JavaScript + Angular

Part 4

AngularJS: templates, directives, filters

Author: Andrey Kucherenko


ANGULARJS
by Google

templates

```
<!DOCTYPE html>
<html ng-app="app">
  <head>
    <meta charset="utf-8">
    <title>JS+Angular Classes: Lesson 4</title>
    <script src="lib/boundle.js" charset="utf-8"></script>
  </head>
  <body ng-controller="MyConroller">
    <h1>JS+Angular Classes: Lesson 3</h1>
    <p>{{'Hello ' + framework + 'JS!'}}</p>
  </body>
</html>
```

directives

```
angular.module('app', []).directive('myCustomer', function()
  return {
    template: 'Name: {{customer.name}} Address: {{customer.ad
  };
});


<div ng-controller="Controller">
  <div my-customer></div>
  <my-customer></my-customer>
</div>
```

components

```
angular.module('myApp').component('hello', {
  templateUrl: 'hello.html',
  controller: HelloController,
  bindings: {
    hero: '='
  }
});

<!-- components match only elements -->
<div ng-controller="mainCtrl as ctrl">
  <b>Hero</b><br>
  <hello hero="ctrl.hero"></hello>
</div>
```

model

```
angular.module('myApp')
    .controller('helloController', function ($scope) {
    $scope.myModel = {
        name: 'test'
    }
});
```

$compile

```
var $compile = ...; // injected into your code
var scope = ...;
var parent = ...; // DOM element where the compiled template

var html = '<div ng-bind="exp"></div>';

// Step 1: parse HTML into DOM element
var template = angular.element(html);

// Step 2: compile the template
var linkFn = $compile(template);

// Step 3: link the compiled template with the scope.
var element = linkFn(scope);

// Step 4: Append to DOM (optional)
parent.appendChild(element);
```

filter

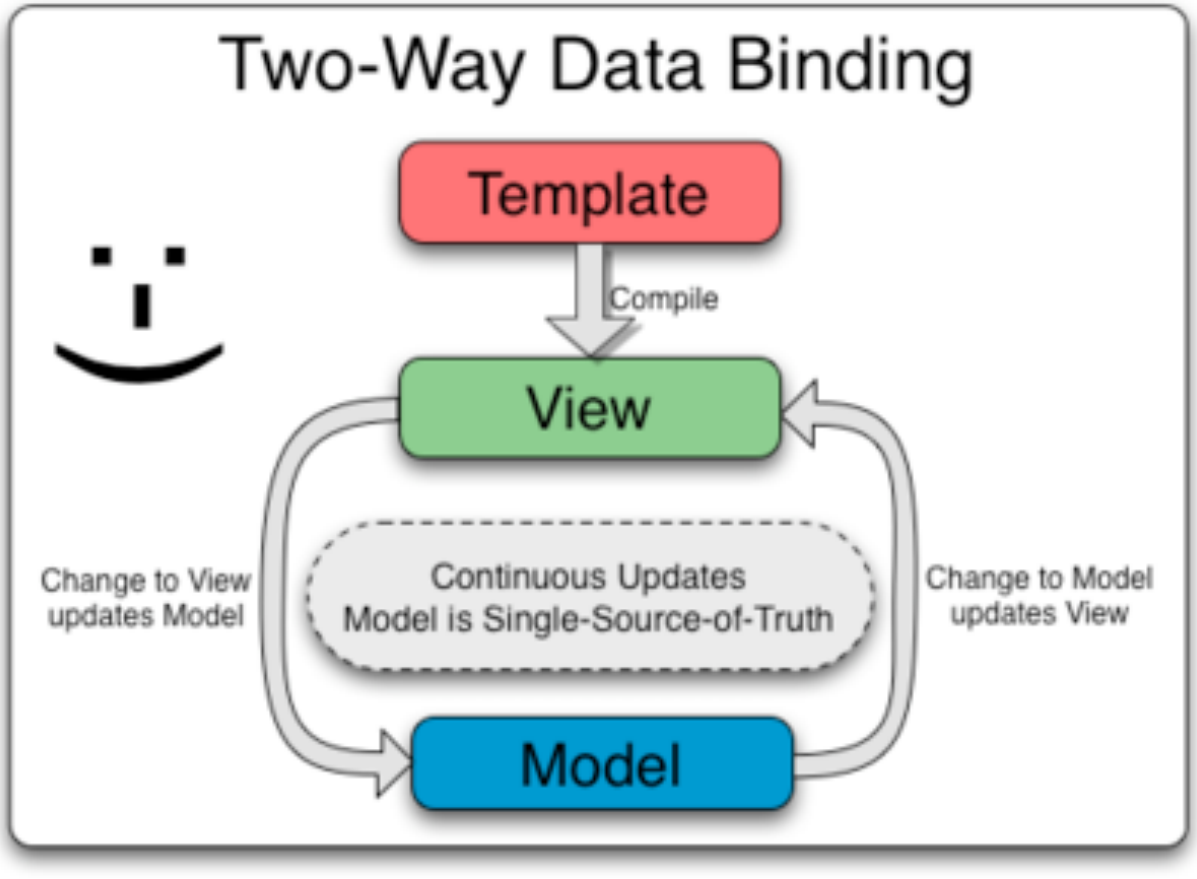
```
{{ expression | filter }}

{{ expression | filter1 | filter2 | ... }}

{{ expression | filter:argument1:argument2:... }}

<div ng-controller="MyController">
  <input ng-model="greeting" type="text"><br>
  No filter: {{greeting}}<br>
  Reverse: {{greeting|reverse}}<br>
  Reverse + uppercase: {{greeting|reverse:true}}<br>
  Reverse, filtered in controller: {{filteredGreeting}}<br>
</div>
```

