Telerik Academy

MVC is Cool

Directives

Angular Basics

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

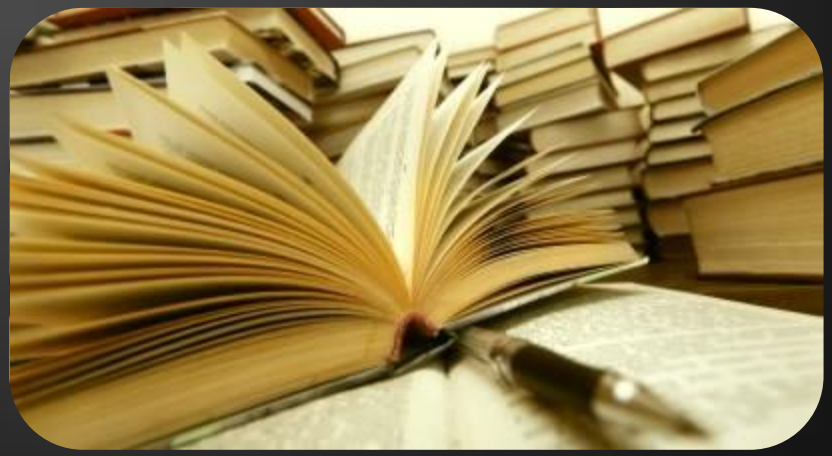
Telerik Academy

Table of Contents

- What is a directive and why we need it
- How to create directives
- Directive specifics for attributes
- jQuery plugins directives

Telerik Academy

What Is a Directive



What Is a Directive

- Directives
  - Extends HTML
  - Make it easier for "dynamic" pages
  - Easier to read document
  - Domain specific tags

What Is a Directive

- Main uses of directives
  - Custom elements (tags)
  - Custom events
  - Observe and react to changes

Telerik Academy

Creating Directives

Creating Directives

- Create directives per module

```
myApp.directive('myDirective', function () {
  return {
     restrict: 'E', // C, A, M, CA, etc.
     template: '<div />' // or templateUrl,
     scope: { // isolated scope - &, @, = },
     link: function(scope, element, attr) {
     // put logic here,  },
     compile: function(element, attr) { // compile },
     replace: true, priority: 0
  };
});
```

Telerik Academy

Directive Attribute Specifics

Attribute Specifics

- Manipulating attribute

```
link: function (scope, element, attrs) {
    console.log(attrs.ngModel);
    attrs.$set('ngModel', 'new value');

    // observe changes to interpolated attribute
    attrs.$observe('ngModel', function(newVal, oldVal) {
      console.log('ngModel has changed to ' + value);
    });
}
```

Telerik Academy

jQuery Plugins Directives

jQuery as Directive

- jQuery plugin directives

```
<input type="text" date-picker />
```

```
myApp.directive('datePicker', function () {
  return {
     restrict: 'A',
     link: function(scope, element, attr) {
        element.datepicker();
     },
  };
});
```

Summary

- What are directives
- Create directives
- Specifics

Telerik Academy

AngularJS

http://academy.telerik.com

Exercises

1. Create AngularJS SPA client for the Tic-Tac-Toe Web API project used in the "Web Services and Cloud" course in lecture "Web API Architecture"
   - Use routing – for every single page
   - Use controllers – games, users, playing, etc.
   - Use services – restful usage, game logic, etc.
   - Use filters – for visualization of the bindings
   - Use directives – for domain specific parts of the HTML