

Copyright © 2014 by Software Craftsmanship Guild.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the Software Craftsmanship Guild. For permission requests, write to the Software Craftsmanship Guild, addressed “Attention: Permissions Coordinator,” at the address below.

Software Craftsmanship Guild

526 S. Main St, Suite 609

Akron, OH 44311



AngularJS

Directives, Forms, & Validation

Software Craftsmanship Guild

Lesson Goals

Introduce the basic concepts behind creating your own angularjs directives.

Learn about some of the directives that can assist with forms and validation.

Angular Directives

Angular makes it relatively easy to add our own directives for use in our applications. Directives can have their own scope, can build complex html templates, and can define events (click, mouse enter, mouse leave, etc.)

Writing your own directives is pretty advanced stuff, you'll need a strong working knowledge of HTML and JavaScript to build interesting things.

Example: Movie Ratings Directive

Let's say that we want to create a directive for our movie application that allows users to dynamically select a rating based on a maximum number of stars provided.

It should support choosing a rating by clicking a star, hovering over stars should animate nicely, and the label should be able to be changed via attributes.

The end result should be something like this:



The ng-model Directive

The key to form validation lies in attaching the ng-model directive to each form field you would like to validate.

ng-model provides two-way data-binding which synchronizes the model values to the view, so that when one changes, the other automatically updates.

Angular Validation

Some browsers have some validation built in, especially for HTML5 controls. When using a framework like angularjs we often want to disable those features (since they don't behave the same across browsers).

To do this, we put a *novalidate* attribute on the form.

```
<form name="reviewForm" ng-controller="ReviewController as reviewCtrl"  
      ng-submit="reviewCtrl.addReview(display.book)" novalidate>
```

Built-In Validation

AngularJS has the following validation attributes built-in for you to use.

Note that angular will also work with `type="email"` and other HTML5 input types automatically.

Class	Usage
required	Makes a field required
ng-minlength	Makes sure a form input is at least n characters
ng-maxlength	Makes sure a form input is less than n characters
ng-pattern	Ensures an input matches a regex pattern
min	Minimum numeric value
max	Maximum numeric value

CSS Classes for Validation

These classes are automatically added to form elements if you use the ng-model directive.

You can add these styles to your CSS file for nice effects.

Class	Usage
ng-pristine	Added when a form field value has not been changed by the user.
ng-dirty	Added when the user changes the value of a form field.
ng-valid	Added when the data in the field is valid
ng-invalid	Added when the data in the form does is not valid

Checking Form Fields Manually

Expression	Description
formName.inputFieldName.\$pristine	true if user has not field
formName.inputFieldName.\$dirty	true if user has modified field
formName.inputFieldName.\$valid	true if the form field is valid
formName.inputFieldName.\$invalid	true if the form field is invalid
formName.inputFieldName.\$error	Contains all of the validations on the form field and if they are valid or invalid

Why do we Care?

So we can show error messages conditionally if a field is dirty and invalid:

```
<fieldset class="form-group">
  <label class="control-label text-danger"
    ng-show="reviewForm.Body.$dirty && reviewForm.Body.$invalid">
    Your review must be at least 25 characters
  </label>
  <textarea ng-model="reviewCtrl.newReview.body" class="form-control" name="Body"
    placeholder="Write a short review of the product..." title="Review" ng-minlength="25"></textarea>
</fieldset>
```

Your review must be at least 25 characters

This is not long enough.|

