Angular: Getting stared Summary

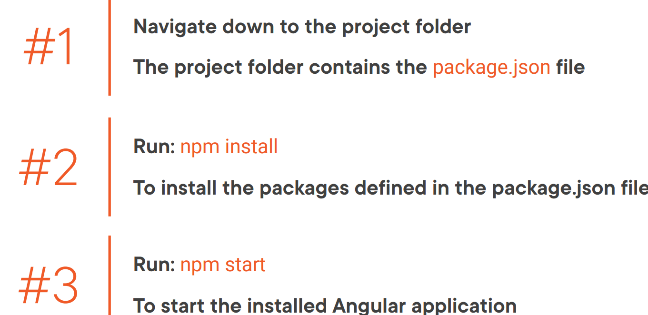
Serviceware  
Mack Hodge

7 July 2022

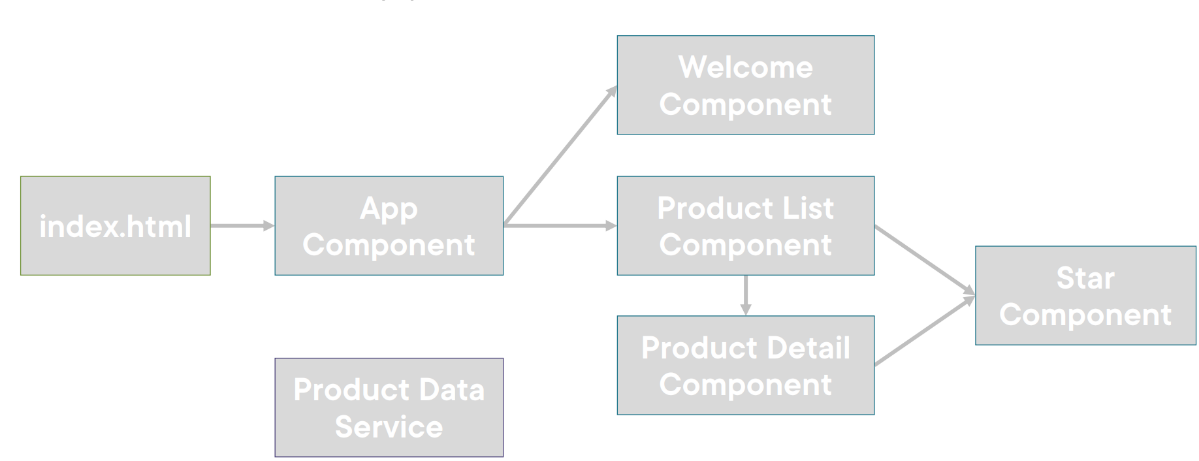
## package.json



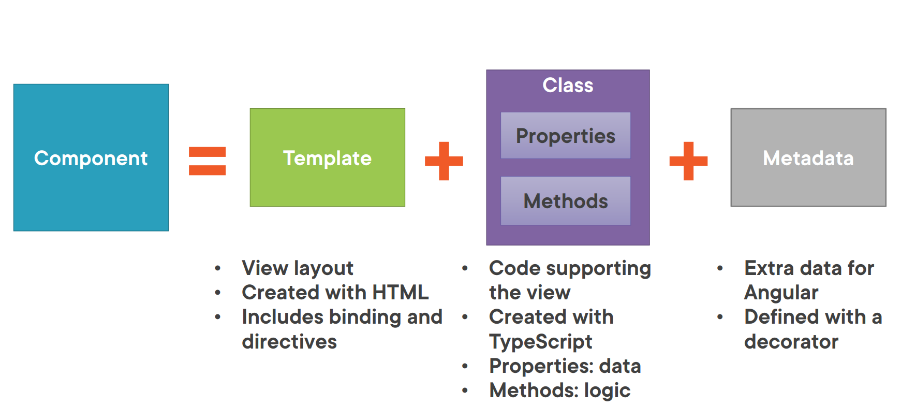
## When Setting Up Existing Angular Code



