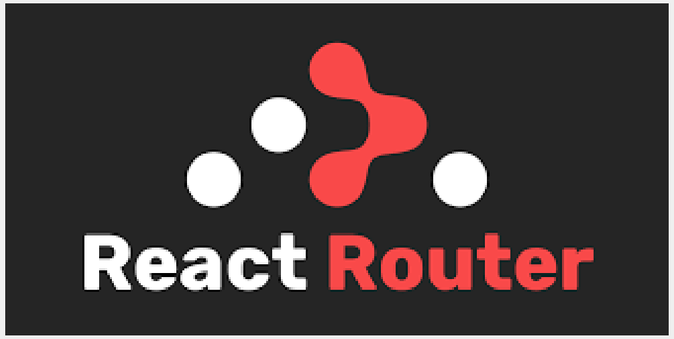
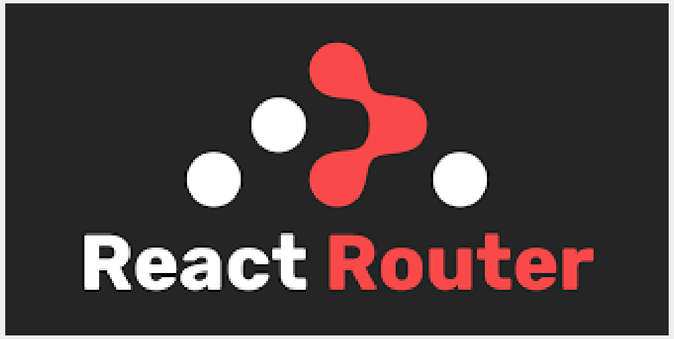
Outline



- Setup Route
- Dynamic Routing
- Nested Route
- Link Component
- NavLink Component
- Navigate Component
- useNavigate Hook
- A Quick look at the Document

What is react-router-dom



A **routing** library for React that provides a way to handle client-side routing within a single-page application.

- define routes for application
- map to different components
- dynamically rendered based on the URL

What is SPA?

SPA is short for **Single Page Application**

- Load single HTML page and dynamically updates its contents as the user interacts with the application without full page refresh.
- Desktop Application Experience
- Better User Experience

Why <a> is not enough?

Normally the <a> tag is used for referencing another html file and it triggers full page reload

Installation

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

Setup Route

Setup Route

main.jsx

src > main.jsx

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

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

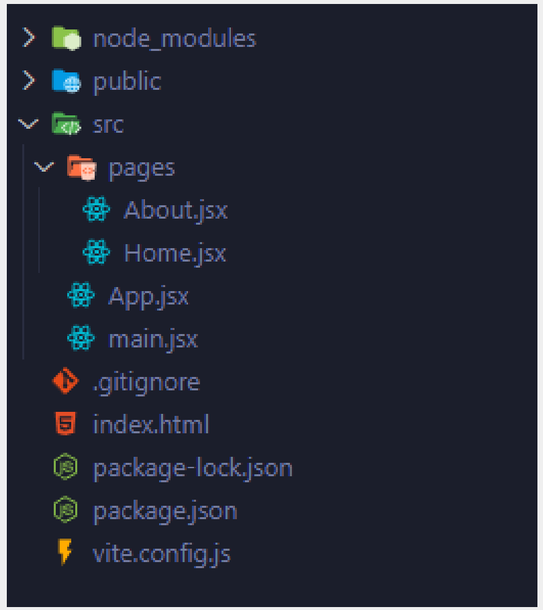
BrowserRouter is context component used for routing on browser.

It stores the current location in the browser's address bar using clean URLs and navigates using the **browser's built-in history stack**.

