AngularJS - Controllers

@kevinderudder

Agenda

- About Controllers
  - Intro
  - Creating controllers
  - Multiple controllers

- $scope
- $http
- Minification

Controllers

- One of the most important parts of an Angular application
  - They “control” what is going to happen in a specific area
- The primary role of a controller is to create a scope object
- Controller areas are created with the `ng-controller` directive

Controllers

```
<body ng-app>
    <section ng-controller="CourseController"></section>
```

Controllers

- Conventions

Used most of the times

```
<body ng-controller="CourseCtrl">
```

```
<body ng-controller="CourseController">
```

What we will use in this course

- Doesn’t matter which one you follow, as long as you are consistent

Controllers

- The ng-controller attribute corresponds with a function that Angular will invoke

```
<body ng-app>
   <section ng-controller="CourseController"></section>
```

```
<script>
    var CourseController = function ($scope) {
        $scope.title = "AngularJS Training";
        $scope.trainer = "Kevin DeRudder";
    }
</script>
```

Controllers

- Angular will automatically invoke the controller function

- Each controller function accepts a number of parameters
  - $scope
  - $http
  - …

- Controllers never change the View (html) directly

The scope object

- This object is your way to communicate with the view

The scope object

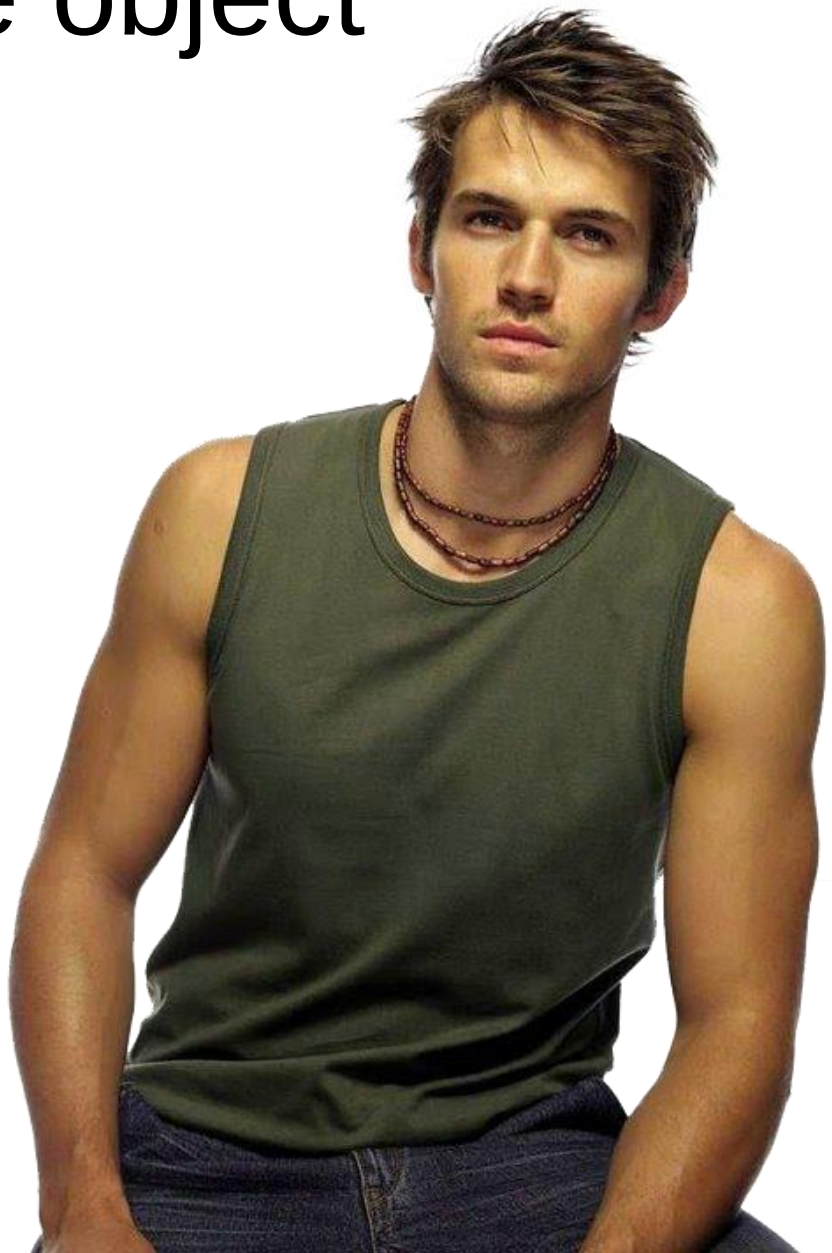
Scope

$\neq$

The scope object

Scope

$\neq$



The Scope Object

- The model is the data that we put in the scope

Demo

CREATING CONTROLLERS

Multiple Controllers

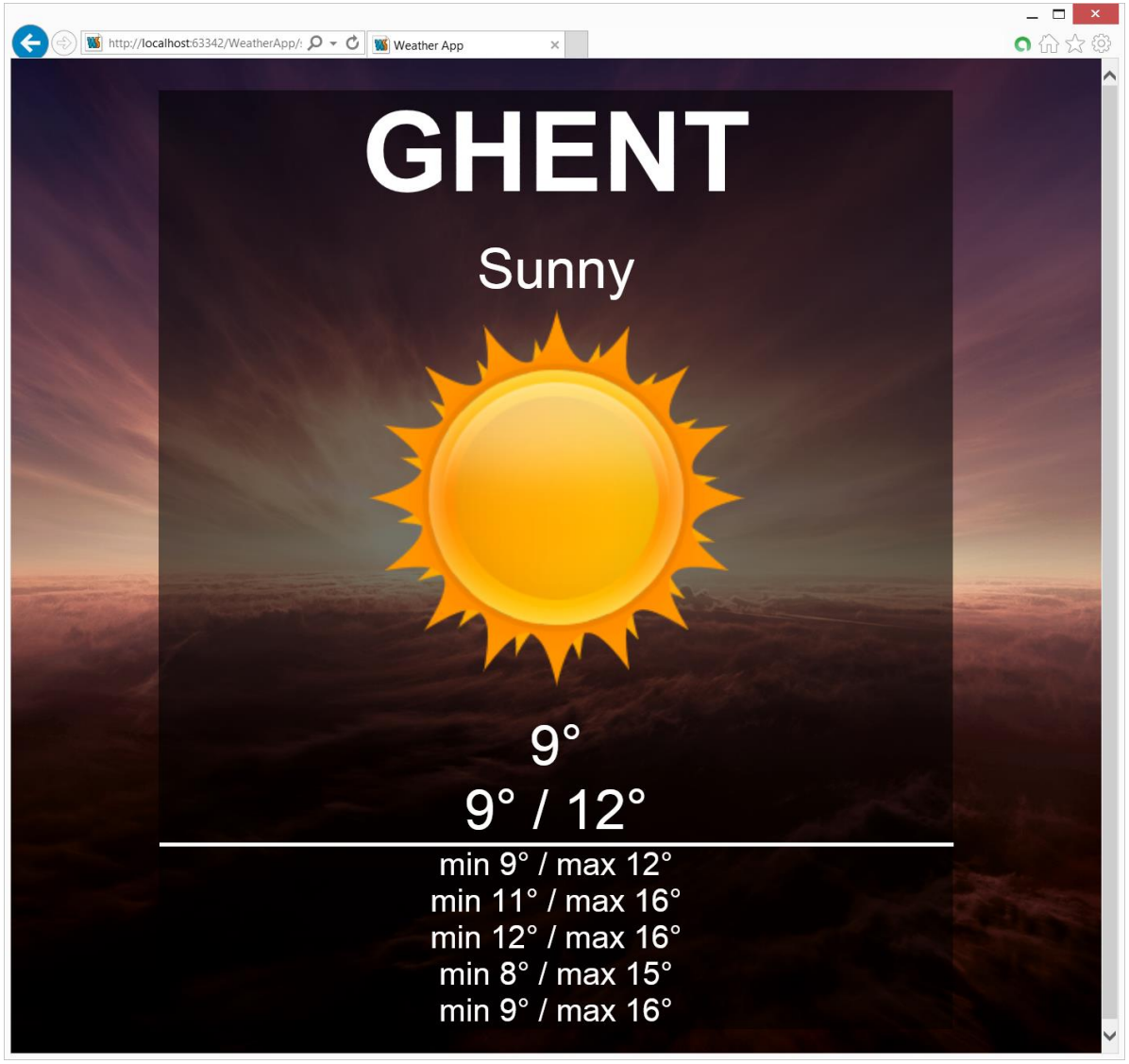
- In large applications you will probably create more than 1 controller

```
<section ng-controller="CourseController">
    <h1>{{title}}</h1>
    <p>{{description}}</p>
</section>
<section ng-controller="AgendaController">
    <table>
        <tr><td></td></tr>
    </table>
</section>
```