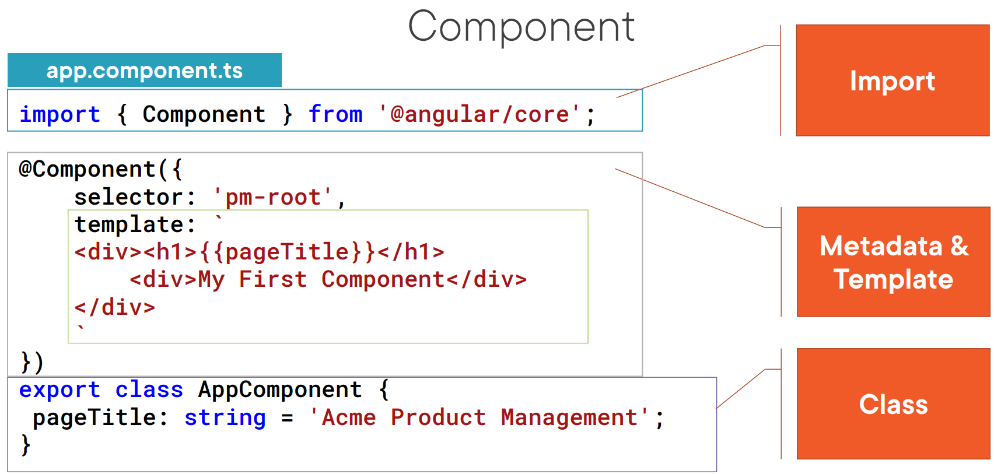
## Application Architecture



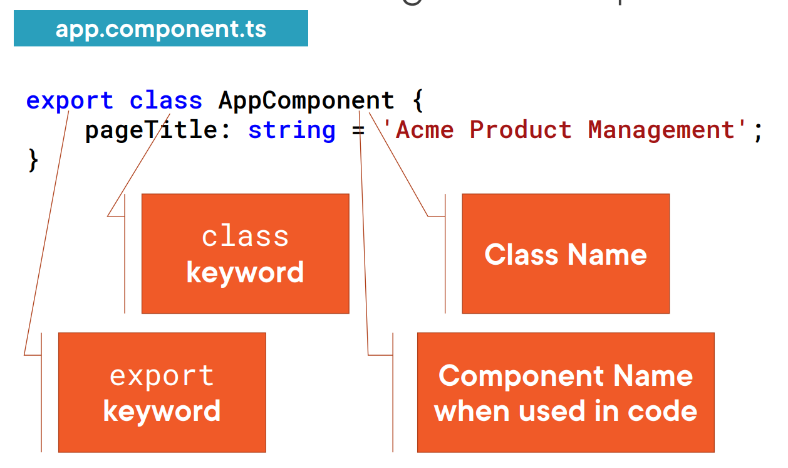
## What is a component?



