

Component Lifecycle Hooks

Torino, 16 November 2017
#webappconf17

Chi siamo



Sono Alessandro Vidotto

lavoro in Coolshop, una società di ICT solution fondata a Torino nel 2005.

Per seguire meglio i clienti distribuiti su scala europea, nel 2015 abbiamo aperto una sede in Olanda.

Cosa Facciamo

Intranet Applications

Mobile Applications

Web and Ecommerce

Project Management / Governance

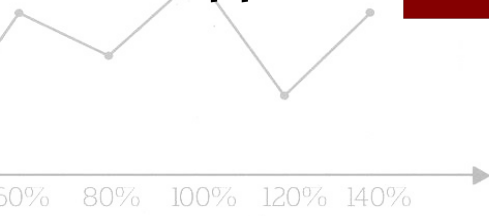
Business Intelligence

Safety Management Systems

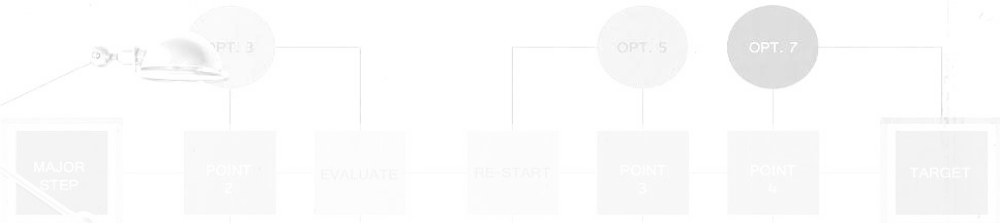
Rapid Prototyping

Quality Assurance

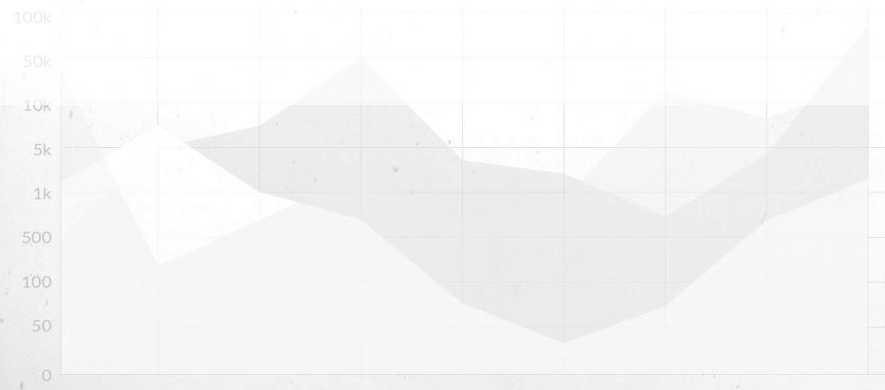




- 70%
- 82%
- 36%
- 51%
- 97%



Components



Directive under steroids

In AngularJS, a Component is a special kind of directive that uses a **simpler configuration** which is suitable for a component-based application structure.

This makes it easier to write an app in a way that's **similar to using Web Components** or using the new Angular's style of application architecture.