Nesting Controllers

```
<section ng-controller="CourseController">
    <h1>{{title}}</h1>
    <p>{{description}}</p>
    <div ng-controller="TrainerController">
        <span>{{trainer}}</span>
    </div>
</section>
```

Exercise


http://localhost:63342/WeatherApp/
Weather App
GHENT
Sunny
9°
9° / 12°
min 9° / max 12°
min 11° / max 16°
min 12° / max 16°
min 8° / max 15°
min 9° / max 16°

AngularJS - Controllers

$HTTP

Before $http

- When you are working with data, you probably will get the data from a server
  - JSON file
  - WEB / REST Service

- In JavaScript, this can be done via the xmlhttprequest object

```
function getData(url) {
    var xmlHttpObject;

    if (window.XMLHttpRequest) {// IE7+,Firefox,Chrome,Opera,Safari
        xmlHttpObject = new XMLHttpRequest();
    }
    else {// code for IE6, IE5
        xmlHttpObject = new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlHttpObject.open('GET',
                       url,
                       /*async*/ true);

    xmlHttpObject.onreadystatechange = function () {
        if (xmlHttpObject.readyState === 4) {
            // do something with the data
        }
    };

    xmlHttpObject.send(null);
}
```

$http

- Provides a way to communicate with a back-end
- $http is an object with methods corresponding to the http verbs
  - GET
  - POST
  - PUT
  - DELETE

$http

And the good thing is that you get all the functionality for free

```
var CourseController = function ($scope, $http) {

}
```

http://www.ptforschools.com/img/PeopleCheering.png

$http

```
var CourseController = function ($scope, $http) {
    $scope.courseDetails = $http.get("http://.../courses/angular");
}
```

This would work if the communication was synchronous

The $http service will return a promise

$http

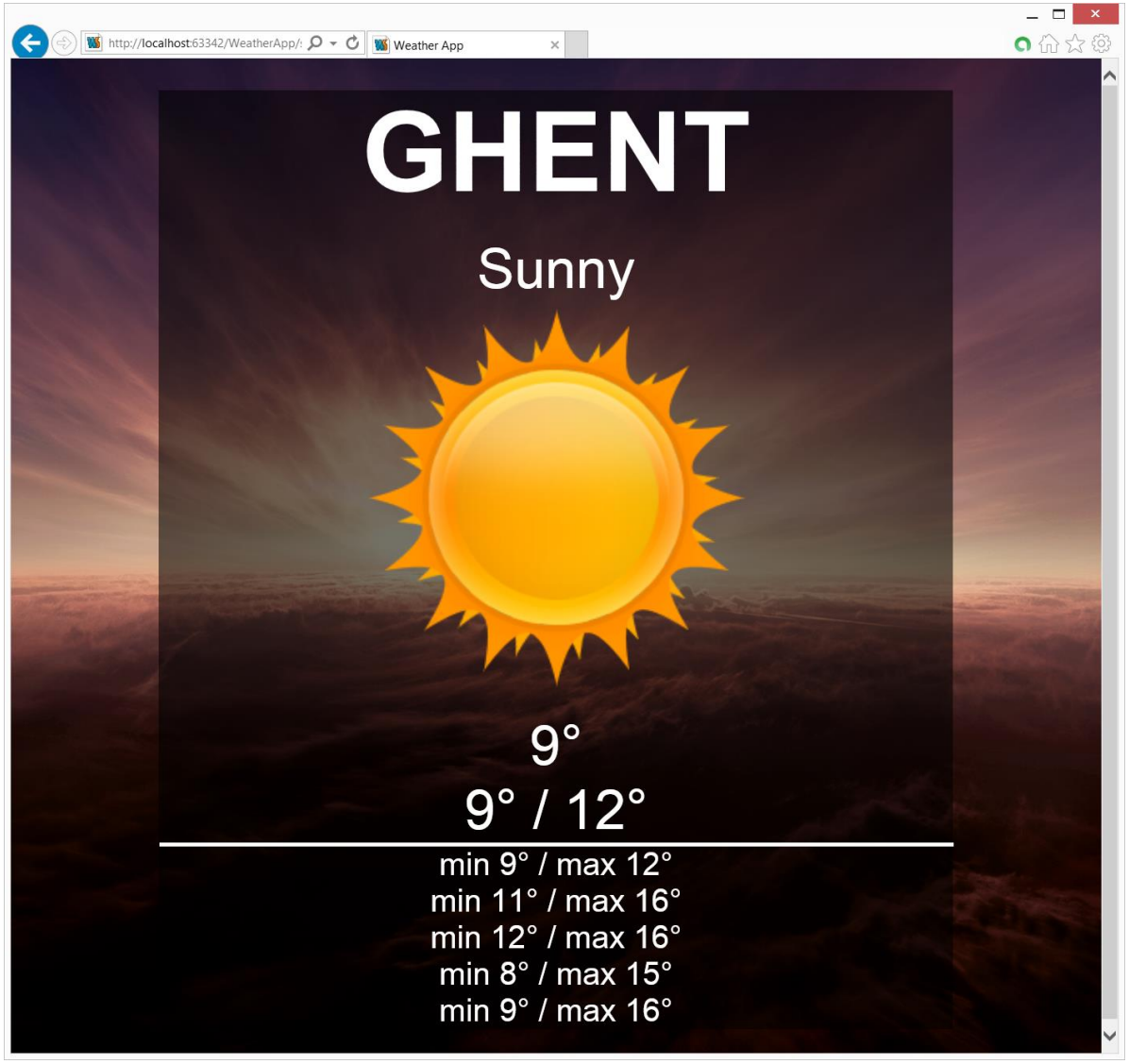
- So, angular returns a promise object when you are trying to get data

```
var CourseController = function ($scope, $http) {
    $http.get("http://www.....com/courses/angular")
       .then(function (response) {
          $scope.courseDetails = response.data;
       });
}
```

