

AngularJS & Bootstrap

Form Validation



HTML default validation

- Browsers have built-in HTML validation
- These work when the type attribute is given to an input element
- We might want better validation than is provided by the web browser

AngularJS validation

- Angular comes with some built-in client-side validations that we might want to use with our directives.
- To use AngularJS validations, we must first turn off Default HTML Validation.

```
<form name="reviewForm" novalidate>
```

Onto AngularJS validation

- Disabling HTML validation with ***novalidate*** attribute has the effect of enabling AngularJS validation.
- Better validation of input type
E.g., validating email address

```
<input name="author" ng-model="reviewCtrl.review.author"  
      type="email" class="form-control"  
      placeholder="you@email.com" title="Email"  
      required/>
```

- We can mark this as a ***required*** field (HTML 5)

Testing that form is valid

- You can query form's validity

`<div>Is reviewForm valid: {{reviewForm.$valid}}</div>`

- Notice that form is given a name attribute
 - Suppose the value of the name attribute is reviewForm
- In an expression if we say **`reviewForm.$valid`**,
\$ allows us to reference a property on the form
- a built-in property from AngularJS

Only want to submit valid form

- Form is valid when all form fields are valid.
- If email address input field or any other form field is invalid, we should not be allowed to submit form.
- How do we accomplish this?

```
<form name="reviewForm"  
      ng-controller="ReviewController as reviewCtrl"  
      ng-submit=" reviewForm.$valid &&  
                reviewCtrl.addReview(product)"  
      novalidate >
```

Not submitting Invalid Form

- How might we give a hint to the user why their form is invalid?
 - By using **css classes** associated with properties that AngularJS provides on forms and their inputs
 - Angular provides properties on forms that help us validate them.
 - They give us various information about a form or its inputs and are applied to a form and inputs.

AngularJS Form Properties

Property	Class	Description
\$valid	ng-valid	<i>Boolean</i> Tells whether an item is currently valid based on the rules you placed.
\$invalid	ng-invalid	<i>Boolean</i> Tells whether an item is currently invalid based on the rules you placed.
\$pristine	ng-pristine	<i>Boolean</i> True if the form/input has not been used yet.
\$dirty	ng-dirty	<i>Boolean</i> True if the form/input has been used.
\$touched	ng-touched	<i>Boolean</i> True if the input has been blurred.

The input classes

```
<input name="author" ng-  
model="reviewCtrl.review.author"  
type="email" required>
```

```
<!-- Source before typing email: -->
```

```
<input name="author" ...class="ng-pristine ng-invalid">
```

```
<!-- Source after typing, with invalid email: -->
```

```
<input name="author" ... class="ng-dirty ng-invalid">
```

```
<!-- Source after typing, with valid email: -->
```

```
<input name="author" ... class="ng-dirty ng-valid">
```

Using provided classes

- So let's highlight the form field using classes after we start typing, ***ng-dirty***
 - Show if a field is valid or invalid. ***ng-valid*** or ***ng-invalid***
- Write some css to highlight form fields
- Why do we include ***ng-dirty***?
 - Because we want to begin highlighting when the user starts typing

css to highlight fields

```
.ng-invalid.ng-dirty {  
    /* Red border for invalid and dirty.*/  
    border-color: #FA787E;  
}
```

```
.ng-valid.ng-dirty {  
    /* Green border for invalid & dirty.*/  
    border-color: #78FA89;  
}
```

HTML5-based type validations

- Web forms usually have rules around valid input
- AngularJS has built-in validations for common input types

```
<input type="email" name="email">
```

```
<input type="url" name="homepage">
```

```
<input type="number" name="quantity"  
      min=1 max=10>
```

- Note: can also define min and max with numbers.

Resources

- <https://scotch.io/tutorials/angularjs-form-validation>
- <http://getbootstrap.com/css/#forms>
- <https://docs.angularjs.org/api/ng/directive/form>
- <https://www.codeschool.com/paths/javascript>