data binding



Controller

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

myApp.controller('GreetingController', ['$scope', ($scope) => {
  $scope.greeting = 'Hola JS!';
}]);
```

$inject

```
var MyController = function($scope, greeter) {
  // ...
}
MyController.$inject = ['$scope', 'greeter'];
someModule.controller('MyController', MyController);

someModule.controller('MyController', function($scope, greeter) {
  // ...
});
```

module

```
<div ng-app="myApp">
  <div>
    {{ 'World' | greet }}
  </div>
</div>

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

service

```
angular.module('finance2', [])
.factory('currencyConverter', function() {
  //.. implementations

  return {
    methos: method,
    data: data
  };
});
```

Getting Start

```
<!DOCTYPE html>
<html ng-app>
  <head>
    <meta charset="utf-8">
    <title>JS+Angular Classes: Lesson 3</title>
    <script src="lib/boundle.js" charset="utf-8"></script>
  </head>
  <body>
    <h1>JS+Angular Classes: Lesson 3</h1>
    <p>{{'Hello' + ' JS!'}}</p>
  </body>
</html>
```

```
import 'angular';
```

```
<!DOCTYPE html>
<html ng-app="app">
  <head>
    <meta charset="utf-8">
    <title>JS+Angular Classes: Lesson 3</title>
    <script src="lib/boundle.js" charset="utf-8"></script>
  </head>
  <body ng-controller="MyConroller">
    <h1>JS+Angular Classes: Lesson 3</h1>
    <p>{{'Hello ' + framework + 'JS!'}}</p>
  </body>
</html>
```

```
import * as angular from 'angular';


angular.module('app', [
    //.. list of dependencies
]).controller('MyConroller', ($scope) => {
  $scope.framework = 'Angular';
  console.log($scope);
});
```

Tasks

- Create angular.js application
- Run html file with angular app
- Create different angular modules
- Initialize your app in html

templates

```
<!DOCTYPE html>
<html ng-app="HolaApp">
  <head>
    <meta charset="utf-8">
    <title>JS+Angular Classes: Lesson 4</title>
    <script src="lib/boundle.js" charset="utf-8"></script>
  </head>
  <body>
    <h1>JS+Angular Classes: Lesson 4</h1>
    <div class="HolaApp">
      <div ng-controller="MyController as ctrl">
        <h2>Hola App</h2>
        <p>{{'Hola ' + framework + 'JS!' + ctrl.fName}}!</p>
      </div>
    </div>
  </body>
</html>
```

```
<button ng-click="count = count + 1" ng-init="count=0">
  Increment
</button>
<span>
  count: {{count}}
</span>

<button ng-click="clickHandler($event)">Test Click</button>

<button ng-controller="MyConroller as ctrl"
        ng-click="ctrl.clickHandler($event)">
    Test Click1
</button>

angular.module('app', [])
    .controller('MyConroller', function ($scope) {
      $scope.clickHandler = (event) => 'Angular';
      this.clickHandler = (event) => 'Angular';
});
```