Advantages of Components

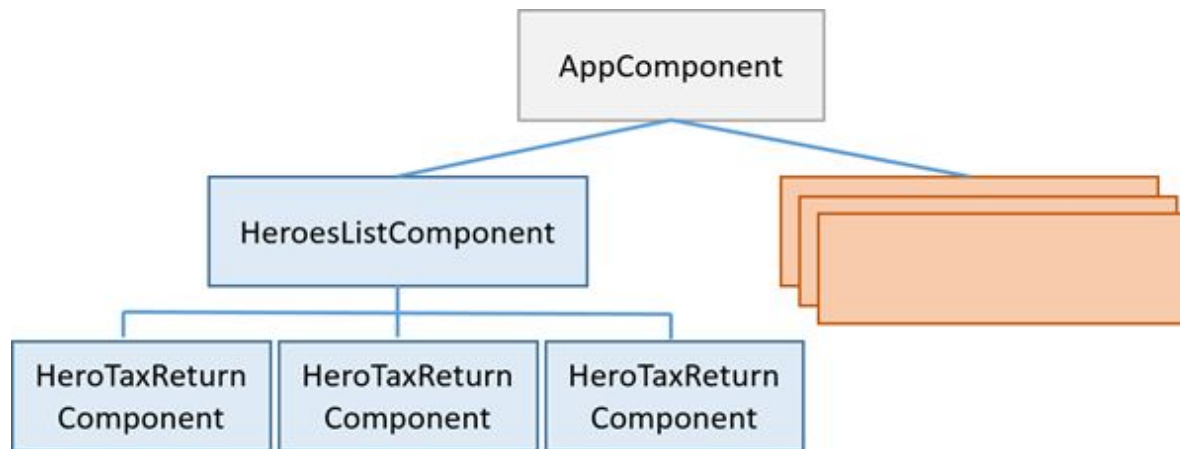
- simpler configuration than plain directives
- promote sane defaults and best practices
- optimized for component-based architecture
- writing component directives will make it easier to upgrade to Angular

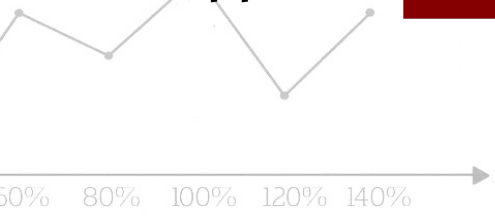
Components are not for

- for directives that need to perform actions in compile and pre-link functions, because they aren't available
- when you need advanced directive definition options like priority, terminal, multi-element
- when you want a directive that is triggered by an attribute or CSS class, rather than an element

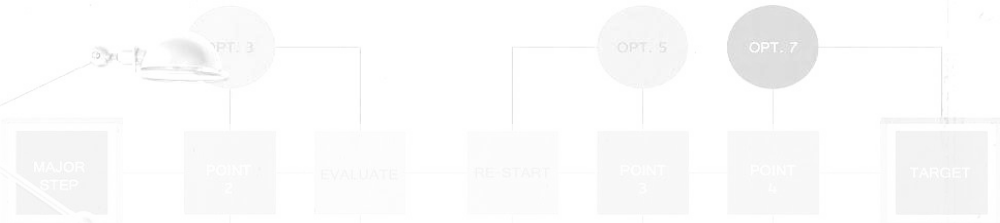
Application is tree of components

An application is a tree of components: Ideally, the whole application should be a tree of components that implement clearly defined inputs and outputs, and **minimize two-way data binding**. That way, it's easier to predict when data changes and what the state of a component is.

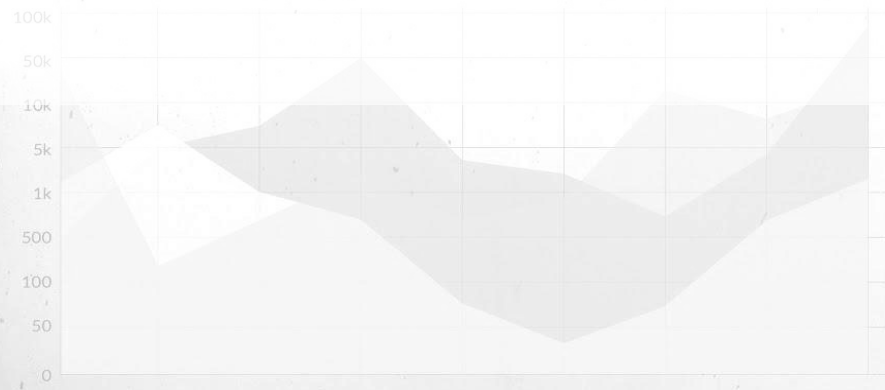




- 70%
- 82%
- 36%
- 51%
- 97%



Lifecycle hooks



Well-defined lifecycle

1.x

\$onInit()

\$onChanges(changesObj)

