



EBook Gratis

APRENDIZAJE angular-material2

Free unaffiliated eBook created from
Stack Overflow contributors.

#angular-
material2

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con material angular2.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración con CLI angular.....	2
Envolviendo todos los módulos juntos.....	3
Instalación y configuración desde el maestro con Angular CLI.....	4
Configurar tema, gestos de gestos e iconos de material.....	5
Capítulo 2: botón md.....	7
Introducción.....	7
Parámetros.....	7
Observaciones.....	7
Examples.....	7
Botones simples.....	7
Capítulo 3: icono-md.....	9
Examples.....	9
Creando un icono.....	9
Usando iconos SVG.....	9
Capítulo 4: md-autocompletar.....	11
Introducción.....	11
Observaciones.....	11
Examples.....	11
Control separado y pantalla.....	11
Obtenga las opciones de md-autocomplete / datos de búsqueda de la API.....	12
Utilice md-autocompletar dentro de una forma reactiva.....	14
Un md-autocompletar para múltiples formControl.....	17
Capítulo 5: md-datepicker.....	20
Introducción.....	20
Observaciones.....	20

Examples.....	20
Enlace de datos con md-datapicker.....	20
Pasar el valor de fecha seleccionado a una función usando \$ event.....	20
Abre el selector de fechas en foco.....	21
Establecer diferente locale para md-datepicker.....	23
Capítulo 6: md-dialog.....	25
Introducción.....	25
Observaciones.....	25
Examples.....	25
Inicializar md-dialog con los datos pasados desde el componente principal.....	25
Capítulo 7: md-table.....	27
Introducción.....	27
Observaciones.....	27
Examples.....	27
Conecte DataSource desde una API externa.....	27
Creditos.....	30

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [angular-material2](#)

It is an unofficial and free angular-material2 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official angular-material2.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con material angular2

Observaciones

Esta sección proporciona una descripción general de qué es angular-material2 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de material angular2, y vincular a los temas relacionados. Dado que la Documentación para angular-material2 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Versiones

Versión	Registro de cambios	Fecha
2.0.0-beta.8	Enlazar	2017-07-06
2.0.0-beta.7	Enlazar	2017-06-19
2.0.0-beta.6	Enlazar	2017-05-25
2.0.0-beta.5	Enlazar	2017-05-13
2.0.0-beta.4	Enlazar	2017-05-12
2.0.0-beta.3	Enlazar	2017-04-07
2.0.0-beta.2	Enlazar	2017-02-15
2.0.0-beta.1	Enlazar	2016-12-23
2.0.0-beta.0	Enlazar	2016-12-22

Examples

Instalación o configuración con CLI angular

En este ejemplo, `@angular/cli` (más reciente) y la última versión de `@angular/material`. Al menos debe conocer los conceptos básicos de Angular 2/4 antes de continuar con los pasos a continuación.

1. Instale el módulo de material angular de `npm` :

```
npm install @angular/material --save
```

2.0.0-beta.3

Esto solo se aplica a las versiones 2.0.0-beta.3 y superiores.

Instala el módulo @angular/animations :

```
npm install @angular/animations --save
```

2.0.0-beta.8

Esto solo se aplica a las versiones 2.0.0-beta.8 y superiores.

Instale el módulo @angular/cdk :

```
npm install @angular/cdk --save
```

2. En su módulo de aplicación, importe los componentes que va a utilizar:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule } from '@angular/router';
import { MatButtonModule, MdSnackBarModule, MdSidenavModule } from '@angular/material';

import { AppComponent } from './app.component';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

@NgModule({
  imports: [
    BrowserAnimationsModule,
    MatButtonModule,
    MdSnackBarModule,
    MdSidenavModule,
    CommonModule,
    // This is optional unless you want to have routing in your app
    // RouterModule.forRoot([
    //   { path: '', component: HomeView, pathMatch: 'full' }
    // ])
  ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

¡Ya está listo para usar material angular en sus componentes!

Nota: Los documentos para componentes específicos estarán disponibles próximamente.

