



React Router

Software Development Bootcamp



Topic

Introduction To React Router



What Is React Router?

React Router is a standard library for routing in React applications.

It enables navigation between different components while maintaining a single-page application (SPA) experience, allowing you to create applications with multiple views and dynamic, client-side routing.



React Router Basics

- Enables dynamic, client-side routing in React apps
- Maintains UI state between route changes
- Provides declarative routing components
- Supports nested routing and URL parameters



Why Use React Router?

Traditional web navigation requires full page reloads. React Router addresses this by:

- Enabling smooth transitions between views
- Maintaining application state during navigation
- Providing a more app-like user experience
- Supporting bookmarkable URLs in SPAs



Core Concepts

- **BrowserRouter:** The wrapper component that enables routing
- **Routes:** Defines the routing configuration
- **Route:** Maps URLs to components
- **Link:** Creates navigation links without page reloads
- **Navigate:** Programmatic navigation component



Topic

Setting Up React Router



Step 1

- Install React Router

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




Step 2

- Wrap your app with **BrowserRouter**

```
import { BrowserRouter } from
'react-router-dom'

function App() {
  return (
    <BrowserRouter>
      { /* Your app content */ }
    </BrowserRouter>
  )
}
```



Step 3

- Basic Routing
 - Routes are defined using the Routes and Route components
 - Each Route maps a URL path to a component
 - The router renders the first matching Route

```
import { Routes, Route } from "react-router-dom";
import About from "../About";
import Contact from "../Contact";

function BasicRouting() {
  return (
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/about" element={<About />} />
      <Route path="/contact" element={<Contact />} />
    </Routes>
  );
}

export default BasicRouting;
```



Navigation Components

- **Link:** Creates clickable navigation links
- **NavLink:** Similar to Link but with active state styling
- **Navigate:** Programmatically redirect users



Hooks In React Router

- `useNavigate`: Programmatic navigation
- `useParams`: Access URL parameters
- `useLocation`: Access current location
- `useSearchParams`: Handle query parameters



Best Practices

- Keep route configurations organized
- Use consistent naming conventions
- Use meaningful URLs



Exercise