

# Angular Forms and Validation

**Deborah Kurata**

Consultant | Developer | Mentor

Level: Intermediate



NAVIGATE THE  
.NET HIGHWAY

# Deborah Kurata

- Independent Consultant | Developer | Mentor
  - Web (Angular), .NET
- Pluralsight Author
  - Angular with TypeScript
  - AngularJS Line of Business Applications
  - Angular Front to Back with Web API
  - Object-Oriented Programming Fundamentals in C#
- Microsoft MVP

---

# Session Materials & Sample Code

<https://github.com/DeborahK/VSLive2015-NY>

---

# Rate Yourself on **Angular Forms**

- New to Angular - **no** proficiency
- Just starting out - **limited** proficiency
- Doing it, not fully understanding it - **working** proficiency
- Been there, done that, can help others - **full** proficiency

# Overview



Building a data entry form

Basic validation

Validation messages

Data Binding Options

Custom validators

Asynchronous validators

Peek at Angular 2 Forms

# DEMO: ANGULAR IN ACTION

```
<form name="movieForm">  
  <fieldset>  
    <legend>Edit Movie Information</legend>  
  </fieldset>  
</form>
```

## Creating a Form

It's just HTML





# Data Binding

- One-Way Binding
  - View is the projection of the model
  - When the model changes, the view reflects the change
- Two-Way Binding
  - View and model data are synchronized
  - Changes to the model are reflected in the view
  - Changes in the view are reflected in the model

# Two-Way Binding

```
<div>
  <label for="inputMovieTitle">
    Movie Title</label>
  <input id="inputMovieTitle"
    type="text"
    placeholder="Movie Title" />
</div>
<div>
  <label for="inputDescription">
    Description</label>
  <textarea id="inputDescription"
    placeholder="Description"
    rows="3" />
</div>
```

# Two-Way Binding

```
<div>
  <label for="inputMovieTitle">
    Movie Title</label>
  <input id="inputMovieTitle"
    type="text"
    placeholder="Movie Title"
    ng-model="vm.movie.title" />
</div>
<div>
  <label for="inputDescription">
    Description</label>
  <textarea id="inputDescription"
    placeholder="Description"
    rows="3"
    ng-model="vm.movie.description" />
</div>
```

# The Result

InStep Movie Hunter   Search by Title   + Add Movie

Edit Movie Information

Movie Title

The Lord of the Rings: TI

Description

While Frodo and  
Sam edge closer  
to Mordor with the

^  
v

# Bootstrap

- Framework for prettifying Web pages
- Developed by Twitter
  - <http://getbootstrap.com/>
- Large third party community



# Bootstrap Grid System

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

# Bootstrap Styles

```
<div class="form-group">
  <label class="col-md-2 control-label"
    for="inputMovieTitle">
    Movie Title</label>
  <div class="col-md-6">
    <input class="form-control"
      id="inputMovieTitle"
      type="text"
      placeholder="Movie Title"
      ng-model="vm.movie.title" />
  </div>
</div>
```

# The Result

InStep Movie Hunter

Search by Title

+ Add Movie

The Lord of the Rings: The Two Towers

## Edit Movie Information

**Movie Title**

The Lord of the Rings: The Two Towers

**Description**

While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.

Save

Cancel



# Validation

InStep Movie Hunter    Search by Title    + Add Movie

The Lord of the Rings: The Two Towers

## Edit Movie Information

Movie Title

Movie title is required.

Description

While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.

Save

Cancel

# Prepare the form

Ensure the  
form has a  
name

Set the  
novalidate  
attribute

```
<form class="form-horizontal"  
      name="movieForm">  
  novalidate>
```

# Validation Techniques

Type Attribute

```
type="email"
```

Required Attribute

```
required
```

HTML Validation Attributes

```
min = "1"
```

Angular Validation Attributes

```
ng-minlength = "3"
```

Custom Angular Directive

```
my-directive=""
```

# Validation Attributes

```
<div class="col-md-8">  
  <input class="form-control"  
    id="inputMovieTitle"  
    name="inputMovieTitle"  
    type="text"  
    placeholder="Title (required)"  
    ng-model="vm.movie.title"  
    required  
    ng-minlength="3"  
    ng-maxlength="50" />  
</div>
```