Envolviendo todos los módulos juntos

También puede envolver fácilmente todos los módulos angulares, que va a utilizar, en un solo módulo:

```
import { NgModule } from '@angular/core';
import { MatButtonModule, MdSnackBarModule, MdSidenavModule } from '@angular/material';
```

```

@NgModule({
  imports: [
    BrowserAnimationsModule,
    MdButtonModule,
    MdSnackBarModule,
    MdSidenavModule
  ],
  exports: [
    MdButtonModule,
    MdSnackBarModule,
    MdSidenavModule
  ]
})
export class MaterialWrapperModule {}

```

Después de eso, simplemente importa tu módulo en el módulo principal de la aplicación:

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule } from '@angular/router';

import { MaterialWrapperModule } from './material-wrapper.module.ts';
import { AppComponent } from './app.component';

@NgModule({
  imports: [
    BrowserAnimationsModule,
    MaterialWrapperModule,
    CommonModule,
    // This is optional, use it when you would like routing in your app
    // RouterModule.forRoot([
    //   { path: '', component: HomeView, pathMatch: 'full' }
    // ])
  ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}

```

Instalación y configuración desde el maestro con Angular CLI

Este ejemplo será cómo instalarlo desde el maestro y también usará `@angular/cli`.

1. Realiza un nuevo proyecto con `@angular/cli`:

```
ng new my-master-app
```

Si no has instalado `@angular/cli`, usa este comando:

```
npm install -g @angular/cli
```

2. Instalar desde el `master`:

```
npm install --save @angular/animations
npm install --save angular/material2-builds
npm install --save angular/cdk-builds
```

3. Siga la misma guía que el anterior.

¡Hecho!

Configurar tema, gestos de gestos e iconos de material.

Tema:

Se **requiere** un tema para que los componentes de material funcionen correctamente dentro de la aplicación.

Angular Material 2 proporciona cuatro temas predefinidos:

- ámbar profundo
- rosa índigo
- gris azulado
- verde púrpura

Si está utilizando **Angular CLI** , puede importar uno de los temas `style.css` en `style.css` .

```
@import "~@angular/material/prebuilt-themes/indigo-pink.css";
```

El tema también se puede agregar usando `<link>` en `index.html` .

```
<link href="node_modules/@angular/material/prebuilt-themes/indigo-pink.css" rel="stylesheet">
```

HammerJS

Agregue HammerJS a la aplicación a través de [CDN](#) o `npm` :

```
npm install --save hammerjs
```

En su módulo raíz, generalmente `app.module.ts` , agregue la declaración de importación:

```
import 'hammerjs';
```

Iconos de material:

A menos que se proporcionen iconos personalizados, Angular Material 2 `<md-icon>` espera iconos de materiales.

En `index.html` añadir:


```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

Lea Empezando con material angular2 en línea: <https://riptutorial.com/es/angular-material2/topic/10828/empezando-con-material-angular2>

Capítulo 2: botón md

Introducción

Este tema incluye ejemplos sobre cómo crear un botón y qué hacen sus directivas y otras cosas.

Parámetros

Atributo	Descripción
md-button	Crea un botón rectangular sin elevación.
md-raised-button	Crea un botón rectangular con elevación.
md-icon-button	Crea un botón circular con un fondo transparente, destinado a contener un icono.
md-fab	Crea un botón circular con elevación, predeterminado al color de acento del tema
md-mini-fab	Igual que <code>md-fab</code> pero más pequeño
disableRipple	Si el efecto de onda para el botón está deshabilitado.

Observaciones

Para obtener más información y más ejemplos, visite [la documentación](#) .

Examples

Botones simples

Para crear un botón, use el atributo `md-button` para un botón plano y `md-raised-button` para un botón elevado:

```
<button md-button>Button</button>
<button md-raised-button>Raised Button</button>
<button md-fab><md-icon>add</md-icon></button>
<button md-mini-fab><md-icon>check</md-icon></button>
<button md-icon-button><md-icon>person_add</md-icon></button>
```

[Demo plunker](#)

Para obtener más información sobre los íconos, vea los documentos en [md-icon](#) .

Lea botón md en línea: <https://riptutorial.com/es/angular-material2/topic/10870/boton-md>

Capítulo 3: icono-md

Examples

Creando un icono

La siguiente es una guía sobre cómo crear un ícono a partir de íconos de diseño de materiales:

1. Cargue la fuente del icono de Google CDN en su `index.html` :

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

Alternativamente, puede importarlo en su `styles.css` :

```
@import url('https://fonts.googleapis.com/icon?family=Material+Icons');
```

2. Úsalo de la siguiente manera:

```
<md-icon>menu</md-icon>
```

Has terminado

Usando iconos SVG

Este ejemplo muestra cómo utilizar los iconos SVG en su aplicación.

