@fpftech.educacional

FPFtech
Escola Tecnológica

FPFtech
Escola Tecnológica

Serviços e Injeção de Dependência

O que são Serviços no Angular?

Imagine que você está construindo uma loja online.

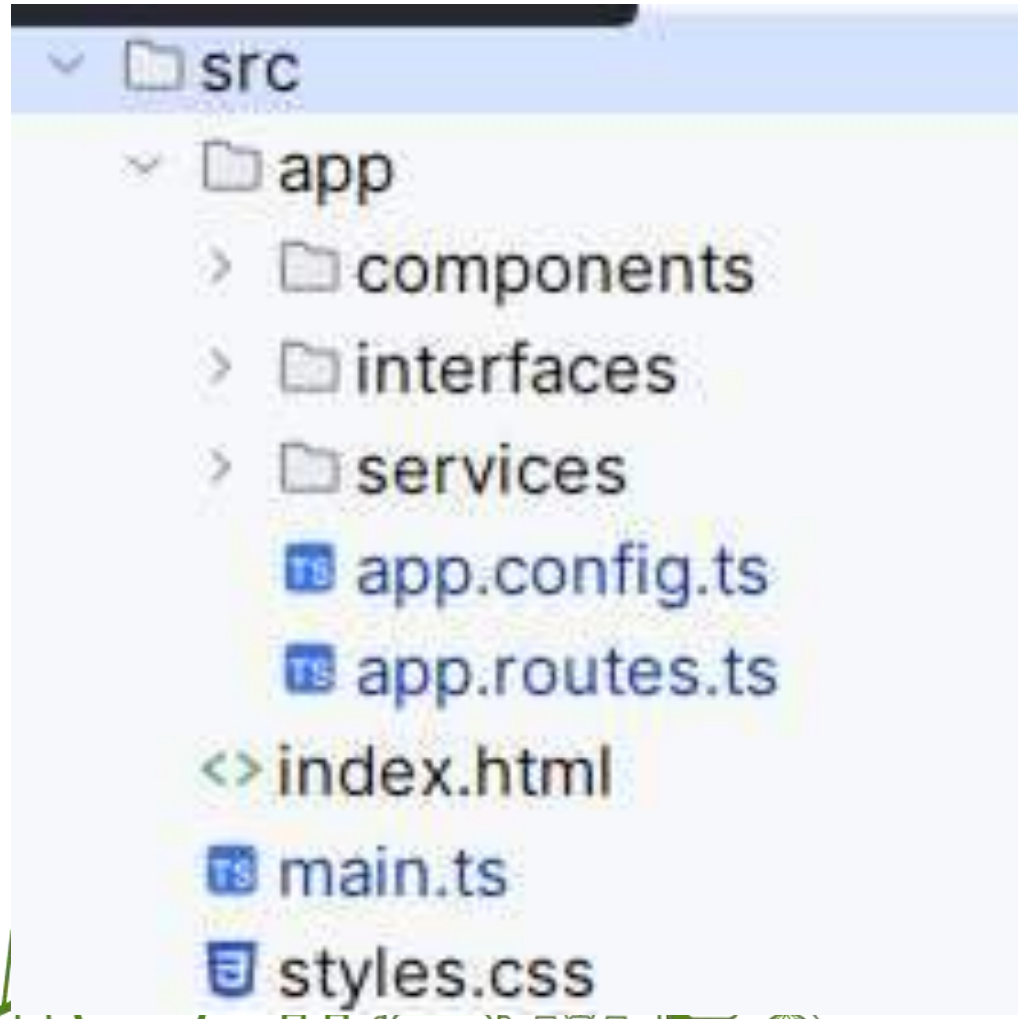
Os componentes são responsáveis por exibir as informações na tela (como mostrar os produtos), mas eles **não devem se preocupar com onde os dados vêm** ou **como são processados**.

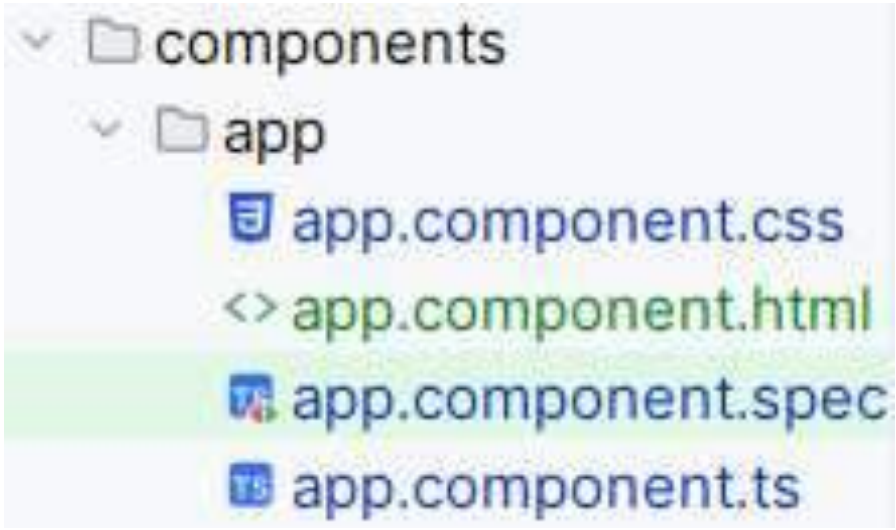
Essa parte fica com **os serviços**.

