CLASE 2: LAB IV

Repaso clase pasada, toda aplicación de angular inicia en el module, este tenía un array en cual íbamos a poner nuestros otros módulos y los importes para módulos externos, bootstrap era en que parte va a iniciar mi aplicación

Lo que debemos hacer en realidad es separar cada cosa en componentes, debe ser bien simple , en este caso un formulario, siempre los componentes se dividen en componentes smart o en tontos

Tenemos varias maneras de comunicarnos entre componentes:

1- @input en el cual en el componente padre

le paso un dato al hijo es decir es de PADRE A

HIJO,

2-@output es cuando el hijo le envia mensajes al

Padre, se le pasa un Evento "( )" y dentro una

Función manejadora con el $event (nos da

Toda la info.Para crear este evento tenemos

Crear el @output() cargar = new eventemiter<any> () y asi transformamos mi cargar en un evento , para poder utilizarlo lo

puedo hacer en cualquier lugar del código :

"this.cargar.emit(persona)" con esto se dispara

la función desde el hijo al padre

En principio el flujo de datos es este y entre hermanos no se comunican

-Creamos dos componentes uno para el form y otro para los datos y luego los conectamos cpn output

-cuando usemos outpur debemos importar el "EventEmitter"

-Ahora el ngif lo haremos desde el padre

“\*ngFor” va a iterar sobre un array , declaramos una variable y luego el "of" es decir de donde va a iterar , en este caso del array datos

Le pasamos el objeto persona desde la clase padre , ese objeto persona es siempre el mismo ya que todos los objetos apuntan a una misma referencia

"..." -> esto lo que hace "persona: persona.persona"

es decir estoy sacando las propiedades de un objeto y guardándolas en un nuevo objeto

Creamos el objeto clase :

“ng generate clas”s clases/persona y ese objeto lo utilizaríamos para lo demás

Resumen:

1-Creamos tres componentes : saludar(clase Padre) , “form “y “data” (clases hijas)

2-Lo que haremos será , dentro del componente “form” en la parte de html tener dos inputs en los cuales ingresaremos “nombre” e “email” y un “botón” , al ingresar datos y pulsar el botón nos dirige hacia una función que tendremos dentro del “form.ts” en la cual ya tendremos un “@Output (llamado ‘cargar’)” creado y dentro de la función crearemos un objeto de tipo Persona (nombre, email) y lo emitiremos mediante ese @Output hacia nuestra clase Padre.

@Output:

@Output() cargar = new EventEmitter<any>();

Función que emite el output:

ngCargar() {

this.persona = new Persona(this.nombre, this.email);

this.cargar.emit(this.persona);

}

3-Desde la clase padre (saludar.html) lo que haremos será crear una etiqueta:

<app-form (cargar)="ngProcesarCarga($event)"></app-form>

En donde “(cargar)” se refiere al @Output de “forms.ts” y “ngProcesarCarga($event)” va a ser la función que crearemos desde “saludar.ts” para poder leer lo que emitio ese @Output es decir para leer el objeto de tipo “Persona”.

\*$event : siempre debemos colocarlo asi para que sepa que es un evento,desde el lado de typescript podremos leerlo con cualquier variable

Una vez que recibimos los datos en nuestra función “ngProcesarCarga($event)” lo vamos a cargar en un “array” de objetos de tipo persona llamado “listado”.

ngProcesarCarga(datos: Persona) {

this.listado.push(datos);

}

4-Volvemos a la clase padre y vamos a enviar ese ”listado” (array) hacia la clase Hija llamada Datos :

<app-datos [datos]="listado"></app-datos>

[datos] es un “@Input” que creamos desde nuestra clase hija al cual le asignamos “listado”

El cuál sería el “array” que cargamos con objetos de tipo Persona desde “saludar.ts”

5-Desde el datos.ts solo tenemos el “@Input datos” cargado con un array de tipo Persona

El cual mostraremos desde datos.html con una tabla utilizando \*ngFor (para recorrer el array) y usando “interpolación” para leer cada atributo de nuestros objetos Persona:

<table class="egt">

<thead>

<th>Nombres</th>

<th>email</th>

</thead>

<tbody>

<tr \*ngFor="let item of datos">

<td>{{item.nombre}}</td>

<td>{{item.email}}</td>

</tr>

</tbody>

</table>