@fpftech.educacional

FPFtech
Escola Tecnológica

FPFtech
Escola Tecnológica

Relembrando ontem

Interface Produto

Transforma o modelo

em Tipagem. {nome: string}

ProdutoService

Manipula dados (fez o “get”)

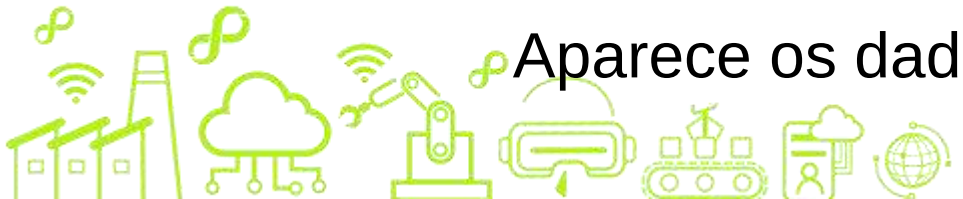
Produto.ts

Injetamos serviço nele,

ngOnInit -> this.produtoService.getProdutos

Produto.html

Aparece os dados em tela via Data Binding



O que sabemos até agora?

1 – Angular baseia-se em **componentes.**

2 – Componentes são subdivididos em **HTML, CSS, TS (TYPESCRIPT).**

3 – Data Binding faz com que dados no TS apareçam no HTML.

4 – Podemos deixar nosso HTML mais dinâmico com **Diretivas**(@If..)

5 – Interfaces transformam Objetos em Tipagem.

6 – Serviços manipulam dados e fazem requisições HTTP.(get,post...)

7 – Serviços são **injetados** no TS, diretamente numa **variável privada**.

Até agora, como fizemos?

Até o presente momento, no **App.ts** nós importávamos um componente.

Em seguida, no **App.html** nós colocávamos esse app como <tag>.

O sistema ficava reclamando dos outros componentes não utilizandos no App.

Hoje vamos resolver isso

Como funciona o roteamento?

O Angular tem um módulo especial de rotas, chamado RouterModule, que faz o controle da navegação.

Esse controle é configurado dentro de um arquivo chamado:

- **app-routes.ts**

Configurando as rotas

```
const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'produtos', component: ProdutosComponent },
  { path: '', redirectTo: 'home', pathMatch: 'full' }
];
```

1 – Quando o usuário acessa */home*, mostre o **HomeComponent**.

2 – Acessa */produtos*, mostre o componente **ProdutosComponent**.

3 – Quando não há caminho, redireciona pra */home*.

Navegando entre páginas

Você usa a diretiva especial **routerLink** nos elementos HTML, como este:

```
<a [routerLink]="['/produtos']">Produtos</a>
```

Isso funciona como um <*a href="..."*>, mas **sem recarregar a página**.

Pode-se também usar queryParams:

```
<a [routerLink]="['/user/bob']" [queryParams]="{debug: true}"
  link to user component
</a>
```

Exemplo

```
<nav>
  <a [routerLink]="['/home']">Home</a>
  |
  <a [routerLink]="['/produtos']">Produtos</a>
</nav>
```

O **router-outlet** é um marcador mágico:

é ali que o Angular renderiza os componentes das rotas.

Sempre que a URL mudar, o componente correspondente é carregado **dentro do router-outlet**.

Como implementar

FPFtech
Escola Tecnológica

1. Atualize seu app.routes.ts

Importe seus componentes e crie os caminhos. Nesse exemplo estamos usando 2 componentes “HomeComponent” e “AboutComponent”

```
import { Routes } from '@angular/router';
import { HomeComponent } from './home.component';
import { AboutComponent } from './about.component';


export const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'about', component: AboutComponent },
  { path: '**', redirectTo: '' } // rota coringa
];
```

