AngularJS - Directives

@kevinderudder

Agenda

- Directives
- Filters
- Validation
- Custom Directives

from the documentation:

A directive is a way to add **new functionality to HTML**

Via an element
```
<ng-form />
```

Via an attribute
```
<img ng-src />
```

Via a class
```
<img class="ng-hide" />
```

Directives

- In a MVC pattern, a model holds information about an object and should not interact with a view
  - Separation of concerns

- Until now, we used binding expressions to visualize a property of a model
  - Also called a “databinding directive”

```
<h1>{{course.title}}</h1>
```

Directives

- Angular comes with a bunch of directives
  - Built-in directives
  - Event directives
  - …

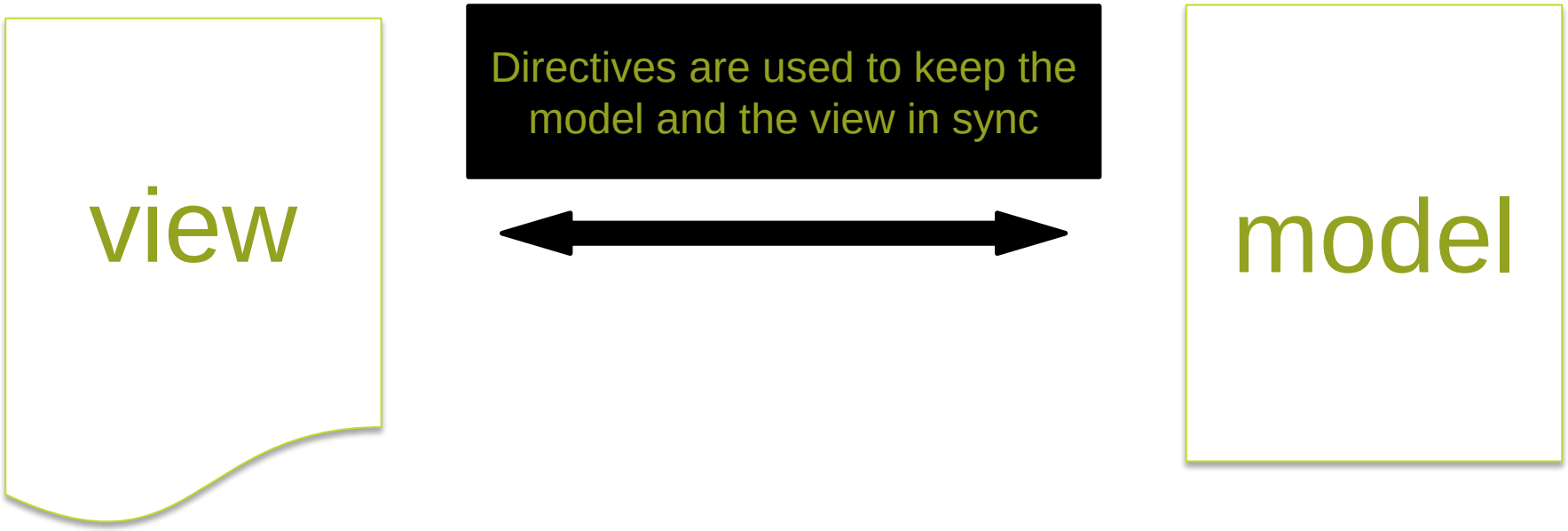
```
<h1>{{course.title}}</h1>
```

Web Components

- Currently draft specification
- A way to create new functionality in browsers
- Angular is already a way to have this kind of functionality

What if we need to push data from the view into the model

```
<form>
    <label for="description">Description</label>
    <textarea id="description"></textarea>
</form>
```


view
Directives are used to keep the model and the view in sync
model

ng-model

- Every Angular application needs a model
  - To pass data from the controller to the view
- Angular creates a default model
  - Use $scope to talk to this model

ng-bind

- Until now, we've used expression to show data

```
<h1>{{course.name}}</h1>
```

- You can have the same result with ng-bind

```
<div ng-bind="title"></div>
```

- Or combine info via ng-bind-tempate

```
<div ng-bind-template="{{title}}: {{trainer}}"></div>
```

ng-bind

- Sometimes when you use expressions and your page doesn’t load that fast you can see the expressions {{}}

- ng-bind only stores the value in memory

```
<div>
    Hello, {{yourName}}
</div>

<div>
    Hello, <span ng-bind="yourName"></span>
</div>
```