# Marking Invalid Elements

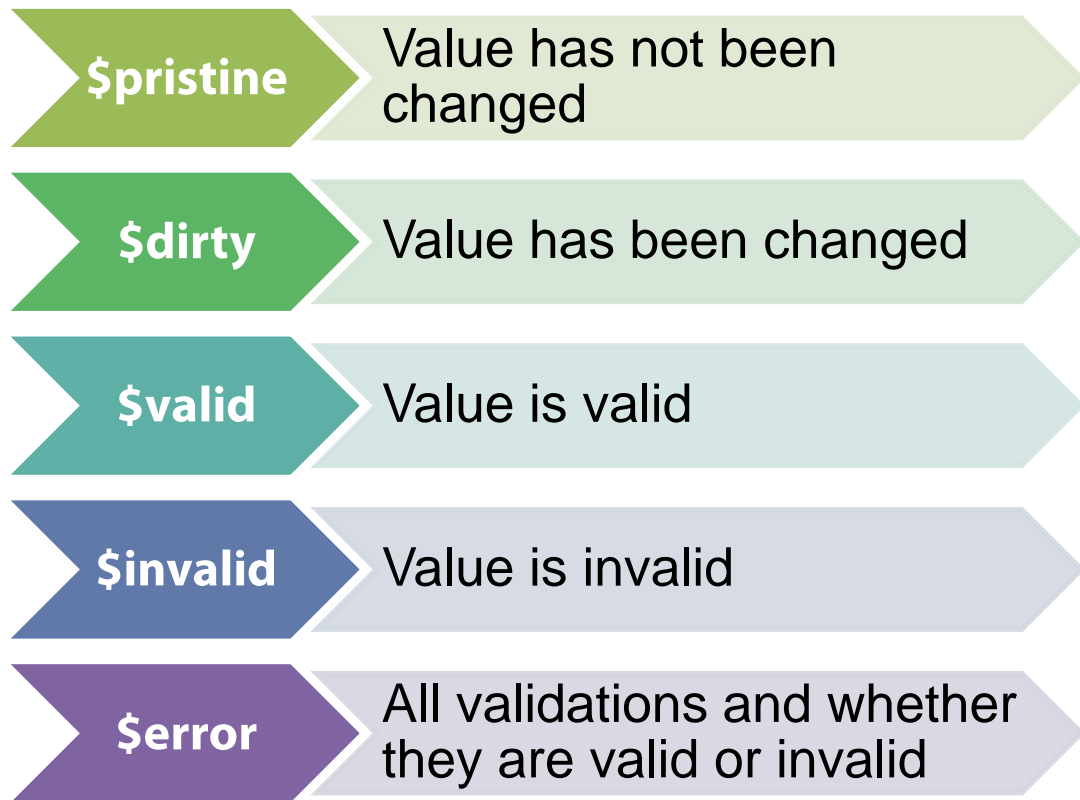
**ngClass  
Directive**

**Bootstrap  
Style**

**Angular  
Validation  
States**

```
<div class="form-group"  
  ng-class="{ 'has-error':  
    movieForm.inputMovieTitle.$invalid &&  
    movieForm.inputMovieTitle.$dirty}">  
  
</div>
```

# Angular Validation States



# Validation States: New in 1.3

**\$touched**

Control was "touched" (on blur)

**\$untouched**

Control never "touched" (blur)

# Validation Messages

InStep Movie Hunter

Search by Title

+ Add Movie

The Lord of the Rings: The Two Towers

## Edit Movie Information

Movie Title

Description

While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.

Save

Cancel



# ngMessages: New in 1.3

## Requires

- angular-messages
- ngMessages

## ng-Messages

- Shows or hides messages
- Similar to a switch or case statement

## ng-Message

- Defines specific cases

```
<span class="help-block">
  <span ng-messages="movieForm.inputMovieTitle.$error">
    <span ng-message="required">movie title is required</span>
    <span ng-message="minlength">Must be at least 3 characters</span>
    <span ng-message="maxlength">Cannot exceed 50 characters</span>
  </span>
</span>
```

# DEMO: FORMS & VALIDATION

# Data Binding Options

- By default, synchronization occurs as the user types
- Use ng-model-options to change when the synchronization occurs
  - {updateOn: 'blur'}
  - {debounce: 1500}
  - {updateOn: 'submit'}

# DEMO: BINDING OPTIONS

# Custom Validator

- Developed as a directive
- `ngModel.$validators`
  - Collection of validators
  - Returns true/false
- Used as an attribute

```
directorMinRatingValidator.js

function directorMinRatingValidator() {
    . . .
}
```

```
movieEditView.html
<input id="inputStarRating"
       type="number"
       ng-model="vm.movie.starRating"
       required
       director-min-rating-validator="vm.movie.director"
       min="1"
       max="5"/>
```



