Frontend Development

AngularJS Forms

Pregatit de: Alexandru Dima   Data: 20.10.2014

Topics

- services
- ng-options
- ng-form

Pregatit de: Alexandru Dima   Data: 20.10.2014

Services

- Objects used for organize and share code in applications;
- Injected in controllers or other services
- Lazy instantiated
- Singleton

TEAMNET

Services

- Most common types of service recipes:
    - Factory – calls the function and injects the returned value;
    - Service – calls the function as a constructor and injects the created object;

TEAMNET

Services

- Factory recipe:

```
myApp.factory('myFactory', function() {

...

});


myApp.controller('myController', ['myFactory',
function(myFactory){

...

});
```

TEAMNET

Services

- Service recipe:

```
myApp.service('myService', ['myFactory', function(myFactory) {

...

});


myApp.controller('myController', [$scope, 'myFactory',
'myService', function($scope, myFactory, myService){

...

});
```

TEAMNET

ng-options

- Built-in AngularJS directive
- Used to generate <option> elements for a <select> element
- Iterates an array or an object and generates a <option> element for each item
- Used with ng-model directive
- Bind the selected option to the model form the ng-model directive

TEAMNET

ng-options

for array data sources:

- label **for** value **in** array
- select **as** label **for** value **in** array
- label **group by** group **for** value in array
- select **as** label **group by** group **for** value **in** array **track by** trackexpr

for object data sources:

- label **for** (key , value) **in** object
- select **as** label **for** (key , value) **in** object
- label **group by** group **for** (key, value) **in** object
- select **as** label **group by** group **for** (key, value) **in** object

TEAMNET

ng-options

```
$scope.colors = [
    {name:'black', shade:'dark'},
    {name:'white', shade:'light'},
    {name:'red', shade:'dark'},
    {name:'blue', shade:'dark'},
    {name:'yellow', shade:'light'}
];

<select ng-model="myColor" ng-options="color.name for color in colors">
</select>

<select ng-model="myColor" ng-options="color.name group by color.shade for color in colors">
</select>
```

TEAMNET

ng-form

- Similar to <form> element from HTML
- Can be used as element or attribute
- Can be nested inside another ng-form directive
- Useful for validation of forms
- 'name' attribute is used to publish to current scope form controller

TEAMNET

ng-form

- Elements:
    - input(text, number, email, checkbox, radio,…)
    - select
    - textarea

ng-form

- Directives:
  - ng-required="[boolean]"
  - ng-disabled="[boolean]"
  - ng-submit="[expression]"
  - ng-dirty
  - ng-valid
  - ng-invalid

TEAMNET

ng-form

```
<form name="myForm" ng-controller="FormController" class="my-form">
  userType: <input name="input" ng-model="userType" required>
  <span class="error" ng-show="myForm.input.$error.required">Required!</span><br>
  userType = {{userType}}<br>
  myForm.input.$valid = {{myForm.input.$valid}}<br>
  myForm.input.$error = {{myForm.input.$error}}<br>
  myForm.$valid = {{myForm.$valid}}<br>
  myForm.$error.required = {{!!myForm.$error.required}}<br>
</form>
```
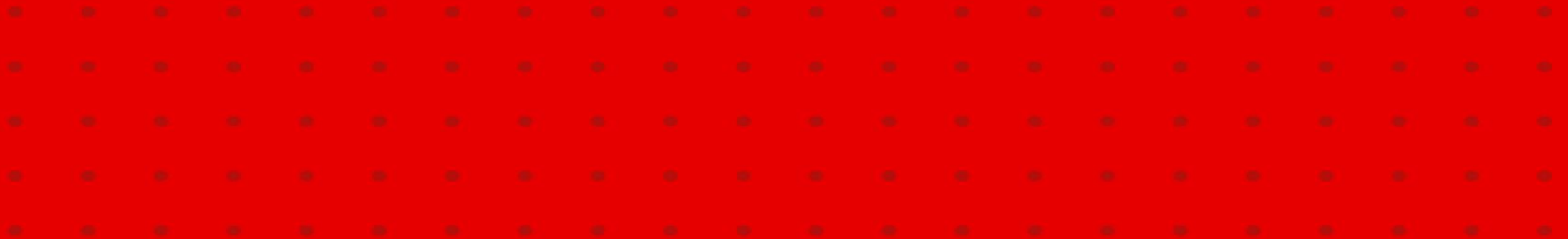
Workshop:
AngularJS Forms

Questions?

Thank you!