



Angular JS Directives 101

Alain Chautard
www.interstate21.com



What is a directive?

- Directives are markers on a DOM element
- They tell AngularJS to attach a specified behavior to that DOM element
- Or even transform the DOM element and its children
- Here is an example of a directive called myDir:

```
<my-dir></my-dir>  
<span my-dir="exp"></span>  
<!-- directive: my-dir exp -->  
<span class="my-dir: exp;"></span>
```

- Angular JS itself is massively made of directives: ng-model, ng-if, etc.

What are directives used for?

- Everything!
- Code reuse, creation of custom components, etc.
- Official guide: <https://docs.angularjs.org/guide/directive>
- Here is a first example:
 - <http://codepen.io/achautard/pen/XJawZg>

Template Directives

- It's possible to use a HTML file template - and of course to access variables from the current scope:

script.js

index.html

my-customer.html

```
angular.module('docsTemplateUrlDirective', [])  
.controller('Controller', ['$scope', function($scope) {  
  $scope.customer = {  
    name: 'Naomi',  
    address: '1600 Amphitheatre'  
  };  
}])  
.directive('myCustomer', function() {  
  return {  
    templateUrl: 'my-customer.html'  
  };  
});
```

Name: Naomi Address: 1600 Amphitheatre

```
<div ng-controller="Controller">  
  <div my-customer></div>  
</div>
```

Name: {{customer.name}} Address: {{customer.address}}

- templateUrl can also be a function:

```
.directive('myCustomer', function() {  
  return {  
    templateUrl: function(elem, attr){  
      return 'customer-'+attr.type+'.html';  
    }  
  };  
});
```

Isolate Scope

- It's possible to have an isolate scope for the directive so that the directive's context is separate from your current scope
- Here is how it works:
 - <http://codepen.io/achautard/pen/Joyqav?editors=101>

Directives that manipulate the DOM

- The main goal of directives is to create custom, reusable independent components
- To do so, directives may have to manipulate the DOM, listen for events, etc.
- This is achieved with the *link* option
- Here is an example:
 - <http://codepen.io/achautard/pen/QwMRzN>

Directives that wrap DOM content

- Sometimes directives are meant to wrap arbitrary content
- To do so, there is the *transclude* option, which tells Angular to use the outside scope
- Here is an example:
 - <http://codepen.io/achautard/pen/emEaoG>

Directives That Communicate

- Directives can use other directives and communicate with them through their controllers:

```
angular.module('docsTabsExample', [])
.directive('myPane', function() {
  return {
    require: ['^myTabs', '^ngModel'],
    restrict: 'E',
    transclude: true,
    scope: {
      title: '@'
    },
    link: function(scope, element, attrs, controllers) {
      var tabsCtrl = controllers[0],
          modelCtrl = controllers[1];

      tabsCtrl.addPane(scope);
    },
    templateUrl: 'my-pane.html'
  };
});
```




Thank you!

Alain Chautard
www.interstate21.com

