**// Product Master Group View Controller**

package com.maan.crm.controller;

import java.util.Collections;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.maan.crm.req.ProductDetailsGroupReq;

import com.maan.crm.res.CommonCrmRes;

import com.maan.crm.res.ProductMasterGroupRes;

import com.maan.crm.service.CRMValidationService;

import com.maan.crm.service.PrintReqService;

import com.maan.crm.service.ProductDetailsService;

import com.maan.crm.util.error.Error;

import io.swagger.annotations.Api;

@RestController

@Api(tags = "PRODUCT DETAILS : Product Details", description = "API's")

@RequestMapping("/master")

public class ProductDetailsController {

@Autowired

private ProductDetailsService entityService;

@Autowired

private CRMValidationService validationService;

@Autowired

private PrintReqService reqPrinter;

@PostMapping("/group/productdetails")

public ResponseEntity<CommonCrmRes> getProcutDetailsGroup(@RequestBody ProductDetailsGroupReq req) {

reqPrinter.reqPrint(req);

CommonCrmRes data = new CommonCrmRes();

List<Error> validation = validationService.validateProductDetailsGroupReq(req);

//// validation

if (validation != null && validation.size() != 0) {

data.setCommonResponse(null);

data.setIsError(true);

data.setErrorMessage(validation);

data.setMessage("Failed");

return new ResponseEntity<CommonCrmRes>(data, HttpStatus.OK);

} else {

/////// Get Product

ProductMasterGroupRes res = entityService.getProcutDetailsGroup(req);

data.setCommonResponse(res);

data.setIsError(false);

data.setErrorMessage(Collections.emptyList());

data.setMessage("Success");

if (res != null) {

return new ResponseEntity<CommonCrmRes>(data, HttpStatus.CREATED);

} else {

return new ResponseEntity<>(null, HttpStatus.BAD\_REQUEST);

}

}

}

// Product Master Group View Service Impl

package com.maan.crm.service.impl;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import java.util.Map;

import java.util.stream.Collectors;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.TypedQuery;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Order;

import javax.persistence.criteria.Root;

import javax.persistence.criteria.Subquery;

import org.apache.commons.lang3.StringUtils;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.modelmapper.ModelMapper;

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

import org.springframework.stereotype.Service;

import com.maan.crm.bean.ProductDetails;

import com.maan.crm.bean.ProductMaster;

import com.maan.crm.repository.ProductDetailsRepository;

import com.maan.crm.req.ProductDetailsGroupReq;

import com.maan.crm.res.ProductDetailsGroupRes;

import com.maan.crm.res.ProductDetailsRes;

import com.maan.crm.res.ProductMasterGroupRes;

import com.maan.crm.res.ProductMasterRes;

import com.maan.crm.res.ViewProductRes;

import com.maan.crm.service.ProductDetailsService;

import com.maan.crm.util.error.Error;

@Service

public class ProductDetailsServiceImpl implements ProductDetailsService {

@PersistenceContext

private EntityManager em;

@Autowired

private ProductDetailsRepository productDetailsRepo;

private Logger log = LogManager.getLogger(ProductDetailsServiceImpl.class);

@Override

public ProductMasterGroupRes getProcutDetailsGroup(ProductDetailsGroupReq req) {

ProductMasterGroupRes groupRes = new ProductMasterGroupRes();

ViewProductRes res = new ViewProductRes();

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

ModelMapper mapper = new ModelMapper();

try {

// Criteria

CriteriaBuilder cb = em.getCriteriaBuilder();

CriteriaQuery<ProductMaster> query = cb.createQuery(ProductMaster.class);

List<ProductMaster> list = new ArrayList<ProductMaster>();

// Find All

Root<ProductMaster> p = query.from(ProductMaster.class);

// Select

query.select(p);

// Effective Date Max Filter

Subquery<Long> effectiveDate = query.subquery(Long.class);

Root<ProductMaster> ocpm1 = effectiveDate.from(ProductMaster.class);

effectiveDate.select(cb.max(ocpm1.get("effectiveDate")));

javax.persistence.criteria.Predicate a1 = cb.equal(p.get("productId"), ocpm1.get("productId"));

effectiveDate.where(a1);

// Amend Id Max Filter

Subquery<Long> amendId = query.subquery(Long.class);

