

specialist ICT learning



AngularJS: Forms

GuruTeam Instructor: Sang Shin

Topics

- Forms
- Two-way databinding between form and model
- Built-in CSS styles
- Form submission
- Form and element states
- Validation
- Show and hide
 - > ng-show, ng-hide
 - > ng-enabled, ng-disabled

Forms

What is a Form and Input Elements?

- A Form is a collection of input elements (input, select, textarea) for the purpose of grouping them together
 - > Input elements are ways for a user to enter data
- Form and input elements provide **validation services**, so that the user can be notified of invalid input before submitting a form
 - > Provides a better user experience than server-side validation alone because the user gets instant feedback on how to correct the error
 - > The value of *ngModel* won't be set unless it passes validation for the input fields
- Typically you want to disable browser's native validation through “novalidate” attribute

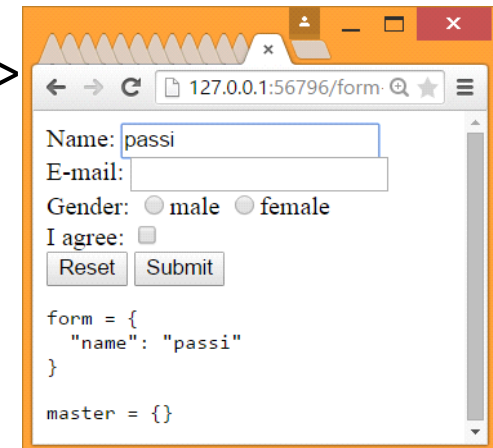
Two-way Databinding: Form and Model

Two-way Databinding: Form and ngModel

- Two-way data-binding works between a form (view) and models – whatever typed in input element is immediately reflected in the model and vice versa

```
<div ng-controller="ExampleController">
  <form novalidate class="simple-form">
    Name: <input type="text" ng-model="user.name" /><br />
    E-mail: <input type="email" ng-model="user.email" /><br />
    Gender: <input type="radio" ng-model="user.gender" value="male" />male
    <input type="radio" ng-model="user.gender" value="female" />female<br />
    <input type="button" ng-click="reset()" value="Reset" />
    <input type="submit" ng-click="update(user)" value="Save" />
  </form>
  <pre>form = {{user | json}}</pre>
```

```
</div>
```



Built-in CSS Styles

Built-in CSS classes with a Form

- To allow styling of form as well as input elements, *ngModel* adds these CSS classes automatically
 - > ng-valid: the model is valid
 - > ng-invalid: the model is invalid
 - > ng-pristine: the control hasn't been interacted with yet
 - > ng-dirty: the control has been interacted with
 - > ng-touched: the control has been blurred
 - > ng-untouched: the control hasn't been blurred
- It is still your job to provide styling rules for the above CSS classes, however

Form and CSS Styling

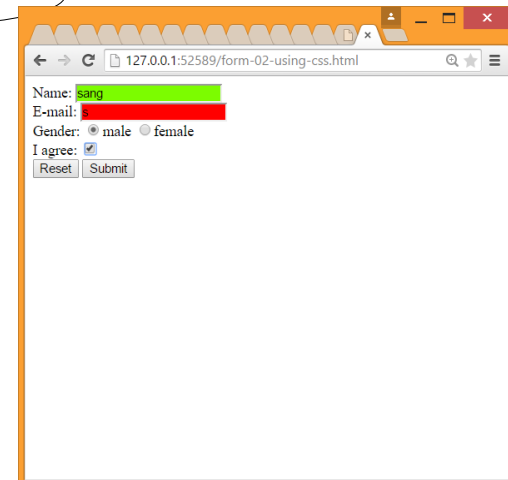
- Uses CSS to display validity of each form control

```
<div ng-controller="ExampleController">
  <form novalidate class="css-form">
    Name: <input type="text" ng-model="user.name" required /><br />
    E-mail: <input type="email" ng-model="user.email" required /><br />
    ...
  </form>
</div>
```

If input element is touched,
and invalid, use red background

```
<style type="text/css">
  .css-form input.ng-invalid.ng-touched {
    background-color: red;
  }
  .css-form input.ng-valid.ng-touched {
    background-color: green;
  }
</style>
```

styling rule



Form Submission

Form Submission using *ng-submit*

- Enables binding angular expressions to “onsubmit” events
- It prevents the default action (which for form means sending the request to the server and reloading the current page), but only if the form does not contain *action* attribute

```
<form ng-submit="submitMethod()">
```

Enter text and hit enter:

```
<input type="text" ng-model="text" name="text" />
```

```
<input type="submit" id="submit" value="Submit" />
```

```
<pre>list={{list}}</pre>
```

```
</form>
```

