



Angular



THE GIST

► AngularJS is a new, powerful, client-side technology that provides a way of accomplishing really powerful things in a way that embraces and extends HTML, CSS and JavaScript, while shoring up some of its glaring deficiencies. It is what HTML would have been, had it been built for dynamic content.

`{{ Two-Way Data-Binding }}`

`{{ Controller }}`

`{{ Iterating Over Properties }}`

`{{ GET Requests }}`

{{ Two-Way Data-Binding }}

Model - View <---> View - Model

```
1 <!doctype html>
2 <html ng-app>
3   <head>
4     <script src="http://code.angularjs.org/angular-1.0.0rc10.min.js"></script>
5   </head>
6   <body>
7     <div>
8       <label>Name:</label>
9       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10      <hr>
11      <h1>Hello, {{yourName}}!</h1>
12    </div>
13  </body>
14 </html>
```

.controller

```
13 <body>
14   <div class="container">div ng-controller="OMDBCtrl">
15     <h1>Angular Demo: Movie Listing</h1>
16     <div class="alert alert-info">
17       <div class="search form">
18         <input type="text" ng-model="search" class="form-control" placeholder="Search movies...">
19         <button ng-click=fetch()>Search</button>
20       </div>
21     </div>
22     <h2>Search results for "{{search}}</h2>
23     <table class="table table-bordered table-striped">=
56   </table>
57 </div>
58 </body>
```

```
1 angular.module('OMDB', [])
2   .controller('OMDBCtrl', function($scope, $http) {
3     $scope.sortType      = 'Title'; // set the default s
4     $scope.sortReverse   = false;  // set the default so
5     $scope.searchMovie = '';      // set the default sear
6     $scope.fetch = function() {
7       console.log($scope.search)
8       $http({
9         method: "GET",
10        url: "http://www.omdbapi.com/?s="+$scope.search
```

\$http GET Request

```
1 // Simple GET request example:  
2 $http({  
3   method: 'GET',  
4   url: '/someUrl'  
5 }).then(function successCallback(response) {  
6   // this callback will be called asynchronously  
7   // when the response is available  
8 }, function errorCallback(response) {  
9   // called asynchronously if an error occurs  
10  // or server returns response with an error status.  
11});
```

```
1 $.ajax({  
2   url:"some url",  
3   method:"get",  
4   dataType: "json",  
5   success: function(data, success, xhr){  
6     console.log(data)  
7   },  
8   error: function(x, y, error){  
9     console.log("error "+ error )  
10    }  
11});|
```

<div ng-repeat = "(key, value) in myObj">

.....

Variable	Type	Details
\$index	number	iterator offset of the repeated element (0..length-1)
\$first	boolean	true if the repeated element is first in the iterator.
\$middle	boolean	true if the repeated element is between the first and last in the iterator.
\$last	boolean	true if the repeated element is last in the iterator.
\$even	boolean	true if the iterator position \$index is even (otherwise false).
\$odd	boolean	true if the iterator position \$index is odd (otherwise false).

The background consists of a series of overlapping, rounded rectangular shapes in various colors, including shades of red, orange, yellow, pink, and purple. These shapes overlap each other in a non-linear fashion, creating a sense of depth and movement.

`{{ ng-Thanks }}`