1. Descargue el conjunto de iconos / íconos SVG (en este caso, estamos obteniendo los íconos de <https://materialdesignicons.com/getting-started>).
2. Guárdelo en la carpeta de `assets` o en otro lugar al que pueda acceder.
3. En su `app.module.ts`, agregue el siguiente código:

```
import { MdIconRegistry } from '@angular/material';
import { DomSanitizer } from '@angular/platform-browser';

export class AppModule {
  constructor(mdIconRegistry: MdIconRegistry, domSanitizer: DomSanitizer){
    // Note that you have to sanitize the resource since Angular will complain that
    // it will cause XSS problems.
    // More info: https://g.co/ng/security#xss

    mdIconRegistry.addSvgIconSet(domSanitizer.bypassSecurityTrustResourceUrl('assets/icons.svg'))
  }
}
```

4. Úsalo a través del atributo `svgIcon` :

```
<md-icon svgIcon="menu"></md-icon>
```

Lea icono-md en línea: <https://riptutorial.com/es/angular-material2/topic/10868/icono-md>

Capítulo 4: md-autocompletar

Introducción

Este tema incluye ejemplos de codificación relacionados con Autocompletar material angular 2 (md-autocompletar)

Observaciones

Estos ejemplos no cubren todas las características de md-autocomplete. Lea la [documentación para](#) obtener más información sobre md-autocomplete.

Examples

Control separado y pantalla

Este ejemplo muestra cómo mostrar una propiedad específica en el menú desplegable pero enlazar con todo el objeto.

autocomplete-overview-example.html:

```
<md-input-container>
  <input mdInput placeholder="State" [(ngModel)]="selection"
    [mdAutocomplete]="auto" [formControl]="stateCtrl">
</md-input-container>

<md-autocomplete #auto="mdAutocomplete" [displayWith]="displayFn">
  <md-option *ngFor="let state of filteredStates | async" [value]="state" >
    {{ state.Country }}
  </md-option>
</md-autocomplete>

<p>Selected Country: {{selection | json}}</p>
<p>Selected Country Id: {{selection?.CountryID}}</p>
```

autocomplete-overview-example.ts:

```
import {Component} from '@angular/core';
import {FormControl} from '@angular/forms';

import 'rxjs/add/operator/startWith';
import 'rxjs/add/operator/map';

@Component({
  selector: 'autocomplete-overview-example',
  templateUrl: 'autocomplete-overview-example.html',
})
export class AutocompleteOverviewExample {
  stateCtrl: FormControl;
```

```

filteredStates: any;

selection: any;

states = [
  { Country: "United States Of America" , CountryID: "1"},
  { Country: "United Kingdom" , CountryID: "2"},
  { Country: "United Arab Emirates" , CountryID: "3"},
];

constructor() {
  this.stateCtrl = new FormControl();
  this.filteredStates = this.stateCtrl.valueChanges
    .startWith(null)
    .map(country => country && typeof country === 'object' ? country.Country : country)
    .map(name => this.filterStates(name));
}

filterStates(val) {
  return val ? this.states.filter(s => s.Country.toLowerCase().indexOf(val.toLowerCase())
=== 0)
    : this.states;
}

displayFn(country): string {
  console.log(country);
  return country ? country.Country : country;
}
}

```

Ejemplo vivo

Obtenga las opciones de md-autocomplete / datos de búsqueda de la API

data.service.ts:

```

import { Injectable } from '@angular/core';
import { Http } from '@angular/http';
import 'rxjs/add/operator/map';

@Injectable()
export class DataService {

  constructor(private http: Http) { }

  fetchData(){
    return this.http.get('https://dinstruct-d4b62.firebaseio.com/.json')
      .map((res) => res.json())
  }
}

```

autocomplete-overview-example.html:

```
<md-input-container>
```

```



```

autocomplete-overview-example.ts:

```

import {Component, OnInit} from '@angular/core';
import {FormControl} from '@angular/forms';

import { DataService } from './data.service';
import 'rxjs/add/operator/startWith';
import 'rxjs/add/operator/map';

@Component({
  selector: 'autocomplete-overview-example',
  templateUrl: './autocomplete-overview-example.html',
})
export class AutocompleteOverviewExample implements OnInit{
  stateCtrl: FormControl;

  filteredSectors: any;

  allSectors;

  constructor(private dataService: DataService) {
    this.stateCtrl = new FormControl();
  }

  ngOnInit(){
    this.dataService.fetchData()
      .subscribe(
        (data) => {
          this.allSectors = data.customers;
          this.filteredSectors = this.stateCtrl.valueChanges
            .startWith(null)
            .map(val => val ? this.filter(val) : this.allSectors.slice());
        }
      );
  }

  filter(name) {
    return this.allSectors.filter(sector => new RegExp(`^${name}`, 'gi').test(sector.name));
  }

  displayFn(sector) {
    return sector ? sector.name : sector;
  }
}

