package com.ouhamza.ecommerce.config;

import com.ouhamza.ecommerce.entity.Country;

import com.ouhamza.ecommerce.entity.Product;

import com.ouhamza.ecommerce.entity.ProductCategory;

import com.ouhamza.ecommerce.entity.State;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.rest.core.config.RepositoryRestConfiguration;

import org.springframework.data.rest.webmvc.config.RepositoryRestConfigurer;

import org.springframework.http.HttpMethod;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

import javax.persistence.EntityManager;

import javax.persistence.metamodel.EntityType;

import java.util.ArrayList;

import java.util.List;

import java.util.Set;

/\*\*

\* @author <a href="https://github.com/ouhamzalhss"> Lhouceine OUHAMZA </a>

\*/

@Configuration

public class MyDataRestConfig implements RepositoryRestConfigurer {

@Autowired

private EntityManager entityManager;

@Override

public void configureRepositoryRestConfiguration(RepositoryRestConfiguration config, CorsRegistry cors) {

HttpMethod[] theUnsupportedHttpMethods = {HttpMethod.POST, HttpMethod.PUT, HttpMethod.DELETE};

//Disable http method for ProductCategory class: POST, PUT, DELETE

disableHttpMethod(Product.class, config, theUnsupportedHttpMethods);

disableHttpMethod(ProductCategory.class, config, theUnsupportedHttpMethods);

disableHttpMethod(Country.class, config, theUnsupportedHttpMethods);

disableHttpMethod(State.class, config, theUnsupportedHttpMethods);

// call an internal helper method

exposeIds(config);

}

private void disableHttpMethod(Class theClass, RepositoryRestConfiguration config, HttpMethod[] theUnsupportedHttpMethods) {

config.getExposureConfiguration()

.forDomainType(theClass)

.withItemExposure((metdata, httpMethods) -> httpMethods.disable(theUnsupportedHttpMethods))

.withCollectionExposure((metdata, httpMethods) -> httpMethods.disable(theUnsupportedHttpMethods));

}

private void exposeIds(RepositoryRestConfiguration config) {

// get all entity classes from the entity manager.

Set<EntityType<?>> entities = entityManager.getMetamodel().getEntities();

// create an array of entity type

List<Class> entityClasses = new ArrayList<>();

// get the entity type from entities

for(EntityType tempEntityType: entities){

entityClasses.add(tempEntityType.getJavaType());

}

// expose the entity ids for the array of entity/domains types.

Class[] domainsTypes = entityClasses.toArray(new Class[0]);

config.exposeIdsFor(domainsTypes);

}

}