This repo is for distribution on `npm` and `bower`. The source for this module is in the [main Angular Material repo (https://github.com/angular/material)](https://github.com/angular/material).
Please file issues and pull requests against that repo.

Installing Angular Material

You can install this package locally either with `npm`, `jspm`, or `bower`.

npm

```
# To install latest formal release
npm install angular-material

# To install latest release and update package.json
npm install angular-material --save

# To install from HEAD of master
npm install http://github.com/angular/bower-material/tarball/master

# To view all installed package
npm list;
```

jspm

```
# To install latest formal release
jspm install angular-material

# To install from HEAD of master
jspm install angular-material=github:angular/bower-material@master

# To view all installed package versions
jspm inspect
```

Now you can use `require('angular-material')` when installing with **npm** or **jspm**, or when using Browserify or Webpack.

bower

```
# To get the latest stable version, use bower from the command line.
bower install angular-material

# To get the most recent, last committed-to-master version use:
bower install angular-material#master

# To save the bower settings for future use:
bower install angular-material --save

# Later, you can use easily update with:
bower update
```

Please note that Angular Material requires **Angular 1.3.x** or higher.

Using the Angular Material Library

Now that you have installed the Angular libraries, simply include the scripts and stylesheet in your main HTML file, in the order shown in the example below. Note that npm will install the files under `/node_modules/angular-material/` and bower will install them under `/bower_components/angular-material/`.

npm

```html
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no" />
    <link rel="stylesheet" href="/node_modules/angular-material/angular-material.css">
</head>
    <body ng-app="YourApp">

    <div ng-controller="YourController">

    </div>

    <script src="/node_modules/angular/angular.js"></script>
    <script src="/node_modules/angular-aria/angular-aria.js"></script>
    <script src="/node_modules/angular-animate/angular-animate.js"></script>
    <script src="/node_modules/angular-material/angular-material.js"></script>
    <script>

        // Include app dependency on ngMaterial

        angular.module( 'YourApp', [ 'ngMaterial' ] )
            .controller("YourController", YourController );

    </script>

</body>
</html>
```

bower

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no" />
    <link rel="stylesheet" href="/bower_components/angular-material/angular-material.css">
</head>
    <body ng-app="YourApp">

    <div ng-controller="YourController">

    </div>

    <script src="/bower_components/angular/angular.js"></script>
    <script src="/bower_components/angular-aria/angular-aria.js"></script>
    <script src="/bower_components/angular-animate/angular-animate.js"></script>
    <script src="/bower_components/angular-material/angular-material.js"></script>
    <script>

        // Include app dependency on ngMaterial

        angular.module( 'YourApp', [ 'ngMaterial' ] )
            .controller("YourController", YourController );

    </script>

</body>
</html>
```

Using the CDN

CDN versions of Angular Material are now available at Google Hosted Libraries (https://developers.google.com/speed/libraries/devguide#angularmaterial).

With the Google CDN, you will not need to download local copies of the distribution files. Instead simply reference the CDN urls to easily use those remote library files.
This is especially useful when using online tools such as CodePen, Plunkr, or jsFiddle.

```
  <head>

    <!-- Angular Material CSS now available via Google CDN; version 0.9.4 used
here -->
    <link rel="stylesheet"
href="//ajax.googleapis.com/ajax/libs/angular_material/0.9.4/angular-
material.min.css">

  </head>
  <body>

    <!-- Angular Material Dependencies -->
    <script
src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.6/angular.min.js"></script>
    <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.6/angular-
animate.min.js"></script>
    <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.6/angular-
aria.min.js"></script>

    <!-- Angular Material Javascript now available via Google CDN; version
0.9.4 used here -->
    <script
src="//ajax.googleapis.com/ajax/libs/angular_material/0.9.4/angular-
material.min.js"></script>

  </body>
```

> Note that the above sample references the 0.9.4 CDN release. Your version will change
> based on the latest stable release version.

Jasmine Testing with Angular Material

If you are using Angular Material and will be using Jasmine to test your own custom application code, you will need to also load two (2) Angular mock files:

- Angular Mocks - **angular-mocks.js** from `/node_modules/angular-mocks/angular-mocks.js`
- Angular Material Mocks - **angular-material-mocks.js** from `/node_modules/angular-material/angular-material-mocks.js`

Shown below is a karma-configuration file (`karma.conf.js`) sample that may be a useful template for your own testing purposes:

```
module.exports = function(config) {

  var SRC = [
    'src/myApp/**/*.js',
    'test/myApp/**/*.spec.js'
  ];

  var LIBS = [
    'node_modules/angular/angular.js',
    'node_modules/angular-animate/angular-animate.js',
    'node_modules/angular-aria/angular-aria.js',
    'node_modules/angular-material/angular-material.js',

    'node_modules/angular-mocks/angular-mocks.js',
    'node_modules/angular-material/angular-material-mocks.js'
  ];

  config.set({

    basePath: __dirname + '/..',
    frameworks: ['jasmine'],

    files: LIBS.concat(SRC),

    port: 9876,
    reporters: ['progress'],
    colors: true,

    autoWatch: false,
    singleRun: true,
    browsers: ['PhantomJS,Chrome']

  });

};
```