

input[date] DIRECTIVE IN ANGULARJS

- **input [date]** is one of the AngularJS input directive in **module ng**.
- AngularJS directive **input [date]** is used to create an HTML input with date validation and transformation.
- The input text must be entered in the format ISO-8601 (i.e. yyyy-MM-dd)
For ex: **2014-15-08**
- In browsers does not support the HTML5 time input, as text element will be used.
- It is important to provide hints to users on the normal input format through the placeholder or label.
- The data model must be date object.
- The timezones is used to read/write the **Date** instance in the model by using ngModelOptions.
- This directive executes at priority level 0.



Syntax for input [date] directive in AngularJS:

```
<input type="date"  
  ng-model="string"  
  [name="string"]  
  [min="string"]  
  [max="string"]  
  [ng-min=" " ]  
  [ng-max=" " ]  
  [required ="string"]  
  [ng-required ="string"]  
  [ng-change ="string"] >
```



Parameter values:

Parameters	Type	Description
ngModel	string	Defines angular expression to data-bind to.
name (optional)	string	Name of the form under which the control is available
min(optional)	string	To set the min validation error key if the value entered is shorter than min. (e.g “{{ minDate date :’yyyy-MM-dd’ }}”)
max(optional)	string	To set the max validation error key if the value entered is greater than max. (e.g “{{ maxDate date : ’yyyy-MM-dd’ }}”)
ngMin (optional)	date, string	To set the min validation constraint to the Date/ISO week string the ngMin expression calculates to. Reminds that it does not set the min attribute.
ngMax (optional)	date, string	To set the max validation constraint to the Date/ISO week string the ngMax expression calculates to. Reminds that it does not set the max attribute.
required (optional)	string	Denotes the required validation error key if the value is not entered.
ngRequired (optional)	string	Sets the required attribute and required validation constraint to the element when the ngRequired expression sets to true.



		Instead of required use ngRequired when we want data-bind to the required attribute.
ngChange (optional)	string	An expression of Angular to be executed when input changes due to user interaction with the input element.



Sample coding for input [date] directive in AngularJS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy AngularJS Tutorial</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js">
    </script>
  </head>
  <body>
    <form ng-app="myApp" name="Form" ng-controller="DateCtrl">
      <h3>input[date] directive example in AngularJS</h3>
      Pick a date in 2016:
      <input type="date" name="input" ng-model="value"
        placeholder="yyyy-MM-dd" min="2016-02-01" max="2016-12-31"
required/>
      <span ng-show="Form.input.$error.required">Required!</span>
      <span ng-show="Form.input.$error.date">Not a valid date!</span>
      <p>Date = {{value | date: "yyyy-MM-dd"}}</p>
      <p>Form.input.$valid = {{Form.input.$valid}}</p>
      <p>Form.input.$error = {{Form.input.$error}}</p>
      <p>Form.$valid = {{Form.$valid}}</p>
      <p>Form.$error.required = {{!!Form.$error.required}}</p>
    </form>
    <script>
      var app = angular.module('myApp', []);
      app.controller('DateCtrl', function($scope) {
        });
    </script>
  </body>
</html>
```

HTML:

```
<form ng-app="myApp" name="Form" ng-controller="DateCtrl">
  <h3>input[date] directive example in AngularJS</h3>
  Pick a date in 2016:
  <input type="date" name="input" ng-model="value"
    placeholder="yyyy-MM-dd" min="2016-02-01" max="2016-12-31"
required/>
  <span ng-show="Form.input.$error.required">Required!</span>
  <span ng-show="Form.input.$error.date">Not a valid date!</span>
  <p>Date = {{value | date: "yyyy-MM-dd"}}</p>
  <p>Form.input.$valid = {{Form.input.$valid}}</p>
  <p>Form.input.$error = {{Form.input.$error}}</p>
  <p>Form.$valid = {{Form.$valid}}</p>
  <p>Form.$error.required = {{!!Form.$error.required}}</p>
</form>
```

- Viewable HTML contents in AngularJS application

Logic:

