

7 . Finding The Path

useEffect

How useEffect called ?

```
import { LOGO_URL } from "../utils/constants";
import { useState, useEffect } from "react";

const Header =() =>
{
    const [btnName, setbtnName]=useState("Login ")

    useEffect((()=> {}),[]);
```

Every time when component render useEffect called

Dependency array changing the behaviour of it's render

In the useEffect dependency array is not mandatory only callback function is mandatory

```
//if no dependancy array => useEffect is called on every render
//if dependancy array is empty = [] => then useEffect called on only initial render (just once)
// if dependancy arrat is [btnName] => called every time btnName is updated
useEffect(()=> {
    console.log("useEffect called");
},[btnName]);
```

Dependency array Not mandatory field

If Don't put - render every Time when component render

If Empty - called on only initial render(just once when component render first time)

If put something - only called when dependency changes

Eg . if our dependency is on btnName so every time btnName changes our useEffect will called

useState

Best practises:

Never create the useState outside the component

Always call the useState on the top

Never use our useState hook or create your state variable inside the if else

Never create the useState variable inside the if else statement or inside the condition or inside the normal function

React-router DOM

Installing React-router-DOM command :

```
npm i react-router-dom
```

1. Import createBrowserRouter

```
6. Exploring the World > JS App.js > AppLayout
1  import React from "react";
2  import ReactDOM from "react-dom/client";
3  import Header from "../src/Components/Header";
4  import Body from "../src/Components/Body";
5  | import {createBrowserRouter} from "react-router-dom";
6  _____
```

2. Routing Configuration

Import browser router, it's take list of path

Path is nothing but object and it contains two things

1. Path
2. element

```
const AppLayout = () => {
  return (
    <div className="app" >
      <Header/>
      <Body/>
    </div>
  );
};

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

We don't have about us page so just create
Page means a big component and it has collection of components

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

RouterProvider is provides to this routing configuration to our app

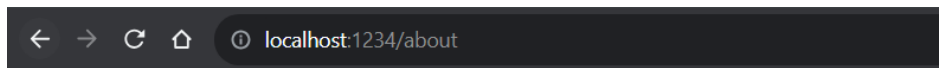
Earlier we rendering our AppLayout directly

```
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<AppLayout/>)
```

Now instead of this we will write

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

So after putting that path it's working

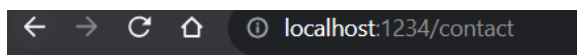


About Us

Food Ordering Application by Nikita!

```
U JS About.js U x JS Body.js U
The Path > src > Components > JS About.js > [e] About
const About = () =>
{
  return(
    <div>
      <h1>About Us</h1>
      <h3>Food Ordering Application by Nikita!</h3>
    </div>
  );
};

export default About;
```

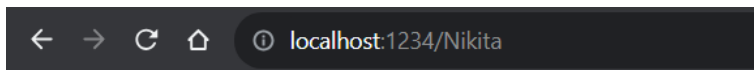


Contact Us

```
const Contact = () =>
{
  return(
    <div>
      <h1>Contact Us</h1>
    </div>
  )
}

export default Contact;
```

/any path



Opps!!!

Something Went Wrong!!

```
const Error = () =>
{
  return(
    <div>
      <h1>Opps!!!</h1>
      <h2>Something Went Wrong!!</h2>
    </div>
  )
}

export default Error;
```

Import all components in app

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

React -router-dom gives the access of hook is `useRouteError`

Whenever we see a Function starting with `use` that means it's hook

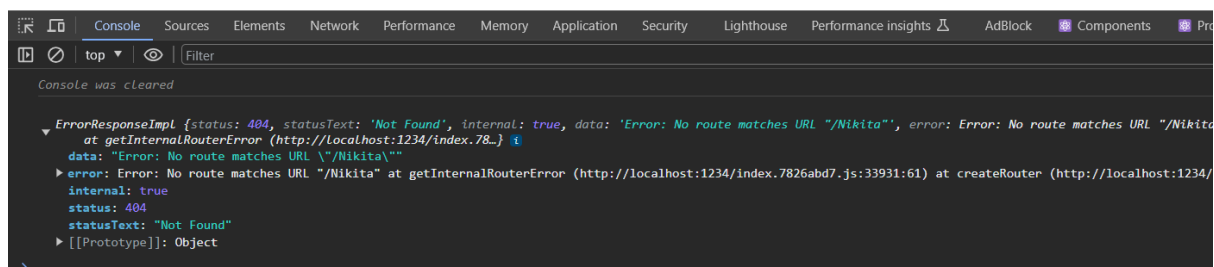
Using the `useRouteError` giving the more information of error



Opps!!!

Something Went Wrong!!

404 : Not Found