Form Submission using ng-submit

```
<form ng-submit="submit()" >
  Enter text and hit enter:<input type="text" ng-model="text" name="text" />
  <input type="submit" id="submit" value="Submit" />
  <pre>list={{list}}</pre>
</form>
<script>
angular.module('submitExample', [])
  .controller('ExampleController', ['$scope', function ($scope) {
    $scope.list = [];
    $scope.text = 'hello';
    $scope.submit = function () {
      if ($scope.text) {
        $scope.list.push(this.text);
        $scope.text = "";
      }
    };
  }]);
</script>
```

Form and Element State

Binding to form and Control state

- Internal states of both form and input elements are available for binding in the view using the standard binding
 - > Custom error message displayed after the user interacted with an input element (i.e. when *\$touched* is set)
 - > Custom error message displayed upon submitting the form (*\$submitted* is set), even if the user didn't interact with a control

Binding to Form and Input Element State

```
<form name="form" class="css-form" novalidate>
```

Name:

```
<input type="text" ng-model="user.name" name="uName" required="" />
```

```
<br />
```

```
<div ng-show="form.$submitted || form.uName.$touched">
```

```
<div ng-show="form.uName.$error.required">Tell us your name.</div>
```

```
</div>
```

E-mail:

```
<input type="email" ng-model="user.email" name="uEmail" required="" />
```

```
<br />
```

```
<div ng-show="form.$submitted || form.uEmail.$touched">
```

```
<span ng-show="form.uEmail.$error.required">Tell us your email.</span>
```

```
<span ng-show="form.uEmail.$error.email">This is not a valid email.</span>
```

```
</div>
```

```
</form>
```

Lab:

Exercise 1: Forms

3307_angularjs_07_forms.zip



Validation

Form Validation

- Angular provides basic implementation for most common HTML5 input types
 - > text, number, url, email, date, radio, checkbox
- Angular also provides some directives for validation
 - > required, pattern, minlength, maxlength, min, max
- You want to turn off default browser validation with “novalidate”

```
<form name="form" ng-submit="submit()" novalidate>
<label>Firstname</label>
<input name="firstname" type="text" ng-model="user.firstname" required minlength=5/>
<label>Age</label>
<input type="number" ng-model="user.age" min=3 max=100 />
<br>Email
<input type="email" ng-model="user.email" />
<br>
<button ng-disabled="form.$invalid" class="btn">Submit</button>
</form>
```

Form Field State

- Angular keeps track of all its input elements (controls) and nested forms as well as the state of them, such as being valid/invalid or dirty/pristine
 - > \$pristine: User has not interacted with the field
 - > \$dirty: User has interacted with the field
 - > \$valid: The field content is valid
 - > \$invalid: The field content is invalid

```
<p>Email:<br>
  <input type="email" name="email" ng-model="email" required>
  <span style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">
  <span ng-show="myForm.email.$error.required">Email is required.</span>
  <span ng-show="myForm.email.$error.email">Invalid email address.</span>
  </span>
</p>
```

Disable Submit Button until form is valid

- *form.\$valid* or *form.\$invalid* indicates validity of the form values

```
<form name="form" ng-submit="submit()" novalidate>
  <label>Firstname</label>
  <input name="firstname" type="text" ng-model="user.firstname" required minlength=5/>
  <label>Age</label>
  <input type="number" ng-model="user.age" min=3 max=100 />
  <br>Email
  <input type="email" ng-model="user.email" />
  <br>
  <button ng-disabled="form.$invalid" class="btn">Submit</button>
</form>
```

Disable submit button
until the form is valid

Lab:

Exercise 2: Form Validation
3307_angularjs_07_forms.zip



Show and Hide

ngShow and ngHide

- ngShow directive shows or hides the given HTML element based on the expression provided to the ngShow attribute

```
<!-- when $scope.myValue is truthy (element is visible) -->  
<div ng-show="myValue"></div>
```

- ngHide directive shows or hides the given HTML element based on the expression provided to the ngHide attribute

```
<!-- when $scope.myValue is falsy (element is visible) -->  
<div ng-hide="myValue"></div>
```

Lab:

Exercise 3: Show and Hide
3307_angularjs_07_forms.zip



Who are GuruTeam?

- Specialist onsite training in Linux, Cloud, Database, Architecture, Software and Web Development Technologies
- Accredited by the LPI, CompTIA, Hortonworks and the Cloud Credential Council to deliver training, examinations and certifications.
- Over 230 courses available
- All GuruTeam instructors have extensive real-world experience in their technologies
- Clients are indigenous Irish Companies and Multinationals
- We can bring high spec preconfigured equipment for deliveries in Ireland, the UK and Europe.



Exceed your expectations...



GURUTEAM 