# DEMO: CUSTOM VALIDATORS

# Asynchronous Validator

- Developed as a directive
- `ngModel.$asyncValidators`
  - Collection of validators
  - Returns promise
- Used as an attribute

```
duplicateTitleValidator.js

function duplicateTitleValidator () {
    . . .
}
```

```
movieEditView.html
<input id="inputMovieTitle"
      type="text"
      ng-model="vm.movie.title"
      required
      duplicate-title-validator="vm.movie.movieId"
      ng-minlength="3"
      ng-maxlength="50"/>
```



# DEMO: ASYNC VALIDATORS



---

# Angular 2 Forms

<https://angular.io/>

<https://github.com/DeborahK/AngularU2015-Angular2Forms>

---

# Angular 2 Form Techniques

Template  
Driven

Similar to 1.x

Imperative  
(Model)  
Driven

Built w/Code

Data Driven

Automatic

# Controller

TypeScript class

Bindable  
Controller  
Properties

Controller  
Methods

```
class checkoutCtrl {  
  model = new CheckoutModel();  
  countries = ['US', 'Canada'];  
  
  onSubmit() {  
    console.log("Submitting:");  
    console.log(this.model);  
  }  
}
```

# Model

TypeScript class

Strongly Typed  
Properties

```
class CheckoutModel {  
  firstName: string;  
  middleName: string;  
  lastName: string;  
  country: string = "Canada";  
  
  creditCard: string;  
  amount: number;  
  email: string;  
  comments: string;  
}
```

# View

Event Binding

Local Variable

Two-way  
BindingProperty  
Binding

```
<h1>Checkout Form</h1>

<form (ng-submit)="onSubmit()" #f="form">
  <p>
    <label for="firstName">First Name</label>
    <input type="text" id="firstName" ng-control="firstName"
      [(ng-model)]="model.firstName" required>
    <show-error control="firstName"
      [errors]="['required']"></show-error>
  </p>
  . . .
  <button type="submit" [disabled]="!f.form.valid">
    Submit
  </button>
</form>
```

# ng-repeat -> \*ng-for

Two-way  
Binding

\*ng-for

Local  
Variable

Property  
Binding

One-Way  
Binding

```
<p>
  <label for="country">Country</label>
  <select id="country" ng-control="country"
    [(ng-model)]="model.country">
    <option *ng-for="#c of countries" [value]="c">{{c}}
    </option>
  </select>
</p>
```

# Key Differences

## Template-Driven

- Controller exposes **data** model

```
class CheckoutCtrl {  
    model = new CheckoutModel();  
    countries = ['US', 'Canada'];  
    ...  
}
```

## Model-Driven

- Controller exposes **form** model

```
class CheckoutCtrl {  
    formModel;  
    countries = ['US', 'Canada'];  
    ...  
}
```

# Key Differences

## Template-Driven

- Controller exposes data model
- Binding & Validation in **View**

```
<input
  type="text"
  id="firstName"
  ng-control="firstName"
  [(ng-model)]="model.firstName"
  required>
```

## Model-Driven

- Controller exposes form model
- Binding & Validation in **Controller**

```
class CheckoutCtrl {
  formModel;
  countries = ['US', 'Canada'];

  constructor(fb: FormBuilder) {
    this.formModel = fb.group({
      "firstName": [ "", validators.required ],
      "lastName": [ "", validators.required ],
      . . .
    });
  }
  . . .
}
```



# Key Differences

## Template-Driven

- Controller exposes data model
- Binding & Validation in View
- View contains **data** bindings

```
<input
  type="text"
  id="firstName"
  ng-control="firstName"
  [(ng-model)]="model.firstName"
  required>
```

## Model-Driven

- Controller exposes form model
- Binding & Validation in Controller
- View contains **control** mappings

```
<input
  type="text"
  id="firstName"
  ng-control="firstName">
```

# Take Away



Use client-side validation where possible

Display friendly and helpful messages

Use directives to create custom & async validation

Keep an eye on Angular 2 progress

# Thank You!

@deborahkurata

deborahk@insteptech.com

<http://blogs.msmvps.com/deborahk>

<https://github.com/DeborahK/VSLive2015-NY>