# Nested Routes

# Outlet concept

1. **Example Structure**

Let’s create an example where we have a Contact page that contains primary content and secondary content sections. Here’s how to do it step-by-step:

1. **1. Component Structure**

Your components might be structured as follows:

css

Copy code

src/

├── components/

│ ├── Home.jsx

│ ├── About.jsx

│ ├── Contact.jsx

│ ├── ContactMain.jsx

│ ├── ContactSecondary.jsx

│ └── Nav.jsx

├── App.jsx

└── NavRouter.jsx

1. **2. Create Components**

Here’s how your components might look:

**Home.jsx** and **About.jsx** can be simple for this example:

jsx

Copy code

// components/Home.jsx

import React from 'react';

const Home = () => {

return <h1>Home Page</h1>;

};

export default Home;

// components/About.jsx

import React from 'react';

const About = () => {

return <h1>About Page</h1>;

};

export default About;

Now, let’s create the Contact page with primary and secondary content.

jsx

Copy code

// components/Contact.jsx

import React from 'react';

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

const Contact = () => {

return (

<div>

<h1>Contact Page</h1>

<nav>

<NavLink to="main">Main Content</NavLink>

<NavLink to="secondary">Secondary Content</NavLink>

</nav>

<Outlet /> {/\* This is where nested routes will render \*/}

</div>

);

};

export default Contact;

And create secondary components:

jsx

Copy code

// components/ContactMain.jsx

import React from 'react';

const ContactMain = () => {

return <div>Main Contact Information</div>;

};

export default ContactMain;

// components/ContactSecondary.jsx

import React from 'react';

const ContactSecondary = () => {

return <div>Secondary Contact Information</div>;

};

export default ContactSecondary;

1. **3. Set Up the Router**

Now, let's set up the routes in your NavRouter:

jsx

Copy code

// components/NavRouter.jsx

import React from 'react';

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

import Home from './Home';

import About from './About';

import Contact from './Contact';

import ContactMain from './ContactMain';

import ContactSecondary from './ContactSecondary';

const NavRouter = () => {

return (

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about" element={<About />} />

<Route path="/contact" element={<Contact />}>

<Route path="main" element={<ContactMain />} />

<Route path="secondary" element={<ContactSecondary />} />

</Route>

</Routes>

);

};

export default NavRouter;

1. **4. Update the Navigation Links**

Ensure that your navigation includes links to each main route, including the Contact page:

jsx

Copy code

// components/Nav.jsx

import React from 'react';

import { NavLink } from 'react-router-dom';

const Nav = () => {

return (

<div className="fixed top-0 left-0 right-0 flex mx-3 gap-5 p-2 rounded-br-lg rounded-bl-lg bg-violet-500 bg-opacity-60 justify-center font-mono text-center">

<NavLink to="/">Home</NavLink>

<NavLink to="/about">About</NavLink>

<NavLink to="/contact">Contact</NavLink>

</div>

);

};

export default Nav;

1. **5. Update the App Component**

Finally, use the NavRouter in your App component:

jsx

Copy code

// App.jsx

import './App.css';

import Nav from './components/Nav';

import NavRouter from './components/NavRouter';

function App() {

return (

<div className="w-screen h-screen bg-fuchsia-700">

<Nav />

<NavRouter />

</div>

);

}

export default App;

1. **Summary**
2. **Create a Parent Component**: In this case, the Contact component acts as a parent for the nested routes.
3. **Use the <Outlet /> Component**: This is where the nested content will render.
4. **Define Child Routes**: Use the Route component to define paths like main and secondary within the Contact route.
5. **Navigation Links**: Add links within the Contact component to allow navigation to the main and secondary content.
6. **How It Works**

* When you navigate to /contact, the Contact component is rendered, showing the main contact page.
* Clicking on **Main Content** or **Secondary Content** in the Contact component will load either ContactMain or ContactSecondary without leaving the Contact page.

This approach provides a clear structure for nested routing while keeping your code organized.

# To keep a default nested route render once we hit parent route

Use navigate replace

**element={<Navigate to="main" replace />}**

import React from 'react'

import { Navigate, Route, Routes } from 'react-router-dom'

import About from './About'

import Contact from './Contact'

import ContactMain from './ContactMain'

import ContactSecondary from './ContactSecondary'

import Home from './Home'

**const** NavRouter = () **=>** {

*// const location=useLocation();*

  return (

    <div>

       <Routes >

        <Route path="/" element={<Home />} />

        <Route path="/about" element={<About />} />

**<Route path="/contact" element={<Contact />}>**

**{*/\* Redirect to the default nested route \*/*}**

**{*/\* use navigate to replace as soon we hit contact page one of the nested rout will be displayed by default \*/*}**

**<Route index element={<Navigate to="main" replace />} />**

**<Route path="main" element={<ContactMain/>} />**

**<Route path="secondarycontact" element={<ContactSecondary/>} />**

**</Route>**

    </Routes>

    </div>

  )

}

export default NavRouter

import React from 'react';

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

**const** Contact = () **=>** {

  return (

    <div className="flex items-center justify-center w-full h-screen bg-violet-900 pt-9 flex-col">

      <h1 className="font-extrabold text-purple-100 text-[3rem]">Contact Page</h1>

      <div className="flex gap-5 px-2 pt-1 mb-0 bg-violet-900 w-[20em] justify-evenly rounded-lg h-10 items-center text-yellow-50">

        <NavLink

          to="main"

          className={({ isActive }) **=>**

            `font-semibold pb-3 border-b-violet-950 rounded-t-lg p-1 ${isActive ? "bg-violet-950 font-semibold" : "bg-violet-700"}`

          }

        >

          main

        </NavLink>

        <NavLink

          to="secondarycontact"

          className={({ isActive }) **=>**

            `font-semibold rounded-t-lg p-1 pb-3  ${isActive ? "bg-violet-950 font-semibold" : "bg-violet-700"}`

          }

        >

          secondaryContact

        </NavLink>

      </div>

      {*/\* Render nested routes here \*/*}

      <Outlet />

    </div>

  );

};

export default Contact;