Demo

$HTTP

Exercise


http://localhost:63342/WeatherApp/
Weather App
GHENT
Sunny
9°
9° / 12°
min 9° / max 12°
min 11° / max 16°
min 12° / max 16°
min 8° / max 15°
min 9° / max 16°

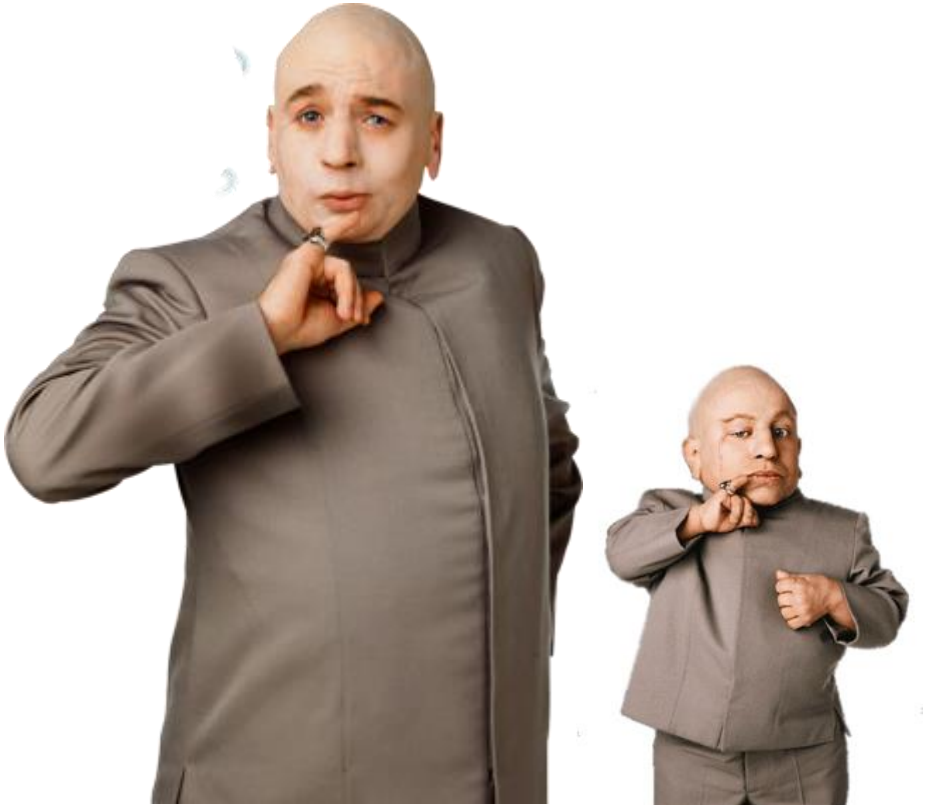
AngularJS - Controllers

MODULES

Minification

- Rewriting your JavaScript so it is as small as it can possibly be
  - Performance

