eBook Gratuit

APPRENEZ angular-material2

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#angular-material2

Table des matières

À propos

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

Chapitre 1: Démarrer avec angular-material2

Remarques

Cette section fournit une vue d'ensemble de ce qu'est un matériau angulaire2 et pourquoi un développeur peut vouloir l'utiliser.

Il convient également de mentionner tous les sujets importants dans angular-material2, et de les relier aux sujets connexes. La documentation de angular-material2 étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

Version	Changelog	Rendez-vous amoureux
2.0.0-beta.8	Lien	2017-07-06
2.0.0-beta.7	Lien	2017-06-19
2.0.0-beta.6	Lien	2017-05-25
2.0.0-beta.5	Lien	2017-05-13
2.0.0-beta.4	Lien	2017-05-12
2.0.0-beta.3	Lien	2017-04-07
2.0.0-beta.2	Lien	2017-02-15
2.0.0-beta.1	Lien	2016-12-23
2.0.0-beta.0	Lien	2016-12-22

Examples

Installation ou configuration avec CLI angulaire

Dans cet exemple, nous utiliserons `@angular/cli` (latest) et la dernière version de `@angular/material` . Vous devez au moins connaître les bases de l'angulaire 2/4 avant de poursuivre les étapes ci-dessous.

1. Installer le module de matériau angulaire à partir de `npm` :

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

2.0.0-beta.3

Cela s'applique uniquement aux versions `2.0.0-beta.3` et supérieures.

Installez le module `@angular/animations` :

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

2.0.0-beta.8

Cela s'applique uniquement aux versions `2.0.0-beta.8` et supérieures.

Installez le module `@angular/cdk` :

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

2. Dans votre module d'application, importez les composants que vous allez utiliser:

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

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

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

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

Vous êtes maintenant prêt à utiliser Angular Material dans vos composants!

Note: Les documents pour des composants spécifiques seront bientôt disponibles.

Envelopper tous les modules ensemble

Vous pouvez également facilement envelopper tous les modules angulaires, que vous allez utiliser, dans un seul module:

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

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

Après cela, importez simplement votre module dans le module principal de l'application:

```
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 {}
```

Installation et configuration à partir de master avec Angular CLI

Cet exemple expliquera comment installer à partir de master et utilisera également `@angular/cli` .

1. Faire un nouveau projet avec `@angular/cli` :

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

   Si vous n'avez pas installé `@angular/cli` , utilisez cette commande:

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

2. Installer depuis le `master` :

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

3. Suivez le même guide que ci-dessus.

Terminé!

Définir des icônes de thème, de support gestuel et de matériel

---

Thème:

*Un thème est **requis** pour que les composants matériels fonctionnent correctement dans l'application.*

Le matériau angulaire 2 fournit quatre thèmes pré-construits:

- deeppurple-amber
- rose indigo
- rose-bleu-gris
- Violet vert

Si vous utilisez **Angular CLI** , vous pouvez importer l'un des thèmes `style.css` dans `style.css` .

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

Le thème peut également être ajouté en utilisant `<link>` dans `index.html` .

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

HammerJS

Ajoutez HammerJS à l'application via CDN ou `npm` :

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

Dans votre module racine, généralement `app.module.ts` , ajoutez l'instruction import:

```
import 'hammerjs';
```

Icônes matérielles:

Sauf si des icônes personnalisées sont fournies, le matériau angulaire 2 `<md-icon>` attend des icônes de matériau.

Dans `index.html` ajoutez:

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

Lire Démarrer avec angular-material2 en ligne: https://riptutorial.com/fr/angular-material2/topic/10828/demarrer-avec-angular-material2

Chapitre 2: bouton md

Introduction

Cette rubrique comprend des exemples sur la façon de créer un bouton et sur ses directives et autres éléments.

Paramètres

Attribut	La description
`md-button`	Crée un bouton rectangulaire sans élévation.
`md-raised-button`	Crée un bouton rectangulaire avec élévation
`md-icon-button`	Crée un bouton circulaire avec un arrière-plan transparent, destiné à contenir une icône
`md-fab`	Crée un bouton circulaire avec élévation, par défaut la couleur d'accent du thème
`md-mini-fab`	Comme `md-fab` mais plus petit
`disableRipple`	Si l'effet d'entraînement pour le bouton est désactivé.

Remarques

Pour plus d'informations et d'autres exemples, visitez le document .

Examples

Boutons simples

Pour créer un bouton, utilisez l'attribut `md-button` pour un bouton plat et le bouton `md-raised-button` pour un bouton élevé:

```
<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>
```

Démonstration Plunker

Pour plus d'informations sur les icônes, consultez la documentation sur `md-icon` .

