Form Validation in 1.3

Deborah Kurata

@deborahkurata | blogs.msmvps.com/deborahk/

What This Module Covers

HTML 5 validator improvements

Validation message directives

Reusable validation messages

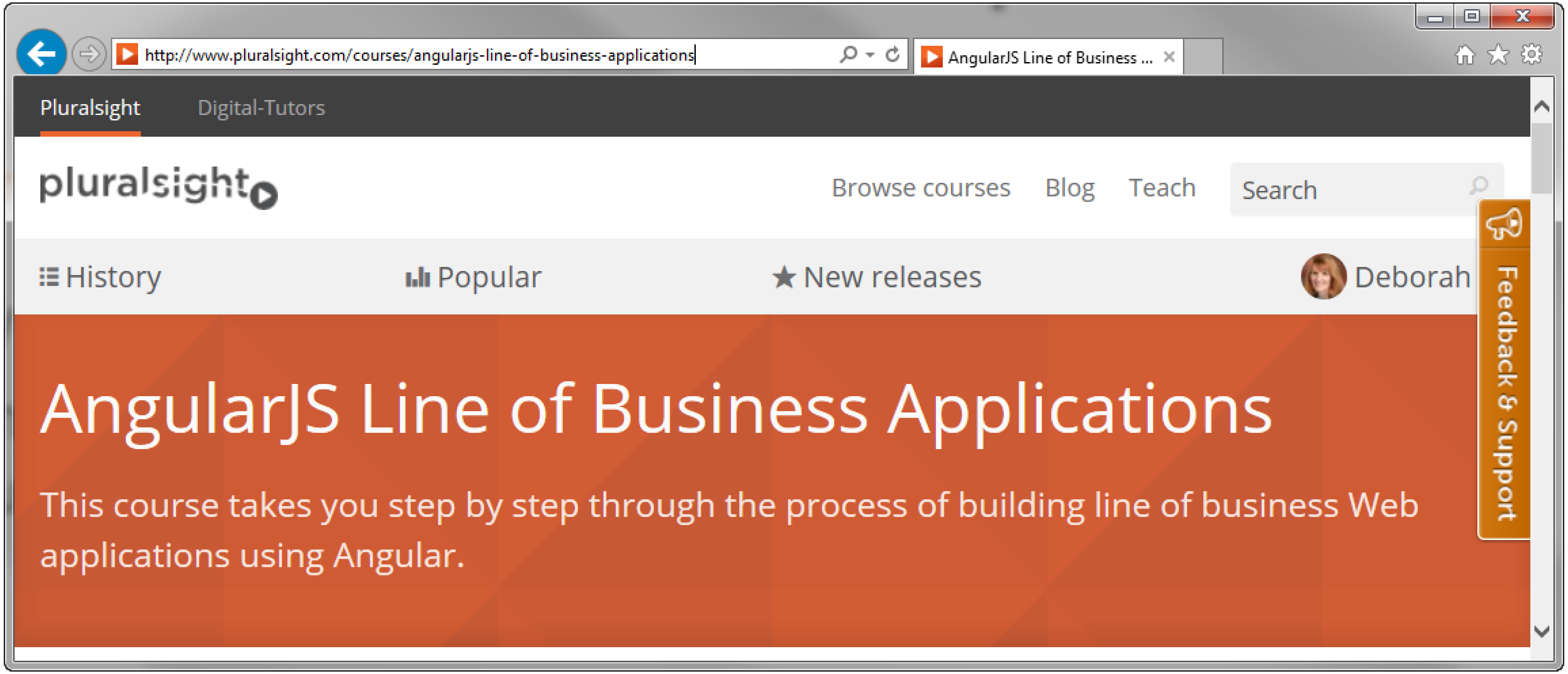
Touch detection

Dynamic element validation

Simplified custom validation

Asynchronous validation

If Form Validation Is New for You



http://www.pluralsight.com/courses/angularjs-line-of-business-applications

AngularJS Line of Business ...

Pluralsight Digital-Tutors

pluralsight

Browse courses Blog Teach Search

History Popular New releases Deborah

AngularJS Line of Business Applications

This course takes you step by step through the process of building line of business Web applications using Angular.

