

ngStickyNotes

Prepare

Please read the following instructions which help you to set up environment to build this project.

1. Install [Node.js \(https://nodejs.org\)](https://nodejs.org) and [npm \(https://docs.npmjs.com/\)](https://docs.npmjs.com/).
2. Install [gulp \(http://gulpjs.com/\)](http://gulpjs.com/) globally:

```
npm install -g gulp
npm install -g gulp-cli
```

3. Install [bower \(http://bower.io/\)](http://bower.io/) globally:

```
npm install -g bower
```

Please pay attention that bower requires [git \(http://git-scm.com/downloads\)](http://git-scm.com/downloads).

4. Download `npm` and `bower` packages:

```
npm install
bower install
```

If you have `Node.js`, `npm`, `bower` and `gulp` you are ready to the next stage.

Build

Here is a list of `gulp` commands which can be used to build different parts of the project.

1. To start NodeJS server and deploy *debug* version locally run this command:

```
gulp serve:debug
```

2. To start NodeJS server and deploy *production* version locally:

```
gulp serve
```

3. To build production scripts:

```
gulp build
```

Scripts will be placed in the `dist` folder in the project root.

4. To build `Readme.html` and `Readme.pdf` based on `Readme.md` file run:

```
gulp readme:html
gulp readme:pdf
```

All building tasks can be found in the `gulp\tasks` folder.

How to use

This component consists of two files `ng-sticky-notes.min.css` and `ng-sticky-notes.min.js` that can be found in the project `dist` folder. All you need is to include these files and dependencies in your html file, put angular directives on your page and run scripts. It can be done in 3 simple steps:

1. Add styles in the page header section:

```
<!-- Awesome Fonts are required for icons -->
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css" rel="stylesheet">
<!-- Reenie Beanie makes written by hand effect -->
<link href="https://fonts.googleapis.com/css?family=Reenie+Beanie:regular" rel="stylesheet">
<!-- Component Styles -->
<link href="styles/ng-sticky-notes.min.css" rel="stylesheet">
```

Add scripts at the end of the page body.

```
<!-- Dependencies -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/3.10.1/lodash.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.8/angular.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.8/angular-cookies.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-translate/2.8.1/angular-translate.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-local-storage/0.2.2/angular-local-storage.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-ui-sortable/0.13.4/sortable.min.js"></script>
<!-- Component Scripts -->
<script src="scripts/ng-sticky-notes.min.js"></script>
```

2. Add `ngStickyNotes` directives in your markup where you want to render notes:

```
!-- Render Add New button -->
<ng-sticky-notes-new></ng-sticky-notes-new>
<!-- Render Clean button -->
<ng-sticky-notes-clean></ng-sticky-notes-clean>
<!-- Render Sticky Notes -->
```

```
<div class="sticky-notes content">
  <ng-sticky-notes header-max-length="headerMaxLength"
    content-max-length="contentMaxLength"
    readonly-enabled="readonlyEnabled"
    drag-and-drop-enabled="dragAndDropEnabled"
    out-drop-enabled="outDropEnabled"
    magnetic-effect-enabled="magneticEffectEnabled">

  </ng-sticky-notes>
</div>
```

3. Run Sticker Notes scripts. If your project is already build with Angular, just make sure that ngStickyNotes module is referenced in your module dependencies.

```
angular.module('MyModule', ['ngStickyNotes'])
```

If you have no angular yet, just add ng-app="ngStickyNotes" attribute at the beginning of the page

```
<html lang="en" ng-app="ngStickyNotes">
```

or run this code that makes a similar thing automatically:

```
angular.bootstrap(document, ['ngStickyNotes']);
```

4. If you want to bind Sticky Notes to the current user profile or sessionId specify cookieKey value in the notesStorageProvider config.

```
angular
  .module('myApp', ['ngStickyNotes'])
  .config(function(notesStorageProvider) {
    notesStorageProvider.setConfig({
      cookieKey: 'uid'
    });
  })
```

5. You are able to sync notes between different instances using a [FireBase \(https://www.firebase.com\)](https://www.firebase.com) DataCloud. For this go to FireBase web site and create a new App. Put the App URL to config and specify a Sync interval (how often would you like changes posted to server).

```
angular
  .module('myApp', ['ngStickyNotes'])
  .config(function(notesStorageProvider) {
    notesStorageProvider.setConfig({
      cookieKey: 'uid',
      fireURL: 'https://sticky-notes-server.firebaseio.com/',
      SYNC_INTERVAL: 30000
    });
  })
```

Dependencies

Here are some explanations regarding project dependencies:

1. font-awesome.min.css is needed for icons.

```
https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css
```

2. Reenie Beanie font family displays text in hand written styles.

```
http://fonts.googleapis.com/css?family=Reenie+Beanie:regular
```

3. underscore or lodash is used for sorting, searching and other utility functions;

```
https://cdnjs.cloudflare.com/ajax/libs/lodash.js/3.10.1/lodash.min.js
```

4. jquery, jquery-ui and angular-ui-sortable are mostly used for drag&drop functionality.

```
https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js
https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.11.4/jquery-ui.min.js
https://cdnjs.cloudflare.com/ajax/libs/angular-ui-sortable/0.13.4/sortable.min.js
```

5. angular is awesome and used for combining all together and data binding.

```
https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.8/angular.min.js
```

6. angular-cookies is required for reading user cookies to bind sticker notes to a specific user id or session id.

```
https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.8/angular-cookies.min.js
```

7. angular-translate is required since the component supports localization.

```
https://cdnjs.cloudflare.com/ajax/libs/angular-translate/2.8.1/angular-translate.min.js
```

8. angular-local-storage is needed to store notes in the browser local storage.

```
https://cdnjs.cloudflare.com/ajax/libs/angular-local-storage/0.2.2/angular-local-storage.min.js
```

Customization

ngStickerNotes component was build with customization and flexibility in mind.

1. Update `variables.scss` to specify fonts, colors and so on.
 2. Make your own buttons to Add/Clean notes using APIs.
- to add a new note raise `ngStickyNotes.New` event on the `$rootScope`

```
$rootScope.$broadcast('ngStickyNotes.New', {});
```

- to remove a specific note raise `ngStickyNotes.Remove` event on the `$rootScope` with the note object:

```
$rootScope.$broadcast('ngStickyNotes.Remove', stickerItem);
```

or

```
$rootScope.$broadcast('ngStickyNotes.Remove', {id: stickerItemId});
```

- to remove all notes raise `ngStickyNotes.RemoveAll` event on the `$rootScope`:

```
$rootScope.$broadcast('ngStickyNotes.RemoveAll');
```

- to refresh a list of notes raise `ngStickyNotes.Refresh` event on the `$rootScope`:

```
$rootScope.$broadcast('ngStickyNotes.Refresh');
```

3. To add more colors for sticky notes just specified your own color palette in the `colorsPaletteProvider`.

```
colorsPaletteProvider.setCustomColors(['#ffc', '#cfc', '#ccf']);
```

4. To support more languages add your own localized constants like `src\scripts\i18n\locale.en-US.js` and register it in the module config this way.

```
angular
.module('ngStickyNotes')
.config(function ($translateProvider, EN_US) {
    $translateProvider.translations('en-US', EN_US);
    $translateProvider.preferredLanguage('en-US');
});
```