



Ground Up

Syed Awase





4. DIRECTIVES



DIRECTIVES

- Building blocks of web applications
- A collection of functionality that can be packaged and dropped into various locations within the application.
- For a user, a collection of functionality that they can interact with it can be called a directive
- Directives represent the transforming of otherwise static html elements into dynamic data driven components



What is a Directive?

- The attribute or set of attributes that we use to declare an HTML node
- The JavaScript that implements that specific functionality



Attaching directive to an element

- There are different ways to attach a directive to an element and how additional data might be passed through
 - Within HTML, use dashes, like this: contact-card
 - Within JavaScript, use CamelCase like this Contactcard.



INSERTING DIRECTIVE INTO DOM

- 4 Way to insert directive into a DOM
- EACM (by specifying restrict property)
 - A by adding an attribute to an element (Most Common)
 - For e.g. <div contact-card="person"></div>
 - E- by creating a new element (custom directive)
 - For e.g. <contact-card></contact-card>
 - No way to pass data to attributes
 - C- by giving an element a classname, (Not used often)
 - For e.g. <div class="contact-card"></div>
 - M by using an HTML comment (Not used often)
 - For e.g. <!- contact-card-->



DIRECTIVES

- \$scope provides the model
- Model does not touch HTML
- Binding Expressions {{message}} Data binding directive - data binding moves model data to view.
- Separation of Concerns the controller and model are concerned with the gathering of the data and view is concerned with the presentation of the data.
- Moving information from the view into the Model. Ex: HTML5 Form.
- Directives allow for indirect model view interaction

```
var MainController = function($scope){
    $scope.message="Hello, Angular"
};

<div>
    <h3>{{message}}</h3>
    </div>
```

```
change the message
<input type="message" name="text"/>
<input type="submit" value="Save"/>
</form>

var MainController = function($scope){
    $scope.message="Hello, Angular"
};
```



DIRECTIVES

Binding Directive

Model Directive

Event Directive

Display Directive

ng-app	ng-bind	ng-blur	ng-change
ng-class	ng-click	ng-copy	ng-dblclick
ng-disabled	ng-focus	ng-hide	ng-href
ng-if	ng-include	ng-init	ng-keydown
ng-keypress	ng-keyup	ng-model	ng-mouseenter
ng-mouseleave	ng-mousemove	ng-mouseover	ng-paste
ng-repeat	ng-style	ng-switch	ng-transclude

