

API > @angular/core



OnDestroy INTERFACE

A lifecycle hook that is called when a directive, pipe, or service is destroyed. Use for any custom cleanup that needs to occur when the instance is destroyed.

```
interface OnDestroy {  
  ngOnDestroy(): void  
}
```

Class implementations

- [NgComponentOutlet](#)
- [AsyncPipe](#)
- [HashLocationStrategy](#)
- [Location](#)
 - [SpyLocation](#)
- [PathLocationStrategy](#)
- [AbstractFormGroupDirective](#)
 - [NgModelGroup](#)
 - [FormGroupName](#)
- [NgModel](#)
- [NgModelGroup](#)
- [RadioControlValueAccessor](#)
- [FormControlDirective](#)
- [FormControlName](#)
- [FormGroupDirective](#)
- [FormArrayName](#)
- [FormGroupName](#)
- [NgSelectOption](#)
- [RouterLinkWithHref](#)
- [RouterLinkActive](#)
- [RouterOutlet](#)
- [RouterPreloader](#)
- [UpgradeComponent](#)

See also

- [Lifecycle hooks guide](#)

Methods

ngOnDestroy()



A callback method that performs custom clean-up, invoked immediately before a directive, pipe, or service instance is destroyed.

ngOnDestroy(): void

Parameters

There are no parameters.

Returns

void

Usage notes

The following snippet shows how a component can implement this interface to define its own custom clean-up method.

```
@Component({selector: 'my-cmp', template: `...`})  
  
class MyComponent implements OnDestroy {  
  ngOnDestroy() {  
    // ...  
  }  
}
```