Feedback & Support

HTML 5 Validator Improvements

Angular 1.2.x

It mostly worked

- Depending on the version of 1.2 you were using

Angular 1.3

It fully works

HTML 5 Validator Improvements

HTML 5 input type validation now works

Validation errors are defined on $error

Displaying Validation Messages

Angular 1.2.x

```
<span class="help-block has-error"
      ng-if="classForm.inputEmail.$dirty">
  <span ng-show="classForm.inputEmail.$error.required">
    Email is required.
  </span>
  <span ng-show="classForm.inputEmail.$error.minlength">
    Instructor's email must be at least 6 characters.
  </span>
  <span ng-show="classForm.inputEmail.$error.email">
    Instructor's email must be a valid email address.
  </span>
</span>
```

Angular 1.3

```
<span class="help-block has-error"
      ng-if="classForm.inputEmail.$dirty"
      ng-messages="classForm.inputEmail.$error">
  <span ng-message="required">
    Email is required.
  </span>
  <span ng-message="minlength">
    Instructor's email must be at least 6 characters.
  </span>
  <span ng-message ="email">
    Instructor's email must be a valid email address.
  </span>
</span>
```

ngMessages Directive

Shows and hides validation messages

Based on $error object

In the order specified

ngMessage Directive

Shows or hides one validation message

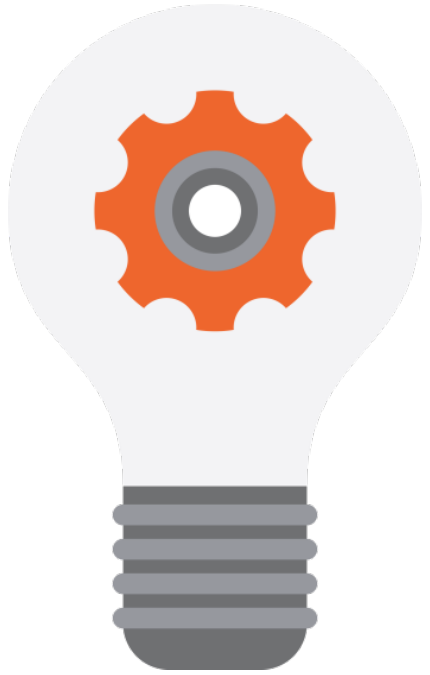
Requires a parent ngMessages directive

Is defined on a child HTML element

angular-messages.js

- ngMessages and ngMessage directives are in a separate .js file
- Include the angular-messages.js file in the project
- Add a script tag for the angular-messages.js file
- And define a dependency on ngMessages

Displaying Validation Messages



Use ngMessages and ngMessage for

- Better control
- Cleaner code

Use them to define message order

Or to display multiple messages

Reusing Validation Messages

Angular 1.2.x

Manual process

Angular 1.3

ng-messages-include="errorMessages.html"

Using ng-messages-include

1. Create an HTML file for the common messages
2. Include the HTML file in the same element as ngMessages
3. Override any messages as required

Creating the Messages File

```
<span ng-message="required">
This item cannot be blank.</span>
<span ng-message="minlength">
You have not met this item's minimum length.</span>
<span ng-message="maxlength">
You have exceeded this item's maximum length.</span>
```

Including the Messages File

```
<span class="help-block has-error"
      ng-if="classForm.inputInstructorEmail.$dirty"
      ng-messages="classForm.inputInstructorEmail.$error"
      ng-messages-include="app/errorMessages.html"
      ng-messages-multiple>
    <span ng-message="email">
        Instructor's email must be a valid email address.
    </span>
</span>
```

Overriding a Message

```
<span class="help-block has-error"
      ng-if="classForm.inputInstructorEmail.$dirty"
      ng-messages="classForm.inputInstructorEmail.$error"
      ng-messages-include="app/errorMessages.html"
      ng-messages-multiple>
    <span ng-message="email">
        Instructor's email must be a valid email address.
    </span>
    <span ng-message="minlength">
    Instructor's email must be at least 6 characters in length.
    </span>
</span>
```

