



## Guía de migración de aplicaciones angular de 1.0.x a 1.1.0

---

14 de Noviembre de 2016

# Índice de contenidos

1	INTRODUCCIÓN .....	3
2	PACKAGE.JSON .....	3
3	ANGULAR-CLI-BUILD.JS.....	5
4	SYSTEM-CONFIG.TS .....	6
5	TABS.....	7
6	TEMA DE LA APLICACIÓN .....	7

---

# 1 Introducción

En este documento se explica paso a paso como actualizar una aplicación web realizada con la versión 1.0.x del framework Ontimize Web a la versión 1.1.0.

1. Actualizar dependencias en el fichero *package.json*.
2. Actualizar el fichero *angular-cli-build.js*.
3. Actualizar el fichero *system-config.ts*.
4. Definir tema de la aplicación.

---

## 2 Package.json

Realizar en el fichero *package.json* las siguientes modificaciones:

- Actualizar versión de los módulos estándar angular2 de 2.0.0 a 2.1.2
- Eliminar módulos de componentes material *@angular2-material/xxx@v2.0.0-alpha.8-2*.
- Añadir módulo *@angular/material@v2.0.0-alpha.10*
- Subir versión de *ontimize-web-ng2* a 1.1.0
- En caso de no haberlo hecho ya, actualizar la versión de *angular-cli* a la versión 1.0.0-beta.10.

El resultado del archivo debe ser como se muestra en el siguiente cuadro.

```
"dependencies": {
  "@angular/common": "2.1.2",
  "@angular/compiler": "2.1.2",
  "@angular/core": "2.1.2",
  "@angular/forms": "2.1.2",
  "@angular/http": "2.1.2",
  "@angular/platform-browser": "2.1.2",
  "@angular/platform-browser-dynamic": "2.1.2",
  "@angular/router": "3.0.0-rc.2",
  "@angular/upgrade": "2.1.2",
  "@angular/material": "v2.0.0-alpha.10",

  "colresizable": "1.6.0",
  "core-js": "^2.4.1",
  "datatables.net": "^1.10.12",
  "datatables.net-buttons": "^1.2.1",
  "datatables.net-colreorder": "^1.3.2",
  "datatables.net-select": "^1.2.0",
  "jquery": "^3.0.0",
  "jquery-ui": "^1.12.0",
  "moment": "^2.13.0",
  "ng2-translate": "2.5.0",
  "pdfmake": "^0.1.18",
  "reflect-metadata": "^0.1.3",
  "rxjs": "5.0.0-beta.12",
  "systemjs": "0.19.38",
  "zone.js": "0.6.23"
},
"devDependencies": {
  "angular-cli": "1.0.0-beta.10",
  "clang-format": "^1.0.35",
  "codelyzer": "0.0.14",
  "ember-cli-inject-live-reload": "^1.4.0",
  "jasmine-core": "^2.4.1",
  "jasmine-spec-reporter": "^2.4.0",
  "karma": "^0.13.15",
  "karma-chrome-launcher": "^0.2.3",
  "karma-jasmine": "^0.3.8",
  "node-sass": "3.7.0",
  "protractor": "^3.3.0",
  "systemjs-builder": "^0.15.33",
  "ts-node": "^0.5.5",
  "tslint": "^3.6.0",
  "typescript": "^2.0.2",
  "typings": "^0.8.1"
}
```

Para realizar las actualizaciones de versiones se recomienda ejecutar la consola con permisos de administrador y previamente borrar las carpetas de las versiones antiguas contenidas en *node\_modules*.

---

## 3 Angular-cli-build.js

Se debe modificar el archivo *angular-cli-build.js* para adaptarlo a las nuevas dependencias. El contenido del fichero será:

```
/* global require, module */

var Angular2App = require('angular-cli/lib/broccoli/angular2-app');

module.exports = function(defaults) {
  return new Angular2App(defaults, {
    vendorNpmFiles: [
      'systemjs/dist/system-polyfills.js',
      'systemjs/dist/system.src.js',
      'zone.js/dist/**/*.+(js|js.map)',
      'core-js/client/**/*.+(js|js.map)',
      'reflect-metadata/**/*.+(js|js.map)',
      'rxjs/**/*.+(js|js.map)',
      '@angular/**/*.+(js|js.map)',
      '@angular/material/**/*.+(js|js.map|css|css.map|scss)',

      'ontimize-web-ng2/*.js',
      'ontimize-web-ng2/ontimize/**/*.js',

      'ng2-translate/**/*.js',
      'moment/**/*.js',
      'jquery/dist/jquery.min.js',
      'jquery-ui/ui/**/*.js',
      'pdfmake/build/*.js',
      'datatables.net/js/jquery.dataTables.js',
      'datatables.net-buttons/js/*.js',
      'datatables.net-colreorder/js/dataTables.colReorder.js',
      'datatables.net-select/js/dataTables.select.js',
      'colresizable/*.js',

      'ontimize-web-ng2/**/*.css'
    ]
  });
};
```

## 4 System-config.ts

En el fichero *system-config.ts* también se tiene que adaptar a las nuevas dependencias. Para ello será necesario eliminar las referencias a la librería antigua *@angular2-material* e incluir la nueva librería *@angular/material*.