Tasks

- Create button with ngClick directive
- Make click counter

```
<span title="{{ attrBinding }}">{{ textBinding }}</span>

<p id="one-time-binding-example">
    One time binding: {{::name}}
</p>
```

Tasks

- Set variable to $scope and show it in html
- Make one-way binding
- Add variable to controller and show it in html
- Change variable on click

```
<input ng-model="foo">

<input type="text" name="userName"
               ng-model="user.name"
               ng-model-options="{ getterSetter: true }" />
```

Tasks

- Create input text and bind with model in controller
- Change value in model
- Change value in DOM

```
<div ng-repeat="(key, value) in myObj"> ... </div>

<div ng-repeat="n in [42, 42, 43, 43] track by $index">
  {{n}}
</div>

<header ng-repeat-start="item in items">
  Header {{ item }}
</header>
<div class="body">
  Body {{ item }}
</div>
<footer ng-repeat-end>
  Footer {{ item }}
</footer>
```

Tasks

- Create array of strings and output in html
- Change array
- Show index of each elements

```
<div class="slide-animate" ng-include="template.url"></div>

<ng-include
  src="string"
  [onload="string"]
  [autoscroll="string"]>
...
</ng-include>
```

Tasks

- Make template in separate html file
- Include html in index.html
- Attach execute onload event

```
<label>
    Click me:
    <input
        type="checkbox"
        ng-model="checked"
        ng-init="checked=true" />
</label><br/>
Show when checked:
<span ng-if="checked" class="animate-if">
  This is removed when the checkbox is unchecked.
</span>
```

Tasks

- Make block of html with ng-if
- Change show status in ng-click
- Check DOM changes

```
<!-- when $scope.myValue is truthy (element is hidden) -->
<div ng-hide="myValue" class="ng-hide"></div>

<!-- when $scope.myValue is falsy (element is visible) -->
<div ng-hide="myValue"></div>

<!-- when $scope.myValue is truthy (element is visible) -->
<div ng-show="myValue"></div>

<!-- when $scope.myValue is falsy (element is hidden) -->
<div ng-show="myValue" class="ng-hide"></div>
```

Tasks

- Make block of html with ng-hide & ng-show
- Change show status in ng-click
- Check DOM changes

```
<div class="animate-switch-container"
  ng-switch on="selection">
   <div class="animate-switch"
     ng-switch-when="settings">Settings Div</div>
   <div class="animate-switch"
     ng-switch-when="home">Home Span</div>
   <div class="animate-switch"
     ng-switch-default>default</div>
</div>
```

Tasks

- Make block of html with ng-switch
- Change show status in ng-click
- Check DOM changes

directive

```
angular.module('app', []).directive('myName', function()
  return {
    template: 'Name: {{name}}'
    //templateUrl: 'path to html'
  };
});

<div ng-controller="MyController">
  <div my-name></div>
</div>
```

```
angular.module('app', []).directive('myName', function() {
  return {
    template: 'Name: {{name}}',
    restrict: 'E' // matches element name
    //restrict: 'A' // matches attrimbute name
    //restrict: 'E' // matches class name
    //restrict: 'M' // matches comment name
    //restrict: 'AE' // matches element & attribue name
  };
});

<div ng-controller="MyController">
  <div my-name></div>
  <my-name></my-name>
  <div class="my-name: exp;"></div>
  <!-- directive: my-directive exp -->
</div>
```

```
export function frameworkNameDirective() {
  return {
    scope: {
      framework: '=info'
    },
    restrict: 'E',
    template: 'Name: {{framework}}'
  };
};

<framework-name info="framework"></framework-name>
```

