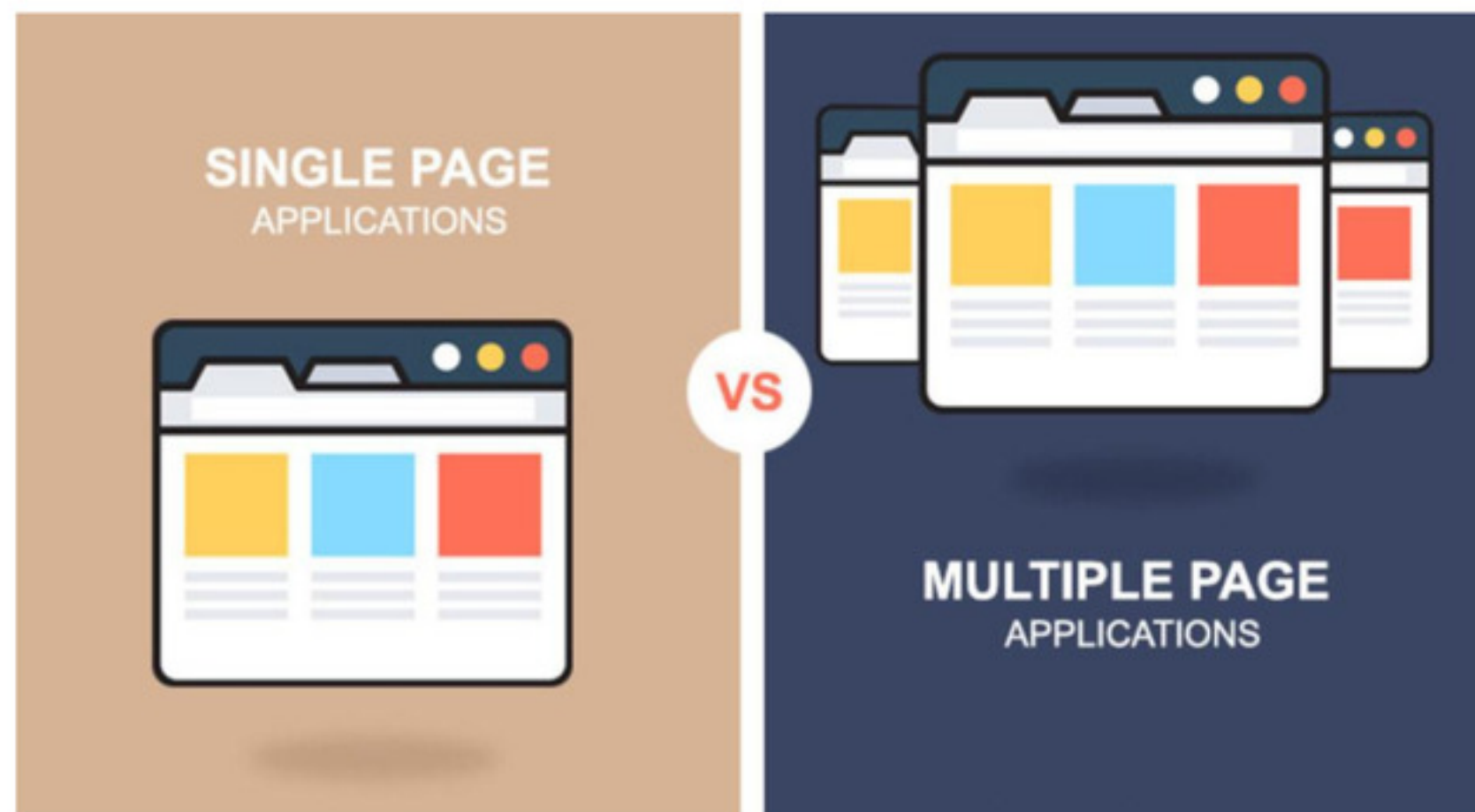
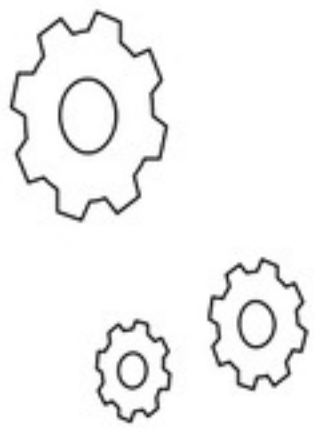
In your components/Navbar/Footer file, add the following code:

```
1 import React from "react";
2 import { Link } from "react-router-dom";
3
4 export default function Footer() {
5   return (
6     <footer className="container d-flex flex-wrap justify-content-between align-items-center py-3 my-4 border-top">
7       <p className="col-md-4 mb-0 text-muted">&copy; 2022 Company, Inc</p>
8
9       <ul className="nav col-md-4 justify-content-end">
10        <li className="nav-item">
11          <Link to="/" className="nav-link px-2 text-muted">
12            Home
13          </Link>
14        </li>
15
16        <li className="nav-item">
17          <Link to="/about" className="nav-link px-2 text-muted">
18            About
19          </Link>
20        </li>
21
22        <li className="nav-item">
23          <Link to="/contact" className="nav-link px-2 text-muted">
24            Contact
25          </Link>
26        </li>
27      </ul>
28    </footer>
29  );
30 }
```




After importing Link, we have to update our navigation bar a bit. Now, instead of using a tag and href, React Router uses Link and to, well, be able to switch between pages without reloading it.

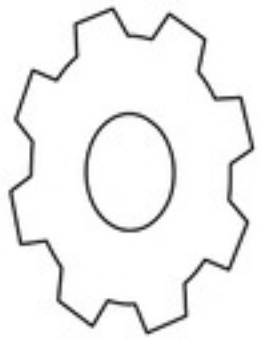
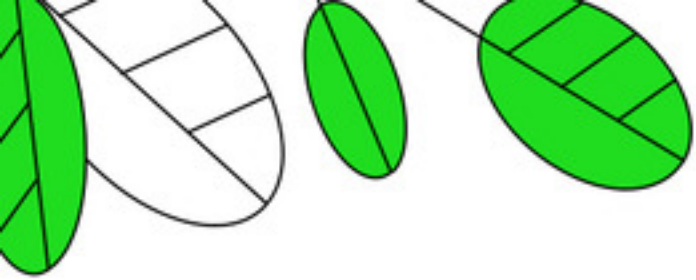




Final Thoughts

React Router is an amazing library that helps us go from a single page to a multi-page application feeling with great usability. (Just keep in mind – at the end of the day, it's still a single page app).





Next Steps

Passing route parameters





Time To Climb

Open Ended Question

We're happy to read your feedback/questions or anything you'd like to share with us

TIME TO PRACTICE

