

React Router

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

Answer: React Router is a standard routing library for React that enables navigation between different components and views in a single-page application (SPA) without reloading the page.

How it handles routing in SPAs:

- In a traditional multi-page app, navigating to a new page causes the browser to load a new HTML document from the server.
- In a React SPA using React Router, routing is handled client-side. This means:
 - The URL in the browser changes,
 - React Router intercepts the change and renders a new component based on the route,
 - The page does not reload, and only the necessary components are updated.
- It uses the History API or hash-based routing to manage navigation and keep the UI in sync with the URL.

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

Answer:

1. BrowserRouter:

BrowserRouter is a wrapper component that enables routing functionality in a React application. It uses the HTML5 History API to keep the UI in sync with the URL. You place this at the top level of your app to allow other routing components like Route and Link to work.

2. Route:

Route is used to define the mapping between a specific path and the

component that should be rendered when the path matches the current URL. It listens to changes in the URL and displays the corresponding component without refreshing the page.

3. Link:

Link is used for navigation in a React app. It works like an anchor (<a>) tag but without reloading the entire page. When a user clicks a Link, it changes the URL and React Router loads the matching Route component smoothly.

4. Switch:

Switch is used to group multiple Route components and render only the first one that matches the current URL. This is useful when you have overlapping routes and want to ensure only one route is rendered. Note that in React Router v6, Switch has been replaced by Routes.