Routing in React (React Router)

• Question 1: What is React Router? How does it handle routing in single-page applications?

**📌 What is React Router?**

**React Router** is a **standard routing library for React** used to manage navigation and rendering of components in a **Single Page Application (SPA)**.

It allows you to:

* Navigate between different components/pages without refreshing the entire page.
* Create routes for different URLs.
* Handle dynamic routing (e.g., /product/:id).
* Manage navigation history (back/forward) in the browser.

**📌 How React Router Handles Routing in SPAs**

In a traditional website:

* Each URL change triggers a **new page reload** from the server.

In a **React SPA** with React Router:

* The entire app is loaded once.
* **React Router intercepts URL changes** and decides **which component to render**, without refreshing the page.

**🔧 Key Components in React Router**

| **Component** | **Purpose** |
| --- | --- |
| <BrowserRouter> | Wraps the app and enables routing using the HTML5 history API. |
| <Routes> | Wraps all the <Route> components. |
| <Route> | Defines a specific path and the component to render. |
| <Link> | Used for navigation without full-page reload (like <a> tag). |
| useNavigate() | Programmatically navigate to different routes inside JavaScript logic. |

**🧠 How it Works Internally**

1. You load your SPA once in the browser.
2. The <BrowserRouter> tracks the URL changes.
3. When a URL changes:
   * React Router **matches the path** to a <Route>.
   * It **renders the matched component**.
   * No actual page reload happens — only the component changes on the screen.

**📝 Example:**

jsx

CopyEdit

import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';

function App() {

return (

<BrowserRouter>

<nav>

<Link to="/">Home</Link> |

<Link to="/about">About</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

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

</Routes>

</BrowserRouter>

);

}

• Question 2: Explain the difference between BrowserRouter, Route, Link, and Switch components in React Router.

React Router is a powerful library used to implement routing in React applications. Here’s a detailed explanation of the key components:

**1. 🔷 BrowserRouter**

* **Definition:** It is the top-level component that enables routing in your React app using the **HTML5 history API**.
* **Role:** Wraps your entire application so that routing can work.
* **Must be used only once** in the application.

jsx

CopyEdit

<BrowserRouter>

<App />

</BrowserRouter>

**2. 🔶 Route**

* **Definition:** Used to **map a URL path to a specific React component**.
* **Role:** Displays a component based on the current URL.
* **Syntax:**

jsx

CopyEdit

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

* **Note:** In React Router v5, it uses component={}. In v6 and above, it uses element={}.

**3. 🟩 Link**

* **Definition:** Used for **navigation** within the app, similar to <a> but without page reload.
* **Role:** Changes the URL and triggers the matching <Route> without refreshing the page.

jsx

CopyEdit

<Link to="/about">Go to About</Link>

* **Why use Link instead of <a>?**
  + <a> reloads the page.
  + <Link> enables SPA behavior (no reload).

**4. 🟨 Switch (For React Router v5)**

* **Definition:** Renders the **first <Route>** that matches the current URL.
* **Role:** Prevents multiple routes from rendering at the same time.

jsx

CopyEdit

<Switch>

<Route path="/about" component={About} />

<Route path="/" component={Home} />

</Switch>

* **Note:** In **React Router v6**, Switch is replaced with <Routes>.

**📌 Summary Table**

| **Component** | **Purpose** | **Version Used In** |
| --- | --- | --- |
| BrowserRouter | Enables routing using browser history | All versions |
| Route | Defines which component to render for a path | All versions |
| Link | Navigates between routes without reloading | All versions |
| Switch | Renders only the first matching route | React Router v5 |
| Routes | Replaces Switch in React Router v6+ | React Router v6+ |

LAB EXERCISE

• Task 1: o Set up a basic React Router with two routes: one for a Home page and one for an About page. Display the appropriate content based on the URL

**🔧 Steps:**

1. Install React Router:

nginx

CopyEdit

npm install react-router-dom

1. In App.js, set up BrowserRouter, Routes, and Route for Home and About.

jsx

CopyEdit

// App.js

import React from 'react';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import Home from './Home';

import About from './About';

function App() {

return (

<BrowserRouter>

<Routes>

<Route path="/" element={<Home />} />

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

</Routes>

</BrowserRouter>

);

}

export default App;

1. Create Home.js:

jsx

CopyEdit

function Home() {

return <h1>Welcome to the Home Page</h1>;

}

export default Home;

1. Create About.js:

jsx

CopyEdit

function About() {

return <h1>About Us</h1>;

}

export default About;

•Task 2: About page. Display the appropriate content based on the URL.

Already done in Step 4 above!  
You simply need to make sure /about URL displays the **About** component.

Task 3 Create a navigation bar using React Router’s Link component that allows users to switch between the Home, About, and Contact pages.

**🔧 Steps:**

1. Add a Navbar.js component:

jsx

CopyEdit

import { Link } from 'react-router-dom';

function Navbar() {

return (

<nav style={{ padding: "10px", background: "#eee" }}>

<Link to="/" style={{ margin: "10px" }}>Home</Link>

<Link to="/about" style={{ margin: "10px" }}>About</Link>

<Link to="/contact" style={{ margin: "10px" }}>Contact</Link>

</nav>

);

}

export default Navbar;

1. Create a Contact.js page:

jsx

CopyEdit

function Contact() {

return <h1>Contact Us</h1>;

}

export default Contact;

1. Update App.js to include Navbar and Contact route:

jsx

CopyEdit

import React from 'react';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import Home from './Home';

import About from './About';

import Contact from './Contact';

import Navbar from './Navbar';

function App() {

return (

<BrowserRouter>

<Navbar />

<Routes>

<Route path="/" element={<Home />} />

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

<Route path="/contact" element={<Contact />} />

</Routes>

</BrowserRouter>

);

}

export default App;

**✅ Final Folder Structure:**

css

CopyEdit

src/

├── App.js

├── Home.js

├── About.js

├── Contact.js

├── Navbar.js