```


Ejemplo vivo

Utilice md-autocompletar dentro de una forma reactiva

Este ejemplo requiere `FormsModule` y `ReactiveFormsModule` . Por favor importarlos en su aplicación / módulo.

```
import {FormsModule, ReactiveFormsModule} from '@angular/forms';
```

input-form-example.html

```
<form class="example-form" (ngSubmit)="submit(addForm.value)" [formGroup]="addForm">
  <md-input-container class="example-full-width">
    <input mdInput placeholder="Company (disabled)" disabled value="Google"
formControlName="company">
  </md-input-container>

  <table class="example-full-width" cellspacing="0"><tr>
    <td><md-input-container class="example-full-width">
      <input mdInput placeholder="First name" formControlName="fname">
    </md-input-container></td>
    <td><md-input-container class="example-full-width">
      <input mdInput placeholder="Long Last Name That Will Be Truncated">
    </md-input-container></td>
  </tr></table>

  <p>
    <md-input-container class="example-full-width">
      <textarea mdInput placeholder="Address" formControlName="address">1600 Amphitheatre
Pkwy</textarea>
    </md-input-container>
    <md-input-container class="example-full-width">
      <textarea mdInput placeholder="Address 2"></textarea>
    </md-input-container>
  </p>

  <table class="example-full-width" cellspacing="0"><tr>
    <td><md-input-container class="example-full-width">
      <input mdInput placeholder="City" formControlName="city">
    </md-input-container></td>

    <td><md-input-container>
      <input mdInput placeholder="State"
        [mdAutocomplete]="auto"
        [formControl]="stateCtrl"
        formControlName="state">
    </md-input-container></td>

    <td><md-input-container class="example-full-width">
      <input mdInput #postalCode maxLength="5" placeholder="Postal Code" value="94043"
formControlName="zip">
      <md-hint align="end">{{postalCode.value.length}} / 5</md-hint>
    </md-input-container></td>
  </tr></table>

  <button md-raised-button type="submit">Submit</button>

  <md-autocomplete #auto="mdAutocomplete" >
```

```

    <md-option *ngFor="let state of filteredStates | async" [value]="state"
(onSelectionChange)="selectState(state, addForm.value)">
        {{ state }}
    </md-option>
</md-autocomplete>

</form>

<p>Form values:</p>
<p>{{ addForm.value | json }}</p>

```

input-form-example.ts:

```

import {Component} from '@angular/core';
import {FormBuilder, FormGroup, FormControl} from '@angular/forms';
import 'rxjs/add/operator/startWith';
import 'rxjs/add/operator/map';

@Component({
  selector: 'input-form-example',
  templateUrl: 'input-form-example.html',
  styleUrls: ['input-form-example.css'],
})
export class InputFormExample {
  stateCtrl: FormControl;
  filteredStates: any;

  addForm: FormGroup;

  state;

  states = [
    'Alabama',
    'Alaska',
    'Arizona',
    'Arkansas',
    'California',
    'Colorado',
    'Connecticut',
    'Delaware',
    'Florida',
    'Georgia',
    'Hawaii',
    'Idaho',
    'Illinois',
    'Indiana',
    'Iowa',
    'Kansas',
    'Kentucky',
    'Louisiana',
    'Maine',
    'Maryland',
    'Massachusetts',
    'Michigan',
    'Minnesota',
    'Mississippi',
    'Missouri',
    'Montana',
    'Nebraska',
    'Nevada',

```

```

    'New Hampshire',
    'New Jersey',
    'New Mexico',
    'New York',
    'North Carolina',
    'North Dakota',
    'Ohio',
    'Oklahoma',
    'Oregon',
    'Pennsylvania',
    'Rhode Island',
    'South Carolina',
    'South Dakota',
    'Tennessee',
    'Texas',
    'Utah',
    'Vermont',
    'Virginia',
    'Washington',
    'West Virginia',
    'Wisconsin',
    'Wyoming',
];

constructor(private fb: FormBuilder) {

    this.addForm = this.fb.group({
        fname: '',
        address: '',
        address2: '',
        city: '',
        "state": this.state,
        zip: '',
        company: '',
        lname: ''
    });
    this.stateCtrl = new FormControl();
    this.filteredStates = this.stateCtrl.valueChanges
        .startWith(null)
        .map(name => this.filterStates(name));
}

filterStates(val: string) {
    return val ? this.states.filter(s => new RegExp(`^${val}`, 'gi').test(s))
        : this.states;
}

submit(form) {
    alert(JSON.stringify(form));
}

selectState(state, form) {
    // console.log(state);
    // console.log(form);
    form.state = state;
}
}

