**package** com.maan.marine.service.impl;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**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.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.maan.marine.error.Error;

**import** com.google.gson.Gson;

**import** com.maan.marine.repository.MenuRepository;

**import** com.maan.marine.req.QuoteNoReq;

**import** com.maan.marine.res.QuoteNoRes;

**import** com.maan.marine.service.KumarMenuService;

@Service

@Transactional

**public** **class** KumarMenuServiceImpl **implements** KumarMenuService{

@Autowired

**private** MenuRepository repository;

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

Gson json = **new** Gson();

/// Validation

@Override

**public** List<Error> validateSearch(QuoteNoReq req) {

// **TODO** Auto-generated method stub

List<Error> errors = **new** ArrayList<Error>();

**try**

{

String applicationid = StringUtils.*isBlank*(req.getApplicaitonId()) ? "" :req.getApplicaitonId();

//loginId

**if** (applicationid.equals("1")) {

**if** (StringUtils.*isBlank*(req.getLoginId())) {

errors.add(**new** Error("Please Enter LoginId", "loginid", "01"));

}

}

//Product Id

**if** (StringUtils.*isBlank*(req.getProductId())) {

errors.add(**new** Error("Please Enter ProductId", "productid", "02"));

}

//Search By

**if** (StringUtils.*isBlank*(req.getSearchBy())) {

errors.add(**new** Error("Please select Search By", "searchby", "03"));

}

//Search Value

**if** (StringUtils.*isBlank*(req.getSearchValue())) {

errors.add(**new** Error("Please Enter Search Value", "searchvalue", "04"));

}

}

**catch** (Exception e) {

e.printStackTrace();

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

**return** errors;

}

**return** errors;

}

///// Search

@Override

**public** List<QuoteNoRes> getAllQuoteNoSearch(QuoteNoReq req) {

List<QuoteNoRes> resList = **new** ArrayList<QuoteNoRes>();

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

**try** {

/// Quote No Search

**if** (req.getSearchValue().equalsIgnoreCase("58893"))

{

// Without Pagination

List<Map<String, Object>> quoteno = repository.getQuoteNo(req.getLoginId(), req.getProductId(), req.getApplicaitonId(),req.getOpenCoverNo(),req.getSearchValue());

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

QuoteNoRes res = **new** QuoteNoRes();

res.setApplicationNo(data.get("APPLICATION\_NO") == **null** ? "" : data.get("APPLICATION\_NO").toString());

res.setBrokerName(data.get("BROKER\_NAME") == **null** ? "" : data.get("BROKER\_NAME").toString());

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

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

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

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

res.setInceptionDate(data.get("INCEPTION\_DATE") == **null** ? "" :sdf.format( data.get("INCEPTION\_DATE")));

res.setLoginId(data.get("LOGIN\_ID") == **null** ? "" : data.get("LOGIN\_ID").toString());

res.setOpenCoverNo(data.get("NVL(A.OPEN\_COVER\_NO,'0')") == **null** ? "" : data.get("NVL(A.OPEN\_COVER\_NO,'0')").toString());

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

res.setPolicyStartDate(data.get("POLICY\_START\_DATE") == **null** ? "" : (data.get("POLICY\_START\_DATE").toString()));

res.setPremium(data.get("PREMIUM") == **null** ? "" : data.get("PREMIUM").toString());

res.setQuotationDate(data.get("QUOTATION\_DATE") == **null** ? "" :(data.get("QUOTATION\_DATE").toString()));

res.setQuoteno(data.get("QUOTE\_NO") == **null** ? "" : data.get("QUOTE\_NO").toString());

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

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

res.setValidityDate(data.get("VALIDITY\_DATE") == **null** ? "" :( data.get("VALIDITY\_DATE").toString()));

res.setLastName(data.get("") == **null** ? "" : data.get("").toString());

res.setExcessPremium(data.get("") == **null** ? "" : data.get("").toString());

res.setDebitNoteNo(data.get("") == **null** ? "" : data.get("").toString());

res.setCreditNoteNo(data.get("") == **null** ? "" : data.get("").toString());

res.setAmendId(data.get("") == **null** ? "" : data.get("").toString());

res.setOcCustId(data.get("") == **null** ? "" : data.get("").toString());

resList.add(res);