```
import {useRouteError } from "react-router-dom";
const Error = () =>
{
  const errr= useRouteError();
  console.log(errr);  // printing error object on console

  return(
    <div>
      <h1>Opps!!!</h1>
      <h2>Something Went Wrong!!</h2>
      <h3>{errr.status} : {errr.statusText} </h3>
    </div>
  )
}

export default Error;
```

Revision of error

If we have Error component
give the

1. Path
2. errorElement

If this path has error it will catch it and show on the page

Putting Configuration inside the children

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

Whenever there is a change in the path so this outlet will be filled with the children according to the path

We trying to build this

```
{/* if path="/" */}  
<Body/>  
{/* if path="/about" */}  
<About/>  
{/* if path =/contact */}  
<Contact/>
```

So when '/' then body will be filled in the outlet ,
When '/about' then About will filled in outlet and similarly
Contact also filled in the outlet

app.js

```
import React from "react";  
import ReactDOM from "react-dom/client";  
import Header from "../src/Components/Header";  
import Body from "../src/Components/Body";  
import About from "../src/Components/About";  
import Contact from "../src/Components/Contact";  
import Error from "../src/Components/Error";  
import {createBrowserRouter, RouterProvider, Outlet} from "react-router-dom";  
  
const AppLayout = () => {  
  return (  
    <div className="app" >  
      <Header/>  
      <Outlet/>  
    </div>  
  );  
};
```

Outlet is filled with the component that is passed as children into parent
(appLayout)

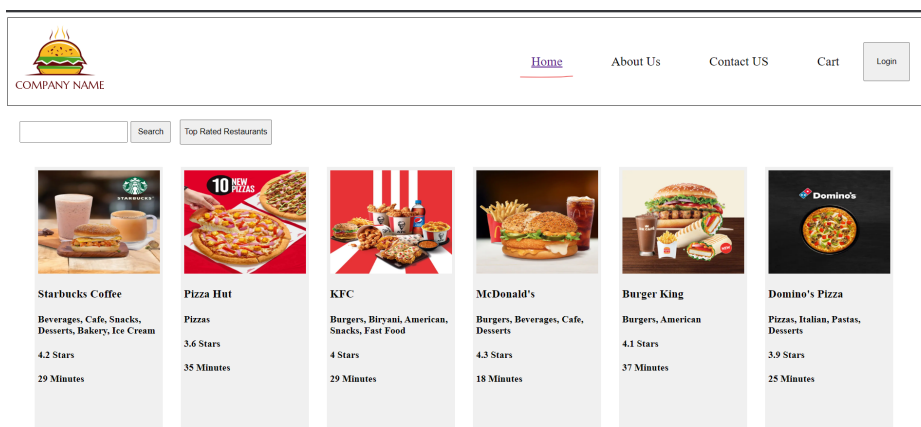
Outlet will be replaced by Another component according to the path

Header.js

```
<div className="header">
  <div className="logo-container">
    <img className="logo"
      src={LOGO_URL} alt="" />
  </div>

  <div className="nav-items">
    <ul>
      <li>
        <a href="/">Home</a>
      </li>
      <li>About Us</li>
      <li>Contact US</li>
      <li>Cart</li>
      <button className="login" onClick={()=>
    {
```

Never use anchor tag here because



When we clicked on that page whole page will get refreshed

There is magic that we do in react

We don't want to refresh our whole page but still we move to new route

We can navigate to the new page without reloading the whole page

We can navigate to different page without reloading the whole page

So for that instead of

``

We used

`<Link to="">`

Header.js

```
import { LOGO_URL } from "../utils/constants";
import { useState } from "react";
import { Link } from "react-router-dom";
```

