



# Geeks México

BLOG DE PROGRAMACIÓN EN ESPAÑOL SOBRE JAVA, FRAMEWORKS, BASES DE DATOS, CÓMPUTO EN LA NUBE, ETC. EN ESPAÑOL Y EN INGLÉS.

[TUTORIALES EN ESPAÑOL](#)[TUTORIALS IN ENGLISH](#)[ABOUT](#)[CONTACT](#)

Anuncios

▲

▼

MLL...  
€0,51

▶

## Databinding en Angular 4 Parte 1

📅 [HACE 2 SEMANAS](#)    💬 [DEJA UN COMENTARIO](#)

Databinding en Angular 4 es utilizado para comunicar los templates (Archivos html) con typescript (La lógica de negocio de nuestra aplicación). Esto nos permitirá generar aplicaciones web dinámicas que puedan generar contenido diferente de acuerdo a los usuarios.

Existen distintos modos de comunicación:

- **Para envío de información** (Typescript->HTML): Permite tomar información de nuestro código typescript, este puede obtenerlo de un servicio web o de un cálculo y desplegarlo en una pantalla HTML, realizar este envío de información es posible a través de las siguientes opciones:
  - String interpolation: **{{data}}**
  - Property binding: **[property]="data"**
- **Para la reacción a eventos del usuario** (HTML->Typescript): Los archivos HTML son la forma en la que el usuario interactúa con una aplicación, por esto es necesario disparar eventos hacia el código que es capaz de ejecutar la lógica de negocio, para hacerlo lo haremos con lo siguiente:
  - Event binding : **(event)="expression"**
- **Combinación de ambos** (Two way binding): También seremos capaces de ejecutar ambos al mismo tiempo con esto seremos capaces de disparar eventos y mostrar información al mismo del siguiente modo:
  - Two way data binding : **[(ngModel)]="data"**

Con esto seremos capaces de realizar la comunicación entre nuestro sitio código typescript y nuestros templates html, veamos algunos ejemplos de cada modo. Se tomará como base para este post el ejemplo [Crear componentes en Angular 4](https://geeks-mexico.com/2017/10/11/crear-componentes-en-angular-4/) (<https://geeks-mexico.com/2017/10/11/crear-componentes-en-angular-4/>).

## String interpolation

String interpolation funciona muy parecido a expression language, con esto es posible tanto declarar variables en el archivo typescript e utilizarlas dentro de nuestro html como

escribir cualquier expresión que pueda ser traducida a un String veamos un ejemplo editando nuestro componente posts.components:

### posts.components.ts

```
1  import { Component } from '@angular/core'
2  @Component({
3    selector: 'app-posts',
4    templateUrl: './posts.component.html'
5  })
6  export class PostsComponent {
7    author='@raidentrance';
8    title='Angular from scratch';
9
10   getTitle(){
11     return this.title;
12   }
13 }
```

*Como se puede observar, dentro de nuestro componente PostsComponent declararemos 2 atributos el atributo author y el atributo title y los inicializaremos con algunos valores por defecto. Es importante mencionar que es posible declarar métodos que devuelvan el valor de alguno de nuestros atributos, muestra de esto es el método getTitle().*

### posts.component.html

```
1  <div align="center">
2    <h3>Post List</h3>
3    <div>
4      <table class="table">
5        <tr>
6          <th>Author Twitter</th>
7          <th>Post name</th>
8          <th>Post date</th>
9        </tr>
10       <tr>
11         <td>{{author}}</td>
12         <td>{{getTitle()}}</td>
13         <td>{{'19/08/1988'}}</td>
14       </tr>
15     </table>
16   </div>
17 </div>
```

*Es momento de analizar el funcionamiento del archivo HTML, como se puede observar en la tabla se utilizarán los símbolos `{{}}` para definir que utilizaremos String interpolation y se colocarán las*

*variables definidas dentro de nuestra clase en typescript, a demás es importante resaltar que la fecha fue declarada como una expresión String, por esto es posible colocarla dentro de nuestro String interpolation y que del mismo modo que podemos obtener el valor de un atributo podemos obtener el resultado de un método.*

## Property binding

Una vez que entendimos como funciona String interpolation es tiempo de entender como funciona property binding, para esto agregaremos un botón a nuestro componente posts y lo modificaremos de acuerdo a un evento dirigido por un componente typescript, veamos el ejemplo:

### posts.component.ts

```
1  import { Component } from '@angular/core'
2  @Component({
3    selector: 'app-posts',
4    templateUrl: './posts.component.html'
5  })
6  export class PostsComponent {
7    author = '@raidentrance';
8    title = 'Angular from scratch';
9    disableNewPost = true;
10
11    constructor() {
12      setTimeout(() => { this.disableNewPost
13    }
14
15    getTitle() {
16      return this.title;
17    }
18  }
```

*Podrás notar que al código anterior se realizaron las siguientes modificaciones:*

- Se le agregó el atributo `disableNewPost` inicializado en `true`, con esto seremos capaces de determinar que un botón que se agregará se encontrará desactivado al principio
- Se declaró un constructor el cual se ejecutará cuando se cree el objeto de nuestra clase

- Se utilizó la función `setTimeout()` la cual definirá que una vez pasados 2 segundos se cambiará el valor de la variable `disableNewPost` a `false` lo cual permitirá que el botón se active después de este periodo de tiempo

## posts.components.html

```

1  <div align="center">
2    <h3>Post List</h3>
3
4    <button class="btn btn-primery" [disabled]:
5    <div>
6      <table class="table">
7        <tr>
8          <th>Author Twitter</th>
9          <th>Post name</th>
10         <th>Post date</th>
11        </tr>
12        <tr>
13          <td>{{author}}</td>
14          <td>{{getTitle()}}</td>
15          <td>{{'19/08/1988'}}</td>
16        </tr>
17      </table>
18    </div>
19  </div>

```

Al principio del archivo HTML se agregó un botón nuevo con el texto *Create*, como ya sabrán el tag `button` contiene una propiedad tipo boolean llamada `disabled`, con esta es posible determinar si el elemento será habilitado o no. Utilizando *property binding* determinaremos el valor de este atributos de acuerdo al valor del atributo `disableNewPost` que se encuentra en nuestro archivo `typescript`, con esto el botón se encontrará des habilitado durante 2 segundos y pasando esto se habilitará de nuevo sin necesidad de actualizar algún elemento del DOM.



En el siguiente post se explicarán los otros modos disponibles en Angular 4 para realizar el databinding en nuestras aplicaciones. Si tienes alguna duda o comentario no dudes en hacerlo en la sección de comentarios o en nuestras redes sociales.

*Autor: Alejandro Agapito Bautista*

*Twitter: @raidentrance*

*Contacto:raidentrance@gmail.com*

Anuncios

