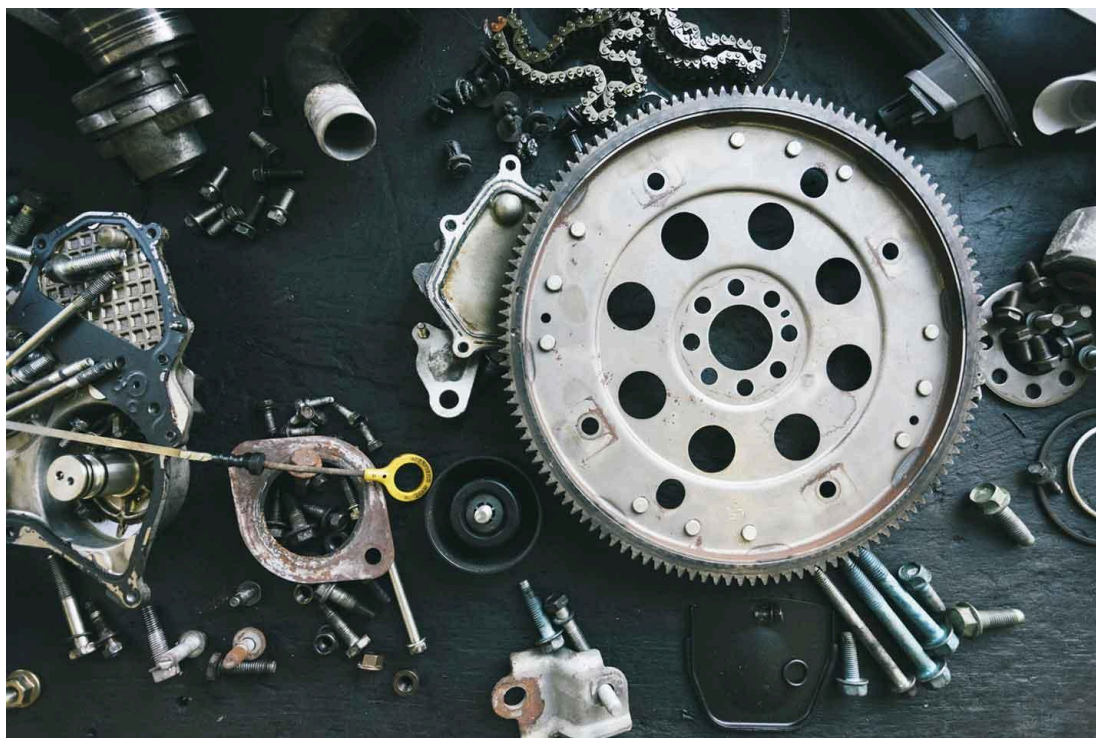


IONIC v8

Slides y Toast



Progresa
Una formación de futuro

ÍNDICE

Introducción Slides	3
Preparación	3
Componente	3
Icono	3
Ejemplo	4
Estilos	4
Visualización	4
Explicación	5
Introducción Toast	5
Servicio	5
Usando el Toast	6
Evento que lo lanza	6
Lógica	7



INTRODUCCIÓN SLIDES

Es un contenedor multisección. Cada sección puede ser desplazada o arrastrada. Contiene cualquier número de componentes Slide.

La documentación de los slides está en la página:

<https://ionicframework.com/docs/angular/slides>

PREPARACIÓN

COMPONENTE

Ahora que ya tenemos las configuraciones básicas, vamos a empezar a crear contenido. Lo primero que vamos a hacer es crear una página para mostrar los slides. En la **terminal**:

```
ionic generate page pages/slides2 --spec=false
```

ICONO

Añadimos en nuestro **app.component.ts** el icono que queremos usar para el menú en la lista de iconos(addIcons).

```
addIcons({grid, logoAppleAppstore, americanFootball, beaker, star,
trash, shareSocial, caretForwardCircle, close, card, pin, car,
logoFacebook, logoTwitter, logoTiktok, logoInstagram, logoYoutube,
logoGithub, add, albums, ellipsis-horizontal-outline});
```

Añadimos en nuestro **menu.json** el nuevo elemento que queremos que aparezca tanto en el menú como en la página de inicio.

```
, {
  "nombre": "Slides 2",
  "ruta": "/slides2",
  "icono": "ellipsis-horizontal-outline"
}
```

