

Cambios en Angular 6

Angular 4

¿Qué pasó con Angular 3?

Todas las librerías de Angular 2 tenían una nomenclatura del tipo angular 2.X.X, salvo @angular/router que tenía versiones 3.X.X, para igualar las versiones de todos los paquetes de Angular 4, decidieron saltar la versión 3 y evitar la incompatibilidad de tener todos los paquetes de angular 3 en su versión 3.X.X, salvo @angular/router con la 4.X.X

Mejoras de eficiencia en Angular 4

Mejoras de la compilación AOT de 100KB.

Hay dos tipos de compilaciones:

- **JIT: Just In Time**
 - La compilación se hace en el navegador en tiempo de ejecución
 - La descarga de la aplicación web incluye la mitad de la librería de angular
 - Los templates no se precompilan, de modo que también lo hace el navegador
- **AOT: Ahead Of Time**
 - La compilación se realiza en el despliegue, con el parámetro `--aot` (`ng build --aot`)
 - La ejecución de la aplicación es más rápida y necesita menos tamaño de descarga

[Documentación de Angular sobre AOT](#)

Paquete animations de Angular

El paquete Animations ya no pertenece al paquete `@angular/core`, sino que pasa a `@angular/platform-browser/animations`.

De este modo, las aplicaciones que no incluyen animaciones, que son muchas, no tienen que descargar el paquete animations que no es pequeño.

Nuevas características

- Typescript 2.0 con sus [nuevas características](#).
 - -strictNullChecks: con esta opción habilitada, sólo es posible asignar null o undefined a un tipo any.
- <template> es ahora <ng-template> se confundía con el tag de HTML5 <template>.
- ngIf tiene la posibilidad de introducir un else de esta forma:

```
<div *ngIf="races.length > 0; else empty"><h2>Races</h2></div>  
<ng-template #empty><h2>No races.</h2></ng-template>
```

Nuevas características

- Nueva sentencia '**as**':

```
<div *ngFor="let pony of ponies | slice:0:2 as total; index as i">  
  {{i+1}}/{{total.length}}: {{pony.name}}  
</div>
```

- Pipe Titlecase

```
<p>{{ 'curso angular' | titlecase }}</p>  
<!-- will display 'Curso Angular' -->
```

- Nuevo validator para formularios: email que se suma a required minLength maxLength y pattern

... Y más novedades

Angular 5

Nuevos eventos en Angular Router

La librería de Router incorpora los siguientes eventos para mejorar el control de acceso y la navegación por las páginas:

- `ActivationStart`
- `ActivationEnd`
- `ChildActivationStart`
- `ChildActivationEnd`
- `GuardsCheckEnd`
- `GuardsCheckStart`
- `ResolveStart`
- `ResolveEnd`

HttpClient por Http

Http ahora está marcada como deprecada y en su lugar, podemos usar HttpClient, con las siguientes ventajas:

- Acceso síncrono y tipado al cuerpo de la respuesta de la API
- JSON se asume como el tipo por defecto de la comunicación
- Los interceptores permiten introducir lógica intermedia en el pipeline de procesamiento de la respuesta
- Los objetos de la petición o respuesta son inmutables
- Dispone de eventos en el progreso de las solicitudes de subida y descarga de la respuesta.
- Disponemos también de verificación post-respuesta.

Internacionalización de pipes

Angular 5 dispone de nuevos pipes relativos a numbers, fecha y moneda que eliminan la necesidad de utilizar l18n para la internacionalización de formatos, supuestamente esta gestión se hará de forma automática y se aplicarán los locales correspondientes a la zona en la que trabajemos

Mejoras en validaciones de formularios

Podemos decidir cuándo se actualizará la validez de un input o incluso un formulario completo: vía onBlur o por onSubmit, en lugar de hacerlo en cada evento que se produzca dentro del input.

```
<input name="nombreCampo" ngModel [ngModelOptions]="{updateOn: 'blur'}">
```

En el formulario:

```
<form [ngFormOptions]="{updateOn: 'submit'}">
```

Mejoras en las animaciones

Se ha mejorado el comportamiento de ciertas animaciones y se han añadido algunas opciones nuevas de transiciones: `:increment` y `:decrement`

Los queries ahora soportan límites negativos, con lo que podremos comenzar por el final del elemento en lugar de por el principio.

Angular 6

Mejoras en el ng-client

ng update te permite mantener actualizadas y sincronizadas las dependencias de tu proyecto. Ejemplo: ***ng update @angular/core***

ng add permite instalar paquetes de dependencias. Ejemplos:

ng add @angular/pwa: Convierte tu web en una PWA y le añade el app manifest y el service worker.

ng add @angular/material: Instala y configura Angular Material, configura los themes e incorpora schematics de nuevos componentes

ng add @angular/elements: Añade las dependencias de Angular Elements así como el polyfill que requieren (document-register-element.js).

Angular Elements

Angular Elements permite generar Custom Elements (una parte de la especificación de Web Components) a partir de componentes Angular.

Esta primera release facilita la carga de componentes Angular externos dentro de una app Angular ya existente, registrándose como Custom Elements. A partir de ahora, no necesitarás inicializar manualmente esos componentes Angular que encuentras en el contenido html estático servido por un CMS, la carga de estos componentes la hace directamente el navegador.

Angular Material

Angular Material v6 trae consigo 4 nuevos componentes:

- **Tree:** Permite mostrar contenido jerárquico.
- **Flexible Overlays:** Permite gestionar el tamaño y posición independientemente del contenido.
- **Badge:** Un componente nuevo para indicar notificaciones.
- **Bottom-Sheet:** Componente tipo menú desplegable desde la parte inferior de la pantalla.

Material starters

- Material Sidenav: Componente de inicio, responsive, que incluye una toolbar y un menu de navegación lateral:

```
ng generate @angular/material:material-nav --name=nombre-de-mi-comp
```

- Material Dashboard: Componente de inicio, que incluye un grid dinámico de tarjetas:

```
ng generate @angular/material:material-dashboard --name=nombre-de-mi-comp
```

- Material Data Table: Componente de inicio, tipo tabla de datos, preconfigurado con un datasource para paginación y ordenación

```
ng generate @angular/material:material-table --name=nombre-de-mi-comp
```

Declaración de providers

Hasta la versión 5, los providers se declaraban o bien en el módulo, o bien en un componente concreto. A partir de Angular 6, se puede especificar en qué nivel hay que introducir el provider desde el propio decorador Provider.

```
@Injectable({  
  providedIn: 'root',  
})
```