package com.maan.marine.service.impl;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Map;

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 org.springframework.transaction.annotation.Transactional;

import com.google.gson.Gson;

import com.maan.marine.entity.DepositeCbcMaster;

import com.maan.marine.repository.DepositeCbcMasterRepository;

import com.maan.marine.repository.MenuRepository;

import com.maan.marine.req.BrokerListReq;

import com.maan.marine.req.ChequeListReq;

import com.maan.marine.req.DepositCbcReq;

import com.maan.marine.req.OpenCoverReq;

import com.maan.marine.res.ChequeListRes;

import com.maan.marine.res.DepositCbcRes;

import com.maan.marine.res.DropDownResponse;

import com.maan.marine.res.OpenCoverRes;

import com.maan.marine.service.KumarMenuService;

@Service

@Transactional

public class KumarMenuServiceImpl implements KumarMenuService {

@Autowired

private MenuRepository repository;

@Autowired

private DepositeCbcMasterRepository cbcrepository;

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

Gson json = new Gson();

///// broker Drop Down

@Override

public List<DropDownResponse> getBrokerList(BrokerListReq req) {

List<DropDownResponse> resList = new ArrayList<DropDownResponse>();

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

try {

List<Map<String, Object>> brokerList = new ArrayList<Map<String, Object>>();

brokerList = repository.getbrokerList(req.getBranchCode(), req.getLoginid());

for (Map<String, Object> data : brokerList) {

DropDownResponse res = new DropDownResponse();

res.setCodeDesc(data.get("COMPANY\_NAME") == null ? "" : data.get("COMPANY\_NAME").toString());

res.setCode(data.get("AGENCY\_CODE") == null ? "" : data.get("AGENCY\_CODE").toString());

// res.setRsabrokerCode(data.get("RSA\_BROKER\_CODE")==null?"":data.get("RSA\_BROKER\_CODE").toString());

resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

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

return null;

}

return resList;

}

///// Open Cover Details

@Override

public List<OpenCoverRes> getOpenCover(OpenCoverReq req) {

List<OpenCoverRes> resList = new ArrayList<OpenCoverRes>();

try {

List<Map<String, Object>> getOpenCover = new ArrayList<Map<String, Object>>();

int limit = StringUtils.isBlank(req.getLimit()) ? 0 : Integer.valueOf(req.getLimit());

int offset = StringUtils.isBlank(req.getOffset()) ? 10 : Integer.valueOf(req.getOffset());

// Between

int start = limit \* offset + 1;

int end = limit \* offset + offset;

List<String> customerId = req.getCustomerId() == null ? Collections.emptyList() : req.getCustomerId();

getOpenCover = repository.getOpenCoverDetails(customerId, req.getBranchCode(), start, end);

for (Map<String, Object> data : getOpenCover) {

OpenCoverRes res = new OpenCoverRes();

res.setCustomerId(data.get("CUSTOMER\_ID") == null ? "" : data.get("CUSTOMER\_ID").toString());

res.setCustomerName(data.get("CUST\_NAME") == null ? "" : data.get("CUST\_NAME").toString());

res.setPolicyNo(data.get("POLICY\_NO") == null ? "" : data.get("POLICY\_NO").toString());

resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

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

return null;

}

return resList;

}

/// Get Deposit CBC

@Override

public DepositCbcRes getdepositcbc(DepositCbcReq req) {

// List<DepositCbcRes> resList = new ArrayList<DepositCbcRes>();

DepositCbcRes res = new DepositCbcRes();

try {

// List<Map<String,Object>> savecbc = new ArrayList<Map<String,Object>>();

List<DepositeCbcMaster> getdepositcbc = cbcrepository.findByCbcNoAndBrokerId(req.getCbcNo(), req.getBrokerId());

for(DepositeCbcMaster data : getdepositcbc)

