

Episode 7 - Finding The Path

Dependancy Array (useEffect) -

How useEffect() is called?

Imp: Dependancy array is not mandatory in useEffect().

Case 1: useEffect(`&`) $\Rightarrow \{$

`console.log ("useEffect called");`
`};`

When you call a useEffect without a dependancy array, It will called every time component renders.

If no dependancy array \Rightarrow useEffect is called on every render

Case 2: useEffect(`<`) $\Rightarrow \{$

`console.log ("useEffect called");`

`, []);`

\hookrightarrow empty dependancy array

If dependancy array is empty \Rightarrow useEffect is called on only Initial Render (just once).

Case 3 : `useEffect(() => {
 console.log ("useEffect called");
 let [btnName] = useState("Initial Value");
 // putting btnName state
 // variable`

It is only being called when the dependency changes and initial render

→ Additional things about `useState()`

- 1) Never use or create state variable outside state variable the component
- 2) It is used to create Local state variable inside component
- 3) Always try to create local state variable on the top of component
- 4) Never create state variables inside if - else

→ React Router

`npm i react-router-dom`

Creating different routes for different pages.

O To create Routing go to app.js

Whenever we have to create routes, firstly create Routing configuration.

- Routing Configuration

- {Create Browser Router} from 'react-router-dom'
- create routing configuration.
- Configuration means that will define what will happen on a specific route.
- createBrowserRouter takes a list of path or url.

- path it is an object and it contains 2 things

1. path: "/about",
element: <About/>

- e.g const appRouter = [
{
path: "/",
element: <AppLayout/>
},
]

- path: "/contact",
element: <Contact/>

-];

e) Passing configuration to render on to the page

- Import `RouterProvider` from 'react-router-dom';
- It will provide the routing configuration to our app.
- eg : `root.render(<RouterProvider router={appRouter} />);`

→ **rafce**

- shortcut to create component in vs code.
- Error: "Unexpected Application Error"
404 not found
- If we provide not defined or not created path/route in our app, it will throw an error
- e.g.: "localhost:1234/akshay"
- This page will created by "react-router-dom" to show error in proper way.
- creating own "Error Page".
- Created using `errorElement` in path object

- e.g const appRouter = createBrowserRouter([
 { path: '/',
 element: <AppLayout>,
 errorElement: <Error/>
 },
]);

- 'react-router-dom' gives us an access to one important hook ie useRouteError

→ useRouteError

- a hook always starts with word 'use' before it.
- useRouteError is hook given by react-router-dom
- A function starting with use is a hook
- Used to display more information about error
- ```
const err = useRouteError();
console.log(err);
```
- Returns an object which contains extra information about the error
- Useful for the Error handling.

## → Children Routes

```
- const appRouter = createBrowserRouter(CC
 {
 path: "/",
 element: <AppLayout1>,
 children: [
 {
 path: "/about",
 element: <About1>,
 },
 {
 path: "/contact",
 element: <Contact1>
 },
],
 errorElement: <Error1>,
 },
);
```

- Rendering children according routes
- 'React-router-dom' is provide {outlet}
- Outlet is a component to render children routes
- ```
const AppLayout = () => {
  return (
    <div>
      <Header>
      <Outlet/>
    </div>
  );
};
```

- Linking children routes to nav items links
- Using Link component from 'react-router-dom'
- Never use `` in react, because it reloads the page and poor user experience
- `<Link to = {"/"}> Home </Link>`

→ Single Page Application (SPA)

- single page application means there is only one page (app) and others are components (Home, about, contact) that are interchanging without making any external call & reloading website
- Every thing in react is a components
- 2 types of routing in React web apps

1) Client side routing

2) Server side routing

Here we developed SPA via client side routing.

Server side routing where, code for about, contact page comes from server by making an call for backend.

- Creating Dynamic Routing for different Restaurants

App.js {

```
path: "/restaurant/:resId",
element: < RestaurantMenu >
}
```

RestaurantMenu.js

- useParams

A hook given by 'react-router-dom' for dynamic routing

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

- fetchMenu

```
const fetchMenu = async () => {
  const data = await fetch(MENU-API+
    resId);
```

```
  const json = await data.json();
  setResInfo(json.data);
}
```