Original URL	Clean URL
`http://example.com/about.html`	`http://example.com/about`
`http://example.com/user.php?id=1`	`http://example.com/user/1`
`http://example.com/index.php?page=name`	`http://example.com/name`
`http://example.com/kb/index.php?cat=1&id=23`	`http://example.com/kb/1/23`

Demo files

Project Stucture



Home page

Home.jsx

src > pages > Home.jsx > ...

```jsx
import React from "react";

function Home() {
  return (
    <>
      <h1>Home page</h1>
    </>
  );
}

export default Home;
```

About page

About.jsx

src > pages > About.jsx > ...

```jsx
import React from "react";

function About() {
  return (
    <>
      <h1>About page</h1>
    </>
  );
}

export default About;
```

Setup Routes in App.jsx

App.jsx

src > App.jsx > ...

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

function App() {
  return (
    <>
      <Routes>
      </Routes>
    </>
  );
}

export default App;
```

Routes contains collection of routes for matching the URL

Declare route with Route

App.jsx

src > App.jsx > ...

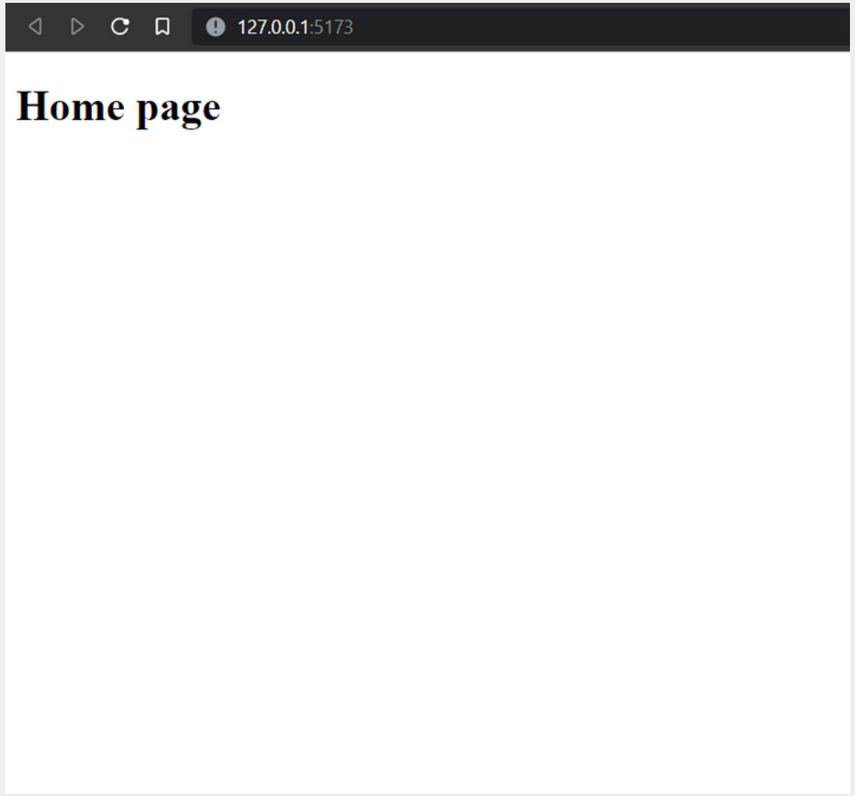
```jsx
import { Route, Routes } from "react-router-dom";
import Home from "./pages/Home";

function App() {
  return (
    <>
      <Routes>
        <Route path="/" element={<Home />} />
      </Routes>
    </>
  );
}