```
<div className="nav-items">
  <ul>
    <li>
      <Link to="/">Home</Link>
    </li>
    <li>
      <Link to="/about">About Us</Link>
    </li>
    <li>
      <Link to="/contact">Contact Us</Link>
    </li>
  </ul>
</div>
```

Link component work exactly same as anchor tag but our whole page will not refreshed

It's does not reload the whole page it's just fetches the data or changing the components

That's why our application are known as single-page-Application

There are two types of Routing in Web Applications

- 1.Client side Routing
- 2.Server side routing

Server Side Routing

We have index.html and all html css Javascript files

And when we click on anchor tag it will reload whole page

When we click on anchor tag of html file Its make network call , fetches that html and render on the page , that is server side Routing

Means we make network call and that page suppose about.html) coming from server that is server side Routing

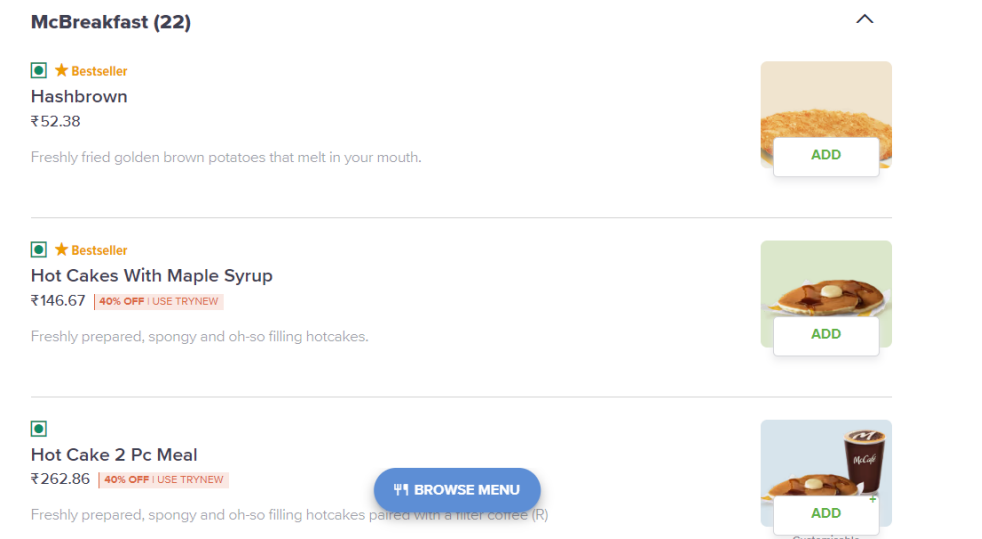
Client side Routing

Using react in Link tag is Client side Routing we are not making any network call Because all the components are already loaded into our app , when we load the app for the first time it already has the code for About Us , when we go to about Us page it's just loads that components it's not making network call

It's only network call made when we Fetch the API , we not fetching new page so that's why it's known as Client side Routing

Dynamic Routing

We trying to making like this page

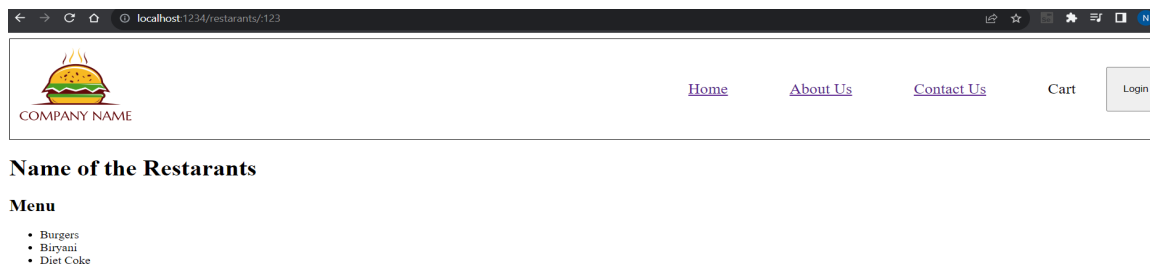


Creating RestarantMenu Component

creating/adding one more route

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

Here we want this id should dynamic
According to the restaurants it should be changed



Here instead of menu we want Actual restaurant data for that need to fetch from API

Fetch the data

Store the data - useState [so whenever state variable update it will automatically update the UI]

Fetch the data

