Ep 7: Finding the path

- Routing

- useEffect is called after every Render of the Component
- But as we have used dependency array [ ], it will change the behaviour of the useEffect.

useEffect:-

// if no dependency array => useEffect is called every Render

// if empty dependgency array => use Effect is called on only Initial Render & just once when Component is Render first time.

// if we pass Something inside dependency array => Then it will only be called when the dependency changes.

use State:-

- Never create useState outside the Component.
  If used it will throw Er: Hook can only be called inside body of Function Component.
- It is used to create local state Variable inside functional Component.
- Try to call on the top, where the fn starts
- Never create UseState Hook inside (if else) - Don't do This. It will create inconsistency in code
- Never Create State Variable Inside (loop) (Conditions) (fn)

React-Router-dom

"npm i react-router-dom" ← install

- For Creating Routes
  ↳ first create Routing-configuration
  ↳ Import {createBrowserRouter} from "react-router-dom"

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

→ on /about path URL element [Component <About/>] will render.

Routerprovider :- will always provide routing configuration to our App.

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

Earlier we were Rendering only Applayout → Now, we will provide Router Configuration which we created using createBrowserRouter to our <App layout/> component using <Routerprovider router = {appRouter}/>

React Router Dom gives access to Special Hook called useRouteError

import { useRouteError } from "react-router-dom".

- Using this hook it gives us better Error handling
- It enables us to define more about Error. Better Error Message

```
<h2> {err.status} : {err.statusText}
</h2>
```

```
const err = useRouteError();
```

Children in Routing (Nested Routes).

```
{ path: "/",
  element: <Applayout/>,
  Children: [
     {
      path: "/about",
      element: <About/>,
     },
     {
      path: "/contact",
      element: <Contact/>,
     },
  ],
  errorElement: <Error/>,
},
```

<outlet/> is a Component
↓
This <outlet/> component will be filled with the children according to the path

While routing to the URL, this Outlet component is Replaced with the Children component According to path behind the Scenes

When you are using React and you want to route to some other page, never use an anchor tag <a href="/"></a>
Because anchor tag will ~~Replace~~ Reload the whole Component.

But,

in React you can Navigate to the page without Reloading the whole page. using <Link/> component. Works exactly Same as anchor tag.

→ it Comes from "react-router-dom"

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

<Link to="/contact"></Link>

Hence React is called Single Page Application

Two Types of Routing in Web Apps

1> Client Side Routing   2> Server Side Routing

---

useParams - React Hook

(import { useParams} from 'react-router-dom';)

is used to get Params from the URL

Params is object

Behind the Scenes <Link></Link> is using anoher tag <a></a>