

Project setup

Keep things organized in folders (Modules are not the default as of 2024). All pages, components, routes, validators, services that belong together should have the same (grand)parent.

Project setup commands - sample

```
# When setting up a new project think about how
you want it to look. Make a short list of commands
to set up your project. Open the new project in
your IDE, if things do not feel right, adjust your
list and run it again. I have added an example
below.
ng new sample
cd sample
ng g c core/home page/c ontainers /homepage
ng g c core/product s/c ontainers /products
ng g c core/product s/c ontainers /product-
edit
ng g c core/product s/c ontainers /product-
edit
ng g c core/product s/c ontainers /product-
add
ng g c core/product s/c omponents /product-
form
ng g s core/product s/service s/product
ng g c core/contact /contact
# Make many more so you get a good feel of how your
decisions will impact the project. Make changes,
delete the project and run you commands again.
# Add a route to the Product sC omponent at
/products
# Use the Product sService in the constr uctor of
the Product sC omponent
```

Life cycle hooks (cont)

ngDoCheck()	Detect and act upon changes that Angular can't or won't detect on its own. Called during every change detection run, immediately after ngOnChanges() and ngOnInit().
ngAfterContentInit()	Respond after Angular projects external content into the component's view / the view that a directive is in. Called once after the first ngDoCheck().
ngAfterViewInit()	Respond after Angular initializes the component's views and child views / the view that a directive is in. Called once after the first ngAfterContentChecked().
ngAfterViewChecked()	Respond after Angular checks the component's views and child views / the view that a directive is in. Called after the ngAfterViewInit() and every subsequent ngAfterContentChecked().
ngOnDestroy()	Cleanup just before Angular destroys the directive/component. Unsubscribe Observables and detach event handlers to avoid memory leaks. Called just before Angular destroys the directive/component.

Life cycle hooks

ngOnChanges()	Respond when Angular (re)sets data-bound input properties. The method receives a SimpleChanges object of current and previous property values. Called before ngOnInit() and whenever one or more data-bound input properties change.
ngOnInit()	Initialize the directive/component after Angular first displays the data-bound properties and sets the directive/component's input properties. Called once, after the first ngOnChanges().



By Robert Broen (rbroen)
cheatography.com/rbroen/

Not published yet.
Last updated 5th August, 2024.
Page 1 of 1.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>