



Conoce a los Guardianes 19/25



RECURSOS

APUNTES

Una aplicación tendrá N cantidad de rutas, pero es posible que muchas de estas tengan restricciones de acceso solo para administradores o determinados roles de usuario.

En estos casos, los **Guards** llegan para ayudarnos a darle seguridad a nuestras rutas.

Cómo securizar Rutas

Con ayuda del CLI de Angular, puedes crear tu primer guardián con el comando

```
ng generate guard <nombre-guard> o en su forma corta ng g g <nombre-guard> .
```

1. Creando el primer guard

Al utilizar este comando, nos hará una pregunta sobre qué interfaz quieres que implemente por defecto:

```
? Which interfaces would you like to implement? (Press <space> to select, <a> to toggle all, <i> to invert selection)
> CanActivate
  o CanActivateChild
  o CanDeactivate
  o CanLoad
```



Cada opción tiene una funcionalidad distinta para cada tipo de Guard. Escoge la primera opción llamada **CanActivate**.

Al auto generar el código, verás tu primer Guard con el siguiente aspecto.

```
// modules/shared/guards/admin.guard.ts
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot } from
 '@angular/router';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AdminGuard implements CanActivate {

  canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot):
  Observable<boolean> | Promise<boolean> | boolean {
    return true;
  }
}
```

Un Guard puede devolver un booleano, una promesa con un booleano o un observable, también con un booleano. Dependiendo la lógica que tengas que aplicar para el caso sea síncrona o asíncrona.

2. Importando el guard

Ahora importa el nuevo Guard el routing de tu aplicación.

```
// app-routing.module.ts
import { AdminGuard } from './modules/shared/guards/admin.guard';

const routes: Routes = [
  {
    path: '',
    loadChildren: () => import('./modules/website/website.module').then(m =>
m.WebsiteModule),
    data: { preload: true },
  },
  {
    path: 'cms',
    loadChildren: () => import('./modules/cms/cms.module').then(m => m.CmsModule),
    canActivate: [ AdminGuard ],
    data: { preload: true },
  }
];
```

Agrégle a las rutas que quieras asegurar `canActivate: [AdminGuard]`.

De esta manera, ya puedes implementar la lógica que necesites para cada Guard. En este caso, permitir el acceso al módulo CMS solo para usuarios administradores.

Contribución creada por: Kevin Fiorentino.

Archivos de la clase

