

Forms & Input Elements



Deborah Kurata

@deborahkurata | blogs.msmvps.com/deborahk/

What This Module Covers



Date-based input types

Data binding options

Rolling back changes

Submission detection

Binding to getters and setters

Date-Based Input Types

Angular 1.2.x

```
<input type="text"  
      id="inputStartDate">
```

Angular 1.3

```
<input type="date"  
      id="inputStartDate">
```

Also:

- "time"
- "datetime-local"
- "month"
- "week"

JSON

```
{  
  "classId": 4,  
  "className": "Middle Earth: A History",  
  "startDate": "2015-01-21",  
  "endDate": "2015-05-10",  
  "instructorEmail": "took@pcc.edu",  
  "description": "Relive Middle Earth adventures through  
tales and songs."  
}
```

Convert to Date

```
vm.class.startDate = new Date(vm.class.startDate);
```

Tue Jan 20 2015 16:00:00
GMT-0800 (Pacific
Standard Time)

2015-01-21

Formatted to a Localized Date

Online Class Scheduler

Edit: Middle Earth: A History

Class Name

Middle Earth: A History

Start Date

01/20/2015

End Date

05/09/2015

Instructor's Email

took@pcc.edu

Description

Relive Middle Earth adventures through tales and songs.

Save Cancel

```
<input  
  class="form-control"  
  type="date"  
  id="inputStartDate"  
  name="inputStartDate"  
  ng-model="vm.class.startDate">
```

Tue Jan 20 2015 16:00:00
GMT-0800 (Pacific
Standard Time)

Date-Based Input Types



`input type="text"` still works

`input type="date"` works better ...

... but requires a `Date` object

Data Binding Options

Angular 1.2.x

No options

Angular 1.3

```
<input  
  type="email"  
  id="inputInstructorEmail"  
  name="inputInstructorEmail"  
  ng-model="vm.class.instructorEmail"  
  ng-model-options="{updateOn: 'blur'}"/>
```


Default Data Binding

Instructor's Email

took

```
<input  
  class="form-control"  
  type="email"  
  id="inputInstructorEmail"  
  name="inputInstructorEmail"  
  ng-model="vm.class.instructorEmail"/>
```

took

Default Data Binding

Instructor's Email

t

Invalid Email Address

```
<input  
  class="form-control"  
  type="email"  
  id="inputInstructorEmail"  
  name="inputInstructorEmail"  
  ng-model="vm.class.instructorEmail"/>
```

Data Binding with Model Options

Instructor's Email

took

```
<input
  class="form-control"
  type="email"
  id="inputInstructorEmail"
  name="inputInstructorEmail"
  ng-model="vm.class.instructorEmail"
  ng-model-options="{updateOn: 'blur'}"/>
```

Data Binding with Model Options

Instructor's Email

took

```
<input  
  class="form-control"  
  type="email"  
  id="inputInstructorEmail"  
  name="inputInstructorEmail"  
  ng-model="vm.class.instructorEmail"  
  ng-model-options="{debounce: 1500}"/>
```

took

Data Binding with Model Options

Instructor's Email

```
<input
  class="form-control"
  type="email"
  id="inputInstructorEmail"
  name="inputInstructorEmail"
  ng-model="vm.class.instructorEmail"
  ng-model-options="{debounce:
    {'default':500, 'blur':0}}"/>
```

Data Binding Options



Use model options for finer control over model updates:

`updateOn`

`debounce`

Use `ngSubmit`

More model options later

Rolling Back Changes

Angular 1.2.x

Manual process

Angular 1.3

`control.$rollbackViewValue();`

Rolling Back Changes

Instructor's Email

took@pcc.edu

took@pcc.edu

```
<input  
  class="form-control"  
  type="email"  
  id="inputInstructorEmail"  
  name="inputInstructorEmail"  
  ng-model="vm.class.instructorEmail"  
  ng-model-options="{updateOn: 'blur'}" />
```


Rolling Back Changes

Instructor's Email

took@pcc.edu

took@pcc.edu

```
<input
  class="form-control"
  type="email"
  id="inputInstructorEmail"
  name="inputInstructorEmail"
  ng-model="vm.class.instructorEmail"
  ng-model-options="{updateOn: 'blur'}"
  ng-keyup="vm.cancelEntry(classForm.inputInstructorEmail, $event)"/>
```

cancelEntry Method

```
vm.cancelEntry = function (control, event) {  
    if (event.keyCode == 27) {  
        control.$rollbackViewValue();  
    }  
};
```

Rolling Back All Form Changes

Set module options at the form level

Add undo/reset button to rollback all of the input elements

```
<form
  class="form-horizontal"
  name="classForm"
  ng-submit="vm.submit()"
  ng-model-options="{updateOn:'submit'}"
  novalidate>

<button
  class="btn btn-default"
  style="width:70px"
  ng-click="classForm.$rollbackViewValue()">
  Reset
</button>
```

Rolling Back Changes



Use `$rollbackViewValue` to roll back an input element to the model value

Also useful for updating the view if the model changes behind the scenes

Can be used at the form level

Only needed when using model options.

Submission Detection

Angular 1.2.x

Detect dirty/pristine, valid/invalid

Angular 1.3

Detect submission
\$submitted

Submission Detection



Use `$submitted` to

Display a message

Perform an operation

After the user has submitted the form

Getter/Setter Function

```
var _startDate = new Date(vm.class.startDate);  
vm.class.dates = {  
    startDate: function(value){  
        if (angular.isDefined(value))  
        {  
            _startDate=value;  
            vm.class.startDate=_startDate;  
        }  
        return _startDate;  
    }  
};
```

Binding to a Getter/Setter Function

Angular 1.2.x

Manual process

Angular 1.3

Defined with a model option

{ getterSetter: true }

Binding to a Getter/Setter Function



Use the `getterSetter` model option to bind to a getter/setter function

Use at the form level to bind to getter/setter functions for all input elements

What This Module Covered



Date-based input types

Data binding options

Rolling back changes

Submission detection

Binding to getters and setters