Material UI Example

Ng new ang-mat-app

ng add @angular/material

ng g c reactive-mat-demo

We will create a separate module to configure Material related modules in a single .ts file for a cleaner implementation. Create a file name material.module.ts under src/app/core folder and the below code:

Material.module.ts

import { NgModule } from '@angular/core';

import { CommonModule } from '@angular/common';

import { MatBadgeModule } from '@angular/material/badge';

import { MatButtonModule } from '@angular/material/button';

import { MatChipsModule } from '@angular/material/chips';

import { MatNativeDateModule } from '@angular/material/core';

import { MatDatepickerModule } from '@angular/material/datepicker';

import { MatFormFieldModule } from '@angular/material/form-field';

import { MatGridListModule } from '@angular/material/grid-list';

import { MatIconModule } from '@angular/material/icon';

import { MatInputModule } from '@angular/material/input';

import { MatListModule } from '@angular/material/list';

import { MatPaginatorModule } from '@angular/material/paginator';

import { MatRadioModule } from '@angular/material/radio';

import { MatSelectModule } from '@angular/material/select';

import { MatSidenavModule } from '@angular/material/sidenav';

import { MatTableModule } from '@angular/material/table';

import { MatToolbarModule } from '@angular/material/toolbar';

import { MatTooltipModule } from '@angular/material/tooltip';

@NgModule({

   imports: [

      CommonModule,

      MatButtonModule,

      MatToolbarModule,

      MatIconModule,

      MatSidenavModule,

      MatBadgeModule,

      MatListModule,

      MatGridListModule,

      MatFormFieldModule,

      MatInputModule,

      MatSelectModule,

      MatRadioModule,

      MatDatepickerModule,

      MatNativeDateModule,

      MatChipsModule,

      MatTooltipModule,

      MatTableModule,

      MatPaginatorModule

   ],

   exports: [

      MatButtonModule,

      MatToolbarModule,

      MatIconModule,

      MatSidenavModule,

      MatBadgeModule,

      MatListModule,

      MatGridListModule,

      MatInputModule,

      MatFormFieldModule,

      MatSelectModule,

      MatRadioModule,

      MatDatepickerModule,

      MatChipsModule,

      MatTooltipModule,

      MatTableModule,

      MatPaginatorModule

   ],

   providers: [

      MatDatepickerModule,

   ]

})

export class AngularMaterialModule { }

app.module.ts

import { NgModule } from '@angular/core';

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

import { AngularMaterialModule } from './core/material.module';

import { AppRoutingModule } from './app-routing.module';

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

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

import { ReactiveMatDemoComponent } from './reactive-mat-demo/reactive-mat-demo.component';

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

import { HttpClientModule } from '@angular/common/http';

import { ArticleListComponent } from './article-list/article-list.component';

@NgModule({

  declarations: [

    AppComponent,

    ReactiveMatDemoComponent,

    ArticleListComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    AngularMaterialModule,

    FormsModule,

    ReactiveFormsModule,

    HttpClientModule

  ],

  providers: [

    provideAnimationsAsync()

  ],

  bootstrap: [AppComponent]

})

export class AppModule { }

ng g c reactive-mat-demo

reactive-mat-demo.component.ts

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

import { FormGroup,FormBuilder,Validators } from '@angular/forms';

import { MatChipInputEvent } from '@angular/material/chips';

import {COMMA,ENTER} from '@angular/cdk/keycodes';

@Component({

  selector: 'app-reactive-mat-demo',

  templateUrl: './reactive-mat-demo.component.html',

  styleUrl: './reactive-mat-demo.component.scss'

})

export class ReactiveMatDemoComponent {

  myForm!:FormGroup;

  selectable=true;

  removable=true;

  addOnBlur=true;

  SubjectArray:Subject []=[];

  GradeArray:any=[

    '1st Grade',

    '2nd Grade',

    '3rd Grade',

    '4th Grade',

    '5th Grade'

  ]

  @ViewChild('chipList',{static:true}) chipList: any;

  readonly seperatorKeysCodes:number[]=[ENTER,COMMA];

  constructor(public fb:FormBuilder){}

  ngOnInit(){

    this.reactiveForm();

  }

  reactiveForm(){

    this.myForm=this.fb.group({

      name:['',Validators.required],

      email:['',Validators.required],

      gender:['Male'],

      dob:['',Validators.required],

      grade:[''],

      subjects:[this.SubjectArray]

    })

  }

  date(e:any)

  {

    var convertDate=new Date(e.target.value).toISOString().substring(0,10);

    this.myForm.get('dob')?.setValue(convertDate,{

      onlyself:true,

    });