Lire bouton md en ligne: https://riptutorial.com/fr/angular-material2/topic/10870/bouton-md

Chapitre 3: md-autocomplete

Introduction

Cette rubrique comprend des exemples de codage liés à la saisie semi-automatique du matériau angulaire 2 (md-autocomplete)

Remarques

Ces exemples ne couvrent pas toutes les fonctionnalités de md-autocomplete. Veuillez lire la documentation pour en savoir plus sur md-autocomplete.

Examples

Contrôle et affichage séparés

Cet exemple montre comment afficher une propriété spécifique dans la liste déroulante mais se lie à l'objet entier.

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;
   }

}
```

Exemple Live

Obtenir les options de md-autocomplete / données de recherche de l'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>
```

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

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

<div>
    <h2>Data :</h2>
    <span>{{ allSectors | json  }}</span>
</div>
```

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;
   }

}
```

Exemple Live

Utiliser md-autocomplete dans un formulaire réactif

Cet exemple nécessite `FormsModule` et `ReactiveFormsModule` . Veuillez les importer dans votre application / module.

```
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%;
}
```

Exemple Live

Un md-autocomplete pour plusieurs formControl

Cet exemple nécessite `FormsModule` et `ReactiveFormsModule` . Veuillez les importer dans votre application / module.

```
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()) === 0)
               : this.states;
  }

}
```

Exemple Live

Lire md-autocomplete en ligne: https://riptutorial.com/fr/angular-material2/topic/10850/md-autocomplete

Chapitre 4: md-datepicker

Introduction

Cette rubrique se concentre sur des exemples liés à md-datepicker.

Remarques

Pour plus de détails, consultez la documentation de `md-datepicker` ici .

Examples

Liaison de données avec 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();
  }

}
```

Démo en direct

Passer la valeur de date sélectionnée à une fonction en utilisant $ 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;
  }
}
```

Démo en direct

Ouvrir datepicker sur focus

Cet exemple inclut également l'utilisation des propriétés:

- min
- max
- commencer à
- 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;

}
```

Démo en direct

Définir des paramètres régionaux différents pour md-datepicker

Cet exemple nécessite l'importation 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);
  }

}
```

Démo en direct

Une liste de code de langue locale peut être trouvée ici .

Lire md-datepicker en ligne: https://riptutorial.com/fr/angular-material2/topic/10876/md-datepicker

Chapitre 5: md-dialog

Introduction

Cette rubrique comprend des exemples de `md-dialog` .

Remarques

Pour plus de détails sur `md-dialog` , consultez la documentation ici .

Examples

Initialiser md-dialog avec les données transmises par le composant parent

Cet exemple nécessite `MdDialogRef` et `MD_DIALOG_DATA` . Veuillez les importer dans le module composant.

```
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;
  }
}
```

Démo en direct

Lire md-dialog en ligne: https://riptutorial.com/fr/angular-material2/topic/10877/md-dialog

Chapitre 6: md-icon

Examples

Créer une icône

Ce qui suit est un guide sur la façon de créer une icône à partir des icônes de conception matérielle:

1. Chargez la police d'icône de Google CDN dans votre `index.html` :

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

   Alternativement, vous pouvez l'importer dans votre `styles.css` :

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

2. Utilisez-le comme suit:

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

Vous avez terminé!

Utiliser des icônes SVG

Cet exemple montre comment utiliser les icônes SVG dans votre application.

1. Téléchargez le icone / icone SVG (dans ce cas, nous obtenons les icônes de https://materialdesignicons.com/getting-started) .

2. Enregistrez-le sous votre dossier de `assets` ou ailleurs, avec lequel vous pouvez accéder.

3. Dans votre `app.module.ts` , ajoutez le code suivant:

   ```
   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. Utilisez-le via l'attribut `svgIcon` :

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

Lire md-icon en ligne: https://riptutorial.com/fr/angular-material2/topic/10868/md-icon

Chapitre 7: md-table

Introduction

Cette rubrique comprend des exemples liés à md-table

Remarques

Pour plus de détails sur `md-table` veuillez consulter la documentation

Examples

Connecter DataSource à partir d'une API externe

S'il vous plaît, faites attention à importer toutes les bibliothèques nécessaires.

Cet exemple utilise `InMemoryDbService` d' `angular-in-memory-web-api` pour fournir les données JSON à partir de l'API simulée.

Démo en direct

service.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);
  }

}
```

composant.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>
```

Pour référence:

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: 'Dynama' },
      { id: 9, name: 'Dr. IQ' },
      { id: 10, name: 'Magma' },
      { id: 11, name: 'Tornado' }
    ];
    return {data};
  }
}
```

Lire md-table en ligne: https://riptutorial.com/fr/angular-material2/topic/10911/md-table

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec angular-material2	Community, Edric, Maciej Treder, Nehal
2	bouton md	Edric
3	md-autocomplete	Nehal
4	md-datepicker	Nehal
5	md-dialog	Nehal
6	md-icon	Edric
7	md-table	Nehal