export default App;
```

Route match the path with URL and render the **jsx** passed through the element attribute

Result when running the App



Current location is "/" or "127.0.0.1:5173/"

Add more route

App.jsx

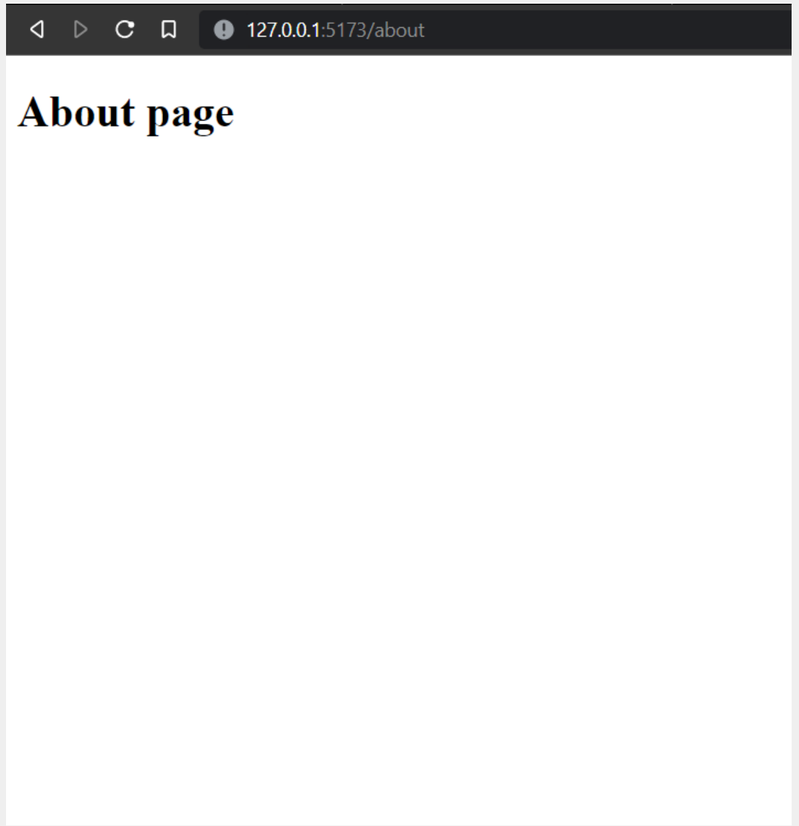
src > App.jsx > ...

```jsx
import { Route, Routes } from "react-router-dom";
import Home from "./pages/Home";
import About from "./pages/About";

function App() {
  return (
    <>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </>
  );
}

export default App;
```

Result when running the App



Current location is "/about" or "127.0.0.1:5173/about"

Let's add navbar

Adding nav

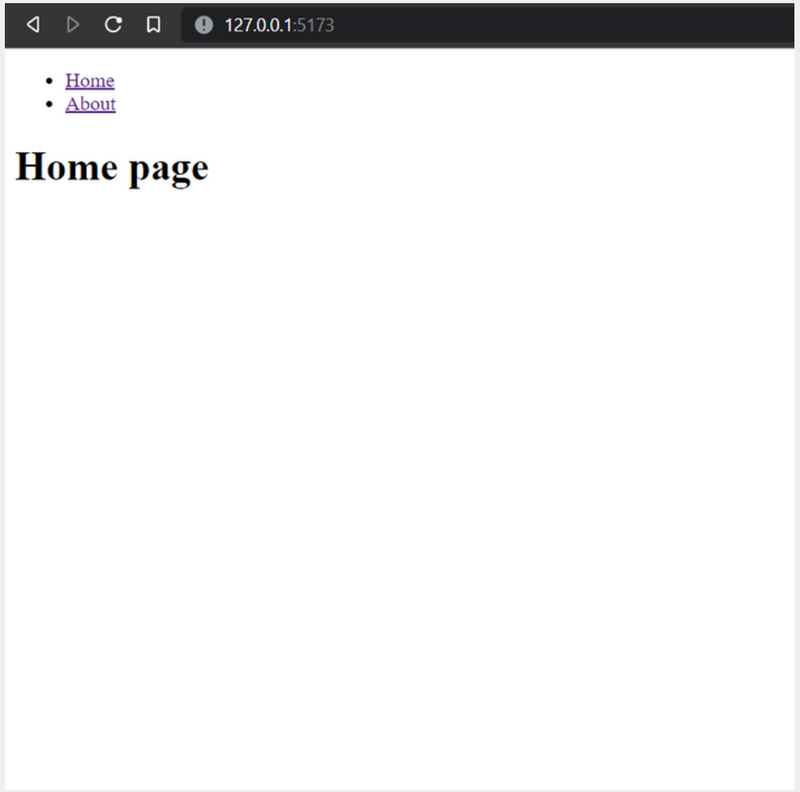
App.jsx

src > App.jsx > ...