```
var app = angular.module('myApp', []);
app.controller('DateCtrl', function($scope) {
});
```

- Controller logic for the AngularJS application.



Code Explanation input [date] directive in AngularJS:

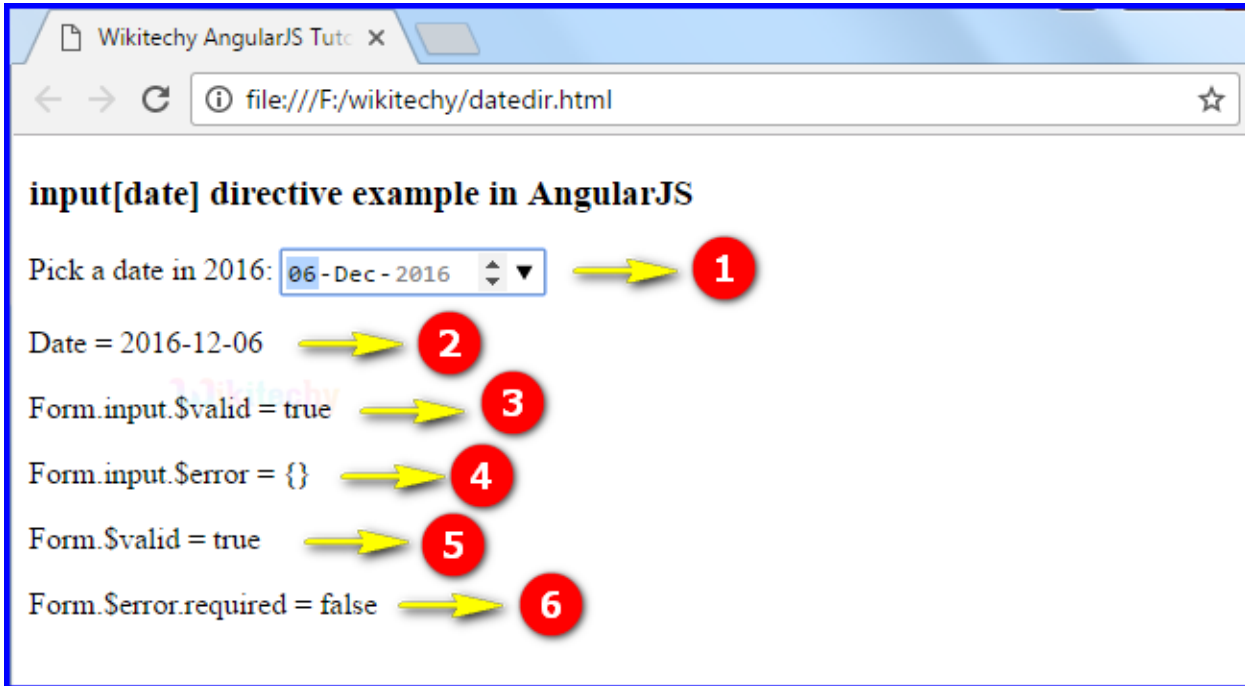
```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy AngularJS Tutorial</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/
      1.5.6/angular.min.js"></script>
  </head>
  <body>
    <form ng-app="myApp" name="Form" ng-controller="DateCtrl">
      <h3>input[date] directive example in AngularJS</h3>
      Pick a date in 2016:
      <input type="date" name="input" ng-model="value"
        placeholder="yyyy-MM-dd" min="2016-02-01" max="2016-12-31"
        required/>
      <span ng-show="Form.input.$error.required">Required!</span>
      <span ng-show="Form.input.$error.date">Not a valid date!</span>
      <p>Date = {{value | date: "yyyy-MM-dd"}}</p>
      <p>Form.input.$valid = {{Form.input.$valid}}</p>
      <p>Form.input.$error = {{Form.input.$error}}</p>
      <p>Form.$valid = {{Form.$valid}}</p>
      <p>Form.$error.required = {{!!Form.$error.required}}</p>
    </form>
    <script>
      var app = angular.module('myApp', []);
      app.controller('DateCtrl', function($scope) {
      });
    </script>
  </body>
</html>
```

1. The **ng-app** specifies the root element ("**myApp**") to define AngularJS application.
2. ng-controller specifies the application controller in AngularJS the controller value is given as "**DateCtrl**".
3. "**date**" is declare the **type** value of the <input> tag.



4. The **ng-model** bind an input field value to AngularJS application variable (“value”).
5. **Placeholder** is used to declare the date format (Like “YYYY-MM-dd”).
6. **min** parameter is used to declare the start date value(2016-02-01)
7. **max** parameter is used to declare the end date value(2016-12-31)
8. ng-show directive is used to hides HTML elements. If the required directive shows an error the content is displayed like “Required”
9. ng-show directive is used to hides HTML elements. If the required directive shows an error the content is displayed like “Not a Valid date”
10. Here the date filter formats a date into specified format (YYYY-MM-dd) and the output will be updated in the <p> tag.
11. **Form.input.\$valid** to checks the correct date format or not. If the user scroll date in the input field then the output will be displays as true otherwise false.
12. **Form.input.\$error** to check whether the valid date format or not .If the date specified in error it throw an exception (Like “required”=“true) otherwise it is an empty curly braces ({}).
13. **Form.\$valid** is used to check whether the form is valid or not and output will be displayed in <p> tag.
14. **Form.\$error.required** is used to check whether date is required or not. If the date is required and output will be displays as false otherwise true.

Sample Output input [date] directive in AngularJS:



input[date] directive example in AngularJS

Pick a date in 2016: → 1

Date = 2016-12-06 → 2

Form.input.\$valid = true → 3

Form.input.\$error = {} → 4

Form.\$valid = true → 5

Form.\$error.required = false → 6

1. If the user scroll the date in the input field.
2. The output displays as **Date=2016-12-06**.
3. The output displays true because it is consider as a valid date.
4. The output displays empty curly braces it means does not thrown any error.
5. The output displays true it is valid date.
6. If the text box does not empty so the output displays as false.



input[date] directive example in AngularJS

Pick a date in 2016: Required!

Date =

```
Form.input.$valid = false
Form.input.$error = {"required":true}
Form.$valid = false
Form.$error.required = true
```

1. If the user does not scroll any value in the input field and the output shows an error **“Required”**.

