**var** mongoose = require('mongoose');  
  
**var** GameSchema = **new** mongoose.Schema({  
 game\_name: String,  
 team: String,  
 organizer: String,  
 game\_loc: String,  
 game\_date: String,  
 game\_time: String,  
 game\_duration: String,  
 contact\_org: String,  
 updated\_date: { type: Date, **default**: Date.now },  
});  
  
module.exports = mongoose.model('Game1', GameSchema);

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

**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) {  
 **const** body = res;  
 **return** body || { };  
 }  
  
 getGames(): Observable<**any**> {  
 **return this**.http.get(apiUrl, httpOptions).pipe(  
 map(**this**.extractData),  
 catchError(**this**.handleError));  
 }  
  
 getGame(id: **string**): Observable<**any**> {  
 **const** url = `${apiUrl}/${id}`;  
 **return this**.http.get(url, httpOptions).pipe(  
 map(**this**.extractData),  
 catchError(**this**.handleError));  
 }  
  
 postGame(data): Observable<**any**> {  
 **return this**.http.post(apiUrl, data, httpOptions)  
 .pipe(  
 catchError(**this**.handleError)  
 );  
 }  
  
 updateGame(id: **string**, data): Observable<**any**> {  
 **const** url = `${apiUrl}/${id}`;  
 **return this**.http.put(url, data, httpOptions)  
 .pipe(  
 catchError(**this**.handleError)  
 );  
 }  
  
 deleteGame(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** { FormsModule, ReactiveFormsModule } **from** '@angular/forms';  
**import** { HttpClientModule } **from** '@angular/common/http';  
  
**import** { AppComponent } **from** './app.component';  
**import** { RouterModule, Routes } **from** '@angular/router';  
**import** { GameComponent } **from** './game/game.component';  
**import** { GameDetailComponent } **from** './game-detail/game-detail.component';  
**import** { GameCreateComponent } **from** './game-create/game-create.component';  
**import** { GameEditComponent } **from** './game-edit/game-edit.component';  
**import** { BrowserAnimationsModule } **from** '@angular/platform-browser/animations';  
  
**import** { AngularFireModule } **from** 'angularfire2';  
**import** { AngularFireAuth } **from** 'angularfire2/auth';  
**import** { firebaseConfig } **from** '../config';  
  
**import** {  
 MatInputModule,  
 MatPaginatorModule,  
 MatProgressSpinnerModule,  
 MatSortModule,  
 MatTableModule,  
 MatIconModule,  
 MatButtonModule,  
 MatCardModule,  
 MatFormFieldModule } **from** '@angular/material';  
  
**const** appRoutes: Routes = [  
 {  
 path: 'games',  
 component: GameComponent,  
 data: { title: 'Game List' }  
 },  
 {  
 path: 'game-details/:id',  
 component: GameDetailComponent,  
 data: { title: 'Game Details' }  
 },  
 {  
 path: 'game-create',  
 component: GameCreateComponent,  
 data: { title: 'Create Game' }  
 },  
 {  
 path: 'game-edit/:id',  
 component: GameEditComponent,  
 data: { title: 'Edit Game' }  
 },  
 { path: '',  
 redirectTo: '/games',  
 pathMatch: 'full'  
 }  
];  
  
@NgModule({  
 declarations: [  
 AppComponent,  
 GameComponent,  
 GameDetailComponent,  
 GameCreateComponent,  
 GameEditComponent  
 ],  
 imports: [  
 RouterModule.*forRoot*(appRoutes),  
 BrowserModule,  
 FormsModule,  
 ReactiveFormsModule,  
 HttpClientModule,  
 BrowserAnimationsModule,  
 MatInputModule,  
 MatTableModule,  
 MatPaginatorModule,  
 MatSortModule,  
 MatProgressSpinnerModule,  
 MatIconModule,  
 MatButtonModule,  
 MatCardModule,  
 AngularFireModule.*initializeApp*(firebaseConfig.fire),  
 MatFormFieldModule  
 ],  
 providers: [AngularFireAuth],  
 bootstrap: [AppComponent]  
})  
**export class** AppModule { }

