**AngularJS Documentation**

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

It can be included in <script> tag in html.

<script src="<http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js>"></script>

**Merits**

* **Unit testing** ready: JavaScript is, usually, very hard to unit test when you have DOM manipulations and business logic together (e.g. jQuery based code). AngularJS keeps DOM manipulation in the HTML and business logic separated. Data and dependencies are $injected as needed.
* **DOM manipulation** where they are used. It decouples DOM manipulation from application logic.
* AngularJS is also excellent for **single-page applications (SPA)**.
* Different **browsers implements** features differently, but fret not. Angular’s directive (or HTML extensions) take care of the differences for you.
* **Global namespace** expressions and method definitions are scoped within controllers, so they do not pollute the global namespace.
* **Data models** are plain old JavaScript objects (POJO).
* Write less code: AngularJS features like directives, filters and automatic data bindings save code writing.AngularJS provides solution for writing modular code and dependencies management.

**Directives**

**Directives** are extensions of HTML markups in form of attributes, element names, CSS class and or even HTML comments. When the AngularJS framework is loaded everything inside ng-app ,it’s compiled by Angular and the directives are bound to data, events and DOM transformations.

**ng-app**

It is a directive which defines an application. It is injected along with the <html> tag.

<html ng-app = “ ”>

It tells AngularJS that the <html> element is the "owner" of an AngularJS **application.**

**Controller**

They are the code that controls certain sections containing the DOM elements in which they are declared. they encapsulate the behaviour , callbacks and $scope models with views.

**ng-model**

It is a directive which binds form elements such as input,select,checkboxes,textarea to a property called **$scope.**

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

The above ng-model can be accessed in the controller by $scope.name

**ng-bind**

It is a directive which binds the **innerHTML** of the <p> element to the application variable **name**.

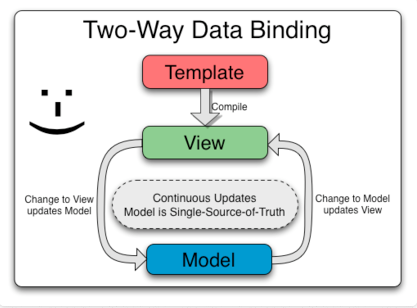
<p ng-bind="name"></p>

**{{ }}**

It is a way of binding models to elements in a HTML. The above mentioned ng-model can be accessed as

<input type="text" ng-model="name"> Hello {{name}}

**AngularJS Data-Binding**

It is an angularjs feature that automatically synchronizes model data with the HTML. Whenever the model data is updated, it gets updated in the HTML also and vice-versa.

**AngularJS Scope**

$scope is an object that contains all data to which HTML is bound. It is the connection between Javascript code(Controllers) and view(HTML).

**AngularJS Models**

Modules are way to encapsulate different parts of application and reuse them to rewrite controller using modules. It can be done in the below mentioned way:

<html ng-app="**app**">

<div ng-controller=”myCtrl”>

First Name: <input type="text" ng-model="firstName"><br>

</div>

</html>

In the controller js file, add the following code

var app = angular.module('**myApp**', []);

app.controller('**myCtrl**', function($scope) {

$scope.name= "John";});

If the controller is added in another JS file, it needs to be included in the script tag.

<script src="myCtrl.js"></script>

**ng-controller**

It is a directive that tells angular what function controller to use for a particular view. Every time AngularJS loads, it reads the ng-controller argument (in this case “**myCtrl**”). Then, it will look for a function in plain old javascript object (POJO) with the same name or for angular.controller matching name.

<div ng-controller="**myCtrl**">

First Name: <input type="text" ng-model="firstName"><br>

</div>

In the controller js file, add the following code

var app = angular.module('**myApp**', []);

app.controller('**myCtrl**', function($scope) {

$scope.name= "John";});

**ng-init**

It is a directive that initializes the application data.

**Code Snippet**

<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>

<p>You wrote: {{ firstName }}</p>

</div>

This will display

You wrote John

**ng-repeat**

It is a directive which repeats the HTML elements.

<div ng-app="app" ng-controller="myCtrl">

<ul>

<li ng-repeat="x in names">

{{ x }}

</li>

</ul>

</div>

In the controller

var app = angular.module('**myApp**', []);

app.controller('**myCtrl**', function($scope) {

$scope.name= ['Jani','Hege','Kai'];

});

**Result**

Jani

Hege

Kai

## **AngularJS Filters**

Filters allow you to format and transform the output of expressions inside the curly braces. AngularJS comes with a bunch of useful filters.

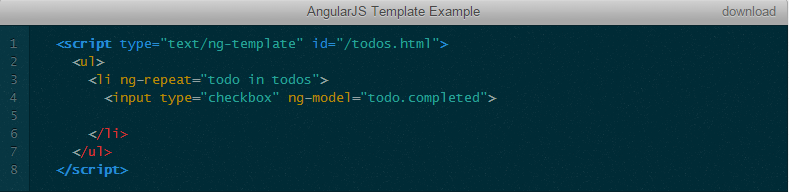
**Built-in Filters**:

* *filter*: search for a given string in an array and return matches.
* *Number*: adds comma-separated 1000’s and two decimal places.
* *Currency*: the same as *Number* and adds a $ in front.
* *Date*: takes a Unix timestamp (e.g. 1288323623006) or date string and output it in the format that you specify (e.g. ‘longDate’ or fragments ‘yyyy’ for four-digit year).
* *JSON*: converts javascript objects to JSON strings.
* *lowercase*/*uppercase*: converts strings to lowercase/uppercase.
* *limitTo*: number of elements from an array to show.
* *orderBy*: order array of objects by key that you specify.

