React is a JavaScript library that is used to create user interfaces. With the aid of React Router, we can also extend it to create multi-page apps. This is a third-party library that allows us to route our React apps. Routing is a critical component of the emerging single page application (SPA), defining what should appear when a user accesses a certain page in the application. When a user clicks an element (link, button, icon, picture, etc.) or enters a URL inside the application, routing allows them to travel between different pages of the application.

**React Router is used to implement routing in React applications.**It is a collection of **navigational components** within your application.

**What is React Router?**

**React Router is a Routing and Navigation Library for**[**React**](https://www.geeksforgeeks.org/react-tutorial/). It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

The current latest verstion is [**React router dom v6**](https://www.geeksforgeeks.org/what-is-react-router-dom/)

**Features**

* **Declarative Routing:** React Router v6 uses the Routes and Route components to define routes declaratively, making the routing configuration simple and easy to read.
* **Nested Routes:** It supports nested routes, allowing for complex and hierarchical routing structures, which helps in organizing the application better.
* **Programmatic Navigation:** The useNavigate hook enables programmatic navigation, allowing developers to navigate between routes based on certain conditions or user actions.
* **Route Parameters:** It provides dynamic routing with route parameters, enabling the creation of routes that can match multiple URL patterns.

**Components of React Router**

**React Router mainly comprises of the below components**

**1. BrowserRouter and HashRouter**

* BrowserRouter: Uses the HTML5 history API to keep your UI in sync with the URL.
* HashRouter: Uses the hash portion of the URL (i.e., window.location.hash) to keep your UI in sync with the URL.

<BrowserRouter>  
 (/\* Your routes go here \*/}  
</BrowserRouter>

**Routes and Route**

* **Routes**: A container for all your route definitions.
* **Route**: Defines a single route with a path and the component to render.

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

**3.**[Link and NavLink](https://www.geeksforgeeks.org/link-and-navlink-components-in-react-router-dom/)

* **Link**: Creates navigational links in your application.
* **NavLink**: Similar to Link but provides additional styling attributes when the link is active.

<nav>  
 <NavLink to="/" activeClassName="active">Home</NavLink>  
 <Link to="/about">About</Link>  
</nav>

**Steps to Implement React Router**

**Step 1: Initialize React Project**

Create React application using the following command.

npx create-react-app react-router-example

cd react-router-example

**Step 2: Install React Router**

Install react-router in your application write the following command in your terminal

npm install react-router-dom@6

**Uses of React Router**

1. **Navigation and Routing:** React Router provides a declarative way to navigate between different views or pages in a React application. It allows users to switch between views without refreshing the entire page.
2. **Dynamic Routing:** React Router supports dynamic routing, which means routes can change based on the application’s state or data, making it possible to handle complex navigation scenarios.
3. **URL Management:** React Router helps manage the URLs in your application, allowing for deep linking, bookmarkable URLs, and maintaining the browser’s history stack.
4. **Component-Based Approach:** Routing is handled through components, making it easy to compose routes and navigation in a modular and reusable way.

**Summary**

React Router is a powerful library for managing navigation in React applications. It enables seamless page transitions without reloading, using components like BrowserRouter, Routes, and Link. It supports dynamic routing, nested routes, and URL parameters, enhancing user experience.

**FAQ’s – React Router**

**What is React Router?**

*React Router is a library for managing navigation and routing in React applications.*

**How do you install React Router?**

*To install react router use this npm commang: $npm install react-router-dom.*

**What is the difference between BrowserRouter and HashRouter?**

*BrowserRouter uses the HTML5 history API, while HashRouter uses the URL hash.*

**How to pass parameters in React Router?**

*Use URL parameters in the route path, e.g., path=”/user/:id”.*

**How do you navigate programmatically?**

*Use the useNavigate hook or history.push().*

**What is Link in React Router?**

*Link is a component for navigating without full page reloads.*

**What is NavLink?**

*NavLink is similar to Link but provides active link styling.*

**How to handle redirects in React Router?**

*Use the Navigate component within Routes.*

**How to create nested routes?**

*Use nested <Route> components within a parent <Route> in Routes.*

**What is useParams in React Router?**

*useParams is a hook that returns route parameters from the URL.*

**React JS Types of Routers**

Till now we only used BrowserRouter as it is the most widely used React router but there are some other types of routers provided with this package

**React JS Types of Routers :**

On the basis of the part of the URL that the router will use to track the content that the user is trying to view, React Router provides three different kinds of routers:

* **Memory Router**
* **Browser Router**
* **Hash Router**

Memory Router:

The memory router keeps the URL changes in memory not in the user browsers. It keeps the history of the URL in memory and it does not read or write to the address bar so the user can not use the browser’s back button as well as the forward button. It doesn’t change the URL in your browser. It is very useful for testing and non-browser environments like React Native.

Syntax:

import { MemoryRouter as Router } from 'react-router-dom';

**Browser Router:**

It uses HTML 5 history API (i.e. pushState, replaceState, and popState API) to keep your UI in sync with the URL. It routes as a normal URL in the browser and assumes that the server is handling all the request URL (eg., /, /about) and points to root index.html. It accepts forceRefresh props to support legacy browsers that don’t support HTML 5 pushState API

**Syntax:**

import { BrowserRouter as Router } from 'react-router-dom';

**Hash Router:**

Hash router uses client-side hash routing. It uses the hash portion of the URL (i.e. window.location.hash) to keep your UI in sync with the URL. The hash portion of the URL won’t be handled by the server, the server will always send the index.html for every request and ignore the hash value. It doesn’t need any configuration in the server to handle routes. It is used to support legacy browsers which usually don’t support HTML pushState API. It is very useful for legacy browsers or you don’t have a server logic to handle the client-side. This route isn’t recommended to be used by the react-router-dom team.

**Syntax:**

import { HashRouter as Router } from 'react-router-dom';