

Objectives

- Understand what a Router is in React
- Learn how to use Routes and Route
- Apply routing in building Single Page Applications (SPA)

1. What is Routing?

Routing is navigation between different pages or views in a web app

In a Single Page Application (SPA), routing allows switching views without reloading the page

Example: "/home" shows the Home page, "/about" shows the About page

2. What is React Router?

- A popular library for handling routing in React applications
- Provides components like BrowserRouter, Routes, and Route

Installation:

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

3. BrowserRouter

Wraps the whole app and enables routing functionality

Example:

```
import { BrowserRouter } from 'react-router-dom';  
  
<BrowserRouter>  
  <App />  
</BrowserRouter>
```

4. Routes and Route

Routes is a container that holds all the Route definitions

Route maps a URL path to a specific React component

Basic Example:

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

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />} />
</Routes>
```

5. Useful Features

Nested Routes:

```
<Route path="/dashboard" element={<Dashboard />}>
  <Route path="settings" element={<Settings />} />
</Route>
```

Dynamic Routes (Parameter):

```
<Route path="/product/:id" element={<ProductDetail />} />
```

Navigation using code:

```
import { useNavigate } from 'react-router-dom';
const navigate = useNavigate();
navigate('/home');
```

6. Summary

- BrowserRouter Wraps app to enable routing
- Routes Container for multiple Routes
- Route Maps path to a React component

Hit: How to Handle 404 Pages in React Router

In React Router v6 and newer, handling a 404 (Not Found) page is simple. You define a Route with a wildcard (*) path at the bottom of your Routes. This acts as a catch-all for any undefined routes.

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Home from './Home';
import About from './About';
import NotFound from './NotFound'; // Your 404 component

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />

```

```
    <Route path="/about" element={<About />} />

    { /* Catch-all 404 route */ }
    <Route path="*" element={<NotFound />} />
  </Routes>
</BrowserRouter>
);
}
```

NotFound Component (Example)

```
const NotFound = () => (
  <div>
    <h1>404 – Page Not Found</h1>
    <p>Sorry, the page you're looking for doesn't exist.</p>
  </div>
);
```

Notes

The `path="*"` must always be placed last inside `<Routes>`, or it might override other routes.

You can also redirect to a 404 route using `<Navigate to="/404" />` if needed.