}

}

/// Policy No Search

**else** **if** (req.getSearchValue().equalsIgnoreCase("STS1001855000100"))

{

List<Map<String, Object>> policyno = repository.getPolicyNo(req.getBranchCode(), req.getProductId(),req.getOpenCoverNo(),req.getSearchValue());

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

QuoteNoRes res = **new** QuoteNoRes();

res.setApplicationNo(data.get("APPLICATION\_NO") == **null** ? "" : data.get("APPLICATION\_NO").toString());

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

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

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

res.setPremium(data.get("PREMIUM") == **null** ? "" : data.get("PREMIUM").toString());

res.setQuotationDate(data.get("QUOTATION\_DATE") == **null** ? "" : data.get("QUOTATION\_DATE").toString());

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

res.setLastName(data.get("LAST\_NAME") == **null** ? "" : data.get("LAST\_NAME").toString());

res.setLoginId(data.get("LOGIN\_ID") == **null** ? "" : data.get("LOGIN\_ID").toString());

res.setQuoteno(data.get("QUOTE\_NO") == **null** ? "" : data.get("QUOTE\_NO").toString());

res.setExcessPremium(data.get("EXCESS\_PREMIUM") == **null** ? "" : data.get("EXCESS\_PREMIUM").toString());

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

res.setInceptionDate(data.get("INCEPTION\_DATE") == **null** ? "" :sdf.format(data.get("INCEPTION\_DATE")));

res.setOpenCoverNo(data.get("OPEN\_COVER\_NO") == **null** ? "" : data.get("OPEN\_COVER\_NO").toString());

res.setDebitNoteNo(data.get("DEBIT\_NOTE\_NO") == **null** ? "" : data.get("DEBIT\_NOTE\_NO").toString());

res.setCreditNoteNo(data.get("CREDIT\_NOTE\_NO") == **null** ? "" : data.get("CREDIT\_NOTE\_NO").toString());

res.setAmendId(data.get("AMEND\_ID") == **null** ? "" : data.get("AMEND\_ID").toString());

res.setBrokerName(data.get("BROKER\_NAME") == **null** ? "" : data.get("BROKER\_NAME").toString());

res.setPolicyStartDate(data.get("POLICY\_START\_DATE") == **null** ? "" : data.get("POLICY\_START\_DATE").toString());

res.setOcCustId(data.get("OC\_CUST\_ID") == **null** ? "" : data.get("OC\_CUST\_ID").toString());

resList.add(res);

}

}

//// Customer Name Search

**else** **if** (req.getSearchValue().equalsIgnoreCase("TEST"))

{

List<Map<String, Object>> customerName = repository.getCustomerName(req.getLoginId(), req.getProductId(), req.getApplicaitonId(), req.getOpenCoverNo(), req.getSearchBy(),req.getSearchValue());

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

QuoteNoRes res = **new** QuoteNoRes();

res.setApplicationNo(data.get("APPLICATION\_NO") == **null** ? "" : data.get("APPLICATION\_NO").toString());

res.setQuoteno(data.get("QUOTE\_NO") == **null** ? "" : data.get("QUOTE\_NO").toString());

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

res.setQuotationDate(data.get("QUOTATION\_DATE") == **null** ? "" : data.get("QUOTATION\_DATE").toString());

res.setPolicyStartDate(data.get("POLICY\_START\_DATE") == **null** ? "" : data.get("POLICY\_START\_DATE").toString());

res.setValidityDate(data.get("VALIDITY\_DATE") == **null** ? "" : data.get("VALIDITY\_DATE").toString());

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

res.setLoginId(data.get("LOGIN\_ID") == **null** ? "" : data.get("LOGIN\_ID").toString());

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

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

res.setOpenCoverNo(data.get("NVL(M.OPEN\_COVER\_NO,'0')") == **null** ? "" : data.get("NVL(M.OPEN\_COVER\_NO,'0')").toString());

res.setPremium(data.get("PREMIUM") == **null** ? "" : data.get("PREMIUM").toString());

res.setInceptionDate(data.get("INCEPTION\_DATE") == **null** ? "" :sdf.format(data.get("INCEPTION\_DATE")));

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

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

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

res.setBrokerName(data.get("BROKER\_NAME") == **null** ? "" : data.get("BROKER\_NAME").toString());

resList.add(res);

}

}

}

**catch** (Exception e) {

e.printStackTrace();

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

**return** **null**;

}

**return** resList;

}

}