```jsx
import { Route, Routes } from "react-router-dom";
import Home from "./pages/Home";
import About from "./pages/About";

function App() {
  return (
    <>
    <nav>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
      </ul>
    </nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </>
  );
}

export default App;
```



Dynamic Routing

Dynamic Route

App.jsx

src > App.jsx > ...

```jsx
import { Route, Routes, Link } from "react-router-dom";
import Home from "./pages/Home";
import About from "./pages/About";
import User from "./pages/User";

function App() {
  return (
    <>
    <nav>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
      </ul>
    </nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/user/:user" element={<User />} />
      </Routes>
    </>
  );
}

export default App;
```

Dynamic Route is a route with path that can be change dynamically.

Use " : " (double colon) to tell react-router that this path contain dynamic route or so call **param**

the "user" is the name of the param

Extract param from URL

User.jsx

src > pages > User.jsx > ...

```jsx
import React from "react";
import { useParams } from "react-router-dom";

function User() {
  const { user } = useParams();
  return (
    <>
      <h1>{user}</h1>
    </>
  );
}

export default User;
```

react-router-dom provides hook called **useParams** that return the key:value of params in the current URL.

we can extract "**user**" param from the URL.

What if URL does not exist in Routes?

Splat route

App.jsx

src > App.jsx > ...

```jsx
import { Route, Routes, Link } from "react-router-dom";
import Home from "./pages/Home";
import About from "./pages/About";
import User from "./pages/User";
import NotFound from "./pages/NotFound";

function App() {
  return (
    <>
    <nav>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
      </ul>
    </nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/user/:user" element={<User />} />
        <Route path="*" element={<NotFound />} />
      </Routes>
    </>
  );
}
```

Splat route for matching any URL that don't match with other declared path

<-- Example for 404 page not found
render this page if user try to enter URL that does not exist in our website

404 Page Not Found

NotFound.jsx

src > pages > NotFound.jsx > ...

```jsx
import React from "react";

function NotFound() {
  return (
    <>
      <h1>404: Page not found</h1>
    </>
  );
}

export default NotFound;
```

Nested Route

Nested Route

```
17      <Routes>
18        <Route path="/" element={<Home />} />
19        <Route path="/about" element={<About />} />
20        <Route path="/user/:user" >
21          <Route index element={<User />} />
22          <Route path="profile" element={<User />}/>
23          <Route path="blog" element={<UserBlog />} />
24        </Route>
25        <Route path="*" element={<NotFound />} />
26      </Routes>
```

Route can be nested

the /user/:user route now have

/user/:user/profile
/user/:user/blog

Why we need nested route?