  }

  add(event:MatChipInputEvent):void{

    const input=event.input;

    const value=event.value;

    //Add Language

    if(value ||''.trim()&& this.SubjectArray.length<5){

      this.SubjectArray.push({name:value.trim()});

    }

    //Reset the Input value

    if(input){

      input.value='';

    }

  }

  //Remove Dynamic languages

  remove(subject:Subject):void{

  const index=this.SubjectArray.indexOf(subject);

  if(index>=0){

    this.SubjectArray.splice(index,1)

  }

  }

  submitForm(){

console.log(this.myForm.value)

  }

}

export interface Subject

{

  name:string;

}

Reactive.component.html

<!-- <form [formGroup]="myForm" (ngSubmit)="submitForm()" novalidate>

    <mat-form-field class="example-full-width">

        <input matInput placeholder="Name" formControlName="Name"/>

    </mat-form-field>

</form> -->

<div class="container">

<mat-sidenav-container>

    <mat-sidenav-content>

        <form [formGroup]="myForm" (ngSubmit)="submitForm()" novalidate>

            <div class="row">

            <!-- Name -->

            <mat-form-field class="example-full-width">

                <input matInput placeholder="Name" formControlName="name">

            </mat-form-field>

        </div>

            <!-- Email -->

            <div class="row">

            <mat-form-field class="example-full-width">

                <input matInput placeholder="email" formControlName="email"/>

            </mat-form-field>

            </div>

            <!-- Gender -->

            <div class="row">

            <div class="misc-bottom-padding">

                <mat-label>Gender:</mat-label>

                <mat-radio-group aria-label="Select an option" formControlName="gender">

                    <mat-radio-button value="Male">Male</mat-radio-button>

                    <mat-radio-button value="Female">Female</mat-radio-button>

                </mat-radio-group>

            </div>

            </div>

            <!-- Date Picker-->

            <div class="row">

            <mat-form-field>

                <input matInput readonly [matDatepicker]="picker" placeholder="Date of Birth"

                formControlName="dob" (dateChange)="date($event)" />

                <mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>

                <mat-datepicker #picker></mat-datepicker>

            </mat-form-field>

            </div>

            <!-- class -->

            <div class="row">

            <mat-form-field>

                <mat-label>Grade</mat-label>

                <mat-select formControlName="grade">

                    <mat-option [value]="gradeArray" \*ngFor="let gradeArray of GradeArray">

                        {{gradeArray}}</mat-option>

                </mat-select>

            </mat-form-field>

            </div>

            <!-- Add Class-->

            <div>

            <mat-form-field class="multiple-items">

                <mat-chip-grid #chipList>

                    <mat-chip \*ngFor="let subjectsArray of SubjectArray"

                     selectable="selectable"

                   [removable]="removable" (removed)="remove(subjectsArray)" >

                {{subjectsArray.name}}

                <mat-icon matChipRemove \*ngIf="removable">cancel</mat-icon>

            </mat-chip>

            <input

            placeholder="Add subjects"

            [matChipInputFor]="chipList"

            [matChipInputSeparatorKeyCodes]="seperatorKeysCodes"

            [matChipInputAddOnBlur]="addOnBlur"

            (matChipInputTokenEnd)="add($event)"