Tasks

- Create own directive with inline template
- Create template in separate file and include it in directive
- Create different restricts and try to insert to html different directives
- Output data from parent controller
- Create variable in directive's scope from scope of controller

```
export function ololoDirective() {

  function link(scope, element, attrs) {
    element.css({color: 'red'});
    element.on('click', (event) => alert(event));
  }

  return {
    link: link,
    template: "<div>!{{name}}</div>"
  };
};
```

```
export function ololoDirective() {

  return {
    restrict: 'E',
    transclude: true,
    scope: {},
    controller: ['$scope', function($scope) {
      var panes = $scope.panes = [];

      $scope.select = function(pane) {
        angular.forEach(panes, function(pane) {
          pane.selected = false;
        });
        pane.selected = true;
      };

      this.addPane = function(pane) {
        if (panes.length === 0) {
          $scope.select(pane);
        }
        panes.push(pane);
      };
    }],
    templateUrl: 'my-tabs.html'
  };
};
```

Tasks

- Make directive with link handler
- Make directive with controller
- Change directive element styles
- Add event handlers in directives

component

```
function HeroDetailController() {

}

angular.module('app').component('heroDetail', {
  templateUrl: 'heroDetail.html',
  controller: HeroDetailController,
  bindings: {
    hero: '='
  }
});

<!-- components match only elements -->
<div ng-controller="mainCtrl as ctrl">
  <b>Hero</b><br>
  <hero-detail hero="ctrl.hero"></hero-detail>
</div>
```

Tasks

- Make component in separate folder
- Make separate template for component
- Create controller in separate file in component's folder
- Initialize new component
- Use directive in component

filter

```
{{ filter_expression | filter : expression : comparator : anyPropertyKey}}

$filter('filter')(array, expression, comparator, anyPropertyKey)

<tr ng-repeat="friend in friends | filter:searchText">
    <td>{{friend.phone}}</td>
</tr>
```

```
{{ currency_expression | currency : symbol : fractionSize}}

$filter('currency')(amount, symbol, fractionSize)

<script>
  angular.module('currencyExample', [])
    .controller('ExampleController', ['$scope', function($scope) {
      $scope.amount = 1234.56;
    }]);
</script>
<div ng-controller="ExampleController">
  <input type="number" ng-model="amount" aria-label="amount"> <br>
  default currency symbol ($): <span id="currency-default">{{amount | currency}}</span><br>
  custom currency identifier (USD$): <span id="currency-custom">{{amount | currency:"USD$"}}</span>
  no fractions (0): <span id="currency-no-fractions">{{amount | currency:"USD$":0}}</span>
</div>
```

```
{{ number_expression | number : fractionSize}}

$filter('number')(number, fractionSize)

angular.module('numberFilterExample', [])
  .controller('ExampleController', ['$scope', function($scope) {
    $scope.val = 1234.56789;
}]);

<div ng-controller="ExampleController">
  <label>Enter number: <input ng-model='val'></label><br>
  Default formatting: <span id='number-default'>{{val | number}}</span><br>
  No fractions: <span>{{val | number:0}}</span><br>
  Negative number: <span>{{-val | number:4}}</span>
</div>
```

Tasks

- Create array of products objects
- Each object should have price, name and count
- Show list of products
- Add filter for name
- Output price with currency
- Output formatted count
- Apply filter in JS and in HTML

```
{{ date_expression | date : format : timezone}}

$filter('date')(date, format, timezone)

<span ng-non-bindable>{{1288323623006 | date:'medium'}}</span>:
    <span>{{1288323623006 | date:'medium'}}</span><br>
<span ng-non-bindable>{{1288323623006 | date:'yyyy-MM-dd HH:mm:ss Z'}}</span>:
   <span>{{1288323623006 | date:'yyyy-MM-dd HH:mm:ss Z'}}</span><br>
<span ng-non-bindable>{{1288323623006 | date:'MM/dd/yyyy @ h:mma'}}</span>:
   <span>{{'1288323623006' | date:'MM/dd/yyyy @ h:mma'}}</span><br>
<span ng-non-bindable>{{1288323623006 | date:"MM/dd/yyyy 'at' h:mma"}}</span>:
   <span>{{'1288323623006' | date:"MM/dd/yyyy 'at' h:mma"}}</span><br>
```

Pet Project