Hello, {{yourName}} is saved in memory

only yourName is saved in memory

ng-repeat

- Like the for-each in other programming languages

- Repeats a portion of HTML depending to the corresponding collection

ng-cloak

- Hides portions or all of your page so that angular has the time to do its work
  - To avoid flashes
  - Or unfinished expressions

```
<head>
    <title></title>
    <style>
        [ng\:cloak], [ng-cloak], [data-ng-cloak],
        [x-ng-cloak], .ng-cloak, .x-ng-cloak {
            display: none !important;
         }
    </style>
</head>
<body ng-cloak>
```

ng-style

- Pretty straightforward

```
<button type="submit" ng-style="buttonStyle">Search</button>
```

```
$scope.buttonStyle = { 'font-family': 'stencil' };
```

ng-class

- Like the style, you can set a class
- ng-class-even and ng-class-odd are pretty useful for tables and rows
  - Can only be used in ng-repeat

ng-show / ng-hide

- Shows or hides HTML based on a condition
  - Thruthy vs Falsy

ng-include

- Renders HTML from another source/file
- Break up a complex page into pieces
- Reuse HTML across your application

ng-form

- HTML spec doesn’t allow a form to be nested
  - Is possible with the ng-form directive

ng-nonbindable

- What if you want to show some text with {{}}

```
<div ng-non-bindable>
   {{ 6+7 }}
</div>
```

Some special directives for older browsers

- ng-disabled
- ng-checked
- ng-multiple
- ng-readonly
- ng-selected

- Add these attributes to an element when you set the to true or false

Directives and Views

FILTERS

Filters

- Evaluate data while rendering the view

- Most basic form of a filter:

```
<tr ng-repeat="course in courses">
    <td>{{course.title}}</td>
    <td>{{course.description}}</td>
    <td>{{course.price | number:2}}</td>
</tr>
```

expression

filter:parameter

Built-in filters

name	example
currency	{{course.price | currency:"EUR"}}
date	{{course.startDate | date:'short'}}
filter	course in courses | filter:searchTerm
json	{{course | json}}
limitTo	course in courses | limitTo:15
lowercase, uppercase	{{course.title | lowercase}}
number	{{course.price | number:2}}
orderBy	course in courses | orderby:'title'

Directives and Views

VALIDATION

Standard Validation

- HTML5 comes with a set of validation techniques
  - Required
  - Email
  - Pattern
  - …
- Angular uses these features to make a form valid

Directives and Views

CUSTOM DIRECTIVES

Custom Directives

- Next to the built-in directives, you can create your own custom directives
  - Makes your HTML cleaner

- To create a custom directive, you need to call the directive method on your module

```
app.directive("customDirective", function () {
});
```

Invoked when the compiler matches the directive for the first time

Template property

- Say that you have a piece of HTML that is used often in your application
  - Make a template out of it

```
angular.module('demo', [])
  .controller('demoController', ['$scope', function ($scope) {
      $scope.course = {
          title: 'AngularJS',
          trainer: 'Kevin DeRudder'
      };
  }])
  .directive('myCourse', function () {
      return {
          template: 'title: {{course.title}} by: {{course.trainer}}'
      };
  });
```

TemplateUrl property

- Same as template, but you are loading the html from an html file

Restrict

- At this point in time, all are custom directives can be used as:
  - Elements
  - Attributes
  - Classes

- Via the restrict, you can specify if you only want to use a directive in a specific way
  - A == only attributes
  - E == only elements
  - C == only class names
  - Combination: AEC

```
.directive('myCourse', function () {
    return {
        restrict: 'E',
        templateUrl: 'courseinfo.html'
    };
});
```

elements vs attributes

Use an **attribute to decorate** an existing element
and use an **element to create a component**

```
angular.module('demo', [])
  .controller('demoController', ['$scope', function ($scope) {
      $scope.course = {
          title: 'AngularJS',
          trainer: 'Kevin DeRudder'
      };
  }])
  .directive('myCourse', function () {
      return {
          template: 'title: {{course.title}} by: {{course.trainer}}'
      };
  });
```

This means that this directive can only be used with the same scope?