          />

                </mat-chip-grid>

            </mat-form-field>

        </div>

            <!-- Submit -->

        <div class="row">

      <div class="button-wrapper">

        <button mat-flat-button color="primary">Submit</button>

      </div>

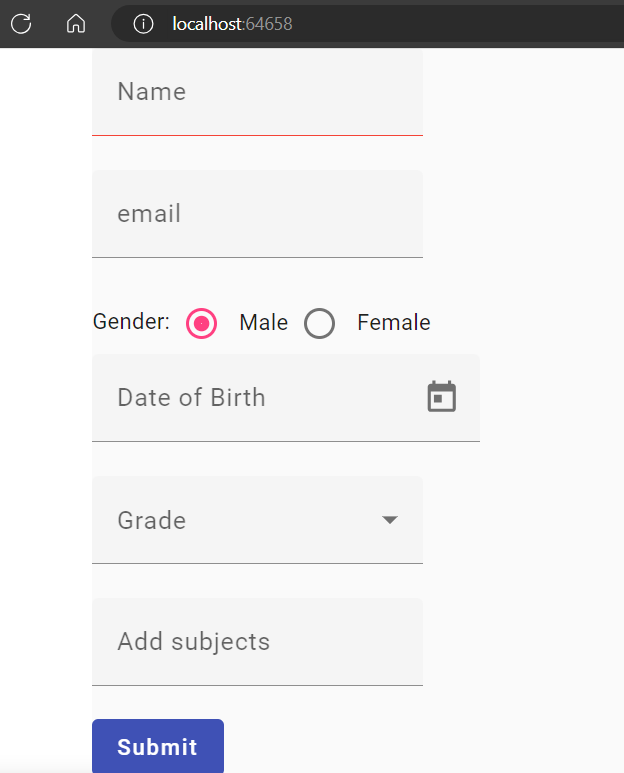
    </div>

        </form>

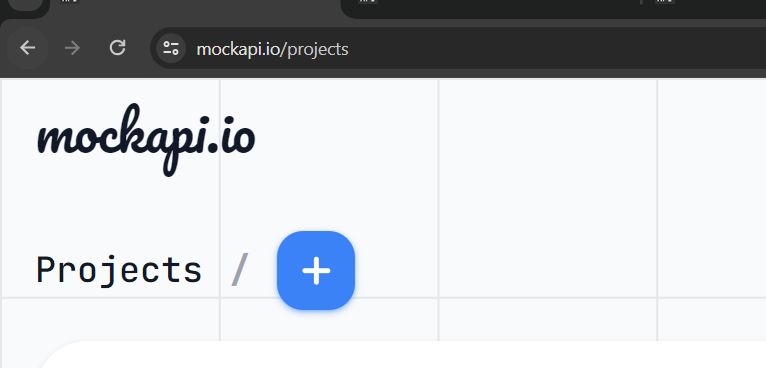
    </mat-sidenav-content>

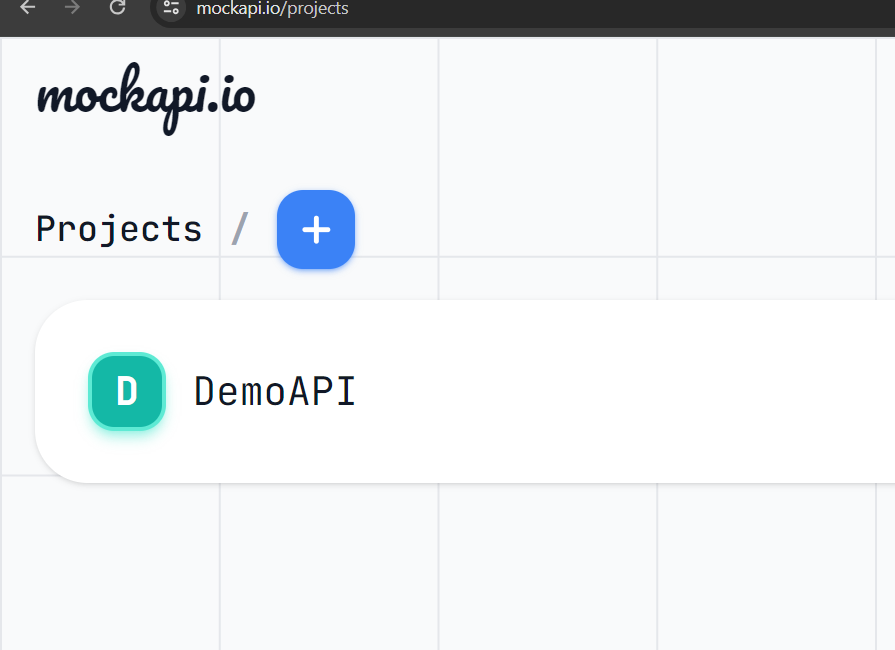
</mat-sidenav-container>

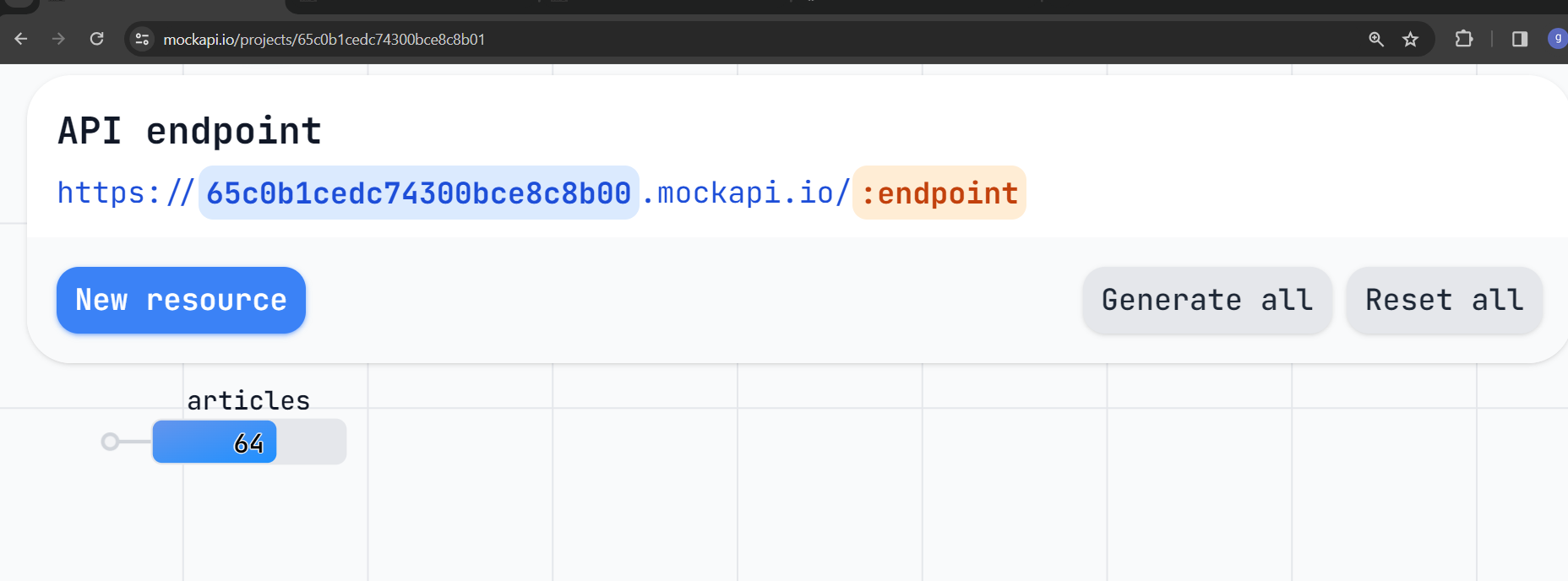
</div>



Fetch Data from any open API







<https://65c0b1cedc74300bce8c8b00.mockapi.io/articles>



Ng g s shared/article

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

@Injectable({

  providedIn: 'root'

})

export class ArticleService {

  constructor(private httpClient:HttpClient) { }

  apiURL="https://65c0b1cedc74300bce8c8b00.mockapi.io/articles";

  getAllArticles(){

    return this.httpClient.get(this.apiURL)

  }

}

ng g c article-list

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

import { ArticleService } from '../shared/article.service';

import { Article } from '../article';

@Component({

  selector: 'app-article-list',

  templateUrl: './article-list.component.html',

  styleUrl: './article-list.component.scss',

})

export class ArticleListComponent {

  allArticle!: any;

  constructor(private articleService: ArticleService) {}

  ngOnInit() {

    this.fetchArticles();

  }

  displayedColumns: string[] = ['createdAt', 'name', 'avatar', 'id'];

