**Display route config in 2 spots: once for the sidebar and once in the main content section. All routes are in the same order they would appear in a <Switch>.**

import React from "react";

import {

BrowserRouter as Router,

Switch,

Route,

Link

} from "react-router-dom";

// Each logical "route" has two components, one for

// the sidebar and one for the main area. We want to

// render both of them in different places when the

// path matches the current URL.

// We are going to use this route config in 2

// spots: once for the sidebar and once in the main

// content section. All routes are in the same

// order they would appear in a <Switch>.

const routes = [

{

path: "/",

exact: true,

sidebar: () => <div>home!</div>,

main: () => <h2>Home</h2>

},

{

path: "/bubblegum",

sidebar: () => <div>bubblegum!</div>,

main: () => <h2>Bubblegum</h2>

},

{

path: "/shoelaces",

sidebar: () => <div>shoelaces!</div>,

main: () => <h2>Shoelaces</h2>

}

];

export default function SidebarExample() {

return (

<Router>

<div style={{ display: "flex" }}>

<div

style={{

padding: "10px",

width: "40%",

background: "#f0f0f0"

}}

>

<ul style={{ listStyleType: "none", padding: 0 }}>

<li>

<Link to="/">Home</Link>

</li>

<li>

<Link to="/bubblegum">Bubblegum</Link>

</li>

<li>

<Link to="/shoelaces">Shoelaces</Link>

</li>

</ul>

<Switch>

{routes.map((route, index) => (

// You can render a <Route> in as many places

// as you want in your app. It will render along

// with any other <Route>s that also match the URL.

// So, a sidebar or breadcrumbs or anything else

// that requires you to render multiple things

// in multiple places at the same URL is nothing

// more than multiple <Route>s.

<Route

key={index}

path={route.path}

exact={route.exact}

children={<route.sidebar />}

/>

))}

</Switch>

</div>

<div style={{ flex: 1, padding: "10px" }}>

<Switch>

{routes.map((route, index) => (

// Render more <Route>s with the same paths as

// above, but different components this time.

<Route

key={index}

path={route.path}

exact={route.exact}

children={<route.main />}

/>

))}

</Switch>

</div>

</div>

</Router>

);

}

Output:

Chart

Description automatically generated