**var** mongoose = require('mongoose');  
  
**var** CustomerSchema = **new** mongoose.Schema({  
 customer\_id: String,  
 customer\_name: String,  
 customer\_email: String,  
 },  
 {collection:'lab7book'}  
);  
  
**const** Customer = mongoose.model('lab7book',CustomerSchema);  
module.exports = Customer;

**var** express = require('express');  
**var** router = express.Router();  
**var** Customer = require('../models/customer.js');  
  
/\* GET ALL CUSTOMERS \*/  
router.get('/', **function** (req, res, next) {  
 Customer.find(**function** (err, products) {  
 **if** (err) **return** next(err);  
 res.json(products);  
 });  
});  
  
/\* GET SINGLE CUSTOMER BY ID \*/  
router.get('/:id', **function** (req, res, next) {  
 Customer.findById(req.params.id, **function** (err, post) {  
 **if** (err) **return** next(err);  
 res.json(post);  
 });  
});  
  
/\* SAVE CUSTOMER \*/  
router.post('/', **function** (req, res, next) {  
 Customer.create(req.body, **function** (err, post) {  
 **if** (err) **return** next(err);  
 res.json(post);  
 });  
});  
  
/\* UPDATE CUSTOMER \*/  
router.put('/:id', **function**(req, res, next){  
 Customer.findByIdAndUpdate(req.params.id, req.body, **function** (err,post){  
 **if** (err) **return** next(err);  
 res.json(post);  
 })  
})  
/\* DELETE CUSTOMER \*/  
router.delete('/:id', **function**(req, res, next){  
 Customer.findByIdAndDelete(req.params.id, **function** (err,post){  
 **if** (err) **return** next(err);  
 res.json(post);  
 })  
})  
module.exports = router;

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/customer-create']">  
 <mat-icon>add</mat-icon>  
 </a>  
</div>  
<div class="example-container mat-elevation-z8">  
 <table mat-table #table [dataSource]="dataSource">  
  
 <!--- Note that these columns can be defined in any order.  
 The actual rendered columns are set as a property on the row definition" -->  
  
 <!-- Title Column -->  
 <ng-container matColumnDef="customer\_id">  
 <th mat-header-cell *\*matHeaderCellDef*>Customer ID</th>  
 <td mat-cell *\*matCellDef*="**let** element" class="id-col"> {{element.customer\_id}} </td>  
 </ng-container>  
  
 <!-- Title Column -->  
 <ng-container matColumnDef="customer\_name">  
 <th mat-header-cell *\*matHeaderCellDef*> Name</th>  
 <td mat-cell *\*matCellDef*="**let** element"> {{element.customer\_name}} </td>  
 </ng-container>  
  
 <!-- Author Column -->  
 <ng-container matColumnDef="customer\_email">  
 <th mat-header-cell *\*matHeaderCellDef*> Email</th>  
 <td mat-cell *\*matCellDef*="**let** element"> {{element.customer\_email}} </td>  
 </ng-container>  
  
 <tr mat-header-row *\*matHeaderRowDef*="displayedColumns"></tr>  
 <tr mat-row *\*matRowDef*="**let** row; columns: displayedColumns;" [routerLink]="['/customer-details/', row.\_id]"></tr>  
 </table>  
</div>

**import** { Component, OnInit } **from** '@angular/core';  
**import** { ApiService } **from** '../api.service';  
**import** { DataSource } **from** '@angular/cdk/collections';  
**import** { Observable } **from** 'rxjs';  
  
@Component({  
 selector: 'app-customer',  
 templateUrl: './customer.component.html',  
 styleUrls: ['./customer.component.css']  
})  
**export class** CustomerComponent **implements** OnInit {  
  
 customers: **any**;  
 displayedColumns = ['customer\_id', 'customer\_name', 'customer\_email'];  
 dataSource = **new** CustomerDataSource(**this**.api);  
  
 **constructor**(**private** api: ApiService) { }  
  
 ngOnInit() {  
 **this**.api.getCustomers()  
 .subscribe(res => {  
 console.log('RES' + res);  
 **this**.customers = res;  
 }, err => {  
 console.log(err);  
 });  
 }  
}  
  
**export class** CustomerDataSource **extends** DataSource<**any**> {  
 **constructor**(**private** api: ApiService) {  
 **super**();  
 }  
  
 connect() {  
 **return this**.api.getCustomers();  
 }  
  
 disconnect() {  
  
 }  
}

.example-container {  
 display: flex;  
 flex-direction: column;  
 max-height: 500px;  
 min-width: 300px;  
 overflow: auto;  
}  
  
.id-col {  
 flex: 0 0 100px **!important**;  
 white-space: unset **!important**;  
}  
  