Scope property

```
.directive('myCourse', function () {
    return {
        restrict: 'E',
        scope: {
              courseInfo : '=info'
        },
        templateUrl: 'courseinfo.html'
    };
});
```

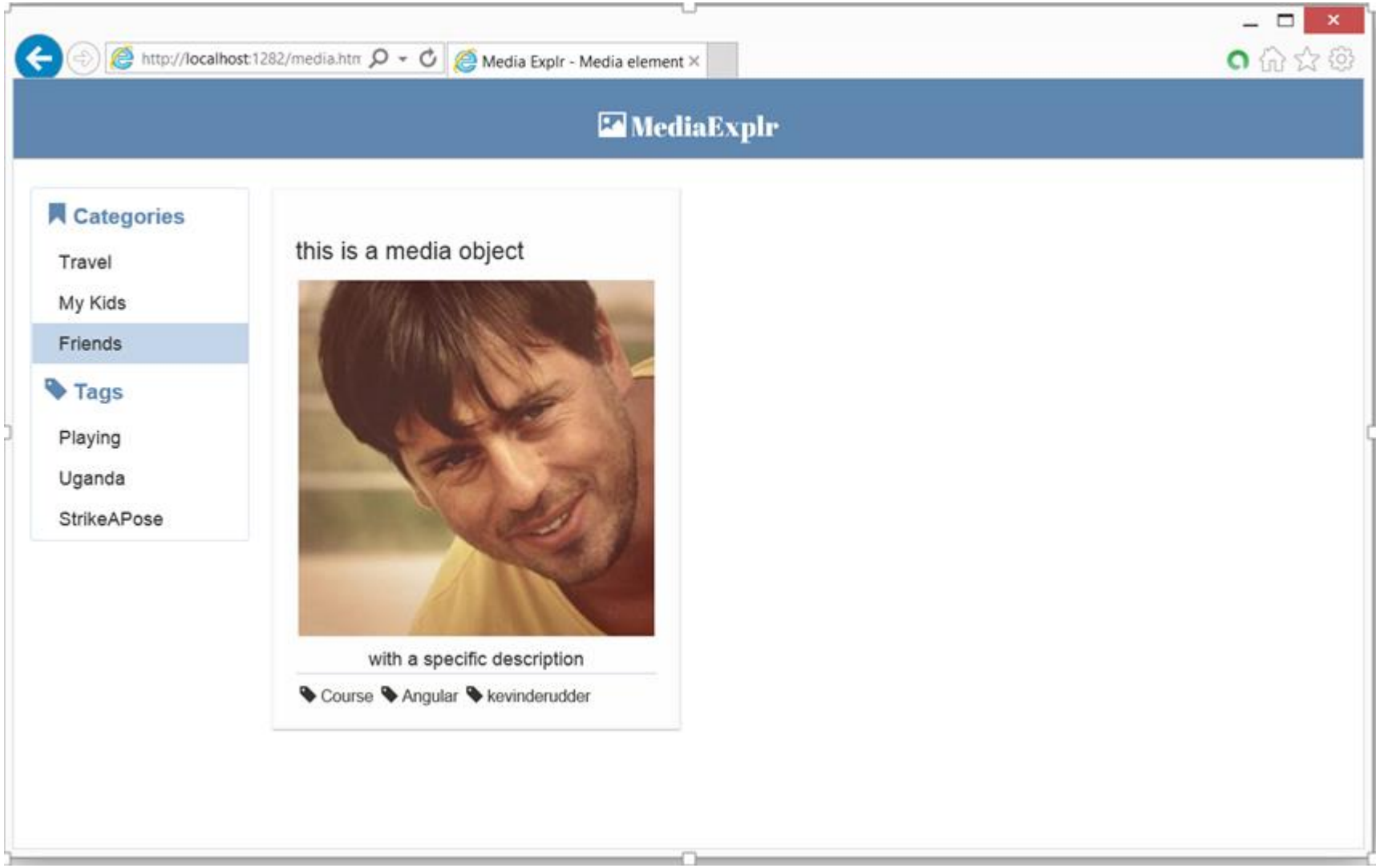
```
<myCourse info="angular"></myCourse>
```

Manipulate the DOM

- If you want to manipulate the DOM, you use the `link` option

- Link takes following parameters
  - Scope
  - Element: jqLite element that matches this directive
  - Attrs: attribute names

Exercise


http://localhost:1282/media.htm
Media Explr - Media element
MediaExplr
Categories
Travel
My Kids
Friends
Tags
Playing
Uganda
StrikeAPose
this is a media object
with a specific description
Course
Angular
kevinderudder