

AngularJS 101

Philadelphia Emerging Technologies for the Enterprise

Ken Rimple

Chariot Solutions

What is AngularJS?

"A JavaScript Model-View-Controller framework that uses Dependency Injection and HTML-based Markup to develop applications"

"A JavaScript MVVM Framework"

"Code in my HTML???"

Why Angular

- D/I makes things very testable
- Great MVC separation of concerns
- Angular has a LARGE, committed community
- Large number of modules ready-to-use

Let's Hack!

- Go to:
- Basic MVC
- Expressions
- Directives (ng-show/hide, ng-if, ng-bind, ng-repeat)
- Routing

Key AngularJS Components

JavaScript Module pattern

- Basic single-module syntax

```
angular.module('myApp', []);
```

- Importing Dependencies

```
angular.module('myApp', ['ngRoute', 'ui.bootstrap'])
```

- Check ngModules.org for a fledgling list of projects

JavaScript object literal notation

- Used everywhere in Angular
- Makes it easy to configure objects

```
var myCustomer = {  
  firstName: 'Happy',  
  lastName: 'Gilmore',  
  roles : [  
    'hockeyPlayer', 'golfer'  
  ]  
}
```

Functional Javascript

- In angular, most methods take a function
- The function returns a JS Literal (or in some cases another function)

Typical "Thingie" setup

```
angular.something('name', function(dep1, dep2) {  
  return {  
    // implementation JS object  
    property: 'myprop',  
    method: function(var) { ... },  
    ...  
  };  
});
```

Controllers

- Goal: define data in a scope
- Another way: They set up the view, then they DIE!
- Use initial caps to define the name (think prototype)

```
angular.controller('CustomerCtrl',
  function($scope) {
    // model data
    $scope.customer = { firstName: 'Ken', ... };

    // method
    $scope.saveCustomer = function(customer) {
      ...
    };

    // event
    $scope.$on('PurchaseEvent',
      function(event, data) {
        ...
      });
  });
```

Defining controllers and view

- There is one default view - all objects within the tag assigned as 'ng-app'
- You can mount controllers in the main view or in any sub-view using 'ng-controller'

The Router

- Views are compiled into templates and rendered by Angular
- You must configure routing

```
angular.module('myApp', ['ngRoute'])
  .config(function($routeProvider) {
    $routeProvider
      .when('/one', {
        template: '<b>One!</b>'
      })
      .when('/two', {
        template: '<b>Two!</b>'
      });
  });
```

On Views

- Views see every object placed into \$scope (or any parent scope of your scope)
- Scopes can be nested since controllers can be nested
- Data in parent scopes can be seen by all children

Views and Scopes

```
<div ng-controller='Controller1'>
  <div ng-controller='Controller2'>
    <div ng-repeat='thing in things'>
      {{ thing.title }}
    </div>
  </div>
</div>
```

- The outer div only sees data in the first controller's scope
- The second div will see all data in controller 2, then in controller 1
- The ng-repeat creates inner scopes for each iteration, and those scopes see controller2, then controller1's scope data
- All views can see the global scope, \$rootScope

Services

- Services are abstractions for your data/business logic
- They are *singletons*
- They can be defined using a factory function (one of four ways)

```
angular.module('myApp')
  .service('myService', function($log) {
    return {
      debugMe: function() {
        $log.debug('I am a service!!!');
      };
    };
  });
```

Services can be injected

- Like Angular's 'dollar sign' services, your services can be injected into your controllers

```
angular.module('MyCtrl', function(myService) {  
    myService.debugMe('ouch, Elliot!');  
});
```

- In this way we can pull out business / data logic from our controllers
- Separation of concerns = easy testing and happy developers

Directives

A directive - jquery

```
directive('changeColor', function() {
  return {
    link: function(scope, element, attrs) {
      element.mouseover(function(event) {
        element.css({
          'background-color': scope.in
        });
      });
      element.mouseout(function(event) {
        element.css({
          'background-color': scope.out
        });
      });
    }
  };
});
```

jQuery/jQuery Lite

- Angular uses jQuery lite by default - a few dozen jQuery methods assigned to '\$' and to your directives
- Dependency Injection for all collaborators