```

input-form-example.css:

```
.example-form {
  width: 500px;
}

.example-full-width {
  width: 100%;
}
```

Ejemplo vivo

Un md-autocompletar para múltiples formControl

Este ejemplo requiere `FormsModule` y `ReactiveFormsModule` . Por favor importarlos en su aplicación / módulo.

```
import {FormsModule, ReactiveFormsModule} from '@angular/forms';
```

autocomplete-overview-example.html:

```
<md-input-container>
  <input mdInput placeholder="State" [mdAutocomplete]="auto" [formControl]="stateCtrl">
</md-input-container>

<p></p>

<md-input-container>
  <input mdInput placeholder="State2" [mdAutocomplete]="auto" [formControl]="stateCtrl2">
</md-input-container>

<p></p>

<md-input-container>
  <input mdInput placeholder="State3" [mdAutocomplete]="auto" [formControl]="stateCtrl3">
</md-input-container>

<md-autocomplete #auto="mdAutocomplete">
  <md-option *ngFor="let state of filteredStates | async" [value]="state">
    {{ state }}
  </md-option>
</md-autocomplete>
```

autocomplete-overview-example.ts:

```
import {Component} from '@angular/core';
import {FormControl} from '@angular/forms';

import 'rxjs/add/operator/startWith';
import 'rxjs/add/operator/map';

@Component({
  selector: 'autocomplete-overview-example',
  templateUrl: 'autocomplete-overview-example.html',
})
export class AutocompleteOverviewExample {
  stateCtrl: FormControl;
  stateCtrl2: FormControl;
```

```
stateCtrl3: FormControl;
filteredStates: any;

states = [
  'Alabama',
  'Alaska',
  'Arizona',
  'Arkansas',
  'California',
  'Colorado',
  'Connecticut',
  'Delaware',
  'Florida',
  'Georgia',
  'Hawaii',
  'Idaho',
  'Illinois',
  'Indiana',
  'Iowa',
  'Kansas',
  'Kentucky',
  'Louisiana',
  'Maine',
  'Maryland',
  'Massachusetts',
  'Michigan',
  'Minnesota',
  'Mississippi',
  'Missouri',
  'Montana',
  'Nebraska',
  'Nevada',
  'New Hampshire',
  'New Jersey',
  'New Mexico',
  'New York',
  'North Carolina',
  'North Dakota',
  'Ohio',
  'Oklahoma',
  'Oregon',
  'Pennsylvania',
  'Rhode Island',
  'South Carolina',
  'South Dakota',
  'Tennessee',
  'Texas',
  'Utah',
  'Vermont',
  'Virginia',
  'Washington',
  'West Virginia',
  'Wisconsin',
  'Wyoming',
];

constructor() {
  this.stateCtrl = new FormControl();
  this.stateCtrl2 = new FormControl();
  this.stateCtrl3 = new FormControl();
  this.filteredStates = this.stateCtrl.valueChanges
```

```
        .startWith(null)
        .map(name => this.filterStates(name));
this.filteredStates = this.stateCtrl2.valueChanges
    .startWith(null)
    .map(name => this.filterStates(name));
this.filteredStates = this.stateCtrl3.valueChanges
    .startWith(null)
    .map(name => this.filterStates(name));
}

filterStates(val: string) {
    return val ? this.states.filter(s => s.toLowerCase().indexOf(val.toLowerCase()) !== -1)
        : this.states;
}
}
```

Ejemplo vivo

Lea md-autocompletar en línea: <https://riptutorial.com/es/angular-material2/topic/10850/md-autocompletar>

Capítulo 5: md-datepicker

Introducción

Este tema se centra en ejemplos relacionados con md-datepicker.

Observaciones

Para obtener más detalles, consulte la documentación de md-datepicker [aquí](#).

Examples

Enlace de datos con md-datapicker

datepicker-overview-example.html:

```
<md-input-container>
  <input mdInput
    [mdDatepicker]="picker"
    [(ngModel)]="date"
    placeholder="Choose a date">
  <button mdSuffix [mdDatepickerToggle]="picker"></button>
</md-input-container>
<md-datepicker #picker></md-datepicker>
<div>
  Date Chosen using 'ngModel':
  <div>{{ date }}</div>
</div>
```

datepicker-overview-example.ts:

```
import {Component, OnInit} from '@angular/core';

@Component({
  selector: 'datepicker-overview-example',
  templateUrl: 'datepicker-overview-example.html'
})
export class DatepickerOverviewExample implements OnInit {

  date;

  ngOnInit() {
    this.date = new Date();
  }
}
```

[Demo en vivo](#)

