

Introduction to React: Static Routing

In this discussion we will be building off the application we started last week. We will cover how to avoid writing repetitive code through the use of components and how to use the same components across different pages.

Task 1: Create the content for the about page

1. Create a file with the path `src/components/about-page.tsx`
2. Now create a component with the content to be displayed on the page

```
import { styled } from 'styled-components';

const AboutContentDiv = styled.div`
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: lightpink;
`;

export default function AboutPage() {
  return (
    <AboutContentDiv>
      <h1>I am a CS391 Instructor!</h1>
      <p>I love teaching this class because building web apps is so fun.</p>
    </AboutContentDiv>
  )
}
```

Task 4: Define routes

1. Edit your `src/App.tsx` file to match the one below. Here we define the content to be displayed at two different paths, the root and /about
 - a. We will install a package to help with this by using the command
`npm i react-router-dom`

```
import './App.css'
import ArtworksListContent from './components/artworks-list-content';
import AboutPage from './components/about-page';
import { createBrowserRouter, Route, Routes, RouterProvider } from 'react-router-dom';

const Root = () => {
  return (
    <div>
      <Routes>
        <Route path="/" element={ <ArtworksListContent /> } />
        <Route path="/about" element={ <AboutPage /> } />
      </Routes>
    </div>
  )
}

const router = createBrowserRouter(
  [{path: "*", Component: Root}]
)

export default function App() {
  return (
    <RouterProvider router={router} />
  )
}
```

Task 5: Test your routes

1. Start your application via the command `npm run dev`
2. Visit your application at <http://localhost:5173>
3. You should see the **ArtworksListContent** component at the root/home page
4. If you visit </about>, you should see the content from the **AboutContent** component

Task 6: Navigation

1. Our routes work, but manually typing in the route is inconvenient. So we will create a header with a navigation element that will be display on all of our pages
2. Create a file with the path `src/components/header.tsx`
3. Now, write the content for this component
 - a. Here we are using the **Link** component from **react-router-dom**. This component wraps the default HTML anchor element

```
import { Link } from "react-router-dom";
import { styled } from "styled-components";

const StyledHeader = styled.header`
  display: flex;
  justify-content: space-between;
  width: 100%;
  position: absolute;
  top: 0;
  left: 0;
`;

const Title = styled.h2`
  padding: 0.5rem;
  margin: 1rem;
`;

export default function Header() {
  return (
    <StyledHeader>
      <Title>React Introduction</Title>
      <nav>
        <Link to="/">Home</Link>
        <Link to="/about">About</Link>
      </nav>
    </StyledHeader>
  )
}
```

- b. The **Link** component is not a default HTML component, so we cannot style it using **styled-components**. If we want to create a styled version of this component, we can wrap it and use inline styling. Then we can replace old instances of the **Link** component with this styled version.

```
const NavLink = ({to, children}: {to: string, children: React.ReactNode}) => {  
  return (  
    <Link style={{padding: "0.5rem", margin: "0.25rem"}} to={to}>  
      {children}  
    </Link>  
  )  
}
```

4. Now that we have created the header, we need to display it on each page. Edit the **Root** component in **/src/App.tsx** to match the following. Don't forget to import the **Header** component.

```
import Header from './components/header';  
  
const Root = () => {  
  return (  
    <div>  
      <Header />  
      <Routes>  
        <Route path="/" element={<ArtworksListContent />} />  
        <Route path="/about" element={<AboutPage />} />  
      </Routes>  
    </div>  
  )  
}
```

5. Your pages should now contain a header with a navigation component, allowing for easy navigation between pages