

How to redirect to another page in ReactJS

To redirect either between 1 page or multiple pages , React-Router is a useful library for handling navigation and redirecting between pages. And useNavigate hook and Route tag of React-Router-dom library are commonly used for Routing.purposes.

Note: React-router-native -> is used for mobile applications

React-router-DOM -> is used for web app.

In this article , we will focus on React-Router-Dom i.e routing in web application.

Real-World example of Routing

Netflix is a popular streaming platform that offers a wide variety of content across different genres like drama, comedy, action, anime, family, crime, and many more. Each genre represents a category with its own collection of movies and TV shows.

When we visit Netflix, we see a homepage with different categories displayed. When we click on any category, it redirects us to a new page specifically dedicated to that category. This allows us to browse and explore the available options within that particular category. For this purpose , **Routing** is an efficient way to handle navigation in a multi-page application like Netflix

For example, when we click on the "Action" category, the application navigates to a new URL like "netflix.com/action".

So, how can we redirect to another page in ReactJS?

Let's see an example to understand it clearly.

Below I have mentioned 3 files : index.js , app.js , ContactUs.js

File 1 : index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(

  <BrowserRouter>
    <App/>
  </BrowserRouter>
);
```

This code snippet is the entry point for a React application

The **<BrowserRouter>** component is a part of the react-router-dom library. It provides the functionality for routing within the application. By wrapping the **<App/>** component with **<BrowserRouter>** , the **<App/>** component becomes ready for handling routing

File 2 App.js

```
import { NavLink, Route } from 'react-router-dom';

function App() {
  return (
    <div>

      <NavLink to="/About" > Contact us </NavLink>
      <Route index element={<Home/>} /> <Route/>
      <Route path="/About" element={<About/>} />

    </div>
  );
}
```

In this file we will handle routing and navigation within a web application.

The **<NavLink>** tag is used to create a navigation link. It renders a clickable link that, when clicked, navigates to the specified route. In this case, the link has "to = "/About" ", which means it will navigate to the "/About" route/path when clicked.

The **<Route>** tag is used to define routes and associate them with specific components to render.

- 1st Route -> The index attribute specifies that it should be rendered when the application's URL matches the root ("/") path. It uses the "element=<Home/>" component to specify the component to render, which in this case is the <Home/> component
- 2nd Route-> It has (to="/About") as path and element="ContactUs" as component to render when this route is accessed

When the user clicks on the "Contact us" link, it will navigate to the "/About" path and render the <ContactUs/> component. The <Home/> component will be rendered by default when the root path ("/") is accessed.

File 3 ContactUs.js

```
import React from 'react'
import { Navigate, useNavigate } from 'react-router-dom'

const ContactUS = () => {
  const navigate = useNavigate();
  function clickHandler() {
    navigate("/About")
  }
  return (
    <div>
      <div>This is Contact us Page</div>
      <button onClick={clickHandler}>Move to ContactUs page</button>
    </div>
  )
}
export default ContactUS
```

Inside the **ContactUs component**, there is a **clickHandler function** that is called when a button is clicked. This function uses the **navigate function**, which is provided by the **useNavigate hook** from the react-router-dom library.

The navigate function is responsible for changing the current page and navigating to a new location. In this case, when the button is clicked, it will navigate to the "/About" page.

Routing for multiple pages

Above we saw a typical single routing example and it's working whenever the user click on a button in react app to move to a new URL.

Similarly we can use routes for multi-page(or component) application and can fmove between different components on clicking the required button in react app.

Note : For multiple route application ,all route should be enclosed inside **Routes tag**

The **<Routes> tag** is used to define a collection of routes. It serves as the top-level container for all the routes in the application.

Here is an example for defining multiple Route's inside Routes tag.

```
<Routes>
  <Route index element={<Home/>} />
  <Route path="/Support" element={ <Support/>} />
  <Route path="/About" element={<About/>} />
  <Route path="/Labs" element={<Labs/>} />
</Route>
</Routes>
```

Overall, the above code snippet defines the routing structure for different paths in the application and specifies the components to render for each path.

Now , we have a clear overview of “how to redirect to another page” in React app using React-Router-DOM”.