Reusing Validation Messages

Create an HTML file containing the messages to reuse

Use ng-message-include to include the messages

Override messages as needed

Touched Detection

Angular 1.2.x

Detect dirty/pristine, valid/invalid

Angular 1.3

- Detect touched
  - $touched
  - $untouched
- Set on blur
- NOTE: Does not necessarily mean "touched"

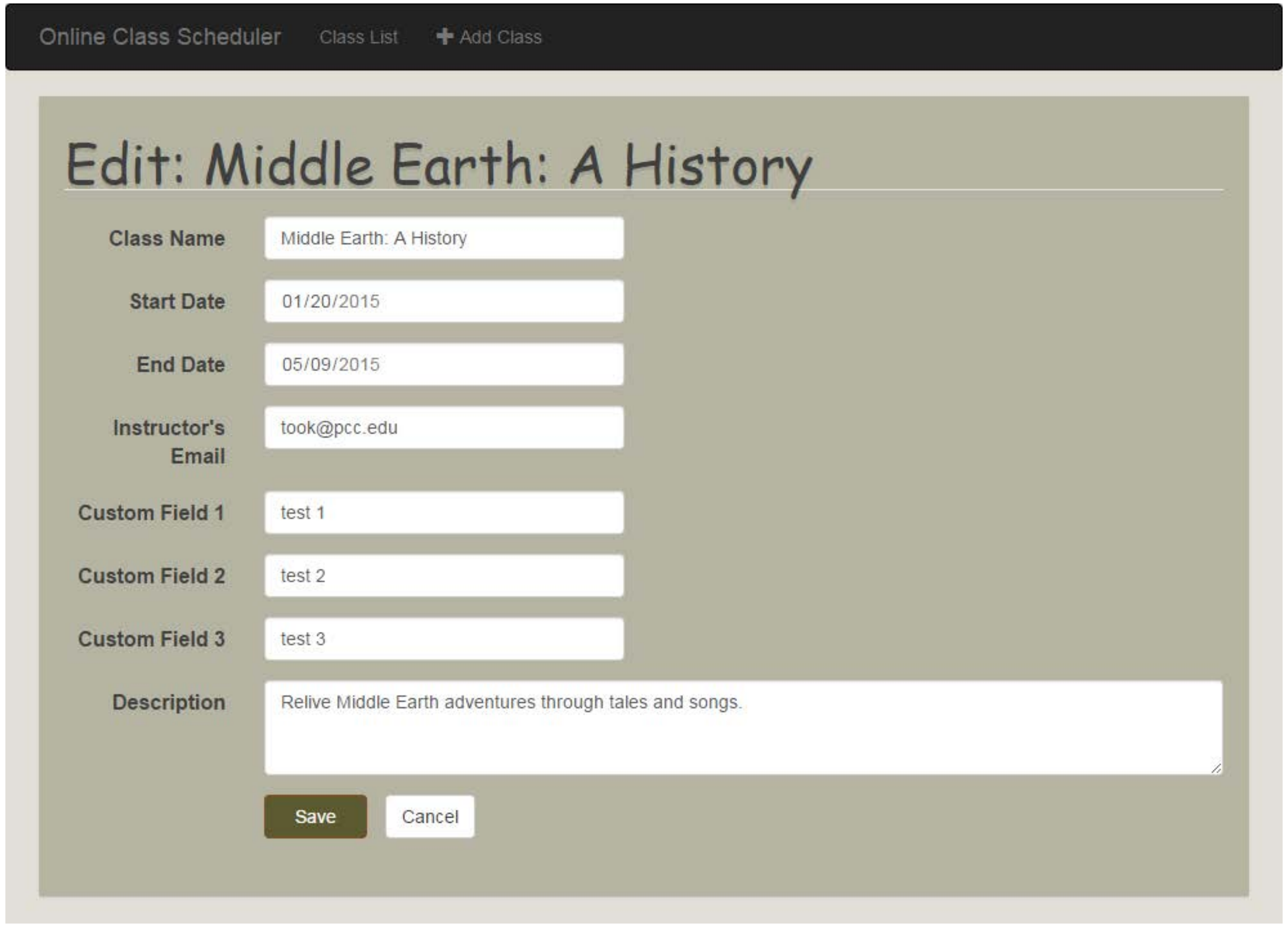
Touched Detection

Use $touched to

- Display a message
- Perform an operation

After the user leaves an input element

Dynamic Element Binding



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
Custom Field 1	test 1
Custom Field 2	test 2
Custom Field 3	test 3
Description	Relive Middle Earth adventures through tales and songs.

```
vm.fields = [
    {"label":"Custom Field 1",
     "data":"test 1"},
    {"label":"Custom Field 2",
     "data":"test 2"},
    {"label":"Custom Field 3",
     "data":"test 3"}
];
```

