

## Routing in React (React Router)

---

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

**Ans.**

React Router is a popular routing library for React that enables navigation between different views (pages) in a Single-Page Application (SPA) without reloading the entire browser page.

What is Routing in a Single-Page Application (SPA)?

In a traditional multi-page website:

- Each URL change loads a new HTML page from the server.

In a SPA (like React apps):

- The app loads once
- URL changes update only the required components
- No full page reload happens

This is exactly what React Router helps achieve.

### What is React Router?

React Router:

- Maps **URLs (paths)** to **React components**
- Updates the UI when the URL changes
- Keeps the app fast and smooth by avoiding page reloads

Example :

/home → Home component

/about → About component

/contact → Contact component

## How React Router Handles Routing in SPAs

### 1. Uses the Browser History API

React Router uses the **HTML5 History API** (pushState, replaceState) to:

- Change the URL
- Track navigation history
- Update components without refreshing the page

### 2. Defines Routes Using <Routes> and <Route>

Each route connects a URL path to a component.

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

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

```
<Route path="/contact" element={<Contact />} />  
</Routes>  
</BrowserRouter>  
);  
}
```

### 3. Navigation Without Page Reload (<Link>)

Instead of <a> tags, React Router uses <Link>:

```
import { Link } from "react-router-dom";
```

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

### 4. Route Matching

React Router:

- Matches the current URL with defined routes
- Renders the **best matching component**

Example:

URL = /about → About component rendered

### 5. Dynamic Routing

Routes can have parameters:

```
<Route path="/users/:id" element={<User />} />
```

Access parameter:

```
import { useParams } from "react-router-dom";
```

```
const { id } = useParams();
```

## 6. Nested Routes

React Router supports layouts and child routes:

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

## Key Advantages of React Router

- No full page reloads
  - Faster navigation
  - Clean and readable URLs
  - Supports dynamic & nested routes
  - Works perfectly with SPAs
-

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

**Ans.**

Here's a clear, interview-ready explanation of the difference between BrowserRouter, Route, Link, and Switch in React Router.

## 1. BrowserRouter

What it is:

- A router wrapper component
- Uses the HTML5 History API

Purpose:

- Enables routing in a React application
- Must wrap the entire app (or routing part)

Example:

```
import { BrowserRouter } from "react-router-dom";
```

```
function App() {
  return (
    <BrowserRouter>
      {/* Routes go here */}
    </BrowserRouter>
  );
}
```

## **2. Route**

What it is:

- A component that maps a URL path to a React component

Purpose:

- Decides which component to render based on the URL

Example (v6):

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

```
<Routes>
```

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

```
</Routes>
```

**Example (v5):**

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

## **3. Link**

What it is:

- A navigation component (replacement for  tag)

Purpose:

- Navigate between routes without page reload

Example:

```
import { Link } from "react-router-dom";
<Link to="/about">About</Link>
```

### Why not <a>?

- <a> refreshes the page
- <Link> keeps SPA behavior

### Key Point:

Link changes the URL without reloading the page.

## 4. Switch (React Router v5 ONLY)

What it is:

- A wrapper that renders only the first matching Route

Purpose:

- Prevents multiple routes from rendering at once

Example (v5):

```
import { Switch, Route } from "react-router-dom";
```

```
<Switch>
  <Route exact path="/" component={Home} />
  <Route path="/about" component={About} />
</Switch>
```

## React Router v6 Update :

| v5 Component | v6 Replacement |
|--------------|----------------|
| Switch       | Routes         |
| component    | element        |
| exact        | Not required   |

### Example (v6):

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

---