Step 3: React router

# Introduction:

Now that we have 2 components in our project, let’s add something called **React router** to navigate our page a bit better.

What is React Router?

**React Router** is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and **keeps the UI in sync with the URL**. We will use the router to **render** (show) us a **one component at a time**.

# Installing React router:

Before we do anything else, we have to **install React Router to our project** as its not included in the create react app. Go to your project folder and open **cmd in the root**. Input the **following command** to install the router.

>npm i -D react-router-dom

In this exercise we have to change our App.js file pretty drastically, we’ll have the router render our components in its own file and add the router to our App.js to showcase a single component at a time.

# Router.jsx:

Let’s star by creating a new file in our pages folder, called **Router.jsx**. At the start of the file, import React and the router using the following statements.

import { Outlet, Link } from "react-router-dom";

import React from "react";

Create a base to our Router component:

export const Router = () => {

  return (

    <div>

    </div>

  )

};

We’ll create our router structure inside the **return tags**. The router structure has **<Outlet>** and **<Link>** elements. The <Outlet> renders the current route selected. The router will use a standard HTML list with <ul> and <li> tags. <Link> is used to **set the URL**. Anytime we link to an internal path, we will use <Link> instead of the standard HTML link: <a href="">.

import { Outlet, Link } from "react-router-dom";

export const Router = () => {

  return (

    <>

        <div>

            <ul>

                <li>

                </li>

                <li>

                </li>

            </ul>

        </div>

        <Outlet />

    </>

  )

};

Create a structure like above and start with adding a link to our Message component, making that our “**Homepage**”. We will do that by using the <Link> tags, like shown below:

<li>

    <Link to="/">Home </Link>

</li>

Next, we’ll add a link to our Button component.

<li>

    <Link to="/buttons">Buttons</Link>

</li>

**Anytime that we make a new component, we have to add it to this file, inside the list.**

# App.js:

Now let’s move to App.js and finish our router there. Start by importing **BrowserRouter**, **Routes** and **Route** using the following statement:

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

And our Router component.

import { Router } from "./pages/Router";

Now we will erase everything inside the App function return tags, you’ll be left with something like this:

import React from "react";

import { Message } from "./pages/Message";

import { Buttons } from "./pages/Buttons";

function App() {

    return (

    );

}

export default App;

Now we’ll start building our router, start by adding a <BrowserRouter> tags and a div element:

return (

    <BrowserRouter>

        <div>

        </div>

    </BrowserRouter>

);

Inside the div, add <Routes> tags:

<Routes>

</Routes>

The first route is our **parent route** which renders the Router component. Every other component will be placed **inside the parent routes** tags.

<Route path='/' element={<Router />}>

</Route>

Now to add our “Homepage”, we will set it as our first component that we created (Message).

<Route path='/' element={<Router />}>

<Route index element={<Message />} />

</Route>

Aside from the parent route and homepage (the only one with index attribute), **every component will be added using the same method**, like the following button route:

<Route path='/' element={<Router />}>

<Route index element={<Message />} />

<Route path='Buttons' element={<Buttons />} />

</Route>

# Browser:

In your browser the view should now look like this:

Graphical user interface, text, application, chat or text message

Description automatically generated

Just click on the component link you want to display:

Graphical user interface, text, application

Description automatically generated

This will be very valuable to us in the future, making the page more clean and pleasant to view.

Our Router.jsx and App.js files should now look like this:

Router.jsx

import { Outlet, Link } from "react-router-dom";

export const Router = () => {

  return (

    <>

        <div>

            <ul>

                <li>

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

                </li>

                <li>

                    <Link to="/buttons">Buttons</Link>

                </li>

            </ul>

        </div>

        <Outlet />

    </>

  )

};

App.js

import React from "react";

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

import { Router } from "./pages/Router";

import { Message } from "./pages/Message";

import { Buttons } from "./pages/Buttons";

function App() {

    return (

        <BrowserRouter>

            <div>

                <Routes>

                    <Route path='/' element={<Router />}>

                        <Route index element={<Message />} />

                        <Route path='Buttons' element={<Buttons />} />

                    </Route>

                </Routes>

            </div>

        </BrowserRouter>

    );

}

export default App;

This exercise was a bit more challenging, but now our project should have links at the top of the page, which we will be using to display **one component at a time**. This will make our page a lot more pleasant to view in the future when we’ll add more components.

In the next exercise we’ll learn about the useEffect hook. **See you there!** 😊