Dynamic Element Binding

Angular 1.2.x

Manual solution

Angular 1.3

Properly binds to dynamically generated input elements

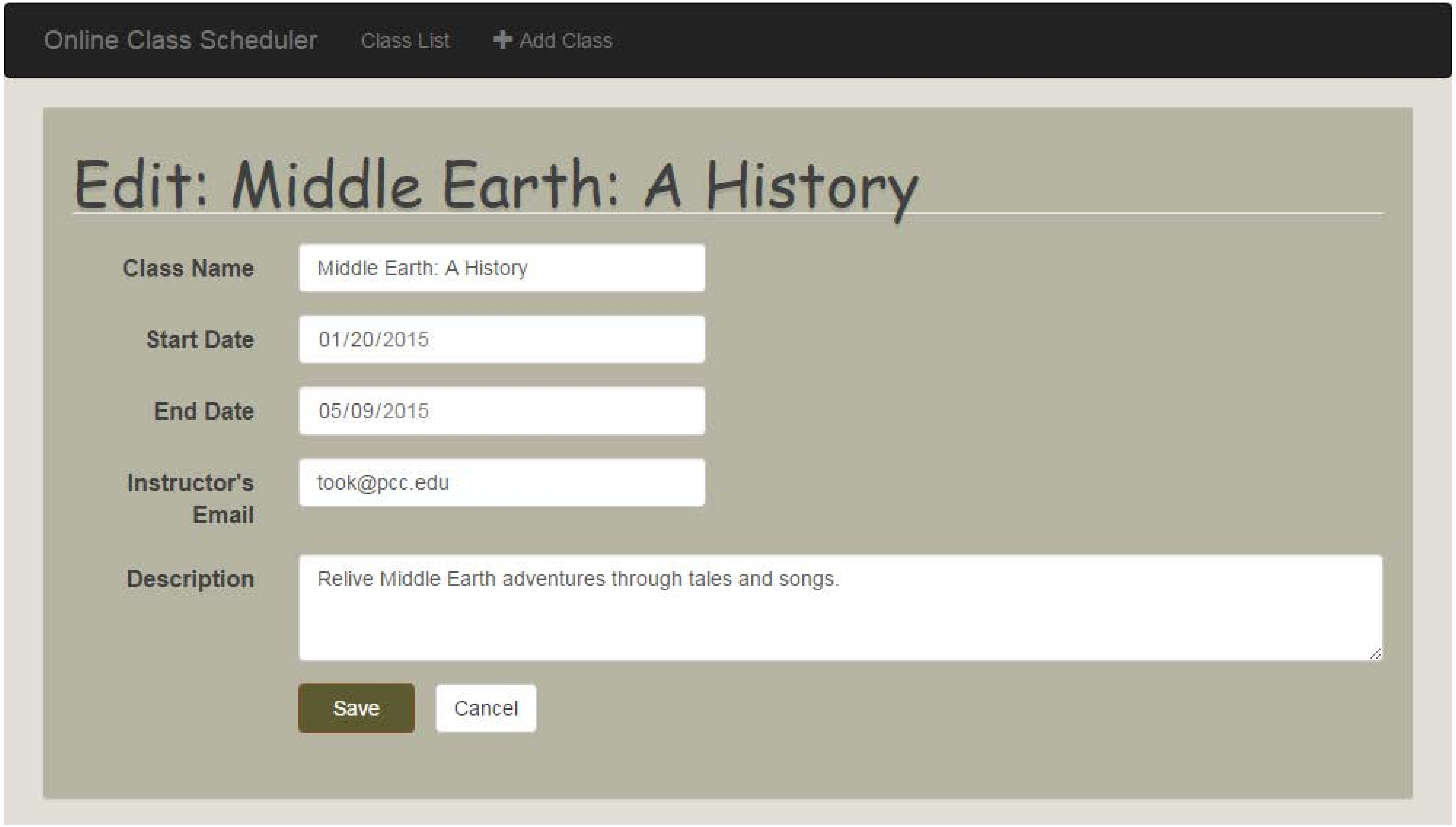
Dynamic Element Binding

If you need to define input elements dynamically …

The binding and validation will now work appropriately

Use reflective calls to access form elements

Custom Validation


Online Class Scheduler | Class List | ✚ Add Class

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.

