GHSV

Angular CRUD

****

Angular CRUD with Web API Tutorial Part #1 - Project Introduction and CRUD overview

- Building new feature called "Products"
- Products module should have it 's own routes
- Product module shuold support lazy loading
- Ability to see all products inventory
- Ability to view a "specific" product for details
- ability to update the product inventory
- Ability to delete the product inventory
- Seach product by category
- Ability to read categories as drop down in UI

- CREATE - create new resource via API , HTTP POST method
- Read - Retrieve existing data from API, HTTP GET
- Update - Update existing data from API, HTTP PUT
- Delete - Delete an existing Resource via API - HTTP Delete

****

Angular CRUD with Web API Tutorial Part #2 - Install Angular CLI in Visual Studio Code

- Check node and npm versions
    - node -v
    - npm - v
- Install Angular CLI
    - npm -g @angular/CLI
- Create New application
    - ng new <project_name>
- Start the angular application
    - ng serve
- The default port of the application is - 4200
- http://localhost:4200
-

**NOTE - You will get this error if there is no angular project - The serve command requires to be run in an Angular project, but a project definition could not be found.

****

Angular CRUD with Web API Tutorial Part #3 - Setup Local JSON Server and Mock API Endpoints

- We can use any framework, any language, any server setup from API
- Threre maybe - Cloud based and mock end points
- JSON Server
    - npm i --save json-server (OR) npm install -g json-server
    - json-server --watch .\db.json
- The default port that run json-server is - 3000, BUT you can configure on your need basis

****

Angular CRUD with Web API Tutorial Part #4 - Install Bootstrap in Angular 9 Visual Studio Code

- You can use any UI framework to build your websites
- The CSS framework
    - Bootstrap
    - Material Design
    - Custom Design
- Installation of Bootstrap framework
    - npm i --save bootstrap jquery popper
    - Above command will install package
    - It will update package.json file with the bootstrap, jquery, popper version details
- IMPORTANT STEP : - In angular.json file you have to update the 'styles' and 'scripts' path of bootstrap & jquery. If this not happend, application will not work
    - angular.json => under build =>

```
            "styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.min.css"
],
"scripts": [
    "node_modules/jquery/dist/jquery.min.js",
    "node_modules/bootstrap/dist/js/bootstrap.min.js"
]
```

- When we make any changes in files - Angular CLI will do HOT load
    - automatic reload of the application

- BUT when we make changes to angular.json - it does not auto reload ***ALWAYS TO RELOAD THE APPLICATION (ng serve)**

- This is ideal way
    - you can also CDN - content delivery network
    - They will provide links and this is not encouraged way of doing things

****

Angular CRUD with Web API Tutorial Part #5 - Feature Modules in Angular App

- Creating feature module helps in many ways
    - It's a great way to group related functionality
    - Easy to maintain and easy to re-use
    - Allows us to implement lazy loading
    - Helps in securing protected pages
- Two ways of approaches
    - Using Modules
    - Using Modules withh Lazy Loading (load when they required)
- Generating modules & components
    - ng generate module <modulename>
    - ng generate component <componentname>
- Create this component inside the orders module
    - import orders module into app module
    - add the module into imports array
- add below line into app.component.html
    - <app-list-orders></app-list-orders>

      This will through exception Since we have not exported the Orders component
- In Order.module.ts add exports array with the component name
    - exports:[ListOrdersComponent]
- ng g m products --route products --module app.module

****

Angular CRUD with Web API Tutorial Part #6 - Generate Components with Module and Routing

- Generate comopnents
    - products
    - create below components under products component

        - prodcuts-by-category
        - create-product
        - view-product
        - view-all-products
        - view-all-products-by-date
        - view-all-products-by-category
        - update-product
        - delete-product

****
****

Angular CRUD with Web API Tutorial Part #7 - Angular Layout Template Tutorials

- Create module called - site-framework : ng g m site-framework
- Create header component inside site-framwork - ng g c header

- Create sidebar component inside site-framwork - ng g c sidebar
- Create footer component inside site-framwork - ng g c footer
- https://startbootstrap.github.io/startbootstrap-shop-homepage/
- View source of above page ->
- import site-framework module into app.module.ts
- IF YOU NOT add <router-outlet> you can not see component in UI

****
****

Angular CRUD with Web API Tutorial Part #8 - CRUD Feature Module Routing

- We are going to update the products feature module's routing file
    - create-product => adding new product
    - list-products => list of all products
    - search?category => list prodcuts by category
    - search?date => list products by date
    - delete-product - delete product
    - view-product/id - view product by id

****
****

Angular CRUD with Web API Tutorial Part #9 - Service For Feature Module

- Import HttpClientModule,HttpClient into App module - '@angular/common/http'
- generate the service - ng g service <service name>
- Import HttpClient into the service
- Inject HttpClient in constructor - constructor (private httpClient:HttpClient) {}
- The best way is to create the service into respective modules

****
****

Angular CRUD with Web API Tutorial Part #10 - Creating Interface Model For Feature Module

- We need to create an interface or Model for Product
    - when the HTTP returns the data we can bind the observable with the model
    - Helps in strict type checking of the response
    - Its good practice to keep all models in one place
- Generate the Interface also known as Model
    - ng generate interrace <interfacename>
- Update the class definition of the interface
    - -export interface Product {

      ```
      id: string;
      categoryId: number;
      productName: string;
      ```

```
                                    description: string;
                                    rating: string;
                                    price: number;
                                    productImg: string;
                                    isAvailable: boolean;
                                    color: string;
                                    reviews: number;
                                                                    }
```

- import above interface in Product service
- Methods should be use interface as below

```
        createProduct(productBody): Observable<Product> {const svcUrl = 
'http://localhost:3000/products';
                    return this.httpClient.post<Product>(svcUrl, productBody); 
//return observable}
```

****
****

Angular CRUD with Web API Tutorial Part #11 - Feature Module CRUD

-

****
****

Angular CRUD with Web API Tutorial Part #12 - Feature Module CRUD Tutorial

--For Create

- Create UI for create product - using bootstrap
- Add the required fileds into that form
- Give all fields attribute name and ngModel
    - this is one way binding
- On Submit event => (submit) = "addNewProduct(addProductForm)"

Common Mistakes

- They do not import forms module in products module and app module .
  In this you will get error - ERROR in src/app/products/create-product/create-product.component.html:3:65 - error NG8003: No directive found with exportAs 'ngForm'.
- Not binding data using ngModel

--For Update

- First we will capture the id of the product which we want to edit
- Read the data of existing data of that product
- use 2-way data mapping to the fields on the screen
    - [(ngModel)] = "productDetails.productName"

- finally call - PUT method to update the product

****

****