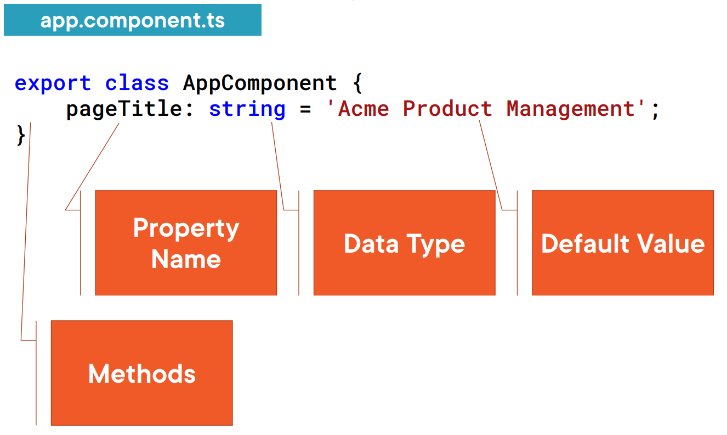
## Component



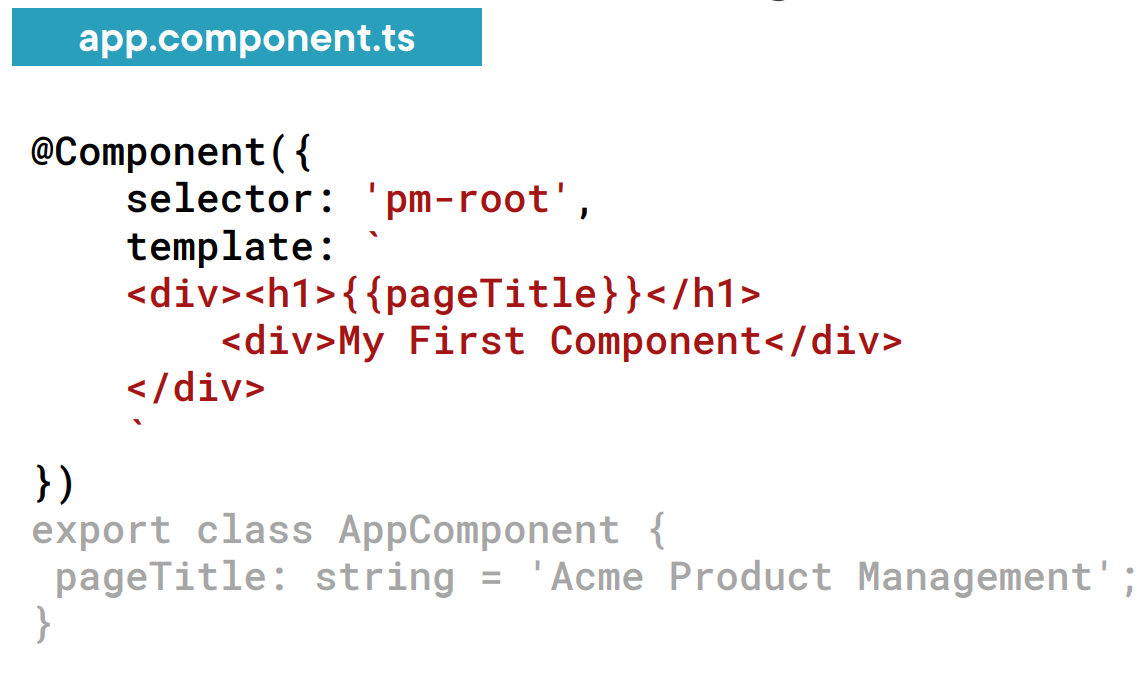
## Creating the Component Class 1



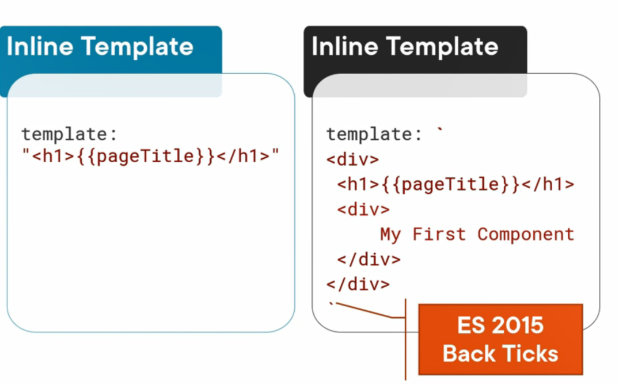
## Creating the Component Class 2



## Defining the Metadata

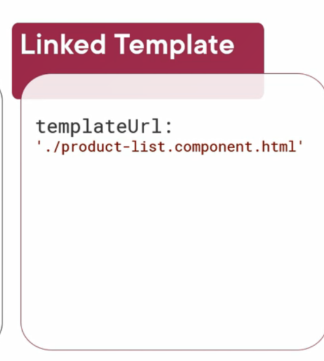


# **Templates, Interpolation, And Directives**



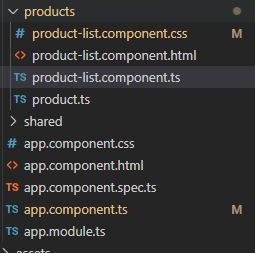
Back ticks   
**Pros**: Make the HTML more readable   
 Template is directly divined with in a component, keeping the code for the view in one file