```
useEffect(()=>
{
  fetchMenu();
},[])

const fetchMenu= async()=>{
  const data= await fetch(
    "https://www.swiggy.com/dapi/menu/pl?page-type=REGULAR_M";
  );
  const json = await data.json();
  console.log(json);
};
```

Store the data - useState

```
const [resInfo, setResInfo]=useState(null);
```

Showing Shimmer UI

```
return resInfo===null ? <Shimmer/> : (
  <div className="Menu">
    <h1>Name of the Restarants</h1>
    <h2>Menu</h2>
    <ul>
      <li>Burgers</li>
      <li>Birryani</li>
      <li>Diet Coke</li>
    </ul>
  </div>
)
```

Showing data on UI

```
};

return resInfo===null ? <Shimmer/> : (
  <div className="Menu">
    <h1>{resInfo?.cards[0]?.card?.card?.info?.name}</h1>
    <h2>Menu</h2>
    <ul>
      <li>Rice</li>
      <li>Biriyani</li>
      <li>Diet Coke</li>
    </ul>
  </div>
)
```

Need to destructure this

Destructuring

```
const {name,cuisines,costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info
```

Problem:

Initially resinfo is null

```
const [resInfo, setResInfo]=useState(null);
```

And we finding name inside it so it's giving error

So getting this error

```
TypeError: Cannot destructure property 'name' of '(intermediate value)' as it is undefined.

RestaurantMenu
src/Components/RestaurantMenu.js:30:18

27 |
28 |
29 |
> 30 |     const { name, cuisines, costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info;
    |           ^
31 |
32 |     if(resInfo===null) return <Shimmer/>;
33 |     return(
View compiled
```

Solution :

```
const { name, cuisines, costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info;

if(resInfo===null) return <Shimmer/>;
return(

  <div className="Menu">
    <h1>{name}</h1>
    <p>{cuisines.join(", ")} - {costForTwoMessage}</p>

    <ul>
      <li>Rice</li>
      <li>Biryani</li>
      <li>Diet Coke</li>
    </ul>
  </div>
)
```

Paste this line on top like this :

```
if(resInfo===null) return <Shimmer/>;

const { name, cuisines, costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info;

return(

  <div className="Menu">
    <h1>{name}</h1>
    <p>{cuisines.join(", ")} - {costForTwoMessage}</p>

    <ul>
      <li>Rice</li>
      <li>Biryani</li>
      <li>Diet Coke</li>
    </ul>
  </div>
)
```

For dynamically showing items on GMA



McDonald's

Burgers, Beverages - ₹400 for two

- Hashbrown
- Hot Cakes With Maple Syrup
- Hot Cake 2 Pc Meal

```

if(resInfo===null) return <Shimmer/>;

const { name, cuisines, costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info;
const {itemCards}=resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards[1]?.card?.card;
//console.log(itemCards);

return(
  <div className="Menu">
    <h1>{name}</h1>
    <p>{cuisines.join(", ")} - {costForTwoMessage}</p>
    <ul>
      <li>{itemCards[0].card.info.name}</li>
      <li>{itemCards[1].card.info.name}</li>
      <li>{itemCards[2].card.info.name}</li>
    </ul>
  </div>
)

```

Here iterating from itemCards we can use map function

```

if(resInfo===null) return <Shimmer/>;

const { name, cuisines, costForTwoMessage } = resInfo?.cards[0]?.card?.card?.info;
const {itemCards}=resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards[1]?.card?.card;
//console.log(itemCards);

return(
  <div className="Menu">
    <h1>{name}</h1>
    <p>{cuisines.join(", ")} - {costForTwoMessage}</p>
    <ul>
      {itemCards.map((item)=>
        <li>{item.card.info.name}</li>)}
    </ul>
  </div>
)

```

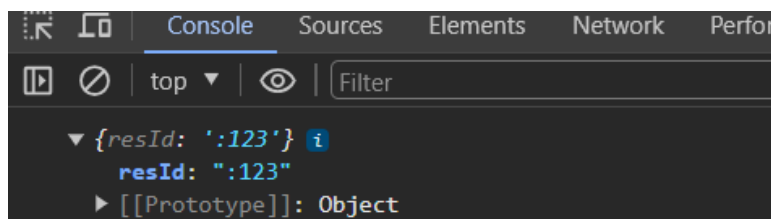
```

