

React Router DOM

Simplifying Navigation

& Dynamic Routing

Created by

Abdulrazak Attar
SDE @ **msg**
global



Embrace the bugs, they make you stronger

swipe >>



What We Will Learn?

- 1. Routing Basics**
- 2. Passing Query Parameters**
- 3. Dynamic Route Paths with Parameterized or Dynamic Segments**

1. Routing Basics


Let's start with a simple example to understand the basics of routing in React Router. Imagine we have a **Header** component with three links: **Home**, **About**, and **Contact_Us**. On clicking each link, we want to display the respective component content.

Here's how we can achieve this using React Router:

1. Install React Router DOM:

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

2. Create the **Header** component with links using the **Link/NavLink** from react-router-dom.



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

export default function Header() {
  return (
    <>
      <Link to={`/`} >Home</Link>
      <Link to={`about`} >About</Link>
      <Link to={`contact`} >Contact-us</Link>
    </>
  );
}
```

3. Set up the layout in the **App component, including the **Header** component and the **Outlet** component for rendering **child routes**.**



```
import {Outlet} from 'react-router-dom'
import Header from './Components/Header/Header'

import { Outlet } from "react-router-dom";
import Header from "./Components/Header/Header";

export default function App() {
  return (
    <>
      <Header />
      <Outlet />
    </>
  );
}
```

NOTE: **<Outlet />** is the JSX element, part of React Router's declarative routing system. it is typically used as a placeholder where child routes can be rendered.

4. Configure the **routes** in the main entry file (**main.jsx** or **index.js**), defining the **paths** and corresponding **components** to render.

```
import React from "react";
import ReactDOM from "react-dom/client";
import { RouterProvider, createBrowserRouter } from "react-router-dom";
import App from "./App";
import About from "./Components/About/About";
import Contact from "./Components/Contact/Contact";
import Home from "./Components/Home/Home";

const router = createBrowserRouter(
  createRoutesFromElements(
    <Route path="/" element={<App />}>
      <Route path="" element={<Home />} />
      <Route path="about" element={<About />} />
      <Route path="contact" element={<Contact />} />
    </Route>
  )
);

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>
);
```

NOTE: Have **Home**, **About** and **Contact_Us** components ready. when we click on specific link on Header, it will redirect to respective component and respective content will be displayed.

2. Passing Query Parameters

Now, let's explore how to pass query parameters from one component and access them in another component using React Router.

Imagine we have an `About` component with a link that redirects to the `User` component, passing along the `userId` and `name` as query parameters.

1. In the `About` component, create a `NavLink` with the desired query parameters appended to the URL. (code in next slide)



```
import React, { useState } from "react";
import { NavLink } from "react-router-dom";

export default function About() {
  let [userId, setUserId] = useState(200);
  let name = "Razak";

  return (
    <>
      <h1>About</h1>
      <NavLink to={`user/?userId=${userId}&name=${name}`}> User </NavLink>
    </>
  );
}
```

2. Configure the route in the main entry file.




```
import React from "react";
```

```
import -----//Note: All relevant imports as their in above slide
```

```
const router = createBrowserRouter(  
  createRoutesFromElements(  
    <Route path="/" element={<App />} >  
      // all the Routes will also be there  
      <Route path="about/user" element={<User/>} />  
    </Route>  
  )  
);
```

```
ReactDOM.createRoot(document.getElementById("root")).render(  
  <React.StrictMode>  
    <RouterProvider router={router} />  
  </React.StrictMode>  
);
```

3. In the User component, use the **useLocation** hook from react-router-dom to access the query parameters and display them



```
import React from "react";
import { useLocation } from "react-router-dom";

export default function User() {
  const location = useLocation();

  let searchParams = new URLSearchParams(location.search);
  let userId = searchParams.get("userId");
  let name = searchParams.get("name");

  return (
    <>
      <h1>User</h1>
      <h4>User: {userId} </h4>
      <h4>Name: {name} </h4>
    </>
  );
}
```

3. Dynamic Route Paths with Parameterized/Dynamic Segments

Sometimes, you might need to create dynamic route paths where a part of the URL is dynamic or parameterized. React Router provides a convenient way to handle this scenario.

Let's consider an example where we want to pass a dynamic `userPlace` parameter (dynamic route path) from the `About` component to the `User` component.

(continue in next slide)

1. In the About component, create a NavLink with the dynamic segment (e.g., user/\${userPlace}).



```
import React, { useState } from "react";

import { NavLink } from "react-router-dom";

export default function About() {
  let [userPlace, setUserPlace] = useState('INDIA')

  return (
    <>
      <h1>About</h1>
      <NavLink to={`user/${userPlace}`}> User </NavLink>
    </>
  );
}
```

2. Configure the route with a dynamic segment in the main entry file (e.g., about/user/:userPlace).



```
import React from "react";
```

```
import -----//Note: All relevant imports as their in above slide
```

```
const router = createBrowserRouter(
```

```
  createRoutesFromElements(
```

```
    <Route path="/" element={<App />}>
```

```
      // all the Routes will also be there
```

```
      <Route path="about/user/:userPlace" element={<User />} />
```

```
    </Route>
```

```
  )
```

```
);
```

```
ReactDOM.createRoot(document.getElementById("root")).render(
```


```
  <React.StrictMode>
```

```
    <RouterProvider router={router} />
```

```
  </React.StrictMode>
```

```
);
```

3. In the User component, use the `useParams` hook from `react-router-dom` to access the dynamic parameter and display it.



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

export default function User() {
  let { userPlace } = useParams();

  return (
    <>
      <h1>User</h1>

      <h4>Location is : {userPlace}</h4>
    </>
  );
}
```

NOTE: Dynamic route path = example: here, `user/userId` , `profile/profileId` are Dynamic route path, where `userId` and `profileId` are dynamic.

Parameterized segment = Dynamic segment = `user/userId` , `profile/profileId` here `userId` and `profileId` are Parameterized segment or Dynamic segment.

And That's A Wrap!!

👉 Follow Me For More
of These In Your Feed.



Abdulrazak Attar

SDE @  MSG
g l o b a l



See ya!!