**import** { Component } **from** '@angular/core';  
  
@Component({  
 selector: 'app-root',  
 templateUrl: './app.component.html',  
 styleUrls: ['./app.component.css']  
})  
**export class** AppComponent {  
 title = 'app';  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/game-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" -->  
  
 <!-- Game Name Column -->  
 <ng-container matColumnDef="game\_name">  
 <th mat-header-cell *\*matHeaderCellDef*> Game Name </th>  
 <td mat-cell *\*matCellDef*="**let** element" class="isbn-col"> {{element.game\_name}} </td>  
 </ng-container>  
  
 <!-- Team Column -->  
 <ng-container matColumnDef="team">  
 <th mat-header-cell *\*matHeaderCellDef*> Team </th>  
 <td mat-cell *\*matCellDef*="**let** element"> {{element.team}} </td>  
 </ng-container>  
  
 <!-- Organizer Column -->  
 <ng-container matColumnDef="organizer">  
 <th mat-header-cell *\*matHeaderCellDef*> Organizer </th>  
 <td mat-cell *\*matCellDef*="**let** element"> {{element.organizer}} </td>  
 </ng-container>  
  
 <!-- Contact Organizer Column -->  
 <ng-container matColumnDef="contact\_org">  
 <th mat-header-cell *\*matHeaderCellDef*> Contact Organizer </th>  
 <td mat-cell *\*matCellDef*="**let** element"> {{element.contact\_org}} </td>  
 </ng-container>  
  
 <tr mat-header-row *\*matHeaderRowDef*="displayedColumns"></tr>  
 <tr mat-row *\*matRowDef*="**let** row; columns: displayedColumns;" [routerLink]="['/game-details/', row.\_id]"></tr>  
 </table>  
</div>

**import** { Component, OnInit } **from** '@angular/core';  
**import** { ApiService } **from** '../api.service';  
**import** { DataSource } **from** '@angular/cdk/collections';  
  
@Component({  
 selector: 'app-book',  
 templateUrl: './game.component.html',  
 styleUrls: ['./game.component.css']  
})  
**export class** GameComponent **implements** OnInit {  
  
 games: **any**;  
 displayedColumns = ['game\_name', 'team', 'organizer', 'contact\_org'];  
 dataSource = **new** BookDataSource(**this**.api);  
  
 **constructor**(**private** api: ApiService) { }  
  
 ngOnInit() {  
 **this**.api.getGames()  
 .subscribe(res => {  
 console.log(res);  
 **this**.games = res;  
 }, err => {  
 console.log(err);  
 });  
 }  
}  
  
**export class** BookDataSource **extends** DataSource<**any**> {  
 **constructor**(**private** api: ApiService) {  
 **super**();  
 }  
  
 connect() {  
 **return this**.api.getGames();  
 }  
  
 disconnect() {  
  
 }  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/games']"><mat-icon>list</mat-icon></a>  
</div>  
<form [formGroup]="gameForm" (ngSubmit)="onFormSubmit(gameForm.value)">  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Name" formControlName="game\_name"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_name').valid && gameForm.get('game\_name').touched">Please enter Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Team Name" formControlName="team"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('team').valid && gameForm.get('team').touched">Please enter Team Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Organizer" formControlName="organizer"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('organizer').valid && gameForm.get('organizer').touched">Please enter Game Organizer Name.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <textarea matInput placeholder="Game Location" formControlName="game\_loc"  
 [errorStateMatcher]="matcher"></textarea>  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_loc').valid && gameForm.get('game\_loc').touched">Please enter Game Location, complete Address.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Day" formControlName="game\_date"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_date').valid && gameForm.get('game\_date').touched">Please enter planned Game Day.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Time" formControlName="game\_time"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_time').valid && gameForm.get('game\_time').touched">Please enter planned Game Time.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Duration" formControlName="game\_duration"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_duration').valid && gameForm.get('game\_duration').touched">Please enter planned Game Duration.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Organizer Contact Number" formControlName="contact\_org"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('contact\_org').valid && gameForm.get('contact\_org').touched">Please enter Contact Number.</span>  
 </mat-error>  
 </mat-form-field>  
 <div class="button-row">  
 <button type="submit" [disabled]="!gameForm.valid" mat-raised-button color="primary"><mat-icon>save</mat-icon></button>  
 </div>  
</form>