EJEMPLO

ESTILOS

A continuación empezamos con unos estilos propios que tendrán nuestras tarjetas que mostraremos en las slides. En [slides2.page.scss](#):

```
.ion-avatar{
  padding: 2px;
  border: solid 2px var(--ion-color-primary);
}
swiper-container{
  padding-top: 10px;
  padding-bottom: 10px;
}
ion-avatar{
  margin-left: auto;
  margin-right: auto;
}
```

VISUALIZACIÓN

Y por último vamos a mostrar las slides en [slides2.page.html](#):

```
<app-header titulo="Slides 2"></app-header>
<ion-content [fullscreen]="true" class="ion-padding">
  <swiper-container [slidesPerView]="4.5"
    [separator]="10">
    @for (a of ['Chibi', 'Chibi2', 'Chibi3', 'Chibi4', 'Chibi5', 'Chibi6']; track $index) {
      <swiper-slide>
        <ion-avatar class="ion-avatar ">
          <ion-img src="/assets/images/chibi.jpg"></ion-img>
        </ion-avatar>
        <ion-label><p class="ion-text-center">{{a}}</p></ion-label>
      </swiper-slide>
    }
  </swiper-container>

  <h2>Página de Slides</h2>
</ion-content>
```



Explicación

```
[slidesPerView]="4.5"
```

Indicamos cuántos slides aparecerán por pantalla.

```
[separator]="10"
```

Indica la distancia de separación entre slides.

INTRODUCCIÓN TOAST

Un toast es una notificación sutil usada normalmente en aplicaciones normales. Puede ser usada para proporcionar información acerca de una operación o enviar un mensaje del sistema. El toast aparece en la parte superior de la aplicación y puede ser eliminada por la propia aplicación o por interacción con el usuario.

SERVICIO

Vamos ahora a añadirle una nueva funcionalidad a nuestros elementos de las slides, para lanzar un toast cuando hagamos click. Vamos a crear un servicio que administre el toast para reutilizarlo en nuestra aplicación. En la **terminal**:

```
ionic generate service services/toast --skip-tests
```

Y en recién creado `toast.service.ts`:

```
import { Injectable } from '@angular/core';
import { ToastController } from '@ionic/angular';

@Injectable({
  providedIn: 'root'
})
export class ToastService {
```

```

constructor(private toastCtrl: ToastController) { }

  async cargarToast(message: string,duration:
number,color: string){
    const toast = await this.toastCtrl.create({
      message,
      duration,
      color,
      position: 'middle'
    });
    await toast.present();
  }
}

```

USANDO EL TOAST

Evento que lo lanza

Y modificamos nuestro html para detectar el click. En [slides2.page.html](#):

```

<app-header titulo="Slides 2"></app-header>
<ion-content [fullscreen]="true" class="ion-padding">
  <swiper-container [slidesPerView]="4.5"
    [separator]="10">
    @for (a of ['Chibi','Chibi2','Chibi3','Chibi4','Chibi5','Chibi6']; track
    $index) {
      <swiper-slide (click)="swiperClick($event,a)" >
        <ion-avatar class="ion-avatar ">
          <ion-img src="/assets/images/chibi.jpg"></ion-img>
        </ion-avatar>
        <ion-label><p
class="ion-text-center">{{a}}</p></ion-label>
      </swiper-slide>
    }
  </swiper-container>
  <h2>Página de Slides</h2>
</ion-content>

```

Lógica

Y por último definimos la función del evento click. En [slides2.page.ts](#):

```
import { Component, CUSTOM_ELEMENTS_SCHEMA, inject, OnInit }
from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { IonAvatar, IonContent, IonHeader, IonImg, IonLabel,
IonTitle, IonToolbar } from '@ionic/angular/standalone';
import { HeaderComponent } from
"../../components/header/header.component";
import { ToastService } from "../../services/toast.service";

@Component({
  selector: 'app-slides2',
  templateUrl: './slides2.page.html',
  styleUrls: ['./slides2.page.scss'],
  standalone: true,
  imports: [IonContent, IonHeader, IonTitle, IonToolbar,
CommonModule, FormsModule, HeaderComponent, IonAvatar, IonImg,
IonLabel],
  schemas: [CUSTOM_ELEMENTS_SCHEMA]
})
export class Slides2Page implements OnInit {
  private toastService = inject(ToastService);

  constructor() { }

  ngOnInit() {

  }

  swiperClick(event: MouseEvent, a: string) {
    this.toastService.cargarToast(a, 1500, 'success');
  }
}
```