Pasar el valor de fecha seleccionado a una función usando \$ event

datepicker-overview-example.html:

```
<md-input-container>
  <input mdInput [mdDatepicker]="picker" placeholder="Choose a date" [(ngModel)]="value">
  <button mdSuffix [mdDatepickerToggle]="picker"></button>
</md-input-container>
<md-datepicker #picker [startAt]="startDate" (selectedChanged)="selectedDate($event)"></md-
datepicker>

<p>ngModel Value: {{value}}</p>

<p>Date from selectedDate(): {{checkDate}}</p>
```

datepicker-overview-example.ts:

```
import {Component} from '@angular/core';

@Component({
  selector: 'datepicker-overview-example',
  templateUrl: 'datepicker-overview-example.html'
})
export class DatepickerOverviewExample {

  value: Date = new Date();

  checkDate: Date;

  selectedDate(date){
    // ngModel still returns the old value
    console.log("ngModel: " + this.value);

    // date passes the newly selected value
    console.log("Selected Value: " + date);
    this.checkDate = date;
  }
}
```

[Demo en vivo](#)

Abre el selector de fechas en foco

Este ejemplo también incluye el uso de propiedades:

- min
- max
- Empieza en
- startView
- touchUi

datepicker-overview-example.html:

```
<h2>Options</h2>
<p>
  <md-checkbox [(ngModel)]="touch">Use touch UI</md-checkbox>
  <md-checkbox [(ngModel)]="filterOdd">Filter odd months and dates</md-checkbox>
```



```

    <md-checkbox [(ngModel)]="yearView">Start in year view</md-checkbox>
</p>
<p>
    <md-input-container>
        <input mdInput [mdDatepicker]="minDatePicker" [(ngModel)]="minDate" placeholder="Min date"
(keydown)="false" (click)="minDatePicker.open()">
        <button mdSuffix [mdDatepickerToggle]="minDatePicker"></button>
    </md-input-container>
    <md-datepicker #minDatePicker [touchUi]="touch"></md-datepicker>
    <md-input-container>
        <input mdInput [mdDatepicker]="maxDatePicker" [(ngModel)]="maxDate" placeholder="Max date"
(keydown)="false" (focus)="maxDatePicker.open()">
        <button mdSuffix [mdDatepickerToggle]="maxDatePicker"></button>
    </md-input-container>
    <md-datepicker #maxDatePicker [touchUi]="touch"></md-datepicker>
</p>
<p>
    <md-input-container>
        <input mdInput [mdDatepicker]="startAtPicker" [(ngModel)]="startAt" placeholder="Start at
date" (keydown)="false" (focus)="startAtPicker.open()">
        <button mdSuffix [mdDatepickerToggle]="startAtPicker"></button>
    </md-input-container>
    <md-datepicker #startAtPicker [touchUi]="touch"></md-datepicker>
</p>
<h2>Result</h2>
<p>
    <button [mdDatepickerToggle]="resultPicker"></button>
    <md-input-container>
        <input mdInput
            #resultPickerModel="ngModel"
            [mdDatepicker]="resultPicker"
            [(ngModel)]="date"
            [min]="minDate"
            [max]="maxDate"
            [mdDatepickerFilter]="filterOdd ? dateFilter : null"
            placeholder="Pick a date"
            (keydown)="false"
            (focus)="resultPicker.open()">
        <md-error *ngIf="resultPickerModel.hasError('mdDatepickerMin')">Too early!</md-error>
        <md-error *ngIf="resultPickerModel.hasError('mdDatepickerMax')">Too late!</md-error>
        <md-error *ngIf="resultPickerModel.hasError('mdDatepickerFilter')">Date unavailable!</md-
error>
    </md-input-container>
    <md-datepicker
        #resultPicker
        [touchUi]="touch"
        [startAt]="startAt"
        [startView]="yearView ? 'year' : 'month'">
    </md-datepicker>
</p>
<p>
    <input [mdDatepicker]="resultPicker2"
        [(ngModel)]="date"
        [min]="minDate"
        [max]="maxDate"
        [mdDatepickerFilter]="filterOdd ? dateFilter : null"
        (focus)="resultPicker2.open()"
        placeholder="Pick a date"
        (keydown)="false">

```

```

<button [mdDatepickerToggle]="resultPicker2"></button>
<md-datepicker
  #resultPicker2
  [touchUi]="touch"
  [startAt]="startAt"
  [startView]="yearView ? 'year' : 'month'">
</md-datepicker>
</p>

