Componente para producto 5/20

RECURSOS MARCADORES

Existen muchas situaciones en donde deberás enviar información de un componente padre a su/s hijo/s, por eso, acá te mostraremos con un ejemplo cómo funciona el **componente para producto**.

Comunicando componente padre a hijo

Un ejemplo real para el uso de la comunicación entre **componente** podría ser para renderizar N cantidad de productos de un catálogo.

Paso 1: Comienza creando una interfaz para tipear el modelo de datos del Producto:

```
// interfaces/producto.interface.ts
export interface Producto {
    id: number;
    name: string;
    precio: number;
    image: string;
}
```

Paso 2: Luego, impórtala en el componente Catálogo que será el componente padre en la comunicación.

```
// components/catalogo/catalogo.component.ts
import { Component } from '@angular/core';
import { Producto } from './producto.interface.ts';

@Component({
  selector: 'app-catalogo',
  templateUrl: './catalogo.component.html',
  styleUrls: ['./catalogo.component.scss']
})
export class CatalogoComponent {

  public productos: Producto[] = [
    {
        id: 1,
        name: 'Automobil de juguete',
        precio: 100,
        image: './image1.jpg'
    },
    {
        id: 2,
        name: 'Muñeca de trapo',
        precio: 180,
        image: './image2.jpg'
    },
    {
        id: 3,
        name: 'Pelota de futbol',
        precio: 120,
        image: './image3.jpg'
    }
```

```
    ];
  }
```

Paso 3: Este componente posee un array de productos para iterar en el HTML inicializando el componente `<app-producto>` por cada objeto en el array.

```
<!-- components/catalogo/catalogo.component.html -->
<app-producto *ngFor="let p of productos"
    [producto]="p"
></app-producto>
```

Paso 4: Finalmente, el componente hijo recibe el producto haciendo uso del decorador `@Input()` y apoyándose también de la interfaz para tipear los datos.

```
// components/producto/producto.component.ts
import { Component, Input } from '@angular/core';
import { Producto } from './producto.interface.ts';

@Component({
  selector: 'app-producto',
  templateUrl: './producto.component.html',
  styleUrls: ['./producto.component.scss']
})
export class ProductoComponent {
```

```
  @Input() producto: Producto;
}
```

Pudiendo mostrar la información del producto en el template del componente hijo:

```
<!-- components/producto/producto.component.html -->
<div>
    <h2>{{ producto.name }}</h2>
    <img [src]="producto.image">
    <p>Precio: {{ producto.precio }}</p>
</div>
```

Será habitual tener la necesidad en tus proyectos de construir componentes más grandes o “contenedores” de muchos otros componentes repetitivos y más pequeños. Es importante buscar este desacople entre componentes de la mejor manera posible.

Ver código fuente del proyecto

Contribución creada con los aportes de Kevin Fiorentino.

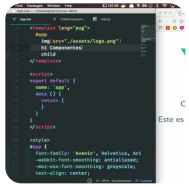
Lecturas recomendadas

GitHub - platzi/angular-componentes at 4-step
https://github.com/platzi/angular-componentes/tree/4-step

Clases relacionadas

Componentes con Outputs

Curso de Angular: Unit Testing para Componentes



Component

Curso Profesional de Vue.js 2

Creación de componentes

Curso Profesional de Vue.js 2