<ul >
  {itemCards.map((item)=>
    <li key={item.card.info.id}>{item.card.info.name} - {"Rs."} {item.card.info.price/100} </li>)}
</ul>

```

Here in api url only difference in restaurant ID

So we need to read (use) resID so for that we use **useParams** Hook



So params is an object with this resID

So instead of param we can write resID (destructuring)


```
import {useState, useEffect } from "react";
import Shimmer from "../Shimmer";
import {useParams} from "react-router-dom";

const RestarantMenu=()=>
{
  const [resInfo, setResInfo]=useState(null);
  const params=useParams();
  console.log(params);

  useEffect(()=>
  {
    fetchMenu();
  },[])
}
```

Pass API url instead of `restaurantId` number pass `resId`

Before

```
72.8296252&restaurantId=23663&catalog_qa=undefined&submitAction=ENTER"
```

After

```
16&lng=72.8296252&restaurantId="+resId+"&catalog_qa=undefined&submitAction=ENTER"
```

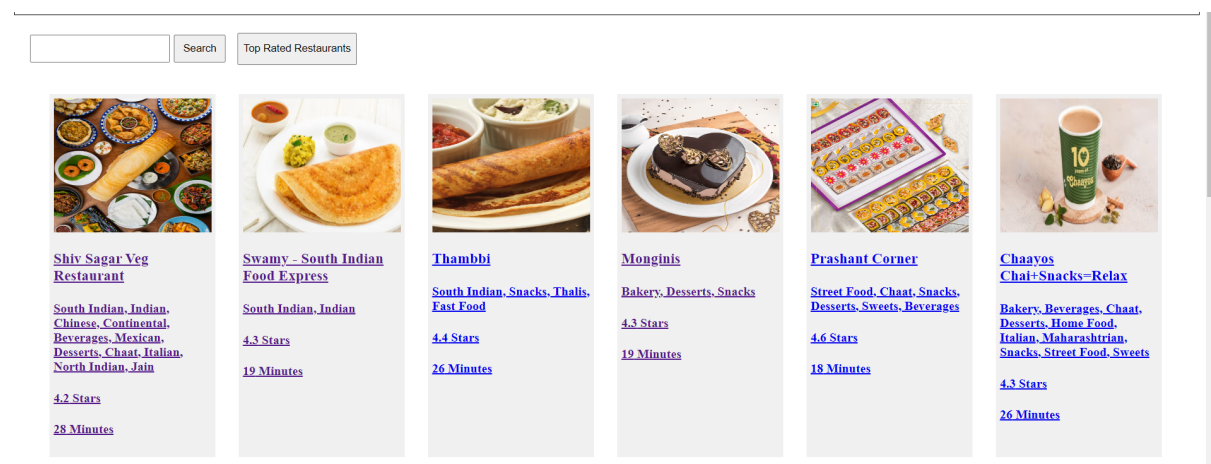
Link is a component which is giving to us by react router dom which is special component and behind the scenes Link is using anchor tag

```
("https://www.swiggy.com/dapi/menu/pl?page-type=REGULAR_MENU&complete-menu=true&lat=19.2183307&lng=72.9780897&restaurantId=221810&catalog_qa=undefined&submitAction=ENTER"
```

We need to Link all restaurant so put Link tag

```
JS App.js U JS RestarantMenu.js U JS Body.js U X JS constants.js U JS Header.js
7.Finding The Path > src > Components > JS Body.js > Body > FilteredRestarants.map() callback
1 import RestaurantCard from "../RestaurantCard";
2 import {useEffect, useState} from "react";
3 import Shimmer from "../Shimmer";
4 import { Link } from "react-router-dom";
5
```

```
{
  FilteredRestarants.map((restaurant)=>
<Link
  key={restaurant.info.id}
  to={"/restaurants/" + restaurant.info.id}>
    <RestaurantCard resData={restaurant} />
  </Link>
)
}
```



Now able to go particular restaurant menu page