

Navigation with React Router

In this lecture, we will learn how to add navigation to a React app using **React Router**. This allows us to create multiple pages within a single-page application.

Installing React Router

First, install React Router with npm:

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

Setting Up Routes

Open `main.jsx` and wrap your app in a `BrowserRouter` :

```
import React from "react"
import ReactDOM from "react-dom/client"
import { BrowserRouter } from "react-router-dom"
import App from "./App"
import "./index.css"

ReactDOM.createRoot(document.getElementById("root")).render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
)
```

Defining Routes

Inside `App.jsx`, use the `Routes` and `Route` components to define paths.

```

import { Routes, Route, Link } from "react-router-dom"

function Home() {
  return <h2>Home Page</h2>
}

function About() {
  return <h2>About Page</h2>
}

function Contact() {
  return <h2>Contact Page</h2>
}

function App() {
  return (
    <div>
      <nav>
        <Link to="/">Home</Link> |{" "}
        <Link to="/about">About</Link> |{" "}
        <Link to="/contact">Contact</Link>
      </nav>

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/contact" element={<Contact />} />
      </Routes>
    </div>
  )
}

export default App

```

Here's what happens:

- The navigation bar uses `Link` to change pages without reloading.
- `Routes` and `Route` define which component shows for each path.

- For example, `/about` displays the `About` component.

Recap

- React Router enables multi-page navigation in React apps.
- Use `BrowserRouter` to enable routing.
- Define routes with `Routes` and `Route`.
- Use `Link` for navigation instead of `<a>` tags.