

Episode 7 - Finding Your Path



useEffect is a hook which is called after the component is rendered

It takes a callback function & a dependency array.

$(() \rightarrow \{\})$ \Rightarrow If we don't have a dependency array it will be called everytime the component renders

$(() \rightarrow \{\}, [])$ \Rightarrow It will be called only once after initial render

$(() \rightarrow \{\}, [count])$ \Rightarrow It will be called after initial render once and will be called everytime count changes

** Never create a component inside a component. As it will be called so many times.

Eg

const AppLayout = () => {

 const Food = () => {

}

What should not be done?

1) Never use a hook inside if statement, for loop

Eg

```
if () {  
  const [value, setvalue] = useState()  
}
```

React gives you a hook which let's you create local state variable inside a component = useState()

2) Never use/create useState outside a function component

Can I use more than one useEffect?
Yes, you can use as many as you want depending on different usecase

Why is CDN a great place to store image?

Because CDN is faster

Use NPM Package Formik

CDN is faster, caches your images, return it very fast, has 100% uptime, optimize the app



React Router

```
import {createBrowserRouter, RouterProvider} from "react-router-dom"

const appRouter = createBrowserRouter([
  {
    path: "/",
    element: <AppLayout />
  },
  {
    path: "/",
    element: <About />
  }
]);

root.render(<RouterProvider router={appRouter} />)
```

React Router Dom also provides a better UI for errors in DOM.

→ Error element in Router \Rightarrow errorElement: <Error/>
which we write in createBrowserRouter objects

```
const appRouter = createBrowserRouter([
  {
    path: "/",
    element: <AppLayout />,
    errorElement: <Error/>
  }
]);
```

Configuration means some information which specifies what will happen in a specific route/path

→ useRouteError is a hook which React Router Dom provides to read & get the information about the type of error we are facing. It can catch error & display it.

```
import {useRouteError} from 'react-router-dom'
```

```
const Error = () => {
  const err = useRouteError();
  <h2> {err.status + ":" + err.statusText} </h2>
```

Should you use <a> tag or <link> tag to navigate to a different page?
We should use <link> tag provided by React Router Dom as <a> will reload the site as we are creating a Single Page Application.

Router provider will provide the routing configuration to our App

<Link> tag is <a> tag in the end so that browser can understand it.

Creating children in createBrowserRoute

```
const AppLayout = () => {
  return (
    <>
      <Header />
      <About /> // if path is /about
      <Body /> // if path is /
      <Contact /> // if path is /contact
      <Footer />
    </>
  )
}
```

This won't work as it will load everything together so we are going to use Outlet

```
const appRoute = createBrowserRouter([
  {
    path: "/",
    element: <AppLayout />,
    errorElement: <Error />,
    children: [
      {
        path: "/about",
        element: <About />,
      }
    ]
})
```

Types of Routing

- 1) Client side Routing → All the page info are already there in the app so there is no need of network
- 2) Server side Routing → All the pages while loading comes from server with network call and it reloads the web pages



Outlet → According to the route, all the children will go to Outlet.
It is a named export component

```
const AppLayout = () => {
  return (
    <>
      <Header />
      <Outlet />
      <Footer />
    </>
  )
}
```

```
const appRouter = createBrowserRouter([
  {
    path: "/",
    element: <AppLayout />,
    errorElement: <Error />,
    children: [
      {
        path: "/",
        element: <Body />,
      },
      {
        path: "/about",
        element: <About />,
      },
      {
        path: "/contact",
        element: <Contact />,
      },
    ],
  },
])
```



Dynamic Routing

When there is a dynamic link the router should take us to that page according to Id

Eg {

 path: "/restaurant/:id", → : means can be changed
 This will come in useParams
 element: <Restaurant />

useParams → It is a hook provided by React Router Dom which gives us Id

How to read a dynamic URL params?

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

→ Object.values convert objects into array of value

```
<Link to={`/restaurant/${restaurant.data.id}}>  
</Link>
```