**Const**: no intelliSense



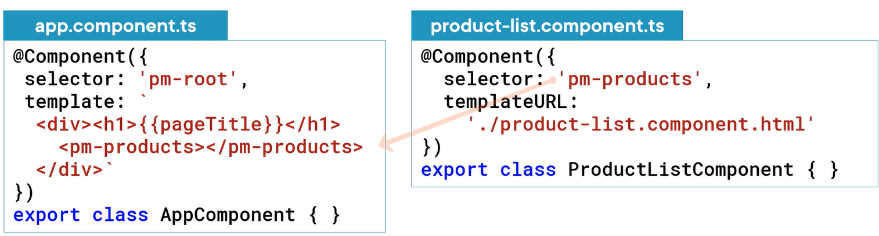
Linked template is a better approach

## Naming convention

****

Use always use .component for a component

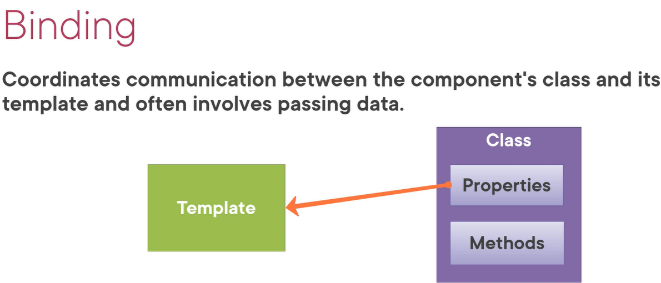
## Using component as a Directive

Selector means that we can use the component as a directive in any other component. Therefore, using this component elsewhere, we can also use its templates and the component properties.

**Quick note**

* Every page needs an app.module.ts that puts everything together
* AngularJs has such thing ?

## Binding with interpolation

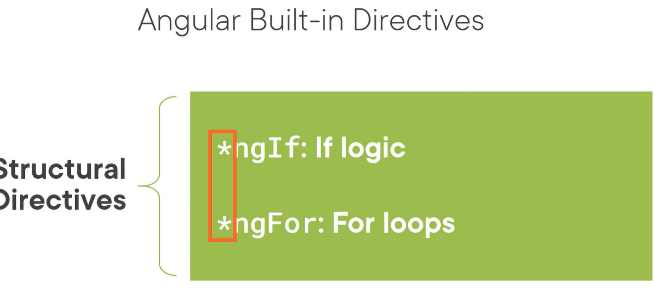


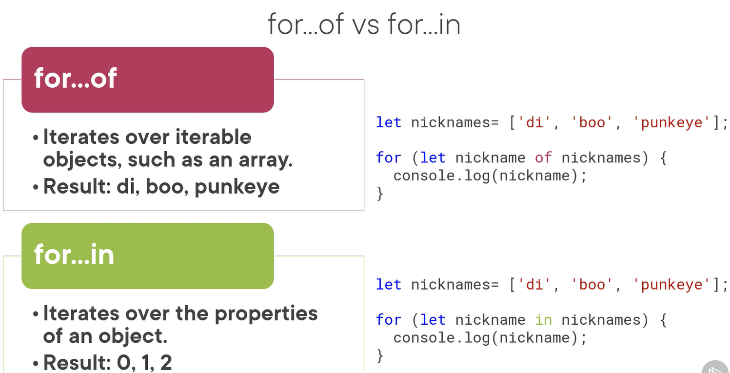


**Quick note**

* Interpolation is a one-way binding
* H1 is an template expressing, both h1 display the same thing