Masss primeiro....

FPFtech
Escola Tecnológica

Vamos arrumar as pastas do nosso projeto





Serviços

Definição de Serviços:

São classes que contêm lógica de negócio ou manipulação de dados.

Eles atuam como "bastidores", enquanto os componentes são o "palco".

Por que usar serviços?

Separação de responsabilidades: Componentes exibem dados, serviços processam dados.

Reutilização de código: Um serviço pode ser usado por vários componentes diferentes sem repetição.

Organização: O código fica mais limpo e mais fácil de manter.

Serviços

Criando um novo serviço:

ng generate service services/produto *#Esse comando cria dois arquivos*

produto.service.ts: *onde você escreve a lógica*

produto.service.spec.ts: *para testes automatizados (não usaremos agora)*

Vamos criar uma pasta **services** e colocar os arquivos novos lá dentro:

Como realizar Injeção de Dependência

Na pasta “Interfaces” crie novo **TS** e chame-o de **Produto.ts**

```ts
export interface Produto {
  nome: string;
  promocao: boolean;
}
```

*Volte pro seu **produto.component.ts** e importe a nova interface*

```ts
import {Produto} from '../../interfaces/produto';


1+ usages
@Component({
```

Serviços

```
import { Injectable } from '@angular/core';
import {Produto} from '../interfaces/produto';

1+ usages  new *
@Injectable({providedIn: 'root'})
export class ProdutoService {
  1+ usages  new *
  getProdutos(): Produto[] {
    return [
    { nome: 'Notebook', promocao: true },
    { nome: 'Mouse', promocao: false },
    { nome: 'Teclado', promocao: true }
  ]};
}
```

Criamos uma função **getProdutos** que faz simulação “mock” como se fosse pegar produtos numa requisição HTTP.

Essa função retorna os Produtos.

O ProdutoService precisa estar anotado com **@Injectable({ providedIn: 'root' })**, que diz ao Angular que ele deve cuidar de sua criação e entrega.

Adicionei somente as linhas azuis.

Injeção de Dependência (Dependency Injection)

Definição:

É uma técnica onde o Angular **entrega automaticamente** uma instância de uma classe (no caso, um serviço) para um componente que precisa dela.

Você não precisa criar o serviço manualmente com **new** — o Angular faz isso para você.

SERVIÇO –(dado)-> COMPONENTE

Como realizar Injeção de Dependência

No componente, usamos o construtor para receber o serviço:

```
jeto> ng g c components/produtos --standalone
component.html (24 bytes)
```

```
import { Component, inject } from '@angular/core';
import {ProdutoService} from '../../services/produto.service';

1+ usages
@Component({
  selector: 'app-produtos',
  imports: [],
  standalone: true,
  templateUrl: './produtos.component.html',
  styleUrl: './produtos.component.css'
})
export class ProdutosComponent {
  private produtoService : ProdutoService = inject(ProdutoService)
}
```

private *é uma forma Angular armazenar serviço dentro do componente.*

Como realizar Injeção de Dependência

Ainda componente

```
  private produtoService : ProdutoService  = inject(ProdutoServi

1 listaProdutos: Produto[] = [];
  no usages
  ngOnInit() : void {
2   this.listaProdutos = this.produtoService.getProdutos();
3 }
}
```

*1 - Foi criado uma nova variável de lista do tipo **produtos[]** vazia.*

*2 – **ngOnInit** é chamado quando o componente é iniciado.*

3 – Aqui estamos pegando o retorno da função (no serviço) e salvando na variável do item 1.

Exercicio: Produtos compostos

Objetivo: Recirar os passos anteriores, só que com esse produtos e fazê-los aparecer na tela:

```
{ nome: 'Notebook', preco: 3000, promocao: true },
{ nome: 'Mouse', preco: 100, promocao: false },
{ nome: 'Teclado', preco: 200, promocao: true },
```

Crie no serviço uma nova função que filtre somente os que estão em promoção:

```
getProdutosEmPromocao() {
  return this.getProdutos().filter(p => p.promocao);
}
```

Notas

Um serviço é como um “cérebro” que guarda as informações e toma decisões.

O componente é o “rosto” que mostra essas informações.

Usamos serviços para separar as tarefas de exibição das tarefas de lógica.

Injeção de Dependência é o jeito do Angular entregar automaticamente essas classes para você — é como se você pedisse algo ao sistema e ele dissesse:

“Toma aqui, já está pronto!”

Obrigado!

FPFtech
Escola Tecnológica

@fpftech.educacional