**import** { Component, OnInit } **from** '@angular/core';  
**import** { Router } **from** '@angular/router';  
**import** { ApiService } **from** '../api.service';  
**import** { FormBuilder, FormGroup, NgForm, Validators } **from** '@angular/forms';  
  
@Component({  
 selector: 'app-book-create',  
 templateUrl: './game-create.component.html',  
 styleUrls: ['./game-create.component.css']  
})  
**export class** GameCreateComponent **implements** OnInit {  
  
 gameForm: FormGroup;  
 game\_name = '';  
 team = '';  
 organizer = '';  
 game\_loc = '';  
 game\_date = '';  
 game\_time = '';  
 game\_duration = '';  
 contact\_org = '';  
  
 **constructor**(**private** router: Router, **private** api: ApiService, **private** formBuilder: FormBuilder) { }  
  
 ngOnInit() {  
 **this**.gameForm = **this**.formBuilder.group({  
 'game\_name' : [**null**, Validators.*required*],  
 'team' : [**null**, Validators.*required*],  
 'organizer' : [**null**, Validators.*required*],  
 'game\_loc' : [**null**, Validators.*required*],  
 'game\_date' : [**null**, Validators.*required*],  
 'game\_time' : [**null**, Validators.*required*],  
 'game\_duration' : [**null**, Validators.*required*],  
 'contact\_org' : [**null**, Validators.*required*]  
 });  
 }  
  
 onFormSubmit(form: NgForm) {  
 **this**.api.postGame(form)  
 .subscribe(res => {  
 **const** id = res['\_id'];  
 **this**.router.navigate(['/game-details', id]);  
 }, (err) => {  
 console.log(err);  
 });  
 }  
}

<div class="button-row">  
 <a mat-raised-button color="primary" [routerLink]="['/games']"><mat-icon>list</mat-icon></a>  
</div>  
<mat-card class="example-card">  
 <mat-card-header>  
 <mat-card-title><h2>{{game.team}}</h2></mat-card-title>  
 <mat-card-subtitle>{{game.organizer}}</mat-card-subtitle>  
 </mat-card-header>  
 <mat-card-content>  
 <dl>  
 <dt>Game Name:</dt>  
 <dd>{{game.game\_name}}</dd>  
 <dt>Team:</dt>  
 <dd>{{game.team}}</dd>  
 <dt>Game Organizer:</dt>  
 <dd>{{game.organizer}}</dd>  
 <dt>Game Location:</dt>  
 <dd>{{game.game\_loc}}</dd>  
 <dt>Game Date:</dt>  
 <dd>{{game.game\_date | date}}</dd>  
 <dt>Game Time:</dt>  
 <dd>{{game.game\_time}}</dd>  
 <dt>Game Duration:</dt>  
 <dd>{{game.game\_duration}}</dd>  
 <dt>Contact Organizer:</dt>  
 <dd>{{game.contact\_org}}</dd>  
 </dl>  
 </mat-card-content>  
 <mat-card-actions>  
 <a mat-raised-button color="primary" [routerLink]="['/game-edit', game.\_id]"><mat-icon>edit</mat-icon></a>  
 <a mat-raised-button color="warn" (click)="deleteBook(game.\_id)"><mat-icon>delete</mat-icon></a>  
 </mat-card-actions>  
</mat-card>

**import** { Component, OnInit } **from** '@angular/core';  
**import** { ActivatedRoute, Router } **from** '@angular/router';  
**import** { ApiService } **from** '../api.service';  
  
@Component({  
 selector: 'app-book-detail',  
 templateUrl: './game-detail.component.html',  
 styleUrls: ['./game-detail.component.css']  
})  
**export class** GameDetailComponent **implements** OnInit {  
  
 game = {};  
  
 **constructor**(**private** route: ActivatedRoute, **private** api: ApiService, **private** router: Router) { }  
  
 ngOnInit() {  
 **this**.getBookDetails(**this**.route.snapshot.params['id']);  
 }  
  
 getBookDetails(id) {  
 **this**.api.getGame(id)  
 .subscribe(data => {  
 console.log(data);  
 **this**.game = data;  
 });  
 }  
  
 deleteBook(id) {  
 **this**.api.deleteGame(id)  
 .subscribe(res => {  
 **this**.router.navigate(['/games']);  
 }, (err) => {  
 console.log(err);  
 }  
 );  
 }  
  