## 

## **AngularJS Templates**

Templates contain HTML and Angular elements (directives, markup, filters or form controls). They can be cached and referenced by an id.

Here’s an example:

## **AngularJS Routes (ngRoute)**

ngRoute module allows changing what we see in the app depending on the URL (route). It, usually, uses templates to inject the HTML into the app.

To access this, it should be installed as a bower component by

**bower install angular-route**

This will download all the files in the bower\_components folder. The min file needs to be included in the <script> tag or it can be downloaded from google cdn.

**Code Snippet For HTML**

<html ng-app="app">

<head>

<title>Hello Controllers in AngularJS</title>

</head>

<body>

<ng-view></ng-view>

<!-- Template -->

<script type="text/ng-template" id="/todos.html">

<ul>

<li ng-repeat="todo in todos">

<input type="checkbox" ng-model="todo.completed">

{{todo.name}}

</li>

</ul>

</script>

</body>

</html>

**Code Snippet For JS**

angular.module('app', ['ngRoute'])

.controller('TodoController', ['$scope', function ($scope) {

$scope.todos = [

{ name: 'AngularJS Directives', completed: true },

{ name: 'Data binding', completed: true },

{ name: '$scope', completed: true },

{ name: 'Controllers and Modules', completed: true },

{ name: 'Templates and routes', completed: true },

{ name: 'Filters and Services', completed: false },

{ name: 'Get started with Node/ExpressJS', completed: false },

{ name: 'Setup MongoDB database', completed: false },

{ name: 'Be awesome!', completed: false },

];

}])

.config(['$routeProvider', function ($routeProvider) {

$routeProvider

.when('/', {

templateUrl: '/todos.html',

controller: 'TodoController'

});

}]);

Now every time on load, it will get redirected to todos.html

**ngView**

It is a directive used by $routeProvider to render HTML into it.Every time the URL changes, it will inject a new HTML template and controller into ngView.

**$http**

**$http** is an AngularJS service for reading data from remote servers.

$http.get(url) is the function to use for reading server data.It reads the data in JSON format.

In the JS file, the following code needs to be added:

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

app.controller('myCtrl', function($scope, $http) {

$http.get("http://www.w3schools.com/")

.success(function(response) {

});

});

## **Cross-Site HTTP Requests**

Requests for data from a different server (than the requesting page), are called **cross-site** HTTP requests.

Cross-site requests are common on the web. Many pages load CSS, images, and scripts from different servers.

In modern browsers, cross-site HTTP requests **from scripts** are restricted to **same site** for security reasons.

The following line, in our PHP examples, has been added to allow cross-site access.

header("Access-Control-Allow-Origin: \*");

**ng-disabled**

The **ng-disabled** directive binds AngularJS application data to the disabled attribute of HTML elements.

If the value of the attribute is true, then it disables the element/button/checkbox/textbox.

**ng-show/ng-hide**

This directive shows or hides the elements based on its value(true/false).

**ng-click**

This directive determines the action to be performed on the click of the element.

**AngularJS Tables**

To display data in the table, we use the ng-repeat directive.

**Code Snippet**

<div ng-app="myApp" ng-controller="myCtrl">

<table>

<tr ng-repeat="x in names">

<td>{{ x.Name }}</td>

<td>{{ x.Country }}</td>

</tr>

</table>

</div>

<script>

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

app.controller(myCtrl, function($scope, $http) {

var url = “”; // need to specify the url which will fetch data

$http.get(url)

.success(function (response) {$scope.names = response.records;});

});

</script>

For sorting the table, we use an **orderBy** filter.

**Code Snippet**

<table>

<tr ng-repeat="x in names | orderBy : 'Country'">

<td>{{ x.Name }}</td>

<td>{{ x.Country }}</td>

</tr>

</table>

**To display the table index, we use $index parameter**

**AngularJS Forms**

<form novalidate>

</form>

The **novalidate** attribute is new in HTML5. It disables any default browser validation.

**Input Validation**

Angularjs forms and controls can provide validation services and notify users of invalid input.

<form ng-app="myApp" ng-controller="validateCtrl"

name="myForm" novalidate>

<p>Username:<br>

<input type="text" name="user" ng-model="user" required>

<span style="color:red" ng-show="myForm.user.$dirty && myForm.user.$invalid">

<span ng-show="myForm.user.$error.required">Username is required.</span>

</span>

</p>

<p>

<input type="submit"

ng-disabled="myForm.user.$dirty && myForm.user.$invalid">

</p>

</form>

The above mentioned parameters have the following functionality:

$dirty - The user hasn’t interacted with the field

$valid - The field content is valid

$invalid - The field content is invalid

$pristine - User has not interacted with the field yet.