- Remove whitespace
- Shorten variables
  & parameters



Problem: angular uses reflection to enable functionality

Before minification

```
angular.module('app').controller('controllerdemo',
    function ($scope, mySvc) {
        $scope.val = mySvc.val;
    }
);
```

After minification

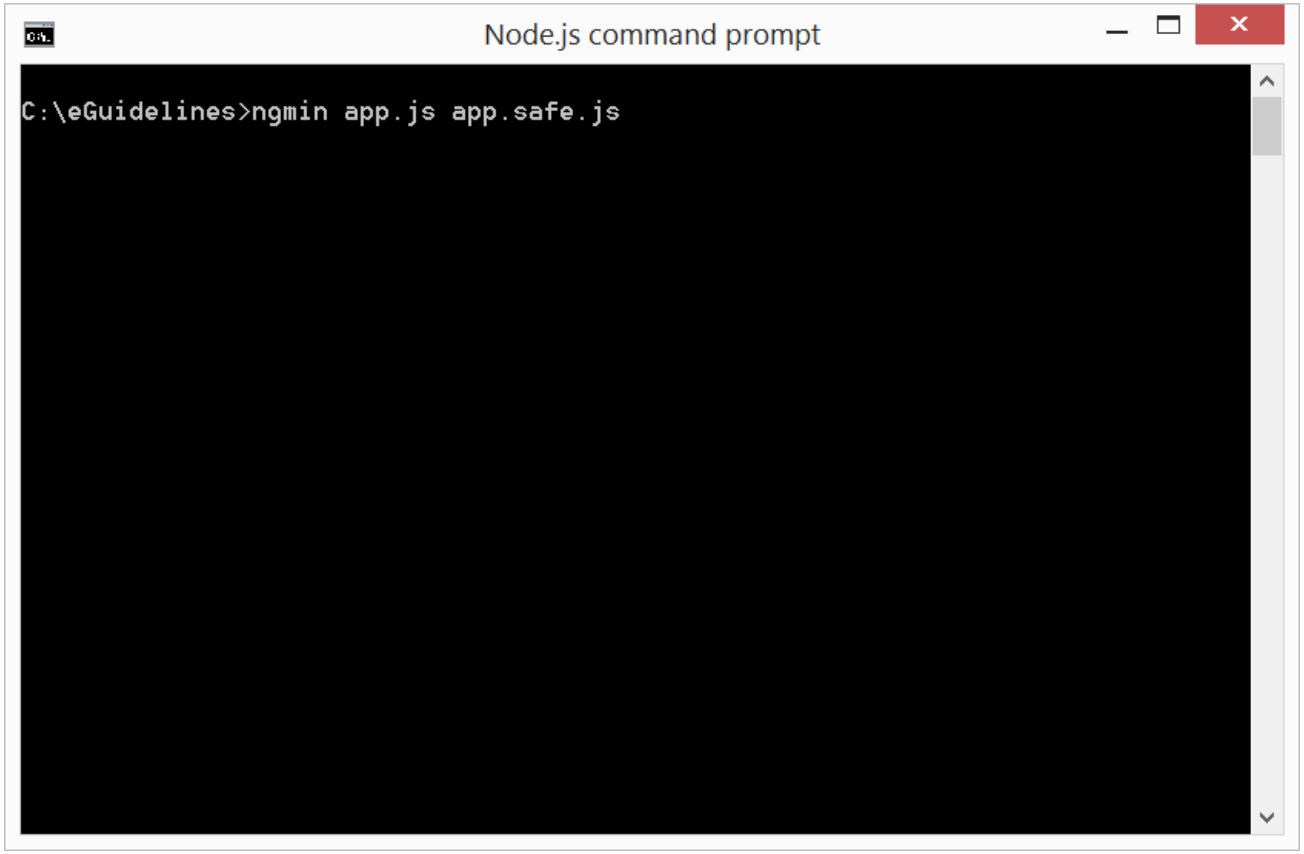
```
angular.module('app').controller('controllerdemo',
    function (a, b) {
        a.val = b.val;
    }
);
```

```
angular.module('app').controller(
   'controllerdemo',
   ['$scope', 'mySvc',
      function (a, b) {
         a.val = b.val;
      }
   ]);
```

Pass your dependencies via the array

ng-min

- Minsafe your code before minification
- Node utility



Controllers

DEMO MINIFICATION