



**FREE eBook**

**LEARNING**

**angularjs-directive**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#angularjs-  
directive**

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with angularjs-directive.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Building a reusable component.....	2
Your first directive.....	3
Success/Error pop-up message using simple link function.....	4
<b>Chapter 2: Commonly Used Directives.....</b>	<b>6</b>
Examples.....	6
ngConfirmClick: Confirm before evaluating expression.....	6
<b>Credits.....</b>	<b>7</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [angularjs-directive](#)

It is an unofficial and free angularjs-directive ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official angularjs-directive.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with angularjs-directive

## Remarks

AngularJS Directives are custom elements in HTML (such as an attribute, element name, comment or CSS class) that tell AngularJS to attach a specified behavior to that DOM element, or even to transform the DOM element and its children. In short, when we create a directive, AngularJS will treat that element differently.

## Examples

### Installation or Setup

Directives comes with the AngularJS library itself. A sample directive can be created as:

```
angular.module('simpleDirective', [])
.directive('helloData', function() {
  return {
    template: 'Hello, {{data}}'
  };
});
```

And can be used as:

#### JS:

```
angular.module('app', ['simpleDirective'])
.controller('Controller', ['$scope', function($scope) {
  $scope.data = 'World';
}])
```

#### HTML

```
<div ng-controller="Controller">
  <div hello-data></div>
</div>
```

Will be compiled as:

Hello, World

### Building a reusable component

Directives can be used to build reusable components. Here is an example of a "user box" component:

## userBox.js

```
angular.module('simpleDirective', []).directive('userBox', function() {
  return {
    scope: {
      username: '=username',
      reputation: '=reputation'
    },
    templateUrl: '/path/to/app/directives/user-box.html'
  };
});
```

## Controller.js

```
var myApp = angular.module('myApp', ['simpleDirective']);

myApp.controller('Controller', function($scope) {
  $scope.user = "John Doe";
  $scope.rep = 1250;
});
```

## myPage.js

```
<html lang="en" ng-app="myApp">
  <head>
    <script src="/path/to/app/angular.min.js"></script>
    <script src="/path/to/app/controllers/Controller.js"></script>
    <script src="/path/to/app/directives/userBox.js"></script>
  </head>

  <body>

    <div ng-controller="Controller">
      <user-box username="user" reputation="rep"></user-box>
    </div>

  </body>
</html>
```

## user-box.html

```
<div>{{username}}</div>
<div>{{reputation}} reputation</div>
```

## Your first directive

Our first element directive will not do much: it will just calculate  $2+2$  and will be called in html like this:

```
<my-calculator></my-calculator>
```

Notice the name of the directive is `myCalculator` (in CamelCase), but in html it's used as `my-calculator` (in lisp-case).

Since we want our directive to be used as html element, we will use `restrict: 'E'`.

Every directive has the template which will be compiled and inserted. Our directive is very simple, so we will insert our html as string into a `template` parameter.

```
// directives/my-calculator.js

angular.module('exampleApp', [])
.directive('myCalculator', function() {
  return {
    restrict: 'E',
    template: '<span> My directive can calculate 2+2: {{2+2}} </span>'
  };
});
```

## HTML

```
<!DOCTYPE html>
<html ng-app="exampleApp">

  <head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
    <script src="my-calculator.js"></script>
  </head>

  <body>
    Here is my first directive:
    <my-calculator></my-calculator>
  </body>

</html>
```

The result will look like this:

Here is my first directive: My directive can calculate 2+2: 4

If you want to play with the live example, go to [plunkr](#).

## Success/Error pop-up message using simple link function

Link function is best way in custom directives to manipulate DOM. It takes three attributes as input (scope, element, attribute) in sequence

scope: its local scope object of directive.

element: html element on which directive is used.

attribute: it gives access to all attributes used in element referred.

```
// on success call or similarly error, warning, info in controller
$scope.message={
  text: "Saved Successfully",
  type: "SUCCESS"
```

```

    };

    <user-info msg="message"> </user-info>    //in html

var mainApp = angular.module("mainApp", []);
mainApp.directive('userInfo', function() {
    var directive = {};
    directive.restrict = 'E';

    directive.scope = {
        message : "=msg"
    },

    directive.link = function(scope, element, attributes) {
        if(scope.message.type==='SUCCESS')
            scope.message.text = 'SUCCESS: '+scope.message.text+' !';
        else if(scope.message.type==='ERROR')
            scope.message.text = 'ERROR: '+scope.message.text+' !';
        else if(scope.message.type==='WARNING')
            scope.message.text = 'WARNING: '+scope.message.text+' !'
        else if(scope.message.type==='INFO')
            scope.message.text = 'INFO: '+scope.message.text+' !'

        element.on('click', function(event) {    //on click of div pop-up will smoothly
close
                $(this).fadeOut();
            });
        },
        directive.template = '<div ng-class={{message.type}}>'+    // one can
create different bg-color as per type of message and width/height
            '<div class="message-text">{{message.text}}</div>'+    //message
text will be printed
            '<div>';

        return directive;
    });

```

Read [Getting started with angularjs-directive online](https://riptutorial.com/angularjs-directive/topic/1855/getting-started-with-angularjs-directive): <https://riptutorial.com/angularjs-directive/topic/1855/getting-started-with-angularjs-directive>

---

# Chapter 2: Commonly Used Directives

## Examples

**ngConfirmClick: Confirm before evaluating expression.**

Description:

Evaluate expression after user's confirmation.

Arguments:

- **ng-confirm-click:(*expression*)** Expression to evaluate when confirmed.
- **ng-confirm-message:(*template*)** Message to be shown in confirm dialog.

Code:

```
Directives.directive("ngConfirmClick", ["$parse", "$interpolate", function ($parse, $interpolate)
{
    return {
        restrict:"A",
        priority:-1,
        compile:function(ele, attr){
            var fn = $parse(attr.ngConfirmClick, null, true);
            return function ngEventHandler(scope, ele) {
                ele.on('click', function (event) {
                    var callback = function () {
                        fn(scope, {$event: "confirm"});
                    };
                    var message = $interpolate(attr.ngConfirmMessage)(scope) || 'Are you
sure?';

                    if(confirm(message)) {
                        if (scope.$root.$$phase) {
                            scope.$evalAsync(callback);
                        } else {
                            scope.$apply(callback);
                        }
                    }
                });
            };
        }
    };
});
```

[Working Example](#)

Read Commonly Used Directives online: <https://riptutorial.com/angularjs-directive/topic/5099/commonly-used-directives>



---

# Credits

S. No	Chapters	Contributors
1	Getting started with angularjs-directive	<a href="#">Asim K T</a> , <a href="#">Charlie H</a> , <a href="#">Community</a> , <a href="#">Daniel</a> , <a href="#">ganqqwerty</a> , <a href="#">Tjaart van der Walt</a> , <a href="#">Vishal Singh</a>
2	Commonly Used Directives	<a href="#">MMhunter</a>