.button-row {  
 margin: 10px 0;  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/customers']">  
 <mat-icon>list</mat-icon>  
 </a>  
</div>  
<form [formGroup]="customerForm" (ngSubmit)="onFormSubmit(customerForm.value)">  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Customer ID" formControlName="customer\_id"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_id').valid && customerForm.get('customer\_id').touched">Please enter Customer ID</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Customer Name" formControlName="customer\_name"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_name').valid && customerForm.get('customer\_name').touched">Please enter Customer Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Customer Email" formControlName="customer\_email"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_email').valid && customerForm.get('customer\_email').touched">Please enter Customer Email</span>  
 </mat-error>  
 </mat-form-field>  
 <div class="button-row">  
 <button type="submit" [disabled]="!customerForm.valid" mat-raised-button color="primary">  
 <mat-icon>save</mat-icon>  
 </button>  
 </div>  
</form>

.example-form {  
 min-width: 150px;  
 max-width: 500px;  
 width: 100%;  
}  
  
.example-full-width {  
 width: 100%;  
}  
  
.example-full-width:nth-last-child() {  
 margin-bottom: 10px;  
}  
  
.button-row {  
 margin: 10px 0;  
}

**import** { Component, OnInit } **from** '@angular/core';  
**import** { Router } **from** '@angular/router';  
**import** { ApiService } **from** '../api.service';  
**import** { FormControl, FormGroupDirective, FormBuilder, FormGroup, NgForm, Validators } **from** '@angular/forms';  
  
@Component({  
 selector: 'app-customer-create',  
 templateUrl: './customer-create.component.html',  
 styleUrls: ['./customer-create.component.css']  
})  
**export class** CustomerCreateComponent **implements** OnInit {  
  
 customerForm: FormGroup;  
 customer\_id: **string** = '';  
 customer\_name: **string** = '';  
 customer\_email: **string** = '';  
  
 **constructor**(**private** router: Router, **private** api: ApiService, **private** formBuilder: FormBuilder) { }  
  
 ngOnInit() {  
 **this**.customerForm = **this**.formBuilder.group({  
 'customer\_id': [**null**, Validators.*required*],  
 'customer\_name': [**null**, Validators.*required*],  
 'customer\_email': [**null**, Validators.*required*]  
 });  
 }  
  
  
 onFormSubmit(form: NgForm) {  
 **this**.api.postCustomer(form)  
 .subscribe(res => {  
 **let** id = res['\_id'];  
 **this**.router.navigate(['/customer-details', id]);  
 }, (err) => {  
 console.log(err);  
 });  
 }  
  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/customers']"><mat-icon>list</mat-icon></a>  
</div>  
<mat-card class="example-card">  
 <mat-card-header>  
 <mat-card-title><h2>{{customer.customer\_id}}</h2></mat-card-title>  
 </mat-card-header>  
 <mat-card-content>  
 <dl>  
 <dt><h4>Phone Number:</h4> {{customer.customer\_name}}</dt>  
 <dt><h4>Email:</h4> {{customer.customer\_email}}</dt>  
 </dl>  
 </mat-card-content>  
 <mat-card-actions>  
 <a mat-raised-button color="primary" [routerLink]="['/customer-edit', customer.\_id]"><mat-icon>edit</mat-icon></a>  
 <a mat-raised-button color="warn" (click)="deleteCustomer(customer.\_id)"><mat-icon>delete</mat-icon></a>  
 </mat-card-actions>  
</mat-card>

.example-card {  
 max-width: 500px;  
}  
  
.button-row {  
 margin: 10px 0;  
}

**import** { Component, OnInit } **from** '@angular/core';  
**import** { ApiService } **from** '../api.service';  
**import** { ActivatedRoute, Router } **from** '@angular/router';  
  
@Component({  
 selector: 'app-customer-detail',  
 templateUrl: './customer-detail.component.html',  
 styleUrls: ['./customer-detail.component.css']  
})  
**export class** CustomerDetailComponent **implements** OnInit {  
  
 customer = {};  
  
 **constructor**(**private** route: ActivatedRoute, **private** api: ApiService, **private** router: Router) { }  
  
 ngOnInit() {  
 **this**.getCustomerDetails(**this**.route.snapshot.params['id']);  
 }  
  
 getCustomerDetails(id) {  
 **this**.api.getCustomer(id)  
 .subscribe(data => {  
 console.log(data);  
 **this**.customer = data;  
 });  
 }  
  
 deleteCustomer(id) {  
 **this**.api.deleteCustomer(id)  
 .subscribe(res => {  
 **this**.router.navigate(['/customers']);  
 }, (err) => {  
 console.log(err);  
 }  
 );  
 }  
  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/customers']">  
 <mat-icon>list</mat-icon>  
 </a>  
</div>  
<form [formGroup]="customerForm" (ngSubmit)="onFormSubmit(customerForm.value)">  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="ID" formControlName="customer\_id"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_id').valid && customerForm.get('customer\_id').touched">Please enter ID</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Name" formControlName="customer\_name"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_name').valid && customerForm.get('customer\_name').touched">Please enter Customer Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Email Address" formControlName="customer\_email"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!customerForm.get('customer\_email').valid && customerForm.get('customer\_email').touched">Please enter Email Address</span>  
 </mat-error>  
 </mat-form-field>  
 <div class="button-row">  
 <button type="submit" [disabled]="!customerForm.valid" mat-raised-button color="primary">  
 <mat-icon>save</mat-icon>  
 </button>  
 </div>  
</form>

