AngularJS ng-model Directive

[« Previous](http://www.w3schools.com/angular/angular_directives.asp)

[Next Chapter »](http://www.w3schools.com/angular/angular_controllers.asp)

The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.

The ng-model Directive

With the ng-model directive you can bind the value of an input field to a variable created in AngularJS.

Example

<div ng-app="myApp" ng-controller="myCtrl">  
    Name: <input ng-model="name">  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
    $scope.name = "John Doe";  
});  
</script>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_model)

Two-Way Binding

The binding goes both ways. If the user changes the value inside the input field, the AngularJS property will also change it's value:

Example

<div ng-app="myApp" ng-controller="myCtrl">  
    Name: <input ng-model="name">  
    <h1>You entered: {{name}}</h1>  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_model_two-way)

Validate User Input

The ng-model directive can provide type validation for application data (number, e-mail, required):

Example

<form ng-app="" name="myForm">  
    Email:  
    <input type="email" name="myAddress" ng-model="text">  
    <span ng-show="myForm.myAddress.$error.email">Not a valid e-mail address</span>  
</form>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_model_validate)

In the example above, the span will be displayed only if the expression in the ng-show attribute returns true.

|  |  |
| --- | --- |
| **Note** | If the property in the ng-model attribute does not exist, AngularJS will create one for you. |

Application Status

The ng-model directive can provide status for application data (invalid, dirty, touched, error):

Example

<form ng-app="" name="myForm" ng-init="myText = 'post@myweb.com'">  
    Email:  
    <input type="email" name="myAddress" ng-model="myText" required></p>  
    <h1>Status</h1>  
    {{myForm.myAddress.$valid}}  
    {{myForm.myAddress.$dirty}}  
    {{myForm.myAddress.$touched}}  
</form>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_model_status)

CSS Classes

The ng-model directive provides CSS classes for HTML elements, depending on their status:

Example

<style>

input.ng-invalid {  
    background-color: lightblue;  
}

</style>  
<body>  
  
<form ng-app="" name="myForm">  
    Enter your name:  
    <input name="myAddress" ng-model="text" required>  
</form>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_model_css)

The ng-model directive adds/removes the following classes, according to the status of the form field:

* ng-empty
* ng-not-empty
* ng-touched
* ng-untouched
* ng-valid
* ng-invalid
* ng-dirty
* ng-pending
* ng-pristine