

Angular IFCD65

Formularios

Hay dos tipos de formularios en angular:

Template-driven forms

Utilizan más etiquetas y atributos

Las validaciones se ponen en el formulario

Reactive Form (AKA data-driven forms)

Se definen formularios puros y se trabaja todo en programación.

Las validaciones se ponen en el ts

Formularios

Creamos un proyecto de formularios
ng new formularios

Creamos los componentes
Creamos 1 formulario de tipo template
ng g c componentes/template --skip-tests

Añadimos la llamada al componente template
Modificamos el app.component.html

```
<div class="contenedor">  
  <h1>Formularios</h1>  
  <hr>  
  <app-template></app-template>  
</div>
```

Formularios

Añadimos **FormsModule** al **@NgModule**

En el app.module.ts hay que importar el **FormsModule**

```
...  
import { FormsModule } from '@angular/forms';  
  
@NgModule({  
  declarations: [  
    AppComponent, TemplateComponent, ReactiveComponent  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    FormsModule  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

Formador: Borja Mulleras Vinzia

Formularios

Escribimos el código HTML
Modificamos template.component.html

```
<h3>Formularios template-drive</h3>
<form [noValidate]="false"> <!-- A true no haría caso de las validaciones de html -->
  <div>
    <label for="nombre">Nombre</label>
    <input id="nombre" name="nombre" type="text" placeholder="Nombre">
  </div>
  <div>
    <label for="apellido">Apellido</label>
    <input id="apellido" name="apellido" type="text" placeholder="apellido">
  </div>
  <div>
    <label for="correo">Correo</label>
    <input id="correo" name="correo" type="email" placeholder="Correo">
  </div>
  <div>
    <button>Guardar</button>
  </div>
</form>
```

Formularios

Directiva ngModel

Añadimos este atributo en el input del formulario sin ningún valor.

Esta directiva le da el control a angular del campo.

Es importante que los campos tengan el campo name

```
<div>  
  <label for="nombre">Nombre</label>  
  <input id="nombre"  
        name="nombre"  
        type="text"  
        placeholder="Nombre"  
        ngModel  
  >  
</div>
```

Formularios

Directiva ngModel

Si examinamos el campo sin meternos dentro, vemos las siguientes clases
ng-untouched ng-pristine ng-valid

A la que nos metemos y escribimos:
ng-valid ng-dirty ng-touched

ng-dirty indica si se ha escrito algo
ng-touched o ng-untouched si está seleccionado el campo o no
ng-valid si el valor es válido. (ejemplo con required)

Formularios

Directiva ngModel

Si en un campo input le añadimos un atributo con **#clases** esto nos permite visualizar en el html las clases de ese atributo.

```
<input id="nombre"  
      name="nombre"  
      type="text"  
      placeholder="Nombre"  
      ngModel  
      required  
      #clases>  
{{clases.className}}
```


Formularios

ngSubmit

En la etiqueta <form> añadimos
(ngSubmit)="guardar()"

Y en template.component.js añadimos el método guardar

```
export class TemplateComponent {  
  public guardar() {  
    console.log('Formulario guardado');  
  }  
}
```

Formularios

ngSubmit

Ahora queremos pasarle al ts los datos del formulario.

(ngSubmit)="guardar(**datosFormulario**)" #datosFormulario="ngForm"

Y en template.component.js añadimos el método guardar

```
export class TemplateComponent {  
  public guardar(formulario:any) {  
    console.log(formulario.value);  
    console.log('Formulario guardado');  
  }  
}
```

La directiva ngForm va a enviar al ts todos los datos del formulario

Formularios

ngSubmit

El tipo de dato no es any, sino NgForm

```
import { Component } from '@angular/core';
import { NgForm } from '@angular/forms';
@Component({
  selector: 'app-template',
  templateUrl: './template.component.html',
  styleUrls: ['./template.component.css']
})
export class TemplateComponent {
  public guardar(formulario:NgForm) {
    console.log(formulario.value);
    console.log('Formulario guardado');
  }
}
```

Formularios

ngSubmit

Ahora nos creamos un interface con los datos del formulario

ng g i modelos/usuario

```
export interface Usuario {  
    nombre:string | null,  
    apellido:string | null,  
    email:string | null  
}
```

Formularios

En template.component.ts hay que añadir en el constructor un usuario

```
...  
import {Usuario} from 'src/app/modelos/usuario';  
...  
public usuario:Usuario;  
  
  constructor() {  
    this.usuario {  
      nombre: "Jorge",  
      apellido: "Ibañez",  
      email:"jorge.ibanez@gmail.com"  
    }  
  }  
  }  
...
```

Formularios

Ahora modificamos en cada campo del formulario la directiva ngModel
`[ngModel]="usuario.nombre"`

```
<div>
    <label for="nombre">Nombre</label>
    <input id="nombre" name="nombre" type="text" placeholder="Nombre"
    [ngModel]="usuario.nombre"
    required #clases>
</div>
<div>
    <label for="apellido">Apellido</label>
    <input id="apellido" name="apellido" type="text" placeholder="apellido"
    [ngModel]="usuario.apellido">
</div>
<div>
    <label for="correo">Correo</label>
    <input id="correo" name="correo" type="email" placeholder="Correo"
    [ngModel]="usuario.email">
</div>
    <button>Guardar</button>
</div>
```

Formularios

Finalmente, debemos asignar los datos del formulario al objeto del usuario

```
public guardar(formulario:NgForm) {  
    //trasladar datos del formulario al objeto  
    this.usuario.nombre=formulario.value.nombre;  
    this.usuario.apellido=formulario.value.apellido;  
    this.usuario.email=formulario.value.email;  
    console.log('Usuario modificado',this.usuario);  
    console.log('Formulario guardado');  
}
```

Pero hay una forma más directa de hacer este paso:

En el formulario ponemos:

```
[(ngModel)]="usuario.nombre"
```

Angular 19

Añadir los formularios necesarios en el proyecto de todo

Instalación de un proyecto

Pasos a seguir para crear nuestro proyecto en producción

PASO 1
ng build

PASO 2
Subimos el contenido de la carpeta dist a un servidor

PASO 3
Editamos el index.html y modificamos la línea base con la ruta de donde se encuentra nuestro proyecto en el servidor

Instalación de un proyecto

```
<!doctype html>
<html lang="en" data-critters-container>
<head>
  <meta charset="utf-8">
  <title>Formularios</title>
  <base href="/borjaMulleras/formularios/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
<link rel="stylesheet" href="styles.ef46db3751d8e999.css"></head>
<body>
  <app-root></app-root>
<script src="runtime.ec122ea712de93fa.js" type="module"></script><script
src="polyfills.219bbaf54b58a23c.js" type="module"></script><script
src="main.d45fde19b8f5bb23.js" type="module"></script></body>
</html>
```

Angular

IFCD65



Formador: Borja Mulleras Vinzia