.example-form {  
 min-width: 150px;  
 max-width: 500px;  
 width: 100%;  
}  
  
.example-full-width {  
 width: 100%;  
}  
  
.example-full-width:nth-last-child() {  
 margin-bottom: 10px;  
}  
  
.button-row {  
 margin: 10px 0;  
}

**import** { Component, OnInit } **from** '@angular/core';  
**import** { Router, ActivatedRoute } **from** '@angular/router';  
**import** { ApiService } **from** '../api.service';  
**import** { FormControl, FormGroupDirective, FormBuilder, FormGroup, NgForm, Validators } **from** '@angular/forms';  
  
@Component({  
 selector: 'app-customer-edit',  
 templateUrl: './customer-edit.component.html',  
 styleUrls: ['./customer-edit.component.css']  
})  
**export class** CustomerEditComponent **implements** OnInit {  
  
 customerForm: FormGroup;  
 id: **string** = '';  
 customer\_id: **string** = '';  
 customer\_name: **string** = '';  
 customer\_email: **string** = '';  
  
 **constructor**(**private** router: Router, **private** route: ActivatedRoute, **private** api: ApiService, **private** formBuilder: FormBuilder) {  
 }  
 ngOnInit() {  
 **this**.getCustomer(**this**.route.snapshot.params['id']);  
 **this**.customerForm = **this**.formBuilder.group({  
 'customer\_id': [**null**, Validators.*required*],  
 'customer\_name': [**null**, Validators.*required*],  
 'customer\_email': [**null**, Validators.*required*]  
 });  
 }  
  
 onFormSubmit(form: NgForm) {  
 **let** id = **this**.route.snapshot.params['id'];  
 console.log(form)  
 **this**.api.updateCustomer(id, form)  
 .subscribe(res => {  
 **this**.router.navigate(['/customer-details', id]);  
 }, (err) => {  
 console.log(err);  
 });  
 }  
  
 getCustomer(id) {  
 **this**.api.getCustomer(id).subscribe(data => {  
 id = data.\_id;  
 **this**.customerForm.setValue({  
 customer\_id: data.customer\_id,  
 customer\_name: data.customer\_name,  
 customer\_email: data.customer\_email,  
 });  
 });  
 }  
}

**import** { Injectable } **from** '@angular/core';  
**import** { Observable, of, throwError } **from** 'rxjs';  
**import** { HttpClient, HttpHeaders, HttpErrorResponse } **from** '@angular/common/http';  
**import** { catchError, tap, map } **from** 'rxjs/operators';  
  
**const** httpOptions = {  
 headers: **new** HttpHeaders({'Content-Type': 'application/json'})  
};  
**const** apiUrl = "/api";  
  
@Injectable({  
 providedIn: 'root'  
})  
**export class** ApiService {  
  
 **constructor**(**private** http: HttpClient) { }  
  
 **private** handleError(error: HttpErrorResponse) {  
 **if** (error.error **instanceof** ErrorEvent) {  
 // A client-side or network error occurred. Handle it accordingly.  
 console.error('An error occurred:', error.error.message);  
 } **else** {  
 // The backend returned an unsuccessful response code.  
 // The response body may contain clues as to what went wrong,  
 console.error(  
 `Backend returned code ${error.status}, ` +  
 `body was: ${error.error}`);  
 }  
 // return an observable with a user-facing error message  
 **return** throwError('Something bad happened; please try again later.');  
 };  
  
 **private** extractData(res: Response) {  
 **let** body = res;  
 **return** body || { };  
 }  
  
 getCustomers(): Observable<**any**> {  
 **return this**.http.get(apiUrl, httpOptions).pipe(  
 map(**this**.extractData),  
 catchError(**this**.handleError));  
 }  
  
 getCustomer(id: **string**): Observable<**any**> {  
 **const** url = `${apiUrl}/${id}`;  
 **return this**.http.get(url, httpOptions).pipe(  
 map(**this**.extractData),  
 catchError(**this**.handleError));  
 }  
  
 postCustomer(data): Observable<**any**> {  
 **return this**.http.post(apiUrl, data, httpOptions)  
 .pipe(  
 catchError(**this**.handleError)  
 );  
 }  
  
 updateCustomer(id: **string**, data): Observable<**any**> {  
 **const** url = `${apiUrl}/${id}`;  
 **return this**.http.put(url, data, httpOptions)  
 .pipe(  
 catchError(**this**.handleError)  
 );  
 }  
  
 deleteCustomer(id: **string**): Observable<{}> {  
 **const** url = `${apiUrl}/${id}`;  
 **return this**.http.delete(url, httpOptions)  
 .pipe(  
 catchError(**this**.handleError)  
 );  
 }  
  