Enhanced Custom Validators

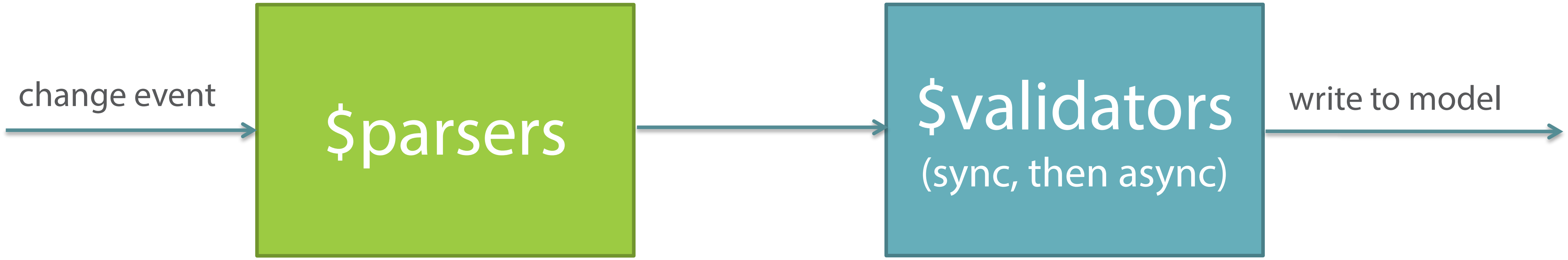
Angular 1.2.x

$parsers and $formatters were used to build custom validation

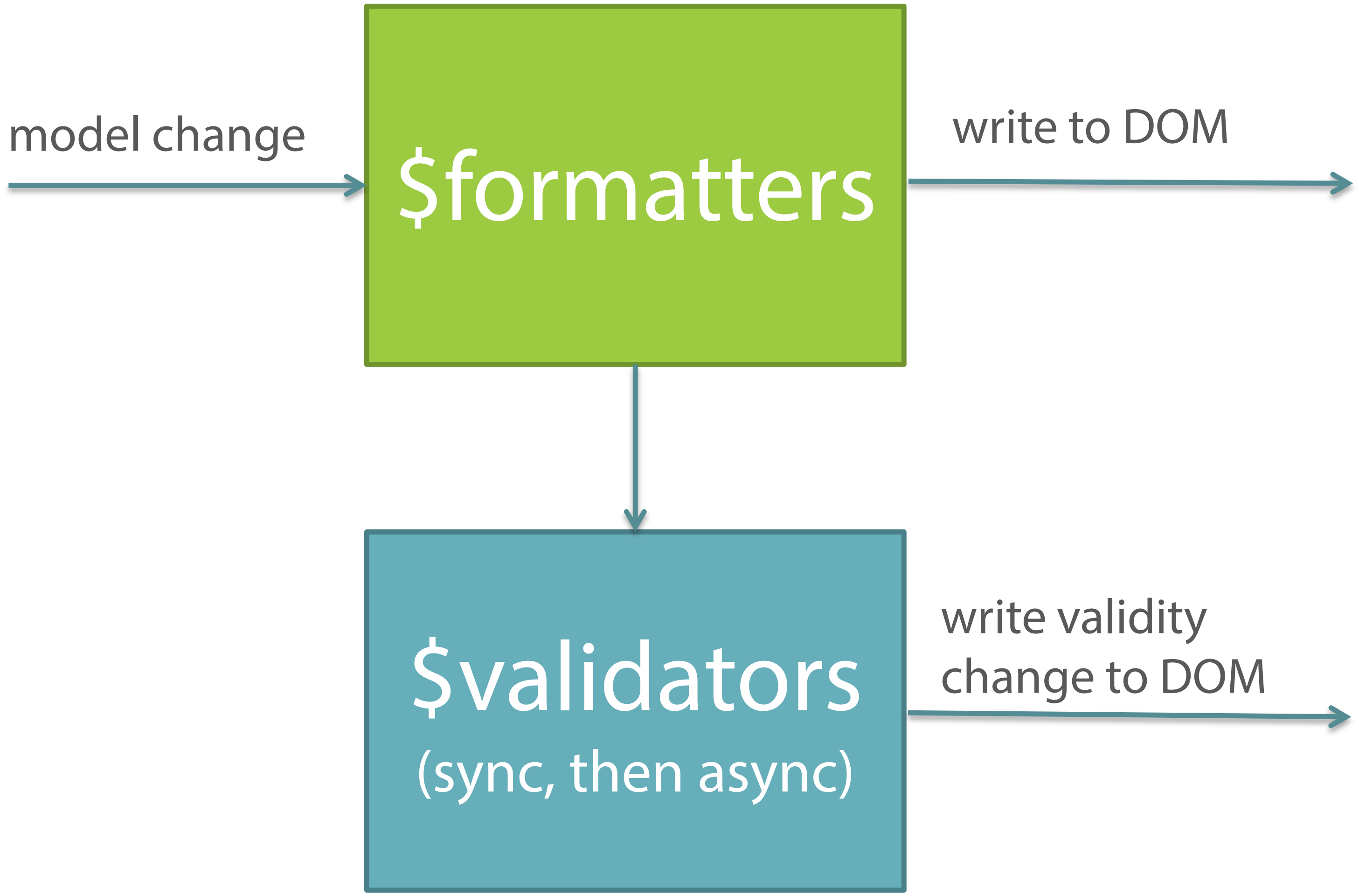
Angular 1.3

