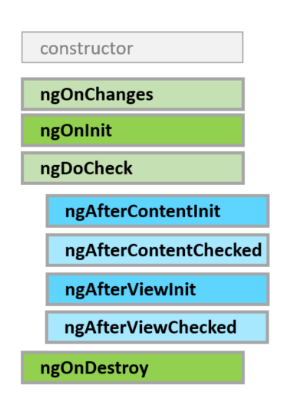
Component Lifecycle

Angular Create Component , renders it , renders its children checks it when its data bound property change and destroys it before removing it from DOM.



Angular Offers Lifecycle hooks that provide visibility into these key life moments. Developers can use them by implementing the interfaces and each interface has a single hook.

LifeCycle Hooks and its Sequence

1. ngOnChanges : When data bound property value changes. Its called before ngOnInit.
2. ngOnInit :Initialize the directive/Component after angular first display data bound properties and sets the directive/component input properties. When component initialization, after first ngOnChanges. It execute only once.
3. ngDoCheck : Detect and act upon changes . it is called after every ngOnChnages and ngOnInit.During every Angular2 Change detection cycle.
4. ngAfterContentInit :Respond after Angular projects external content into components view. After inserting content using <ng-content>.
5. ngAfterContentChecked : Responds after Angular checks the content projected into component.After every check of inserted Content.
6. ngAfterViewInit : Responds after Angular initialize the Components view and child views. Called after ngAfterContentChecked.After Initializing the components views/child views.
7. ngAfterViewChecked : Respond after Anuglar Checks for Component Views.After every Check of the components views/child views.
8. ngOnDestroy : Clean up just before Angular destroys the directive/component. Unsubsscribe observables and detach event handlers to avoid memory leaks.just before Angular 2 destroys the directive/component.

Can we use this in real world applications??

1. OnInit

* To perform complex initialization shortly after construction.
* To set up the component after Angular sets the input properties.

Rule :

Don’t fetch data in component constructor . it should only be used to initialize the things.

ngOnInit : it’s a good place for a component to fetch its initial data.

OnDestroy

* Put Clean up logic in ngOnDestroy .
* This logic must be run before Angular destroys the directive.

It can be use to notify the another part of the application that component is going away.

OnChanges

* Angular calls its ngOnChanges method whenever it detects changes to input properties of the component. This hooks iterates over the changed properties and logs them.

DoCheck

* Use the doCheck hook to detect and act upon changes.
* It can be used to capture and compare the current state against previous values.