```

datepicker-overview-example.ts:

```

import {Component, OnInit} from '@angular/core';
import {MdDatepicker} from '@angular/material';

@Component({
  selector: 'datepicker-overview-example',
  templateUrl: 'datepicker-overview-example.html'
})
export class DatepickerOverviewExample implements OnInit {

  touch: boolean;
  filterOdd: boolean;
  yearView: boolean;
  minDate: Date;
  maxDate: Date;
  startAt: Date;
  date: Date;
  dateFilter = (date: Date) => date.getMonth() % 2 == 1 && date.getDate() % 2 == 0;

}

```

[Demo en vivo](#)

Establecer diferente locale para md-datepicker

Este ejemplo requiere la importación de `DateAdapter` .

```

import {DateAdapter} from '@angular/material';

```

datepicker.component.html:

```

<md-input-container>
  <input mdInput [mdDatepicker]="picker" placeholder="Choose a date">
  <button mdSuffix [mdDatepickerToggle]="picker"></button>
</md-input-container>
<md-datepicker #picker></md-datepicker>

<p></p>
<div>
  <button md-raised-button (click)="setLocale('en')">English - US</button>

  <button md-raised-button (click)="setLocale('es')">Spanish</button>

  <button md-raised-button (click)="setLocale('zh')">Chinese</button>

  <button md-raised-button (click)="setLocale('nl')">Dutch</button>

```

```
<button md-raised-button (click)="setLocale('bn')">Bengali</button>

<button md-raised-button (click)="setLocale('hi')">Hindi</button>

<button md-raised-button (click)="setLocale('ar')">Arabic</button>
</div>
```

datepicker.component.ts:

```
import {Component} from '@angular/core';
import {DateAdapter} from '@angular/material';

@Component({
  selector: 'datepicker-overview-example',
  templateUrl: './datepicker-overview-example.html',
  styleUrls: ['./datepicker-overview-example.css'],
})
export class DatepickerOverviewExample {

  constructor(private dateAdapter: DateAdapter<Date>) {
    this.dateAdapter.setLocale('en');
  }

  setLocale(val) {
    console.log(val);
    this.dateAdapter.setLocale(val);
  }

}
```

[Demo en vivo](#)

Una lista de código de idioma de configuración regional se puede encontrar [aquí](#) .

Lea md-datepicker en línea: <https://riptutorial.com/es/angular-material2/topic/10876/md-datepicker>

Capítulo 6: md-dialog

Introducción

Este tema incluye ejemplos de `md-dialog`.

Observaciones

Para encontrar más detalles sobre `md-dialog`, consulte la documentación [aquí](#).

Examples

Inicializar md-dialog con los datos pasados desde el componente principal

Este ejemplo requiere `MdDialogRef` y `MD_DIALOG_DATA`. Por favor importarlos en el módulo de componentes.

```
import {MdDialog, MdDialogRef, MD_DIALOG_DATA} from '@angular/material';
```

input-overview-example.html:

```
<md-input-container>
  <input mdInput
    [(ngModel)]="id"
    placeholder="Value passed to md-dialog">
</md-input-container>

<p></p>

<button md-raised-button
  (click)="openDialog(id)">
  Open Dialog
</button>
```

input-overview-example.ts:

```
import {Component, Inject, Input, OnInit } from '@angular/core';
import {MdDialog, MdDialogRef, MD_DIALOG_DATA} from '@angular/material';

@Component({
  selector: 'input-overview-example',
  templateUrl: 'input-overview-example.html'
})
export class InputOverviewExample {

  id: any;

  @Input() isChecked: boolean;

  constructor(public dialog: MdDialog) {}
```

```

    openDialog(value) {
      let dialogRef = this.dialog.open(DialogResultExampleDialog, {
        data: {
          id: value
        }
      });
      dialogRef.afterClosed().subscribe(result => {
        console.log(result);
      });
    }
  }

@Component({
  selector: 'dialog-result-example-dialog',
  template: `<p md-dialog-title>Confirm Toggle </p>
    <p md-dialog-content>Id passed from component: {{ this.passedId }}</p>
    <md-dialog-actions>
      <button md-button color="primary"
(click)="dialogRef.close('Cancel') ">Cancel</button>
      <button md-button color="primary"
(click)="dialogRef.close('continue') ">Continue</button>
    </md-dialog-actions>
  `
})
export class DialogResultExampleDialog implements OnInit {

  passedId: string;

  constructor(@Inject(MD_DIALOG_DATA) private data: { id: string },
    public dialogRef: MdDialogRef<DialogResultExampleDialog>) {}

  ngOnInit(){
    this.passedId = this.data.id;
  }
}

```

[Demo en vivo](#)

Lea md-dialog en línea: <https://riptutorial.com/es/angular-material2/topic/10877/md-dialog>

Capítulo 7: md-table

Introducción

Este tema incluye ejemplos relacionados con md-table.

