Smart List

1. The nodejs will start the server:

app.js:

const mongoose = require("mongoose");

const app = express();

const resetdb = require("./models/initDB");

mongoose

.connect(

//"mongodb+srv://max:QuBqs0T45GDKPlIG@cluster0-ntrwp.mongodb.net/node-angular?retryWrites=true"

"mongodb://smartlistmean:Mazda5Gray@ds026658.mlab.com:26658/smartlistmean", { useNewUrlParser: true }

)

.

// resetdb.resetDB();

.

app.use("/api/checklistItems", checklistItemsRoutes);

module.exports = app;

2. browse to

**index.html** 🡪 <app-root>=**app.component**.ts 🡪 <app-**header**>+ <router-**outlet**> 🡪

Click on header menu will rout to component’s url, which will be rendered inner the outlet tag:

**checklist-items-list.component.**ts 🡪 will call the service to get data – **ChecklistItemsService.** **getChecklistItems()**

(Will return **ChecklistItem** from Model)

The service will open http request to backend server to appropriate URL:

**"**http://localhost:3000/api/checklistItems" + queryParams

**The backend server** will get the request and search for matching url route definition in:

**app.js** 🡪 app.use("/api/checklistItems", checklistItemsRoutes);

const checklistItemsRoutes = require("./routes/checklistItems");

🡪 ~\backend\routes\checklistItems.js will get the request and switch to the appropriate function depend on Http Method

const ChecklistItems = require("../models/checklistItem");

.

.

router.get("", (req, res, next) => {

Get data from database (Mongo) schema/model:

ChecklistItems.find().where('checklistId').equals(ObjectId(checklistId)) ;

🡪 ~\backend\models\checklistItem.js

const mongoose = require('mongoose'), Schema = mongoose.Schema;

const checklistSchema = mongoose.Schema({

2.