**AngularJS Global Functions**

angular.lowercase - converts a string to lowercase

angular.uppercase - converts a string to uppercase

angular.isString - returns True if the reference is String

angular.isNumber - returns True if the reference is Number

**Code Snippet**

<div ng-app="myApp" ng-controller="myCtrl">

<p>{{ x1 }}</p>

<p>{{ x2 }}</p>

<p>{{ x3 }}</p>

<p>{{ x4 }}</p>

<p>{{ x5 }}</p>

</div>

<script>

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

app.controller('myCtrl', function($scope) {

$scope.x1 = "JOHN";

$scope.x2 = angular.lowercase($scope.x1); // results john

$scope.x3 = angular.uppercase($scope.x1); // results JOHN

$scope.x4 = angular.isString($scope.x1); // results true

$scope.x5 = angular.isNumber($scope.x1); // results false

});

</script>

**Angular Bootstrap**

This is a directive which provides HTML,CSS and Javascript framework. To install the angularjs bootstrap, run the following command in the command prompt by traversing to the project folder.

**bower install angular-bootstrap**

To access the module, inject the following module in the controller and add the min.js file and css file in the script tag of html.

**angular.module('myModule', ['ui.bootstrap']);**

**Angular Accordion**

The accordion directive builds on top of the collapse directive to provide a list of items, with collapsible bodies that are collapsed or expanded by clicking on the item's header.

For further reference, refer to [Code Snippet](http://fiddle.jshell.net/7vRV9/).

**Angular Alert**

This directive can be used to both to generate alerts from static and dynamic model data by using ng-repeat directive.

For further reference, refer to [Code Snippet](http://jsfiddle.net/internoma/YCY9e/).

**Angular Carousel**

It helps to automatically cycle through the slides at a given rate.

For further reference, refer to [Code Snippet](http://jsfiddle.net/mylastof/srzncta1/).

**Angular Collapse**

This directive helps to hide and show an element with a css effect.

Add the following code snippet in html:

<div ng-controller="CollapseDemoCtrl">  
 <button type="button" class="btn btn-default" ng-click="isCollapsed = !isCollapsed">Toggle collapse</button>  
 <hr>  
 <div collapse="isCollapsed">  
 <div class="well well-lg">Some content</div>  
 </div>  
</div>

Add the following piece of code in controller(js file)

angular.module('ui.bootstrap.demo').controller('CollapseDemoCtrl', **function** ($scope) {  
 $scope.isCollapsed **=** false;  
});

**Angular Datepicker**

This directive shows the dates that come from the other than the main month being displayed. User can select any date as per their convenience.

For further reference, refer to the following [Code Snippet](http://jsfiddle.net/cletourneau/kGGCZ/).

**Angular Modal(Pop Up)**

**$modal** is a service to quickly create AngularJS-powered modal windows. Creating custom modals is straightforward: create a partial view, its controller and reference them when using the service.

For further reference, refer to the following [Code Snippet](http://jsfiddle.net/alexsuch/RLQhh/).

**Angular Pagination**

A lightweight pagination directive that is focused on ... providing pagination & will take care of visualising a pagination bar and enable / disable buttons correctly!

For further reference, refer to following [Code Snippet](http://jsfiddle.net/27USb/light/).

**Angular Popover**

A lightweight, extensible directive for fancy popover creation. The popover directive supports multiple placements, optional transition animation, and more.

For further reference, refer the following [Code Snippet](http://jsfiddle.net/KyleMit/VUZhL/).

**Angular Timepicker**

This is a directive which is used to select the time and provides user convenience.

For further reference, refer the following [Code Snippet](http://www.dnasir.com/github/angular-timepicker/demo.html).

**Angular DateTimePicker**

This directive provides both the feature for selecting a date and time simultaneously in a single module.

For further reference, refer the following [Code Snippet](http://codepen.io/osales/pen/ADbHg).

**$location**

The $location service parses the URL in the browser address bar (based on [window.location](https://developer.mozilla.org/en/window.location)) and makes the URL available to your application. Changes to the URL in the address bar are reflected into the $location service and changes to $location are reflected into the browser address bar.

The $location service:

* Exposes the current URL in the browser address bar, so you can
  + Watch and observe the URL.
  + Change the URL.
* Maintains synchronization between itself and the browser's URL when the user
  + Changes the address in the browser's address bar.
  + Clicks the back or forward button in the browser (or clicks a History link).
  + Clicks on a link in the page.
* Represents the URL object as a set of methods (protocol, host, port, path, search, hash).

**Syntax for $location :**

**$location.path(‘/path to traverse’);**

**AngularJS Factory/Service**

Angular services are substitutable objects that are wired together using [dependency injection (DI)](https://docs.angularjs.org/guide/di). You can use services to organize and share code across your app.

Angular services are:

* Lazily instantiated – Angular only instantiates a service when an application component depends on it.
* Singletons – Each component dependent on a service gets a reference to the single instance generated by the service factory.

For usage of the same, refer to the following [Code Snippet](http://jsfiddle.net/vitconte/tG46g/light/).