Observaciones

Para más detalles sobre `md-table` por favor verifique la [documentación](#).

Examples

Conecte DataSource desde una API externa

Por favor, tenga en cuenta la importación de todas las bibliotecas necesarias.

Este ejemplo utiliza `InMemoryDbService` de `angular-in-memory-web-api` para proporcionar los datos JSON de la API simulada.

[Demo en vivo](#)

servicio.ts:

```
import { Injectable } from '@angular/core';
import { Headers, Http } from '@angular/http';
import 'rxjs/add/operator/toPromise';

@Injectable()
export class AppState {

  private headers = new Headers({'Content-Type': 'application/json'});
  private apiUrl = 'api/data';

  constructor(private http: Http) { }

  fetchFilterFields() {
    console.log(this.apiUrl);
    return this.http.get(this.apiUrl)
      .toPromise()
      .then(response => response.json().data)
      .catch(this.handleError);
  }

  private handleError(error: any): Promise<any> {
    console.error('An error occurred', error); // for demo purposes only
    return Promise.reject(error.message || error);
  }
}
```

componente.ts:

```

import {Component} from '@angular/core';
import {DataSource} from '@angular/cdk';
import {BehaviorSubject} from 'rxjs/BehaviorSubject';
import {Observable} from 'rxjs/Observable';
import 'rxjs/add/observable/merge';

import { AppState } from './shared.service';

@Component({
  selector: 'md-table-example',
  templateUrl: 'select-form-example.html',
  styleUrls: ['select-form-example.css']
})
export class SelectFormExample implements OnInit {

  displayedColumns = ['id', 'name'];
  dataSource: ExampleDataSource | null;

  constructor(private appState: AppState){ }

  ngOnInit(){
    this.dataSource = new ExampleDataSource(this.appState);
  }

}

export interface UserData {
  id: string;
  name: string;
}

export class ExampleDataSource extends DataSource<any> {
  constructor(private appState: AppState) {
    super();
  }

  subject: BehaviorSubject<Hero[]> = new BehaviorSubject<Hero[]>([]);

  connect(): Observable<Hero[]> {
    console.log('connect');
    if (!this.subject.isStopped)
      this.appState.fetchFilterFields()
        .then(res => {
          this.subject.next(res)
        });
    return Observable.merge(this.subject);
  }

  disconnect() {
    this.subject.complete();
    this.subject.observers = [];
  }
}

```

component.html:

```

<h2> Material Table </h2>

<div *ngIf="dataSource" class="example-container mat-elevation-z8">
  <md-table #table [dataSource]="dataSource">

```

```

<!-- ID Column -->
<ng-container cdkColumnDef="id">
  <md-header-cell *cdkHeaderCellDef> ID </md-header-cell>
  <md-cell *cdkCellDef="let row"> {{row.id}} </md-cell>
</ng-container>

<!-- Name Column -->
<ng-container cdkColumnDef="name">
  <md-header-cell *cdkHeaderCellDef> Name </md-header-cell>
  <md-cell *cdkCellDef="let row"> {{row.name}} </md-cell>
</ng-container>

<md-header-row *cdkHeaderRowDef="displayedColumns"></md-header-row>
<md-row *cdkRowDef="let row; columns: displayedColumns;"></md-row>
</md-table>
</div>

```

Para referencia:

in-memory-data-service.ts:

```

import { InMemoryDbService } from 'angular-in-memory-web-api';
export class InMemoryDataService implements InMemoryDbService {
  createDb() {
    const data = [
      { id: 1, name: 'Ironcast' },
      { id: 2, name: 'Mr. Nice' },
      { id: 3, name: 'Narco' },
      { id: 4, name: 'Bombasto' },
      { id: 5, name: 'Celeritas' },
      { id: 6, name: 'Magneta' },
      { id: 7, name: 'RubberMan' },
      { id: 8, name: 'Dynamia' },
      { id: 9, name: 'Dr. IQ' },
      { id: 10, name: 'Magma' },
      { id: 11, name: 'Tornado' }
    ];
    return {data};
  }
}

```

Lea md-table en línea: <https://riptutorial.com/es/angular-material2/topic/10911/md-table>

Creditos

S. No	Capítulos	Contributors
1	Empezando con material angular2	Community , Edric , Maciej Treder , Nehal
2	botón md	Edric
3	icono-md	Edric
4	md-autocompletar	Nehal
5	md-datepicker	Nehal
6	md-dialog	Nehal
7	md-table	Nehal