## Adding logic with directives

****

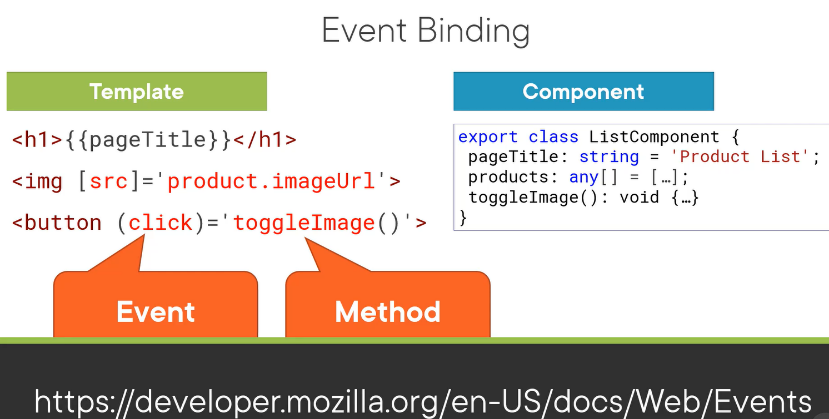


**Quick note**

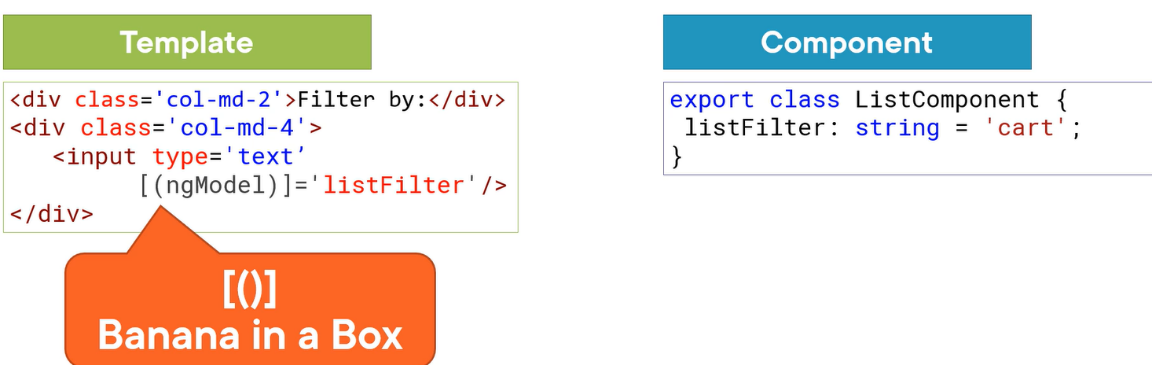
* We can also use angular build in directive
* Notice the \*
* Ng if length == 0 the dom is not shown
* BrowserModule is imported in the app module
* ngFor
* for ..of result in its properties for …in result in its index in the list
* Define local variable with let