\$doCheck()

\$onDestroy()

\$postLink()

2/4

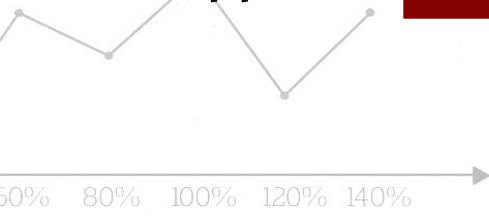


Boost performance - onInit

Called on each controller after all the controllers on an element have been constructed and had their bindings initialized (and before the pre & post linking functions for the directives on this element). **This is a good place to put initialization code for your controller.**

Boost performance - onChange

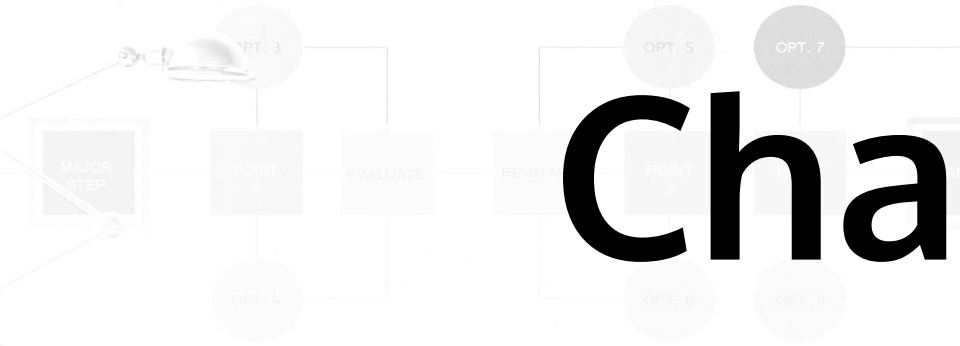
Called whenever **one-way bindings are updated**. The `changesObj` is a hash whose keys are the names of the bound properties that have changed, and the values are an object of the form `{ currentValue, previousValue, isFirstChange() }`. Use this hook to **trigger updates within a component** such as cloning the bound value to prevent accidental mutation of the outer value.



70%
82%
36%
51%
97%

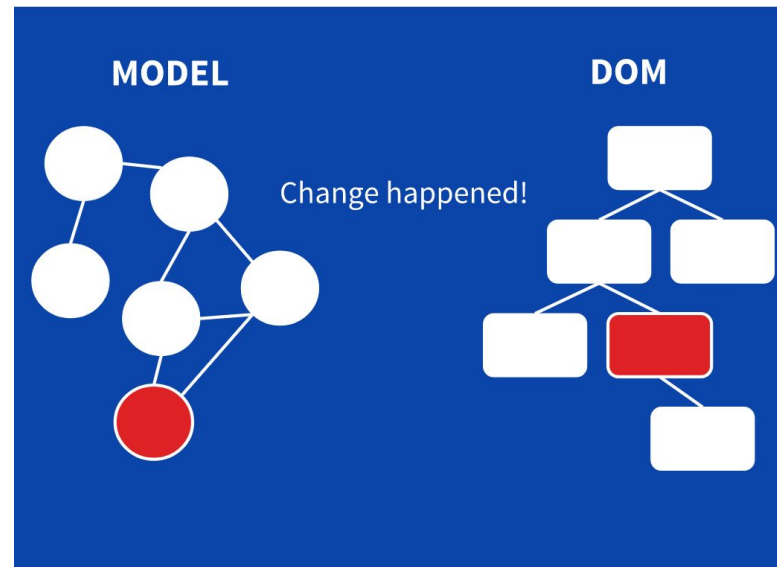


Change detection



Take control

In Angular you can take complete control on the “digest cycle” disabling the component change detector.



<https://blog.thoughttram.io/angular/2016/02/22/angular-2-change-detection-explained.html>

WebAppConf 

coolshop

Grazie

alessandro@coolshop.it

www.coolshop.it

Torino, 16 November 2017

#webappconf17