{

res.setAccountType(data.getAccountType()== null ?"" : data.getAccountType().toString());

res.setCbcBelonging(data.getCbcBelonging()==null ?"": data.getCbcBelonging().toString());

res.setCustomerId(data.getCreatedBy()==null ?"": data.getCustomerId().toString());

res.setCustomerOption(data.getCustomerOption()==null ?"": data.getCustomerOption());

res.setDepositType(data.getDepositeType()==null ?"": data.getDepositeType());

res.setOpenCoverNo(data.getOpencoverNo()==null?"": data.getOpencoverNo());

res.setProductId(data.getProductId()==null?"":data.getProductId());

res.setRsaCustomerCode(data.getRsaCustomerCode()==null?"":data.getRsaCustomerCode());

res.setUpdatedBy(data.getUpdatedBy()==null ?"": data.getUpdatedBy());

res.setApiDeposit(data.getApiDeposit()==null?"": data.getApiDeposit());

res.setBrokerId(data.getBrokerId()==null?"":data.getBrokerId());

res.setCbcNo(data.getCbcNo()==null?"": data.getCbcNo());

res.setCreatedBy(data.getCreatedBy()==null?"": data.getCreatedBy());

res.setDepositAmount(data.getDepositAmount()==null?"": data.getDepositAmount().toString());

res.setDepositUtilised(data.getDepositUtilised()==null?"":data.getDepositUtilised().toString());

res.setRefundAmount(data.getRefundAmount()==null?"":data.getRefundAmount().toString());

res.setStatus(data.getStatus()==null?"": data.getStatus().toString());;

// resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

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

return null;

}

return res;

}

//// Cheque List

@Override

public List<ChequeListRes> getChequeList(ChequeListReq req) {

List<ChequeListRes> resList = new ArrayList<ChequeListRes>();

ModelMapper mapper = new ModelMapper();

try {

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

int limit = StringUtils.isBlank(req.getLimit()) ? 0 : Integer.valueOf(req.getLimit());

int offset = StringUtils.isBlank(req.getOffset()) ? 10 : Integer.valueOf(req.getOffset());

// Between

int start = limit \* offset + 1;

int end = limit \* offset + offset;

List<Map<String, Object>> depositmaster = new ArrayList<Map<String, Object>>();

depositmaster = cbcrepository.GetChequeList(start,end);

for (Map<String, Object> data : depositmaster) {

ChequeListRes res = new ChequeListRes();

res.setChequeAmount(data.get("CHEQUE\_AMOUNT") == null ? "" : data.get("CHEQUE\_AMOUNT").toString());

res.setChequeBank(data.get("CHEQUE\_BANK") == null ? "" : data.get("CHEQUE\_BANK").toString());

res.setChequeBankName(data.get("CHEQUE\_BANK\_NAME") == null ? "" : data.get("CHEQUE\_BANK\_NAME").toString());

res.setChequeDate(data.get("CHEQUE\_DATE") == null ? "" : data.get("CHEQUE\_DATE").toString());

res.setChequeDate1(data.get("CHEQUE\_DATE1") == null ? "" : data.get("CHEQUE\_DATE1").toString());

res.setChequeNo(data.get("CHEQUE\_NO") == null ? "" : data.get("CHEQUE\_NO").toString());

res.setDepositNo(data.get("DEPOSIT\_NO") == null ? "" : data.get("DEPOSIT\_NO").toString());

res.setEffectiveDate(data.get("EFFECTIVE\_DATE") == null ? "" : data.get("EFFECTIVE\_DATE").toString());

res.setPaymentType(data.get("PAYMENT\_TYPE") == null ? "" : data.get("PAYMENT\_TYPE").toString());

res.setReceiptNo(data.get("RECEIPT\_NO") == null ? "" : data.get("RECEIPT\_NO").toString());

res.setRemarks(data.get("REMARKS") == null ? "" : data.get("REMARKS").toString());

res.setStatus(data.get("STATUS") == null ? "" : data.get("STATUS").toString());

res.setCbcNo(data.get("CBC\_NO") == null ? "" : data.get("CBC\_NO").toString());

resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

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

return null;

}

return resList;

}

}