}

**import** { BrowserModule } **from** '@angular/platform-browser';  
**import** { NgModule } **from** '@angular/core';  
**import** { RouterModule, Routes } **from** '@angular/router';  
**import** { FormsModule, ReactiveFormsModule } **from** '@angular/forms';  
**import** { HttpClientModule } **from** '@angular/common/http';  
  
**import** { AppComponent } **from** './app.component';  
**import** { CustomerComponent } **from** './customer/customer.component';  
**import** { CustomerDetailComponent } **from** './customer-detail/customer-detail.component';  
**import** { CustomerCreateComponent } **from** './customer-create/customer-create.component';  
**import** { CustomerEditComponent } **from** './customer-edit/customer-edit.component';  
**import** { BrowserAnimationsModule } **from** '@angular/platform-browser/animations';  
  
**import** {  
 MatInputModule,  
 MatPaginatorModule,  
 MatProgressSpinnerModule,  
 MatSortModule,  
 MatTableModule,  
 MatIconModule,  
 MatButtonModule,  
 MatCardModule,  
 MatFormFieldModule } **from** '@angular/material';  
  
**const** appRoutes: Routes = [  
 {  
 path: 'customers',  
 component: CustomerComponent,  
 data: { title: 'Customer List' }  
 },  
 {  
 path: 'customer-details/:id',  
 component: CustomerDetailComponent,  
 data: { title: 'Customer Details' }  
 },  
 {  
 path: 'customer-create',  
 component: CustomerCreateComponent,  
 data: { title: 'Create Customer' }  
 },  
 {  
 path: 'customer-edit/:id',  
 component: CustomerEditComponent,  
 data: { title: 'Edit Customer' }  
 },  
 { path: '',  
 redirectTo: '/customers',  
 pathMatch: 'full'  
 }  
];  
  
@NgModule({  
 declarations: [  
 AppComponent,  
 CustomerComponent,  
 CustomerDetailComponent,  
 CustomerCreateComponent,  
 CustomerEditComponent  
 ],  
 imports: [  
 RouterModule.*forRoot*(appRoutes),  
 BrowserModule,  
 FormsModule,  
 ReactiveFormsModule,  
 HttpClientModule,  
 BrowserAnimationsModule,  
 MatInputModule,  
 MatTableModule,  
 MatPaginatorModule,  
 MatSortModule,  
 MatProgressSpinnerModule,  
 MatIconModule,  
 MatButtonModule,  
 MatCardModule,  
 MatFormFieldModule  
 ],  
 providers: [],  
 bootstrap: [AppComponent]  
})  
**export class** AppModule { }

**var** createError = require('http-errors');  
**var** express = require('express');  
**var** path = require('path');  
**var** favicon = require('serve-favicon');  
**var** logger = require('morgan');  
  
**var** mongoose = require('mongoose');  
mongoose.connect('mongodb+srv://lab7:lab7@cluster0-ydpjq.mongodb.net/book?retryWrites=true')  
 .then(() => console.log('connection successful'))  
 .catch((err) => console.error(err));  
  
**var** apiRouter = require('./routes/customer');  
  
**var** app = express();  
  
app.use(logger('dev'));  
app.use(express.json());  
app.use(express.urlencoded({extended: **false**}));  
app.use(express.static(path.join(\_\_dirname, 'dist/lab7')));  
app.use('/customers', express.static(path.join(\_\_dirname, 'dist/lab7')));  
app.use('/customer-details/:id', express.static(path.join(\_\_dirname, 'dist/lab7')));  
app.use('/customer-create', express.static(path.join(\_\_dirname, 'dist/lab7')));  
app.use('/customer-edit/:id', express.static(path.join(\_\_dirname, 'dist/lab7')));  
app.use('/api', apiRouter);  
  
// catch 404 and forward to error handler  
app.use(**function** (req, res, next) {  
 next(createError(404));  
});  
  
// error handler  
app.use(**function** (err, req, res, next) {  
 // set locals, only providing error in development  
 res.locals.message = err.message;  
 res.locals.error = req.app.get('env') === 'development' ? err : {};  
  
 // render the error page  
 res.status(err.status || 500);  
 res.send(err.status);  
});  
  
module.exports = app;