New $validators simplifies custom validation

Pipeline: View to Model



Pipeline: Model to View



$validators

Collection of validators

Key => Validator name

Value => Validation function

Parameter => Model Value

Returns => True or False

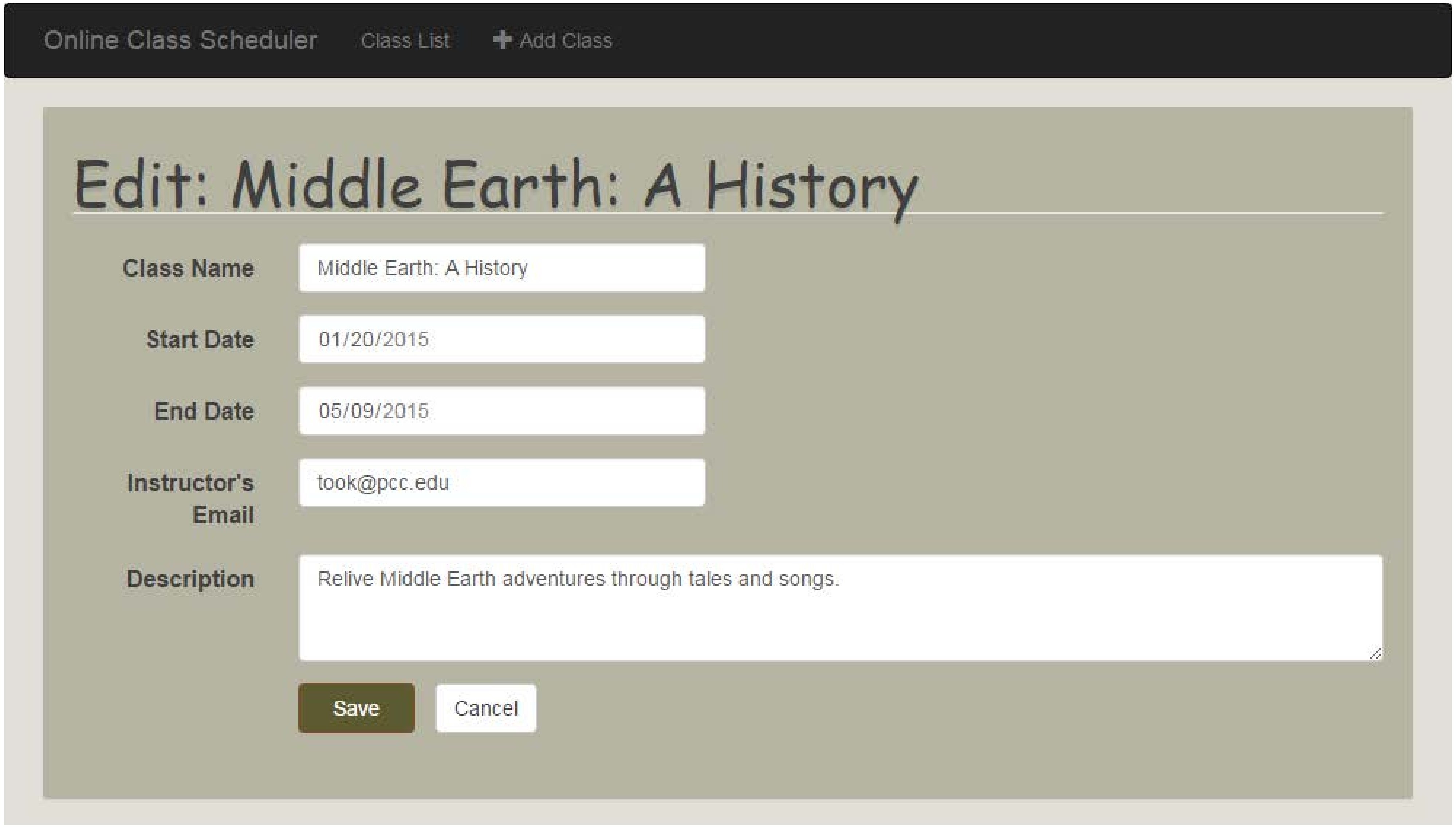
```
ngModel.$validators.dateRangeValidator =
    function (modelValue) {
       return Date.parse(modelValue) >=
              Date.parse(scope.beginDate);
    };
```