# Data Binding & Pipes

## Handling Events



## Handling input with two-way binding



## Pipes

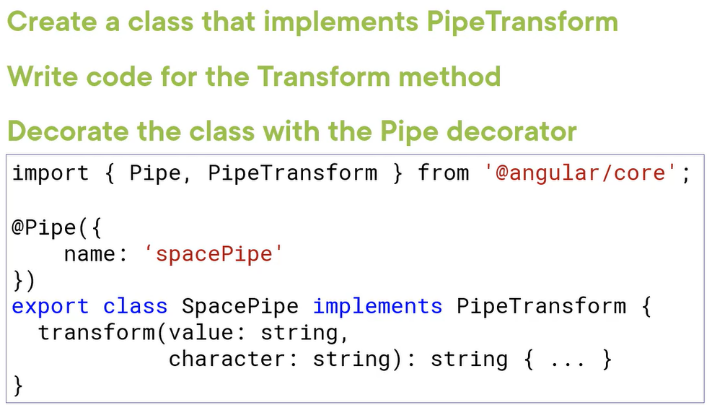


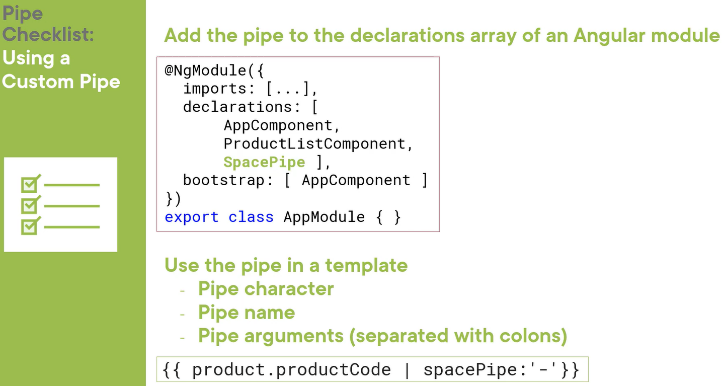
Note add form to the app module

# More on Components



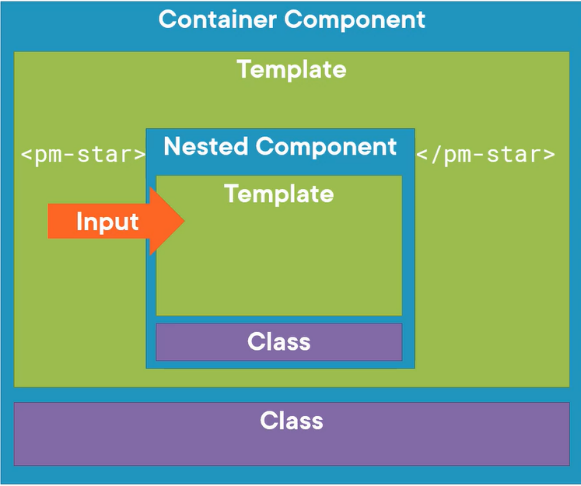
Lifecycle hook?





# Building Nested component

## Passing data to a nested component

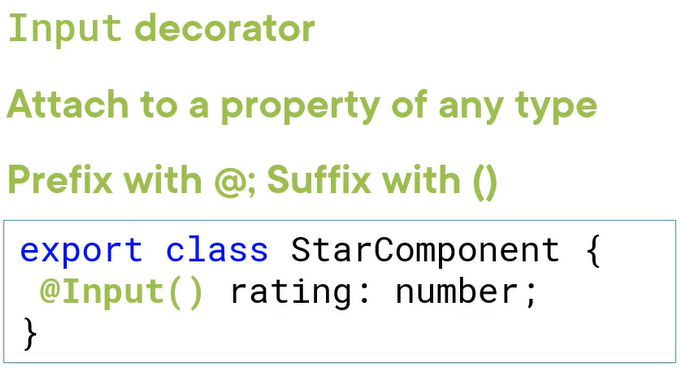


The main container passes input to the nested couponed using input properties

Note

* We can think of the properties decorated with the @input or @Output decorators as the public API of the component

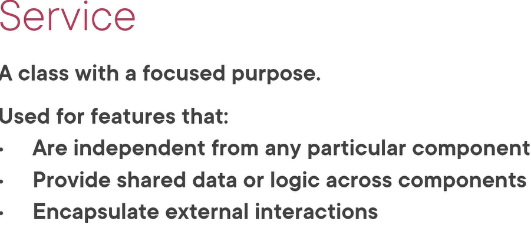
**Summery**



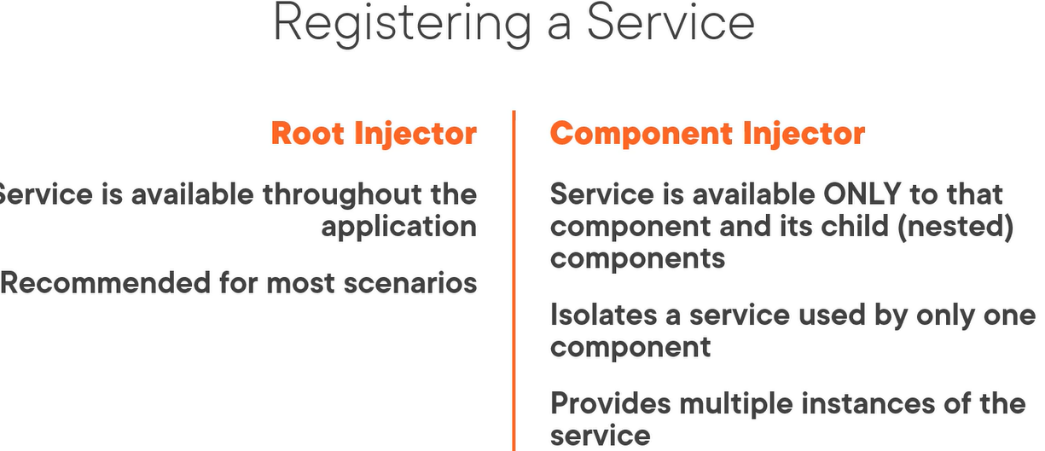
**-Property binding**

# Services and dependency Injection

**What is a service?**



## Registering the Service



**Note**

* A sserice us a class with
* We can register an instance of the service class called a singleton, angular provides an angular injector
* A component inject the services as a dependency when its needs it