UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 11: Angular - LocalStorage

Profesor: Ing. Leandro Rodolfo Gil Carrano
Email: leangilutn@gmail.com

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

@input

@input

El paso de información de padres a hijos se realiza por medio de propiedades del componente. La información se bindea desde el template, usando los atributos de la etiqueta del componente.

Para ello, las propiedades del componente se deben decorar con @Input, de modo que Angular sea capaz de saber que éstas son capaces de inicializarse, o modificarse, desde fuera. Este es el escenario que vamos a abordar en este artículo.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

@input

En el componente hijo se debe declarar las variables utilizando @input. En el caso de que la variable tenga asignado un valor, dicho valor se tomara por default.

```
import { Component, OnInit, Input } from '@angular/core';

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

export class ProductoComponent implements OnInit {
  @Input()
  producto_id=10;
  @Input()
  producto_nombre;

  constructor() {
```

@input

El componente padre al llamar al componente hijo debe enviar las variables definidas en el hijo como directivas

```
<p>
  home works!
  <app-producto
  [producto_id]="producto_home_id"
  [producto_nombre]="producto_home_nombre"
  >


  </app-producto>
</p>
```

@input

Si modificamos las variables del componente padre, se modifica el hijo

```
<button (click)="actualizar_datos()">Modificar Producto</button>
```

```
actualizar_datos(){
  this.producto_home_id=30;
  this.producto_home_nombre="Producto modificado"
}
```

@output

@output

También puede surgir la necesidad de que los hijos comuniquen a los padres cambios en la información que ellos manejan. En estos casos Angular utiliza eventos. Es decir, cuando el hijo tiene un dato y quiere hacerlo llegar al padre, genera un evento que puede ser capturado en el padre para realizar aquellas acciones que sean necesarias

@output

En el evento hijo declaramos un objeto emiter

```
import { Component, OnInit, Input,Output, EventEmitter } from '@angular/core';

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

export class ProductoComponent implements OnInit {

  @Output()
  propagar = new EventEmitter<string>();
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

@output

El hijo emite evento

```
<button (click)="onPropagar()">Emitir Evento</button>
```

```
onPropagar(){
  this.propagar.emit("Evento emitido");
}
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

@output

El padre debe recibir el evento emitido

```
<app-producto

(propagar)="procesaPropagar($event)"
>

</app-producto>
```

```
procesaPropagar(mensaje) {
  console.log(mensaje);
}
```

LocalStorate

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

LocalStorage

Almacenamiento local en el navegador, dispone de los siguientes métodos:

```
//Guarda el valor en la clave especificada
localStorage.setItem(key:string,  value:string);
//Obtiene el valor dada una clave
localStorage.getItem(key:string);
//Elimina el valor de una clave
localStorage.removeItem(key:string);
//Limpia todo el almacenamiento
localStorage.clear();
```

SessionStorage

Mismo concepto que localstorage, pero los datos se pierden cuando se cierra la pagina:

```
//Guarda el valor en la clave especificada
sessionStorage.setItem(key:string,  value:string);
//Obtiene el valor dada una clave
sessionStorage.getItem(key:string);
//Elimina el valor de una clave
sessionStorage.removeItem(key:string);
//Limpia todo el almacenamiento
sessionStorage.clear();
```

LocalStorage

Ejemplo de guardado de json, solo se pueden almacenar string por lo cual debemos utiliza la funciona stringfy para guardar y parse para leer:

```
localStorage.setItem('usuarios',  JSON.stringify(data));
let usr2 = localStorage.getItem('usuarios');
console.log(JSON.parse(usr2));
```