

UseEffect expects two things

Callback function and a dependency array.

First component is rendered and then useEffect is called

If any one of the state is getting updated, then it will render every time.

But if there is empty dependency, then useEffect is called on initial render.

If useEffect has no dependency, it will be called on every render.

We will see react-router

npm i react-router-dom

```
Chapter 04 - Talk is Cheap, show me the code > App.js > ...
24 import Body from "../Body";
25 import Footer from "../Footer";
26 import { createBrowserRouter, RouterProvider } from "react-router-dom/dist";
27 import About from "../Chapter 05 - Let's get Hooked!/About";
28 const AppLayout = () => (
29   <React.Fragment>
30     <Header />
31     <Body />
32     <Footer />
33   </React.Fragment>
34 );
35 const appRouter = createBrowserRouter([
36   { path: "/", element: <AppLayout /> }, // new
37   { path: "/about", element: <About /> },
38 ]);
39 const root = ReactDOM.createRoot(document.getElementById("root"));
40 console.log(root);
41 root.render(<RouterProvider router={appRouter} />);
42
```

useRouteError Hook:

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

const Error = () => {
  const err = useRouteError();
  console.log(err);
  return <div>Error</div>;
};

export default Error;
```

console.log(err) is giving this

```
{
```

```
"status": 404,  
"statusText": "Not Found",  
"internal": true,  
"data": "Error: No route matches URL \"/899\"",  
"error": {}  
}
```

Single page Application

Client side routing and server side routing

Server side routing is getting different routing from client side

But what we are doing in our react application is client side routing meaning that routing is happening via client side

Under the table, Link tag is anchor tag because browser can understand anchor tag

Lets Explore React -Router dom

```
import Error from "../chapter 07-finding the path/error";  
import Contact from "../Contact";  
import { Outlet } from "react-router-dom/dist";  
const AppLayout = () => (  
  <React.Fragment>  
    <Header />  
    <Outlet />  
    <Footer />  
  </React.Fragment>  
);  
const appRouter = createBrowserRouter([  
  {  
    path: "/",  
    element: <AppLayout />,  
    errorElement: <Error />,  
    children: [  
      { path: "/", element: <Body /> },  
      { path: "/about", element: <About /> },  
      { path: "/contact", element: <Contact /> },  
    ],  
  },  
], // You, 1 second ago • Uncommitted changes  
);  
const root = ReactDOM.createRoot(document.getElementById("root"));  
console.log(root);  
root.render(<RouterProvider router={appRouter} />);
```

In the above code snippet, header and footer will be same.

Only thing that will change is

<Outlet/> and that is going to change as per children.

All the children there will go to Outlet according to Route

This outlet is a named component which is doing routing for us

```
import React from "react";
import { useParams } from "react-router-dom";
const RestrauntMenu = () => {
  debugger;
  const param = useParams();
  return (
    <>
    | <h1>{param.id}</h1>
    </>
  );
};

export default RestrauntMenu;
```