

MVC is
cool



Controllers and Markup

Angular Basics

Telerik Software Academy
Learning & Development Team
<http://academy.telerik.com>



Table of Contents

- ◆ Controllers and \$scope object
- ◆ Markup & Binding
- ◆ Filters
- ◆ Validation



Controllers and \$scope



Controllers and \$scope

- ◆ **Controllers in Angular**

- Have all the logic about a page
- Control the \$scope object

- ◆ **\$scope object**

- Two-way bound to the view
- Tracks any changes in view and updates itself
- Tracks any changes in JS and updates the view

Controllers and \$scope

◆ Controllers

```
myApp.controller('myController',  
  function myController($scope, /* services */) {  
    $scope.hello = "Hi, guys!";  
  }  
);
```

◆ View (HTML)

```
<div ng-app="myApp">  
  <div ng-controller="myController">  
    <h1>{{hello}}</h1>  
  </div>  
</div>
```

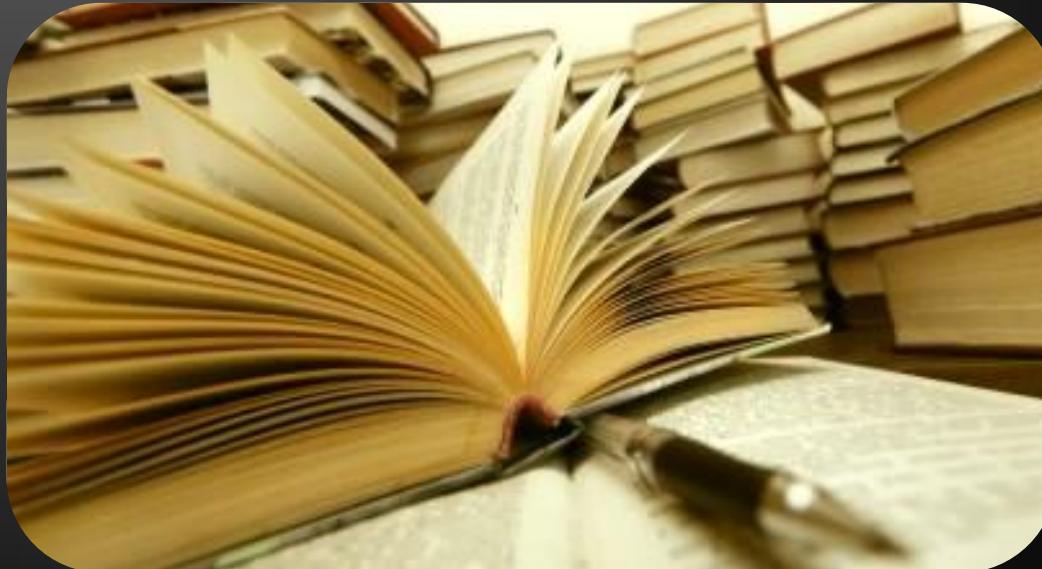
◆ Values

```
myApp.value('someValue', {  
    name: 'Ivaylo',  
    number: 5,  
    date: new Date()  
});
```

◆ Constants

```
myApp.constant('PI', 3.14);
```

Directives



◆ Directives

- Adds additional markup to the view
- Extends HTML functionality
- Three ways to declare
- Start with ng (from Angular)

```
<ng-form />  
<div ng-form />  
<div class="ng-form" />
```

◆ Other Directives

`ngApp` // which module "compiles" the element

`ngRepeat` // repeats an element

`ngBind` // same as curly brackets

`ngBindTemplate` // template to bind

`ngBindHtml` // binds HTML directly (sanitizes it)

`ngHide` // hides an element when true

`ngShow` // shows an element when true

`ngCloak` // does not show until compiled

`ngStyle` // styles as JS object

`ngClass` // puts styling by class

`ngClassEven` // class on all even elements

`ngClassOdd` // class on all uneven elements

◆ Other Directives

```
ngDisabled // disables an element  
ngChecked // checks an element  
ngMultiple // used with select element  
ngReadonly // can't be changed  
ngSelected // selected option from select element  
ngForm // allows nesting of forms  
ngSubmit // disables the default submit  
ngHref // url bound to the $scope  
ngSrc // image source bound to the $scope  
ngNonBindable // tells angular to ignore  
ngPluralize // easy edit of text  
ngSwitch // easy switch HTML templates
```

◆ Event Directives

```
ngClick // mouse click
```

```
ngDblClick // double mouse click
```

```
ngMousedown // mouse down
```

```
ngMouseenter // mouse over the element
```

```
ngMouseleave // mouse leaves the element
```

```
ngMouseup // mouse up
```

```
ngChange // changing a value
```