Around 50 directives currently available

Custom Directives

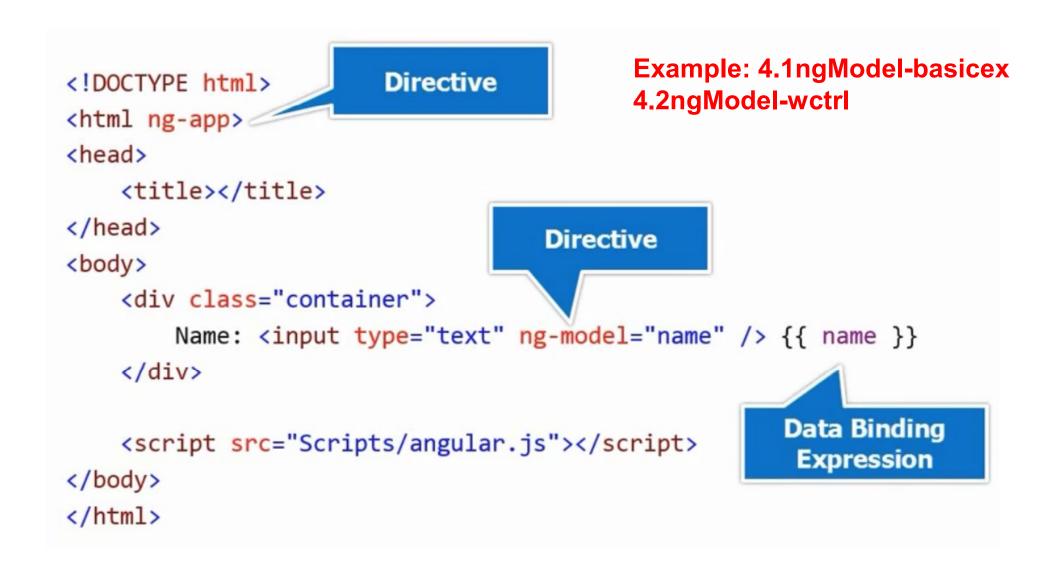


ngModel

- ngModel directive binds an
 - Input, select, textarea or custom form control to a property on the scope using NgModelController, which is created and exposed by this directive.
 - ngModel is responsible for
 - Binding the view into the model, which other directives such as input, textarea or select require
 - Providing validation behaviour(i.e. Required, number, email, url)
 - Keeping the state of control(valid/invalid, dirty/pristine, touched/untouched, validation errors).
 - Setting related css classes on the element(ng-valid, ng-invalid, ng-dirty, ng-pristine, ng-touched, ng-untouched, ng-empty, ng-not-empty) including animations.
 - Registering the control with its parent form



Using directives and Data Binding Syntax





ngInit

Example: 4.3nglnit-ex

 The ngInit directive allows you to evaluate an expression in the current scope.

```
<div ng-app="" ng-init="quantity=1;cost=5">
Total in dollar: {{ quantity * cost }}
expression: {{5*100}}
expression: {{5/100}}
```

Note: If you have assignment in ngInit along with a <u>filter</u>, make sure you have parentheses to ensure correct operator precedence:

<div ng-init="test1 = (\$index | toString)"></div>



ngRepeat

Examples: 4.4 to 4.12

 The ngRepeat directive instantiates <u>a template once per item</u> from a collection. Each template instance gets its own scope, where the given loop variable is set to the current collection item, and \$index is set to the item index or key

https://docs.angularjs.org/api/ng/directive/ngRepeat



ngRepeat

Examples: 4.13 to 4.14

 Special Properties are exposed on the local scope of each template

Variable	Туре	Details
\$index	<u>number</u>	iterator offset of the repeated element (0length-1)
\$first	<u>boolean</u>	true if the repeated element is first in the iterator.
\$middle	<u>boolean</u>	true if the repeated element is between the first and last in the iterator.
\$last	<u>boolean</u>	true if the repeated element is last in the iterator.
\$even	<u>boolean</u>	true if the iterator position \$index is even (otherwise false).
\$odd	<u>boolean</u>	true if the iterator position \$index is odd (otherwise false).



Event Directives

Examples: 4.15

- ng-click: allows users to specify custom behaviour when an element is clicked.
- ng-show: The ngShow directive shows or hides the given HTML element based on the expression provided to the ngShow attribute.
- ng-hide: The ngHide directive shows or hides the given HTML element based on the expression provided to the ngHide attribute.
- ng-disabled: This directive sets the disabled attribute on the element if the <u>expression</u> inside ngDisabled evaluates to truthy.



Filters

Examples: 4.16-4.18

Basic format: expression | filterName:parameter

Name	
currency	{{amount currency:"USD\$"}}
date	{{startDate date:'short'}}
filter	repo in repos filter:searchTerm
json	{{ repo json }}
limitTo	repo in repos limitTo:10
lowercase, uppercase	{{ user.name uppercase }}
number	{{ stars number }}
orderBy	repoin repos filter:searchTerm orderBy:'name'



