http://localhost:9090/product

Spring --------------Controller

@CrossOrigin(origins = "http://localhost:4200")

restart spring

Angular

services

product

components

product-list

product-add

models

product.ts

1)

H:\alstrom-product-app> ng g s services/product

PS H:\alstrom-product-app> ng g c components/product-list

PS H:\alstrom-product-app> ng g c components/product-add

PS H:\alstrom-product-app> ng g class models/product --type=model

2. update app.module.ts

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

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule \*\*\*\*\*

],

3. update models/product.ts

export class Product {

productId?: number;

productName?: string;

quantityOnHand?: number;

price?: number;

}

4. open services/product.service.ts

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

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

import { Observable } from 'rxjs';

import { Product } from '../models/product.model';

const productUrl = "http://localhost:9090/product"

@Injectable({

providedIn: 'root'

})

export class ProductService {

//DI

constructor(public http:HttpClient) { }

getProducts() :Observable<Product[]>{

return this.http.get<Product []>(productUrl).pipe();

}

}

5. product-list.component.ts

//

constructor(public productService:ProductService) { }

products : Product[] = [] ;

ngOnInit(): void {

this.productService.getProducts().subscribe((data: any[])=>{

console.log("###Products recieved from spring :")

console.log(data);

this.products = data;

})

}

6.app.component.html ( remove all the stuffs)

<h1>Alstrom Product -App</h1>

<app-product-list></app-product-list> \*\*\* selector of product-list.component.ts

Save all the files and run in the browser

------------------Error Handling in REST

**import { retry, catchError } from 'rxjs/operators';**

**import { Observable, throwError } from 'rxjs';**

getProducts(): Observable<Product[]> {

return this.http.get<Product[]>(productUrl)

.pipe(

**retry(1),**

**catchError(this.errorHandler)**

)

}

**errorHandler(error: { error: { message: string; }; status: any; message: any; }) {**

**let errorMessage = '';**

**if (error.error instanceof ErrorEvent) {**

**// Get client-side error**

**errorMessage = error.error.message;**

**} else {**

**// Get server-side error**

**errorMessage = `Error Code: ${error.status}\nMessage: ${error.message}`;**

**}**

**console.log(errorMessage);**

**return throwError(errorMessage);**

**}**

----------------open product.component.ts

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

import { Product } from 'src/app/models/product.model';

import { ProductService } from 'src/app/services/product.service';

@Component({

  selector: 'app-product-list',

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

  styleUrls: ['./product-list.component.css']

})

export class ProductListComponent implements OnInit {

**errorMessage ?:string;**

  //

  constructor(public productService:ProductService) { }

  products : Product[] = [] ;

  ngOnInit(): void {

      this.productService.getProducts().subscribe((data: any[])=>{

        console.log("###Products recieved from spring :")

        console.log(data);

        this.products = data;

      },**err => this.errorMessage = err)**

  }

}

Open product.component.html

      <font color=red>{{ errorMessage }}</font>