

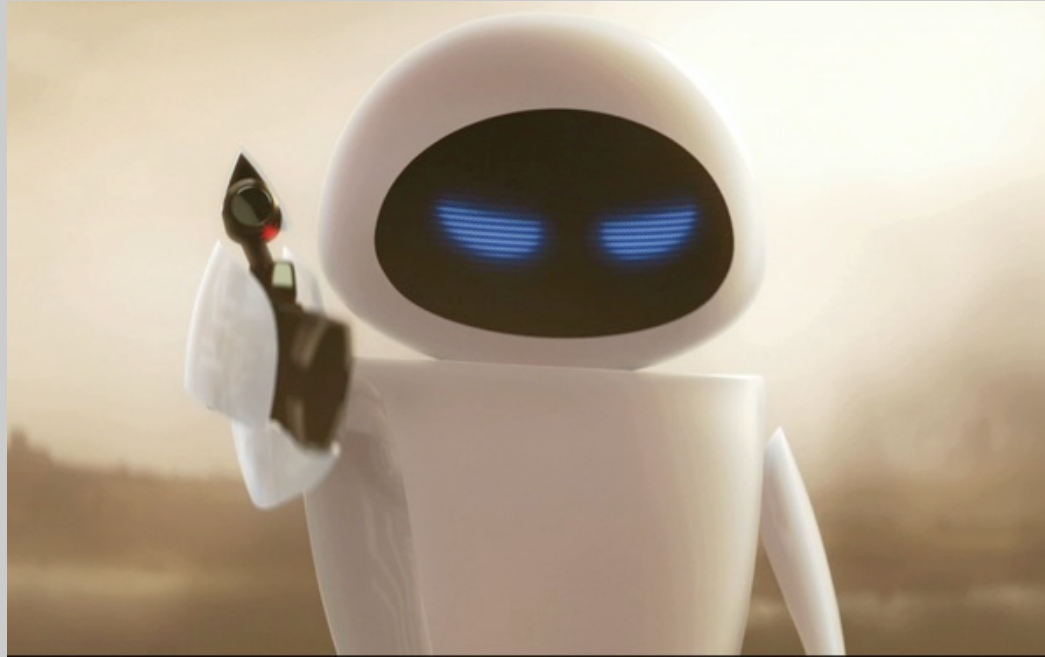


Go Deep with Directives

About me

- Lead Software Engineer at Omnyx
- jQuery -> BackboneJS -> AngularJS 😊
- Github: dcf417
- Twitter: @DCFaint

Directive?





Batteries Included

AngularJS ships with a bunch of directives that you are already using:

- ngApp
- ngController
- ngModel
- ngRepeat
- The list goes on...



How does it work?

Two important steps:

- Compile
- Link

Custom Directives

```
var myModule = angular.module(...);
myModule.directive('directiveName', function (injectables) {
  return {
    restrict: 'A',
    template: '<div></div>',
    templateUrl: 'directive.html',
    replace: false,
    priority: 0,
    transclude: false,
    scope: false,
    terminal: false,
    require: false,
    controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ... },
    compile: function compile(tElement, tAttrs, transclude) {
      return {
        pre: function preLink(scope, iElement, iAttrs, controller) { ... },
        post: function postLink(scope, iElement, iAttrs, controller) { ... }
      }
    },
    link: function postLink(scope, iElement, iAttrs) { ... }
  };
});
```

Wait slow down

Basic Layout

```
var myModule = angular.module(...);  
myModule.directive('directiveName', function (injectables) {  
  return {  
    restrict: 'AEC',  
    template: '<div></div>',  
    replace: false  
  };  
});
```

Demo #1

Show me something cool!



Is this the only place for my template?

TemplateUrl

```
templateUrl: 'templates\directive.html'
```

- Embedded in a script tag

```
<script type="text/ng-template" id="templates\directive.html">  
  <h1>This is my template content.</h1>  
</script>
```

- Separate location entirely (i.e. templates folder on the server)

Demo #2

- Yay, more awesomeness!



Scope

Three options:

- Use parent
- Create child
- Isolated

scope: false

scope: true

scope: { }

Isolated?

Binding

- One-way expression (top down)
- Two-way binding (reference)
- One-way behavior (bottom up)

```
scope: {  
  type: '@type',  
  props: '=props',  
  addShape: '&addShape'  
}
```

```
<shape type="{{type}}" props="props" addShape="addShape()">
```

Demo #3

- It's show time!



Link function

- Binds the template of a directive to the correct scope

```
myModule.directive('directiveName', function (injectables) {  
  return {  
    ...  
    link: function (scope, element, attrs) { ... }  
  };  
});
```

Demo #4

- Sweet, more!



Transclude

Wrap arbitrary content within your directive

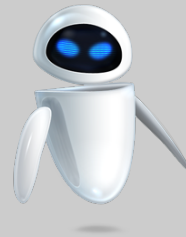
- Transclude contents of directive
- Transclude whole directive

```
transclude: true
```

```
transclude: 'element'
```


Demo #5

- More, really?



Controller and Require

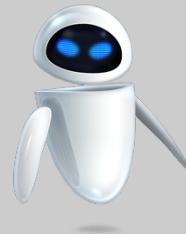
- Controller: data and functions are accessible to other directives
- Require: allows you to pass a specified controller into your link function

```
myModule.directive('outerDirective', function
(injectables) {
  return {
    ...
    controller: function(scope, element, attrs) {
      this.publicFunc = function () { ... };
    }
  };
});
```

```
myModule.directive('innerDirective', function
(injectables) {
  return {
    ...
    require: '^outerDirective'
    link: function(scope, element, attrs, controller) {
      controller.publicFunc();
    }
  };
});
```

Demo #6

- Do we have time for this?



Thank you!