Enhanced Custom Validators

It is now easier to build custom validation functions

Use the new $validators

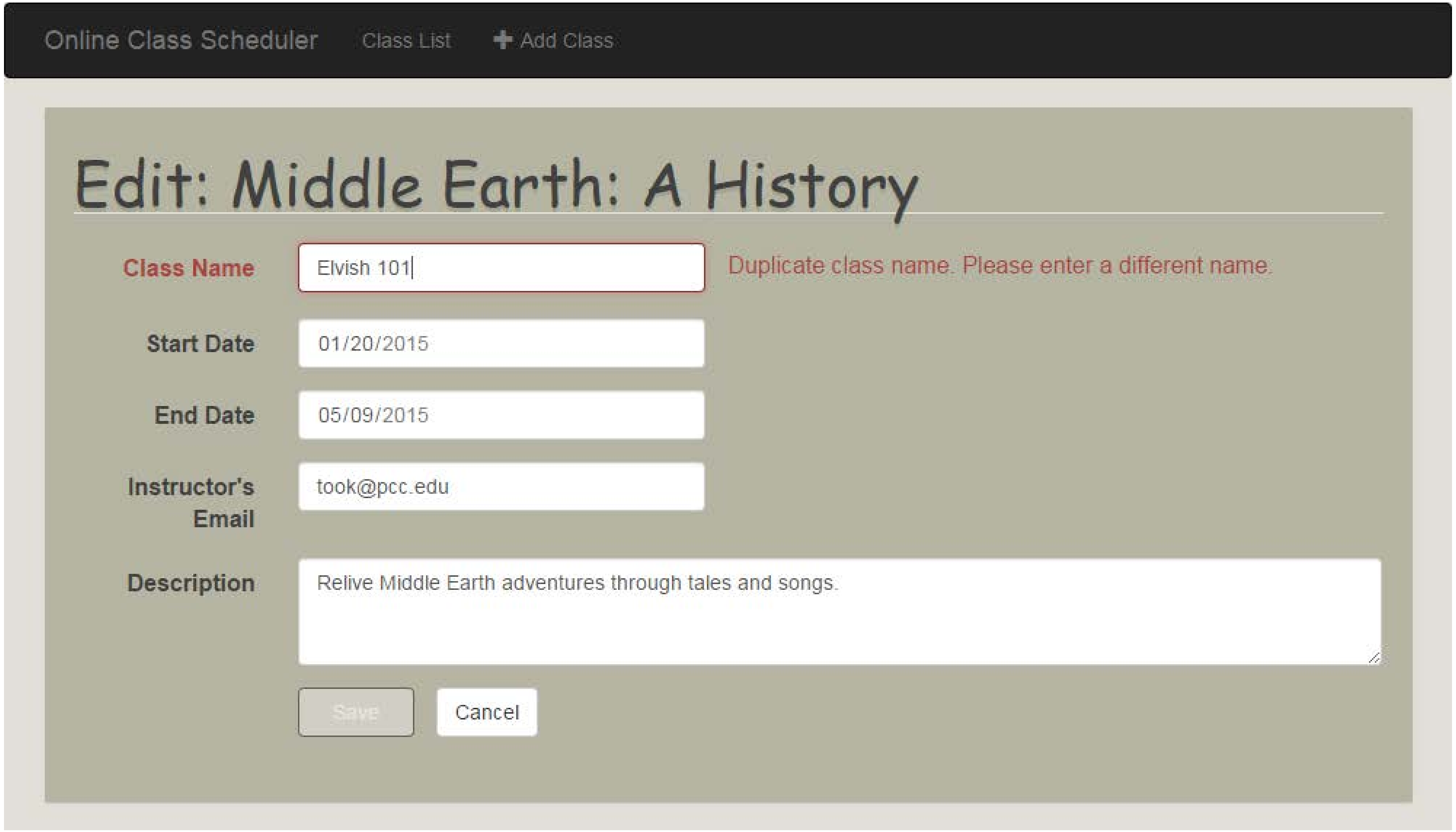
Asynchronous Validation



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.

Asynchronous Validation

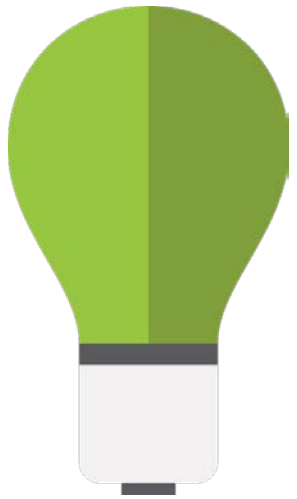


Edit: Middle Earth: A History

Class Name	Elvish 101
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.

Duplicate class name. Please enter a different name.

Asynchronous Validation Is Great For



Duplicate value checking

Validation against server-side business rules

Asynchronous Validation

Angular 1.2.x

Manual process

Angular 1.3

New $asyncValidators simplifies asynchronous validation

$asyncValidators

Collection of validators

Key => Validator name

Value => Validation function

Parameter => Model Value

Returns => A promise

$pending

```
ngModel.$asyncValidators.duplicateClassName =
    function (modelValue) {
       var defer = $q.defer();
       classResource.get({ classId: classId,
                           className: modelValue },
         function (response) {
               defer.reject("Exists");
         },
         function (response) {
                defer.resolve();
         });
      return defer.promise;
   };
```

Asynchronous Validation

It is now easier to build custom asynchronous validation functions

Use the new $asyncValidators

Use the new $pending status as needed

What This Module Covered

HTML 5 validation improvements

Validation message directives

Reusable validation messages

Touch detection

Dynamic element validation

Simplified custom validation

Asynchronous validation