}

<div class="button-row">  
 <a mat-raised-button color="primary" (click)="bookDetails()"><mat-icon>show</mat-icon></a>  
</div>  
<form [formGroup]="gameForm" (ngSubmit)="onFormSubmit(gameForm.value)">  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Name" formControlName="game\_name"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_name').valid && gameForm.get('game\_name').touched">Please enter Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Team Name" formControlName="team"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('team').valid && gameForm.get('team').touched">Please enter Team Name</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Organizer" formControlName="organizer"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('organizer').valid && gameForm.get('organizer').touched">Please enter Game Organizer Name.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <textarea matInput placeholder="Game Location" formControlName="game\_loc"  
 [errorStateMatcher]="matcher"></textarea>  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_loc').valid && gameForm.get('game\_loc').touched">Please enter Game Location, complete Address.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Day" formControlName="game\_date"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_date').valid && gameForm.get('game\_date').touched">Please enter planned Game Day.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Time" formControlName="game\_time"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_time').valid && gameForm.get('game\_time').touched">Please enter planned Game Time.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Game Duration" formControlName="game\_duration"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('game\_duration').valid && gameForm.get('game\_duration').touched">Please enter planned Game Duration.</span>  
 </mat-error>  
 </mat-form-field>  
 <mat-form-field class="example-full-width">  
 <input matInput placeholder="Organizer Contact Number" formControlName="contact\_org"  
 [errorStateMatcher]="matcher">  
 <mat-error>  
 <span *\*ngIf*="!gameForm.get('contact\_org').valid && gameForm.get('contact\_org').touched">Please enter Contact Number.</span>  
 </mat-error>  
 </mat-form-field>  
 <div class="button-row">  
 <button type="submit" [disabled]="!gameForm.valid" mat-raised-button color="primary"><mat-icon>save</mat-icon></button>  
 </div>  
</form>

**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-book-edit',  
 templateUrl: './game-edit.component.html',  
 styleUrls: ['./game-edit.component.css']  
})  
**export class** GameEditComponent **implements** OnInit {  
 gameForm: FormGroup;  
 id:**string** = '';  
 game\_name: String = '';  
 team: String = '';  
 organizer: String = '';  
 game\_loc: String = '';  
 game\_date: String = '';  
 game\_time: String = '';  
 game\_duration: String = '';  
 contact\_org: String = '';  
  
 **constructor**(**private** router: Router, **private** route: ActivatedRoute, **private** api: ApiService, **private** formBuilder: FormBuilder) { }  
  
 ngOnInit() {  
 **this**.getBook(**this**.route.snapshot.params['id']);  
 **this**.gameForm = **this**.formBuilder.group({  
 'game\_name' : [**null**, Validators.*required*],  
 'team' : [**null**, Validators.*required*],  
 'organizer' : [**null**, Validators.*required*],  
 'game\_loc' : [**null**, Validators.*required*],  
 'game\_date' : [**null**, Validators.*required*],  
 'game\_time' : [**null**, Validators.*required*],  
 'game\_duration' : [**null**, Validators.*required*],  
 'contact\_org' : [**null**, Validators.*required*]  
 });  
 }  
  
 getBook(id) {  
 **this**.api.getGame(id).subscribe(data => {  
 **this**.id = data.\_id;  
 **this**.gameForm.setValue({  
 game\_name: data.game\_name,  
 team: data.team,  
 organizer: data.organizer,  
 game\_loc: data.game\_loc,  
 game\_date: data.game\_date,  
 game\_time: data.game\_time,  
 game\_duration: data.game\_duration,  
 contact\_org: data.contact\_org  
 });  
 });  
 }  
  
 onFormSubmit(form:NgForm) {  
 **this**.api.updateGame(**this**.id, form)  
 .subscribe(res => {  
 **let** id = res['\_id'];  
 **this**.router.navigate(['/game-details', id]);  
 }, (err) => {  
 console.log(err);  
 }  
 );  
 }  
  
 bookDetails() {  
 **this**.router.navigate(['/game-details', **this**.id]);  
 }  
}