```
17    <Routes>
18      <Route path="/" element={<Home />} />
19      <Route path="/about" element={<About />} />
20      <Route path="/user/:user" element={<User />} />
21      <Route path="/user/:user/profile" element={<User />} />
22      <Route path="/user/:user/blog" element={<UserBlog />} />
23      <Route path="*" element={<NotFound />} />
24    </Routes>
```

So you don't have to do something like the example

index route

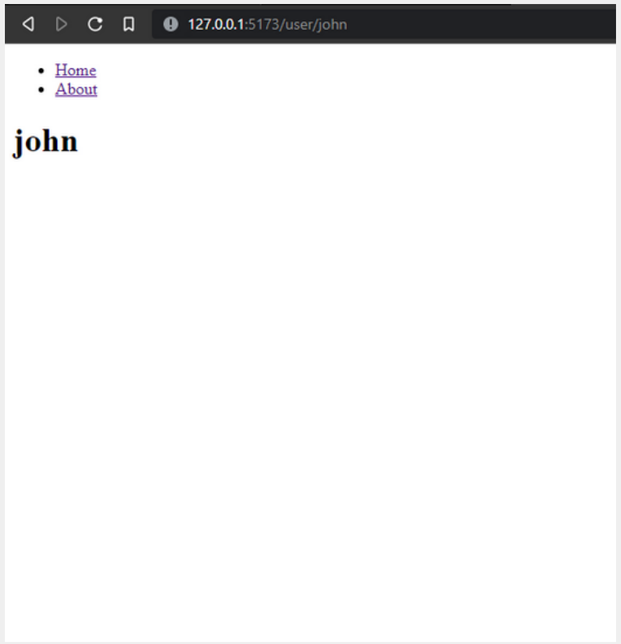
```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />} />
  <Route path="/user/:user" >
    <Route index element={<User />} />
    <Route path="profile" element={<User />}/>
    <Route path="blog" element={<UserBlog />} />
  </Route>
  <Route path="*" element={<NotFound />} />
</Routes>
```

use index attribute for setting the route to be default route

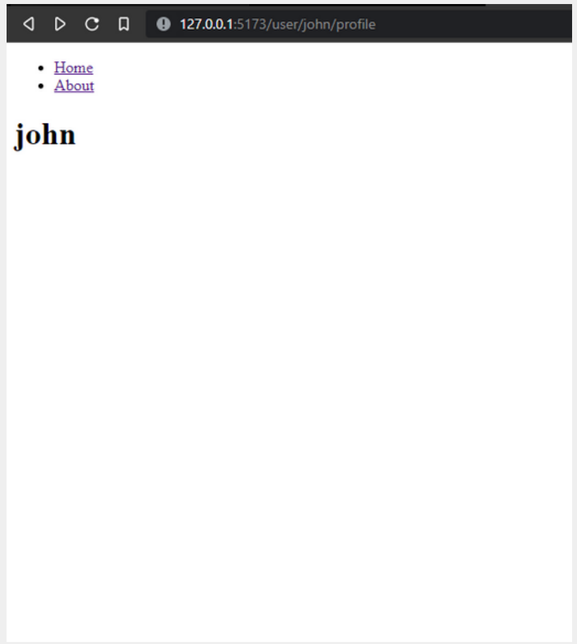
the path="/" that render Home component can be set to path="*" as well

Nested Route

/user/john



/user/john/profile



/user/john/blog

»

Link Component

Link Component

```
11    <nav>
12      <ul>
13        <li><Link to="/">Home</Link></li>
14        <li><Link to="/about">About</Link></li>
15      </ul>
16    </nav>
```

Link component is just an <a> tag but instead of passing the "href" attribute you pass in the "to" attribute to tell react-router-dom where this Link navigates to

https://reactrouter.com/en/main/components/link

NavLink Component

NavLink Component

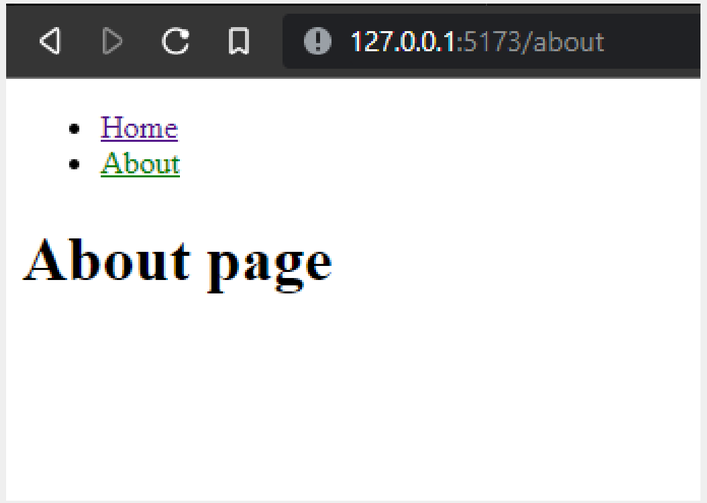
```jsx
<NavLink
  style={({ isActive }) => {
    return isActive ? { color: "green" } : {};
  }}
  to="/"
>
  Home
</NavLink>
```

Special kind of Link component that knows whether or not it is active

It accepts function in "style" and "class" attribute to decide what class or style it will apply.

https://reactrouter.com/en/main/components/nav-link

NavLink Component



https://reactrouter.com/en/main/components/nav-link

Navigate Component

Navigate Component

NotFound.jsx

src > pages > NotFound.jsx > ...