El contenido a modificar de ese fichero es el siguiente:

```
/** Map relative paths to URLs. */
const map: any = {
  ...
  // Add this
  '@angular/material': 'vendor/@angular/material/material.umd.js'

  // Remove this
  '@angular2-material/button': 'vendor/@angular2-material/button/button.umd.js',
  '@angular2-material/button-toggle': 'vendor/@angular2-material/button-toggle/button-toggle.umd.js',
  '@angular2-material/card': 'vendor/@angular2-material/card/card.umd.js',
  '@angular2-material/checkbox': 'vendor/@angular2-material/checkbox/checkbox.umd.js',
  '@angular2-material/core': 'vendor/@angular2-material/core/core.umd.js',
  '@angular2-material/grid-list': 'vendor/@angular2-material/grid-list/grid-list.umd.js',
  '@angular2-material/icon': 'vendor/@angular2-material/icon/icon.umd.js',
  '@angular2-material/input': 'vendor/@angular2-material/input/input.umd.js',
  '@angular2-material/list': 'vendor/@angular2-material/list/list.umd.js',
  '@angular2-material/menu': 'vendor/@angular2-material/menu/menu.umd.js',
  '@angular2-material/progress-bar': 'vendor/@angular2-material/progress-bar/progress-bar.umd.js',
  '@angular2-material/progress-circle': 'vendor/@angular2-material/progress-circle/progress-circle.umd.js',
  '@angular2-material/radio': 'vendor/@angular2-material/radio/radio.umd.js',
  '@angular2-material/sidenav': 'vendor/@angular2-material/sidenav/sidenav.umd.js',
  '@angular2-material/slide-toggle': 'vendor/@angular2-material/slide-toggle/slide-toggle.umd.js',
  '@angular2-material/slider': 'vendor/@angular2-material/slider/slider.umd.js',
  '@angular2-material/tabs': 'vendor/@angular2-material/tabs/tabs.umd.js',
  '@angular2-material/toolbar': 'vendor/@angular2-material/toolbar/toolbar.umd.js',
  '@angular2-material/tooltip': 'vendor/@angular2-material/tooltip/tooltip.umd.js'
};
```

---

## 5 Tabs

El componente ‘tab’ de la librería `@angular/material` ha cambiado su notación, por lo que es necesario actualizar las pantallas que incluyan este componente.

Notación antigua:

```
<md-tab-group layout-fill>
  <md-tab>
    <template md-tab-label>{{ 'CUSTOMERS' | oTranslate }}</template>
    <template md-tab-content>
      <!-- content of your tab -->
      <div layout="column" layout-padding>
        ...
      </div>
    </template>
  </md-tab>
</md-tab-group>
```

Nueva notación:

```
<md-tab-group layout-fill>
  <md-tab label="{{ 'CUSTOMERS' | oTranslate }}">
    <!-- content of your tab -->
    <div layout="column" layout-padding>
      ...
    </div>
  </md-tab>
</md-tab-group>
```

---

## 6 Tema de la aplicación

En esta versión del framework se permite definir la gama de colores (tema) que se aplican a la aplicación. Para ello se pueden configurar dos opciones:

- Utilizar un tema predefinido en la librería `@angular/material`
- Definir un tema propio.

Sea cual sea la opción escogida, se debe configurar en el archivo `app.scss` de la aplicación.

## a) Tema predefinidos

En la librería `@angular/material` existen una serie de temas predefinidos que se pueden incorporar directamente a la aplicación. Para ello en el archivo `app.scss` deberemos hacer lo siguiente:

```
/* Temas predefinidos.
   Escoger uno de la siguiente lista de imports
*/
/*@import 'node_modules/@angular/material/core/theming/prebuilt/deeppurple-amber.scss';*/
/*@import 'node_modules/@angular/material/core/theming/prebuilt/indigo-pink.scss';*/
/*@import 'node_modules/@angular/material/core/theming/prebuilt/pink-bluegrey.scss';*/
/*@import 'node_modules/@angular/material/core/theming/prebuilt/purple-green.scss';*/

/*Transferir al framework el tema para que se configuren los componentes de acuerdo a la
gama de colores seleccionada
*/
@import 'node_modules/ontimize-web-ng2/ontimize/components/theming/all-theme.scss';

@include o-material-theme($theme);

/* Extra style definition of app */
```

## b) Tema propio

Para definir un tema propio crearemos un archivo de estilos separado, en el que se definirán los colores del tema, y después lo importaremos en nuestro fichero de estilos principal de la aplicación (`app.scss`). El contenido del archivo `app.scss` será el siguiente:

```
/* Temas propio.*/
@import 'my-custom-app-theme.scss';

/*Transferir al framework el tema para que se configuren los componentes de acuerdo a la
gama de colores seleccionada
*/
@import 'node_modules/ontimize-web-ng2/ontimize/components/theming/all-theme.scss';

@include o-material-theme($theme);

/* Extra style definition of app */
```

Nuestro tema propio lo definimos en el archivo `my-custom-app-theme.scss`, cuyo contenido es el siguiente:



```
@import 'node_modules/@angular/material/core/theming/all-theme';

// Include non-theme styles for core.
@include md-core();

/* Custom color definition */
$md-custom-primary: (
  50: #bdf5b3,
  100: #82eb6f,
  200: #57e53e,
  300: #33c11a,
  400: #2ca617,
  500: #258b13,
  600: #1e700f,
  700: #17550c,
  800: #103a08,
  900: #081f04,
  A100: #bdf5b3,
  A200: #82eb6f,
  A400: #2ca617,
  A700: #17550c,
  contrast: (
    50: $black-87-opacity,
    100: $black-87-opacity,
    200: $black-87-opacity,
    300: white,
    400: white,
    500: $white-87-opacity,
    600: $white-87-opacity,
    700: $white-87-opacity,
    800: $white-87-opacity,
    900: $white-87-opacity,
    A100: $black-87-opacity,
    A200: white,
    A400: white,
    A700: white,
  )
);

// Define a theme.
$primary: md-palette($md-custom-primary);
/* md-amber is a predefined color */
$accent: md-palette($md-amber, A200, A100, A400);

/* Light theme */
$theme: md-light-theme($primary, $accent);

// Include all theme styles for the components.
@include angular-material-theme($theme);
```