Como implementar

FPFtech
Escola Tecnológica

2. *app.component.ts*

```ts
import { Component } from '@angular/cor
import {RouterLink, RouterOutlet} from

1+ usages  jonatas.lopes
@Component({
  selector: 'app-root',
  imports: [RouterOutlet, RouterLink],
  templateUrl: './app.component.html',
  standalone: true,
  styleUrl: './app.component.css'
})
export class AppComponent {
  title :string  = 'meu-projeto';
}
```

3. *app.component.html*

```html
<nav>
  <a [routerLink]="['/home']">Home</a>
  |
  <a [routerLink]="['/produtos']">Produtos</a>
</nav>
<hr /> <!--linha horizontal-->
<router-outlet></router-outlet>
```

Agora é sua vez

Adicione todos os componentes vistos até agora.

- 1. Modifique as rotas;
- 1.1. Importe os componentes nas rotas;
- 1.2. crie os caminhos pra cada componente.
- 2. Modifique o App.ts
- 2.1. Delete todas as importações dos componentes até agora.
- 2.2. Importe somente RouterOutlet e RouterLInk
- 2.3. Adicione RouterOutlet e RouterLInk na variável “imports”
- 3. Atualize o App.HTML
- 3.1. Crie um <nav> que tenha links <a> pra cada rota e o <router-outlet>

Desafio: Lista Dinâmica de Tarefas

Objetivo: Criar uma lista onde o usuário possa:

Ver todas as tarefas

Clicar para marcar como concluída (opcional, para os avançados)

Tarefas concluídas aparecem com um estilo diferente

1. Crie uma lista com(pelo menos) 3 tarefas:

   Estudar Angular, Fazer Exercicio, Fazer Atividade

2. (opcional) Faça uma função que faça o Toggle de Tarefa Concluída.

   Toggle é **true=>false** e **false=>true**

3. Faça o HTML e o CSS serem dinâmicos.

Fazendo o Desafio: Lista Dinâmica de Tarefas

1 – Crie o componente “tarefaComponent” = *ng g c components/tarefa*

2 – Crie a Interface Tarefa:

export interface Tarefa {titulo:string, descricao:string, concluido:boolean}

3 – Crie o serviço “tarefaService” = *ng g s services/tarefa*

4 – No tarefaService, importe a interface e crie a função getTarefas();

Obs: a função retorna uma lista de tarefas: Tarefa[]

Crie as tarefas mockadas.

5 – Em “tarefaComponent” importe a interface, o serviço e o inject.

5.1. – Variável tarefaService, injete o serviço. Outra variável listaTarefas tipo Tarefa[] = [];

5.2. – ngOnInit roda o serviço e salva em listaTarefas. Crie outra função toggleConcluido()

6 – No tarefa.HTML, atualize pra mostrar a lista de Tarefas.

7 – Atualize rotas, adicione o tarefaComponent lá.

7.1. – Atualize o App.HTML, com o link <a> para o novo componente. Rode a aplicação.

Notas

Angular SPA = uma só página, vários conteúdos dinâmicos.

Roteamento é como trocar "cenas" dentro da mesma página.

Cada componente é uma "página" que o Angular mostra conforme a URL.

routerLink funciona como um botão de troca de tela, sem precisar atualizar tudo.

router-outlet é o palco onde as páginas aparecem.

O que sabemos até agora?

1 – Angular baseia-se em **componentes.**

2 – Componentes são subdivididos em **HTML, CSS, TS (TYPESCRIPT).**

3 – Data Binding faz com que dados no TS apareçam no HTML.

4 – Podemos deixar nosso HTML mais dinâmico com **Diretivas**(@If..)

5 – Interfaces transformam Objetos em Tipagem.

6 – Serviços manipulam dados e fazem requisições HTTP.(get,post...)

7 – Serviços são **injetados** no TS, diretamente numa **variável privada**.

8 – Rotas com **RouterLink** e aparecem na tela pelo **<router-outlet>**

Obrigado!

FPFtech
Escola Tecnológica

@fpftech.educacional