Navegação com React Router

Labenu_

O que vamos ver hoje?

- Introdução à Rotas
- React Router
  - Configuração
  - Hooks!

Rotas

Labenu_

Rotas 🔀

- Indicam qual página será acessada
- Trazem semântica para o usuário
- Otimização do mecanismo de busca (SEO)

Rotas - Exemplos 🔀

https://github.com/facebook/react

https://www.npmjs.com/package/axios

http://localhost:3000/admin

- Domínio
- Rota

o conjunto todo é a URL

React Router 🔀

Labenu_

React Router 🔀

- Fará o gerenciamento das rotas
- Cada **rota** direciona para um **componente**
- Exemplos:

"/home" ⇒ **HomePage**

"/admin" ⇒ **AdminPage**

"/" ⇒ **HomePage**

React Router - Instalação 🔀

- Para instalá-la:

```
npm install react-router-dom
```

Documentação

Configuração do Router

Labenu_

Configuração - Preparando 🧩

- Criar a pasta pages
- Cada página será um componente
- Começaremos com HomePage e AdminPage
- Vamos lá!

Configuração - App.js 🧩

- O App.js começará a agrupar configurações globais
- Vamos configurar um exemplo!

Configuração - App.js 🧩

```
function App( ) {
  return (
     <BrowserRouter>
       <Routes>
         <Route path={"/"} element={<HomePage />} />
         <Route path={"/admin"} element={<AdminPage />} />
       </Routes>
     </BrowserRouter>
  );
}
```

Configuração - Rota inválida 🧩

- Caso a rota da URL seja inválida, renderizamos a página de erro

- Vamos criá-la e configurá-la no App!

Configuração final - App.js 🧩

```
function App( ) {
  return (
     <BrowserRouter>
       <Routes>
         <Route path={"/"} element={<HomePage />} />
         <Route path={"/admin"} element={<AdminPage />} />
         <Route path={"*"} element={<ErrorPage />} />
       </Routes>
     </BrowserRouter>
  );
}
```

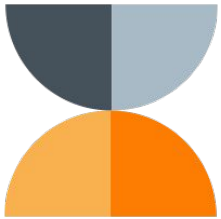
Exercício 1

5 min

- Crie um site com roteamento de 3 páginas, utilizando o React Router:
  - ❑ Página inicial
  - ❑ Página de perfil
  - ❑ Página de erro

Pausa para relaxar 😴

5 min



- `<BrowserRouter>` envolve tudo relativo à roteamento
- `<Routes>` troca entre as opções de tela e garante que apenas 1 será renderizada
- `<Route>` relaciona uma rota (path) ao componente (element) que será renderizado

Hooks!

Labenu_

Hooks do Router - useNavigate 🎣

- Bibliotecas também podem ter hooks!
- Utilizamos o hook **useNavigate** para navegarmos entre as nossas rotas
- Vamos configurá-lo!

Hook useNavigate 🧩

```
export default function Header( ) {
   const navigate = useNavigate( )

   return <header>
      <button onClick={( ) => navigate("/")}>Home</button>
      <button onClick={( ) => navigate("/admin")}>Admin</button>
   </header>
}
```

Hook useNavigate 🎣

- O que acontece se passarmos o argumento da rota errado? Vamos testar?
- Seria ideal centralizar o valor em uma única função para evitarmos erro de typo (digitação)
- Bora lá!

Hook useNavigate - coordinator.js 🧩

```
export function goToAdminPage(navigate) {
    navigate("/admin")
}

export function goToHomePage(navigate) {
    navigate("/")
}
```

Hook useNavigate - Header.js 🧩

```
export default function Header( ) {
   const navigate = useNavigate( )

   return <header>
      <button onClick={( ) => goToHomePage(navigate)}>Home</button>
      <button onClick={( ) => goToAdminPage(navigate)}>Admin</button>
   </header>
}
```

Exercício 2

10 min

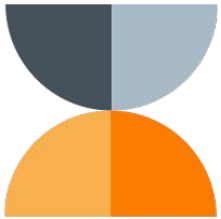
- Agora monte a mudança de rota no clique dos botões do cabeçalho:

  - ❑ Botão de ir para página inicial

  - ❑ Botão de ir para página de perfil

Pausa para relaxar 😴

5 min



- Utilizamos o hook **useNavigate()** para navegarmos entre páginas

- É interessante centralizar as funções de navegação em um arquivo separado

Hooks do Router - useParams 🎣

- Utilizamos o hook **useParams** para acessarmos o valor de um path param
- Primeiro precisamos configurar uma rota que terá path param
- Conseguimos acessar o valor recebido na rota

Hook useParams- App.js 🧩

```
function App( ) {
  return (
     <BrowserRouter>
       <Routes>
         <Route path={"/"} element={<HomePage />} />
         <Route path={"/video/:id"} element={<VideoPage />} />
         <Route path={"/admin"} element={<AdminPage />} />
         <Route path={"*"} element={<ErrorPage />} />
       </Routes>
     </BrowserRouter>
  );
}
```

Hook useParams- App.js 🧩

```
export default function VideoPage( ) {
   const { id } = useParams( )

   return <main>
      <Header />
      <h1>Video {id}</h1>
   </main>
}
```

Exercício 3

5 min

- Agora configure uma rota que utiliza path param:

  - ❏ Crie uma nova página de produto específico por id “/product/:id”

  - ❏ Não se preocupe em aplicar navegação de rota nela

Observação sobre o surge ⚠️

- Quando usamos o Router, para subir o site usando o surge, precisamos executar os 4 comandos abaixo:

```
npm run build

cd build

cp index.html 200.html

surge
```

Organizando o App.js

Labenu_

Organizando o App.js ⚠️

- É comum centralizar configurações no App.js ou index.js

- Conforme a aplicação evolui, precisamos separar as lógicas em arquivos externos

- Extrairemos toda a configuração no App.js do React Router para um componente externo <Router>

Router.js 🧩

```
export default function Router( ) {
 return (
    <BrowserRouter>
      <Routes>
        <Route path={"/"} element={<HomePage />} />
        <Route path={"/video/:id"} element={<VideoPage />} />
        <Route path={"/admin"} element={<AdminPage />} />
        <Route path={"*"} element={<ErrorPage />} />
      </Routes>
    </BrowserRouter>
 );
}
```

Hook useParams- App.js 🧩

```
function App() {
 return (
    <>
      <Router />
    </>
 );
}
```

Resumo

Labenu_

Resumo 📓

- Para reproduzir o comportamento natural de navegação no browser usamos a lib react-router-dom

- Possui 3 componentes principais:

```
<BrowserRouter>
    <Routes>
        <Route path={"/"} element={<HomePage />} />
    </Routes>
</BrowserRouter>
```

Resumo 📓

- A lib também possui hooks:
    - `useNavigate()`
      nos permite navegar entre componentes
    - `useParams()`
      nos permite acessar o valor de path params

Resumo 📓

- É recomendado organizar nossos arquivos em pastas
- **pages** centraliza nossos componentes de páginas
- **routes** centraliza nossa configuração do React Router

Router.js

componente de configuração de rotas

coordinator.js

arquivo com todas as funções de navegação de rotas

Dúvidas? 🧐

Labenu_

Obrigado(a)!