```jsx
import React from "react";
import { Navigate } from "react-router-dom";

function NotFound() {
  return (
    <>
      {/* <h1>404: Page not found</h1> */}
      <Navigate to="/" />
    </>
  );
}

export default NotFound;
```

A <Navigate> element changes the current location when it is rendered. It's a component wrapper around useNavigate, and accepts all the same arguments as props.

In the code on the left instead of showing "404: Page not found" we redirect user to "/"

useNavigate hook

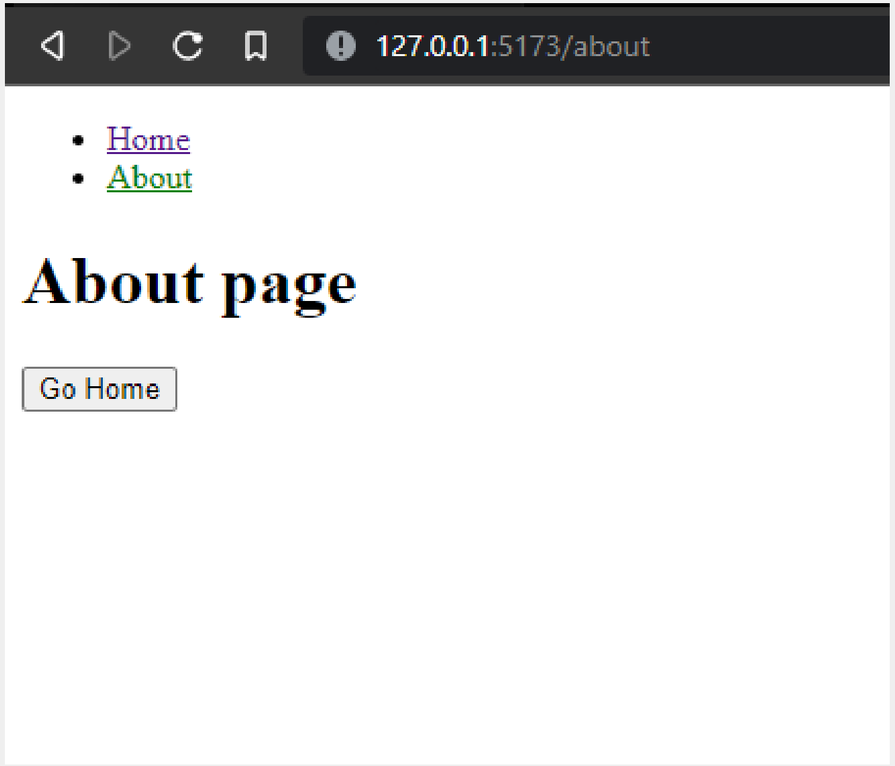
useNavigate

The useNavigate hook returns a function that lets you navigate programmatically.

Code on the left will redirect user to "/" when the param "user" is "gotohome"

useNavigate hook can be used when you want to navigate to another page by trigger them with **events** or **conditions**

useNavigate



Quick Look At the Document

https://reactrouter.com/en/main