  dataSource = this.allArticle;

  fetchArticles() {

    this.articleService.getAllArticles().subscribe((data) => {

      this.allArticle = data;

      console.log(this.allArticle);

    });

  }

}

Article-list.component.html

<table mat-table [dataSource]="allArticle" class="mat-elevation-z8">

    <!--- Note that these columns can be defined in any order.

          The actual rendered columns are set as a property on the row definition" -->

    <!-- Position Column -->

    <ng-container matColumnDef="id">

      <th mat-header-cell \*matHeaderCellDef> No. </th>

      <td mat-cell \*matCellDef="let element"> {{element.id}} </td>

    </ng-container>

    <!-- Name Column -->

    <ng-container matColumnDef="name">

      <th mat-header-cell \*matHeaderCellDef> Name </th>

      <td mat-cell \*matCellDef="let element"> {{element.name}} </td>

    </ng-container>

    <!-- Weight Column -->

    <ng-container matColumnDef="createdAt">

      <th mat-header-cell \*matHeaderCellDef> CreatedAt </th>

      <td mat-cell \*matCellDef="let element"> {{element.createdAt}} </td>

    </ng-container>

    <!-- Symbol Column -->

    <ng-container matColumnDef="avatar">

      <th mat-header-cell \*matHeaderCellDef> Avatar </th>

      <td mat-cell \*matCellDef="let element">

        <img src="{{element.avatar}}" height="100px" width="100px"> </td>

    </ng-container>

    <tr mat-header-row \*matHeaderRowDef="displayedColumns"></tr>

    <tr mat-row \*matRowDef="let row; columns: displayedColumns;"></tr>

  </table>

