LABORATÓRIO ADICIONAL

Adicionando navegação

Objetivos:

- Escrever uma URL na barra de endereços e navegar para a página do produto correspondente
- Clicar em links na página para navegar na single-page application
- Clicar no voltar e avançar do navegador para navegar na história do navegador intuitivamente

Associando um caminho URL com um componente

A aplicação já usa o Router para navegar para o ListaProdutosComponent. Nessa seção vamos mostrar como definir a rota para mostrar detalhes individuais do produto.

1. Gere um novo componente para os detalhes dos produtos. Utilize o comando generate no terminal

   $ ng generate component detalhe-produtos

2. No module do app, adicione uma rota para os detalhes do produto, com um 'path' de 'produtos/produtoId' e o novo component

$ { path: 'produtos/:produtotId', component: DetalheProdutosComponent },

3. No HTML da lista de produtos, vamos modificar o link do nome do produto incluindo um *routerLink* com o id do produto como parâmetro

```
$ <a
        [title]="produto.nome + ' detalhes"
        [routerLink]="['/produtos', produto.id]">
        {{ produto.nome }}
      </a>
```

A diretiva *RouterLink* ajuda a customizar a tag <a>. Nesse caso, a rota, ou URL, contém um segmento fixo, /produtos. O último segmento é variável, colocando a propriedade id no produto atual.

4. Verifique que a rota funciona como esperado quando o nome do produto é clicado. A aplicação deve mostrar o DetalheProdutos, que já está com um texto padrão.

Note também que a URL mudou

Visualizando detalhes do produto

O DetalheProdutosComponent manuseia a visualização de cada produto. O Router mostra componentes baseado na URL do navegador e na suas rotas definidas.

Nessa seção, vamos usar o Router para combinar os dados do produto com a informação da rota. Assim poderemos mostrar os detalhes específicos de cada produto.

1. No typescript do DetalheProdutos, importe ActivatedRoute e o array de produtos

$ import { ActivatedRoute } from '@angular/router';

$ import { Produto, produtos} from '../produtos;

2. Defina a propriedade produto

$ produto: Produto | undefined;

3. Injete ActivatedRoute no constructor adicionando um argumento privado dentro dos parênteses.

$ constructor(private route: ActivatedRoute) { }

ActivatedRoute é específica para cada componente que o Router carrega.

ActivatedRoute contém informações sobre a rota e os parâmetros da rota.

Injetando ActivateRoute, você está configurando o componente para usar o serviço.

4. No método ngOnInit(), extraia o produtoId dos parâmetros da rota e encontre o produto correspondente no array

```
ngOnInit() {
  const routeParams = this.route.snapshot.paramMap;
 const produtoIdRoute = Number(routeParams.get('productId'));

  this.produto = produtos.find(produto => produto.id === produtoIdRoute);
}
```

Os parâmetros da rota correspondem as variáveis do caminho que você definiu na sua rota. Para acessar os parâmetros da rota, utilize route.snapshot, que é o ActivatedRouteSnapshot que contém a informação sobre a rota ativada em um momento particular do tempo.

A URL que corresponde a rota provém o id do Produto. Nossa aplicação usuária o id do produto para mostrar os detalhes de cada produto.

5. Atualize o DetalheProdutosComponent HTML para mostrar os detalhes dos produtos com o *ngIf. Se o produto existir, a renderizará o nome, preço e descrição.

```
<h2>Detalhes do Produto</h2>

<div *ngIf="produto">
  <h3>{{ produto.nome }}</h3>
  <h4>{{ produto.preco | currency }}</h4>
  <p>{{ produto.descricao }}</p>
</div>
```

A linha <h4> usa uma pipe, currency. Essa pipe transforma um número em uma string de moeda. Uma pipe é uma maneira de transformar dados no seu template HTML.