|  |  |
| --- | --- |
| File | Desc |
| SmartList |  |
| ~\backend |  |
| ~\backend\angular |  |
| ~\backend\dataFromScraper |  |
| ~\backend\dataFromScraper\bakingmad.json |  |
| ~\backend\dataFromScraper\bakingmad2.json |  |
| ~\backend\images |  |
| ~\backend\models |  |
| ~\backend\models\checklist.js |  |
| ~\backend\models\checklistItem.js |  |
| ~\backend\models\group.js |  |
| ~\backend\models\initDB.js |  |
| ~\backend\models\user.js |  |
| ~\backend\routes |  |
| ~\backend\routes\checklistItems.js |  |
| ~\backend\routes\checklists.js |  |
| ~\backend\routes\groups.checklists.js |  |
| ~\backend\routes\groups.HLL.js |  |
| ~\backend\routes\groups.js |  |
| ~\backend\routes\users.js |  |
| **~\backend\app.js** |  |
| ~\backend\npm-debug.log |  |
| ~\e2e |  |
| ~\node\_modules |  |
| ~\server |  |
| ~\src |  |
| ~\src\app |  |
| ~\src\app\components |  |
| ~\src\app\components\checklistItems |  |
| ~\src\app\components\checklistItems\checklist-items-create |  |
| ~\src\app\components\checklistItems\checklist-items-list |  |
| ~\src\app\components\checklistItems\checklist-items-list\ checklist-items-list.component.css |  |
| ~\src\app\components\checklistItems\checklist-items-list\ checklist-items-list.component.html | <table mat-table **[dataSource]="dataSource"** matSort>  <!-- Checkbox Column -->  <ng-container matColumnDef="select">  <th mat-header-cell \*matHeaderCellDef>  <mat-checkbox (click)="$event.stopPropagation()"  (change)="$event ? masterToggle() : null"  [checked]="selection.hasValue() && isAllSelected()"  [indeterminate]="selection.hasValue()">  </mat-checkbox>  </th>  <td mat-cell \*matCellDef="let row">  <mat-checkbox (click)="$event.stopPropagation()"  (change)="toggle(row)"  [checked]="row.isDone">  </mat-checkbox>  </td>  </ng-container>  <!-- Progress Column -->  <ng-container matColumnDef="title">  <th mat-header-cell \*matHeaderCellDef mat-sort-header> Title </th>  <td mat-cell \*matCellDef="let row"> {{row.title}} </td>  </ng-container> |
| ~\src\app\components\checklistItems\checklist-items-list\ checklist-items-list.component.ts | import { Component, OnInit, OnDestroy, Input, Output ,ViewChild} from "@angular/core";  import {MatPaginator, MatSort, MatTableDataSource} from '@angular/material';  @Component({  selector: "app-checklistItem-list",  templateUrl: "./checklist-Items-list.component.html",  styleUrls: ["./checklist-Items-list.component.css"]  })  export class ChecklistItemsListComponent implements OnInit, OnDestroy {  **dataSource**: MatTableDataSource<ChecklistItem>;  **// ChecklistItemsService service will created as a singleton instance and will be injected automatically to constratcor!!!**  constructor(public **checklistItemsService**: ChecklistItemsService,  public route: ActivatedRoute,) {}  ngOnInit() {  this.**checklistItemsService**.getChecklistItems(this.checklistItemsPerPage, this.currentPage, this.checklistId);  this.checklistItemsSub = this.checklistItemsService  .getChecklistItemsUpdateListener()  .subscribe((checklistItemsData: {**checklistItems**: ChecklistItem[], checklistItemsCount: number}) => {  this.isLoading = false;  this.totalChecklistItems = checklistItemsData.checklistItemsCount;  this.checklistItems = checklistItemsData.checklistItems;  **this.dataSource = new MatTableDataSource(this.checklistItems);**  });  }); |
| ~\src\app\components\checklists |  |
| ~\src\app\components\email |  |
| ~\src\app\components\groups |  |
| ~\src\app\components\header |  |
| ~\src\app\components\header\header.component.css |  |
| ~\src\app\components\header\header.component.html | **routerLink="/group-list"** will tell the router (in app-routing.module.ts) to render the tag **<router-outlet></router-outlet> (**in app.component.html**) as a Component** which declared in { path: 'group-list', component: GroupListComponent ,canActivate: [AuthGuard]},  <mat-toolbar color="primary">  <span>  <a routerLink="/">Smart List</a>  </span>  <span class="spacer"></span>  <ul>  <li>  <a mat-button **routerLink="/group-list"** **routerLinkActive**="mat-accent">Groups</a>  </li>  </ul> |
| ~\src\app\components\header\header.component.ts |  |
| ~\src\app\components\login |  |
| ~\src\app\components\shared |  |
| ~\src\app\components\signup |  |
| ~\src\app\components\Statistics |  |