Root<ProductMaster> ocpm2 = amendId.from(ProductMaster.class);

amendId.select(cb.max(ocpm2.get("amendId")));

javax.persistence.criteria.Predicate a2 = cb.equal(p.get("productId"), ocpm2.get("productId"));

javax.persistence.criteria.Predicate a3 = cb.equal(p.get("effectiveDate"), ocpm2.get("effectiveDate"));

amendId.where(a2, a3);

// Order By

List<Order> orderList = new ArrayList<Order>();

orderList.add(cb.asc(p.get("productName")));

// Where

javax.persistence.criteria.Predicate n1 = cb.equal(p.get("status"), "Y");

javax.persistence.criteria.Predicate n2 = cb.equal(p.get("effectiveDate"), effectiveDate);

javax.persistence.criteria.Predicate n3 = cb.equal(p.get("amendId"), amendId);

javax.persistence.criteria.Predicate n4 = cb.equal(p.get("productId"), req.getProductId());

query.where(n1, n2, n3, n4).orderBy(orderList);

// Get Result

TypedQuery<ProductMaster> result = em.createQuery(query);

list = result.getResultList();

ProductMaster data = list.get(0);

res = mapper.map(data, ViewProductRes.class);

res.setEntryDate(sdf.parse(sdf.format(list.get(0).getEntryDate())));

Integer amend = data.getAmendId();

Date eff = data.getEffectiveDate();

Integer pro = data.getProductId();

List<ProductDetails> products = productDetailsRepo.findByAmendIdAndEffectiveDateAndProductIdOrderByCategoryIdAsc(amend, eff, pro);

List<ProductDetailsGroupRes> productDetails = new ArrayList<ProductDetailsGroupRes>();

if (products !=null && products.size()>0 ) {

// Group By Category

Map<Integer, List<ProductDetails>> groupByCategory = products.stream().filter(o -> o.getCategoryId() !=null ) .collect(Collectors.groupingBy(o -> o.getCategoryId()));

//Map<String, Long> result = items.stream().collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));

for (Integer categorId : groupByCategory.keySet() ) {

ProductDetailsGroupRes productDetailRes = new ProductDetailsGroupRes();

List<ProductDetails> filterByCategory = groupByCategory.get(categorId);

productDetailRes.setCategoryId(categorId.toString() );

productDetailRes.setCategoryName(filterByCategory.get(0).getCategoryName() );

List<String> pointsRes = new ArrayList<String>();

for(ProductDetails productData : filterByCategory ) {

if(StringUtils.isNotBlank(productData.getDetails()) ) {

pointsRes.add('\u2022' + " " + productData.getDetails());

}

}

productDetailRes.setPointsList(pointsRes);

productDetails.add(productDetailRes);

}

}

// Response

groupRes.setProduct(res);

groupRes.setProductDetailsList(productDetails);

} catch (Exception e ) {

e.printStackTrace();

log.info("Exception is ---> " + e.getMessage());

return null;

}

return groupRes ;

}

}

// Product Master Group View Req

package com.maan.crm.req;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

@Data

public class ProductDetailsGroupReq {

@JsonProperty("ProductId")

private String productId ;

@JsonProperty("InsuranceId")

private String insuranceId ;

}

// Product Master Group View Res

package com.maan.crm.res;

import java.util.List;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

@Data

public class ProductMasterGroupRes {

@JsonProperty("Product")

private ViewProductRes product ;

@JsonProperty("ProductDetailsList")

private List<ProductDetailsGroupRes> productDetailsList ;

}

// View Product Master Res

package com.maan.crm.res;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

@Data

public class ViewProductRes {

@JsonProperty("ProductId")

private Integer productId;

@JsonProperty("ProductName")

private String productName;

@JsonProperty("Price")

private double price;

@JsonProperty("Quantity")

private Integer quantity;

@JsonProperty("ManufactureYear")

private String manufactureYear;

@JsonProperty("ClassTypeId")

private Integer classTypeId;

@JsonProperty("ClassType")

private String classType;

@JsonProperty("InsCompanyId")

private String insCompanyId;

@JsonFormat(pattern = "dd/MM/yyyy")

@JsonProperty("EntryDate")

private Date entryDate;

@JsonProperty("Status")

private String status;

}

// Product Details Group Res

package com.maan.crm.res;

import java.util.List;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

@Data

public class ProductDetailsGroupRes {

@JsonProperty("CategoryId")

private String categoryId ;

@JsonProperty("CategoryName")

private String categoryName ;

@JsonProperty("PointsList")

private List<String> pointsList ;

}