ng-include

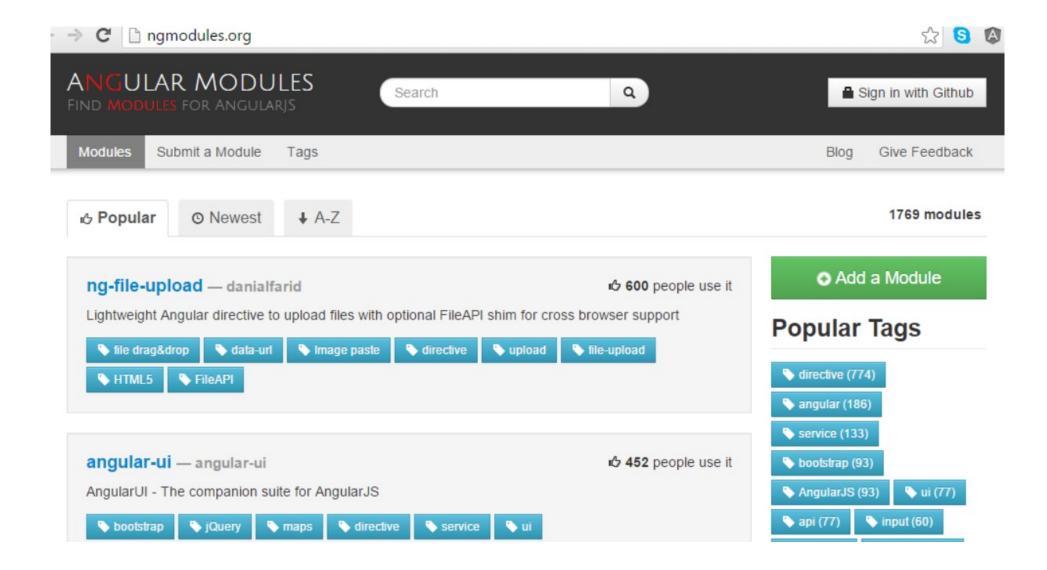
 Fetches, compiles and includes an external HTML fragment.

```
<!-- include the partial/template html using ngInclude --> <div ng-include="'userdetails.html'" ng-show="user"></div>
```

In chrome: We need to provide relative url with reference to the root directory

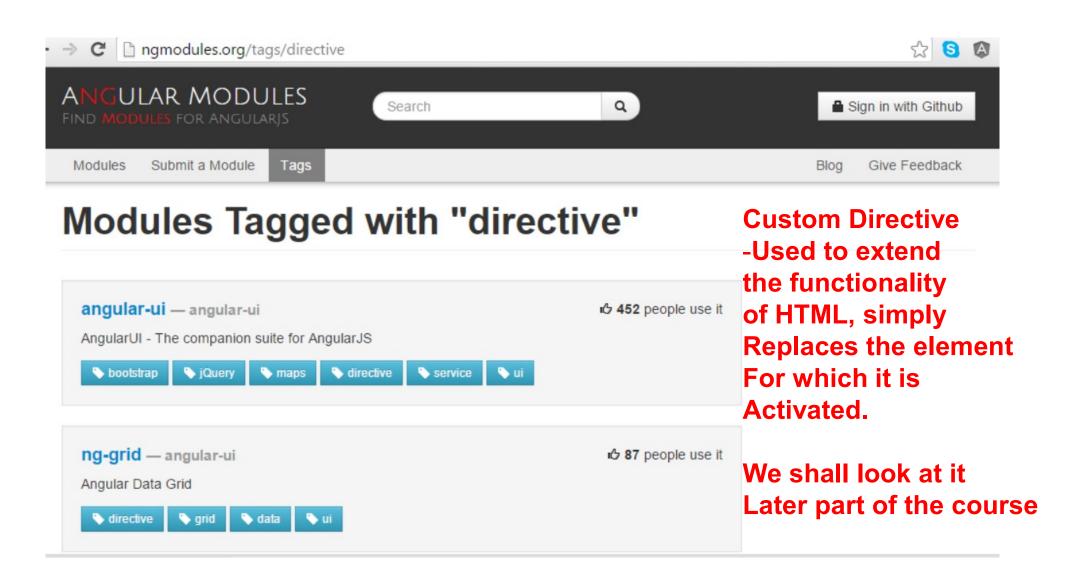


3rd Party





3rd Party Directives





Summary

- Controllers setup the model, view consumes the model and directives are intermediary
- Directives are building blocks of angular web applications
- 4 ways to insert directive into a DOM EACM
- Directives are intermediary and Angular provides one way ({{expr}})and two-way binding approaches (ngModel)
- Directives can be categorized into binding directive, model directives, event directives and display directives.
- We can create or include a 3rd party directive i.e. a custom directive