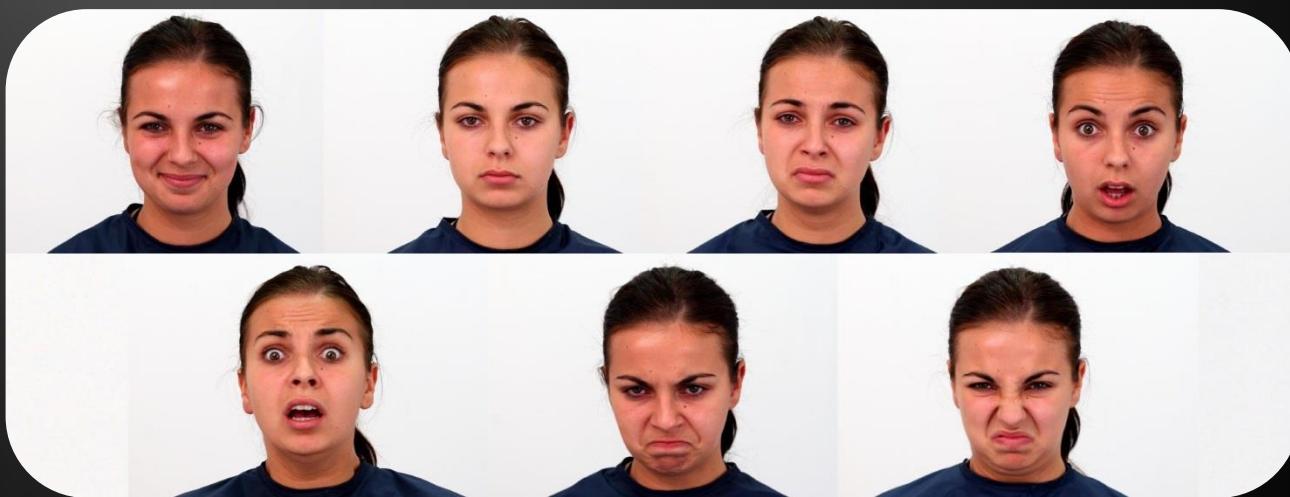
```
<input type="checkbox" ng-change="handleChange()"  
ng-model="property" />
```

- ◆ If you support old IE versions
 - ◆ Polyfill JSON.stringify (google it)
 - ◆ No NG tags (only attributes)

```
<ng-pluralize count="..." when="..."></ng-pluralize>
```

```
<div ng-pluralize count="..." when="..."></div>
```

Expressions



- ◆ You can write expressions
 - Use curly brackets
 - You can evaluate some JS expressions
 - You can create arrays

```
{{ expression }}  
{{ name }}  
{{ amount * 100 + 3 }}  
{{ number in [1, 2, 3, 4, 5, 6, 7, 8, 9] }}
```

Filters



- ◆ Filters:

- ◆ Can modify the output
- ◆ Can format output
- ◆ Can sort data
- ◆ Can filter data

◆ Using filters

```
{{ expression | filter }}
```

◆ Built-in filters

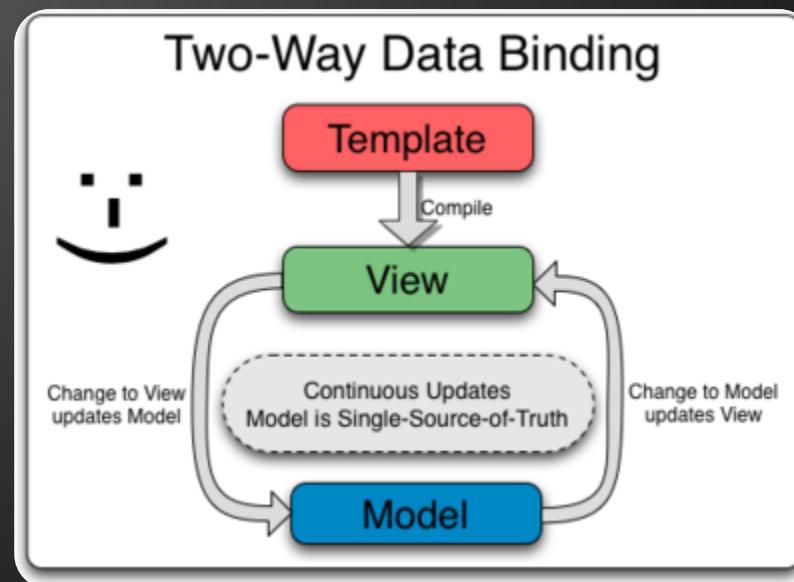
- ◆ uppercase, lowercase
- ◆ number
- ◆ currency
- ◆ json
- ◆ orderBy, limitTo, filter



◆ Writing custom filters

```
module.filter('name', function() {  
    return function(input /*,filter parameters*/) {  
        // modify input  
        return modifiedOutput;  
    }  
});
```

Two Way Binding



Two Way Binding

◆ Using ngModel

- On input
- On select
- On textarea



```
<input type="text" ng-model="object.property" />
```

```
<input type="text" ng-model="property" />
```

```
<input type="text" ng-model="object.container.property" />
```

Validation



- ◆ Default validation

- ◆ Required – makes property required
- ◆ ngPattern – regex pattern
- ◆ Form Properties – \$valid / \$invalid
- ◆ CSS Classes – classes can be styled

- ◆ Controllers – main logic
- ◆ \$scope – bound to the view
- ◆ Markup – HTML++
- ◆ Binding – two way
- ◆ Filters – sort, filter, format
- ◆ Validation – validate user input

Questions?

1. Make addition of albums and band members available in the new artist form
2. Validate the whole form in a better way
3. Implement the ng-Cloak directive
4. Display an icon for each album using a filter
5. *Make editing of existing artist possible through the same page of creating artist (use routes and same controller)