| ~\src\app\components\two-ways-bindings |  |
| ~\src\app\directives |  |
| ~\src\app\directives\d3-chart.directive.ts |  |
| ~\src\app\directives\google-places.directive.ts |  |
| ~\src\app\Model |  |
| ~\src\app\Model\checklist.model.ts |  |
| ~\src\app\Model\checklistItem.model.ts | export interface **ChecklistItem** {  id: string;//PK  title: string;  description: string;  imagePath: string;  checklistId: string;//FK  isDone: boolean;  } |
| ~\src\app\Model\group.model.ts |  |
| ~\src\app\Model\user.ts |  |
| ~\src\app\Pipes |  |
| ~\src\app\Pipes\HLLPipe.ts |  |
| ~\src\app\services | import { HttpClient } from "@angular/common/http";  @Injectable({ providedIn: "root" })  export class **ChecklistItemsService** {  **private checklistItems: ChecklistItem[] = [];**  private checklistItemsUpdated = new Subject<{ checklistItems: ChecklistItem[]; checklistItemsCount: number }>();  constructor(private http: HttpClient, private router: Router) {}  **getChecklistItems**(checklistItemsPerPage: number, currentPage: number, checklistId: string) {  const queryParams = `?pagesize=${checklistItemsPerPage}&page=${currentPage}&checklistId=${checklistId}`;  console.log(checklistId);  this.http  .get<{ message: string; checklistItems: any; maxChecklistItems: number }>(  "http://localhost:3000/api/checklistItems" + queryParams  )  .pipe(  map(checklistItemsData => {  return {  **checklistItems**: checklistItemsData.checklistItems.map(checklistItems => { |
| ~\src\app\services\auth-guard.service.ts |  |
| ~\src\app\services\auth.service.ts |  |
| ~\src\app\services\checklistItems.service.ts |  |
| ~\src\app\services\checklists.service.ts |  |
| ~\src\app\services\groups-hll.service.ts |  |
| ~\src\app\services\groups.service.ts |  |
| ~\src\app\services\socket.service.ts |  |
| ~\src\app\services\users.service.ts |  |
| ~\src\app\validators |  |
| ~\src\app\app-routing.module.ts | import { NgModule } from "@angular/core";  import { RouterModule, Routes } from "@angular/router";  import { SignupComponent } from './components/signup/signup.component';  import { EmailComponent } from './components/email/email.component';  import { GroupCreateComponent } from './components/groups/group-create/group-create.component';  import { GroupListComponent } from './components/groups/group-list/group-list.component';  const routes: Routes = [  //{ path: '', redirectTo: 'login-email', pathMatch: 'full' },  { path: '', component: EmailComponent },  { path: 'login', component: LoginComponent },  { path: 'signup', component: SignupComponent },  { path: 'login-email', component: EmailComponent },  //Groups  { path: 'group-create', component: GroupCreateComponent ,canActivate: [AuthGuard]},  { path: 'group-edit/:groupId', component: GroupCreateComponent ,canActivate: [AuthGuard]},  { path: 'group-list', component: GroupListComponent ,canActivate: [AuthGuard]}, |
| ~\src\app\app.component.css |  |
| ~\src\app\app.component.html | <app-header></app-header>  <main>  **<router-outlet></router-outlet>**  </main> |
| ~\src\app\app.component.spec.ts |  |
| ~\src\app\app.component.ts |  |
| ~\src\app\app.module.ts | import { AppComponent } from "./app.component";  import { D3ChartDirective } from './directives/d3-chart.directive';  @NgModule({  declarations: [  AppComponent,  D3ChartDirective  ],  imports: [  AgmCoreModule.forRoot({  apiKey: "AIzaSyCnqVDFETbV",  libraries: ["places"]  }),  BrowserModule,  AppRoutingModule,  ReactiveFormsModule,  BrowserAnimationsModule,  MatInputModule,  AngularFireAuthModule,  ],  providers: [AuthGuard],  bootstrap: [AppComponent]  })  export class AppModule {} |
| ~\src\assets |  |
| ~\src\environments |  |
| ~\src\browserslist |  |
| ~\src\favicon.ico |  |
| ~\src\index.html | <app-root></app-root>  This is the selector selector: 'app-root', in ~\src\app\app.component.ts |
| ~\src\karma.conf.js |  |
| ~\src\main.ts |  |
| ~\src\polyfills.ts |  |
| ~\src\styles.css |  |
| ~\src\test.ts |  |
| ~\src\tsconfig.app.json |  |
| ~\src\tsconfig.spec.json |  |
| ~\src\tslint.json |  |
| ~\.editorconfig |  |
| ~\.gitignore |  |
| ~\angular.json |  |
| ~\how-to-use.txt |  |
| ~\npm-debug.log |  |
| ~\package-lock.json |  |
| ~\package.json |  |
| ~\README.md |  |
| ~\server.js |  |
| ~\start.bat |  |
| ~\tsconfig.json |  |
| ~\tslint.json |  |
| .gitignore |  |