**Campaign Master**

package com.maan.crm.service.impl;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

import java.util.List;

import java.util.Optional;

import java.util.Random;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.transaction.Transactional;

import org.apache.commons.lang3.StringUtils;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.jsoup.internal.StringUtil;

import org.modelmapper.ModelMapper;

import org.modelmapper.internal.Errors;

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

import org.springframework.stereotype.Service;

import com.google.gson.Gson;

import com.maan.crm.bean.CampaignDetails;

import com.maan.crm.bean.CampaignDetailsId;

import com.maan.crm.bean.CampaignMaster;

import com.maan.crm.bean.CampaignMasterId;

import com.maan.crm.bean.ClientDetails;

import com.maan.crm.bean.LeadDetails;

import com.maan.crm.repository.CampaignDetailsRepository;

import com.maan.crm.repository.CampaignMasterRepository;

import com.maan.crm.repository.ClientDetailsRepository;

import com.maan.crm.repository.LeadQuoteDetailsRepository;

import com.maan.crm.repository.LeadRepository;

import com.maan.crm.req.CampaignDetailsGetAllReq;

import com.maan.crm.req.CampaignDetailsGetReq;

import com.maan.crm.req.CampaignDetailsReq;

import com.maan.crm.req.CampaignDetailsSaveReq;

import com.maan.crm.req.CampaignGetReq;

import com.maan.crm.req.CampaignMasterSaveReq;

import com.maan.crm.res.CampaignDetailsRes;

import com.maan.crm.res.CampaignMasterSaveRes;

import com.maan.crm.res.CampaignRes;

import com.maan.crm.res.CampaignSaveRes;

import com.maan.crm.service.CampaignService;

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

@Service

@Transactional

public class CampaignServiceImpl implements CampaignService {

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

Gson json = new Gson();

@Autowired

private CampaignDetailsRepository repo;

@Autowired

private CampaignMasterRepository masterrepo;

@Autowired

private ClientDetailsRepository clientrepo;

@Autowired

private LeadRepository leadrepo;

// Get All Campaign Master

@Override

public List<CampaignDetailsRes> campaignDetails(CampaignDetailsReq req) {

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

ModelMapper mapper = new ModelMapper();

try {

Date today = new Date();

Calendar cal = new GregorianCalendar();

cal.setTime(today);

cal.set(Calendar.HOUR\_OF\_DAY, 0);

cal.set(Calendar.MINUTE, 0);

today = cal.getTime();

String status = "Y";

List<CampaignMaster> campaigndetails = masterrepo

.findByEndDateGreaterThanEqualAndStatusOrderByStartDateAsc(today, status);

for (CampaignMaster data : campaigndetails) {

CampaignDetailsRes res = new CampaignDetailsRes();

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

resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return resList;

}

// Validation for Campaign Master

@Override

public List<Error> validateCampaignMaster(CampaignMasterSaveReq req) {

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

try {

if (req.getBranchCode() == null || StringUtils.isBlank(req.getBranchCode())) {

errors.add(new Error("01", "Branch Code", "Please Enter Branch Code"));

} else if (req.getBranchCode().length() > 20) {

errors.add(new Error("01", "Branch Code", "Please Enter Branch Code within 20 Characters"));

}

/\*

\* if(req.getCampaignId()==null ||

\* StringUtils.isBlank(req.getCampaignId().toString())) { errors.add(new

\* Error("02","Campaign Id", "Please Enter Campaign Id")); }

\*/ if (req.getCampaignName() == null || StringUtils.isBlank(req.getCampaignName())) {

errors.add(new Error("03", "Campaign Name", "Please Enter Campaign Name"));

} else if (req.getCampaignName().length() > 20) {

errors.add(new Error("03", "Campaign Name", "Please Enter Campaign Name within 20 Characters"));

}

if (req.getClassTypeId() == null || StringUtils.isBlank(req.getClassTypeId().toString())) {

errors.add(new Error("04", "Class Type Id", "Please Enter Class Type Id"));

}

if (req.getCreatedBy() == null || StringUtils.isBlank(req.getCreatedBy())) {

errors.add(new Error("05", "Created By", "Please Enter Created By"));

} else if (req.getCreatedBy().length() > 100) {

errors.add(new Error("05", "Created By", "Please Enter Created By within 100 Characters"));

}

if (req.getDescription() == null || StringUtils.isBlank(req.getDescription())) {

errors.add(new Error("06", "Description", "Please Enter Description"));

} else if (req.getDescription().length() > 300) {

errors.add(new Error("06", "Description", "Please Enter Description within 300 Characters"));

}

if (StringUtils.isBlank(req.getEndDate().toString())) {

errors.add(new Error("07", "End Date", "Please Enter End Date"));

}

if (req.getInsCompanyId() == null || StringUtils.isBlank(req.getInsCompanyId())) {

errors.add(new Error("08", "Insurance Id", "Please Enter Insurance Id"));

} else if (req.getInsCompanyId().length() > 10) {

errors.add(new Error("08", "Insurance Id", "Please Enter Insurance Id within 10 Characters"));

}

if (req.getOffer() == null || StringUtils.isBlank(req.getOffer())) {

errors.add(new Error("09", "Offer", "Please Enter Offer"));

} else if (req.getOffer().length() > 10) {

errors.add(new Error("09", "Offer", "Please Enter Offer within 10 Characters"));

}

if (req.getPolicyTypeId() == null || StringUtils.isBlank(req.getPolicyTypeId().toString())) {

errors.add(new Error("10", "Policy Type Id", "Please Enter Policy Type Id"));

}

if (req.getProduct() == null || StringUtils.isBlank(req.getProduct())) {

errors.add(new Error("11", "Product", "Please Enter Product Id"));

} else if (req.getProduct().length() > 100) {

errors.add(new Error("11", "Product", "Please Enter Product within 100 Characters"));

}

if (req.getPromoCode() == null || StringUtils.isBlank(req.getPromoCode())) {

errors.add(new Error("12", "Promo Code", "Please Enter Promocode"));

} else if (req.getPromoCode().length() > 100) {

errors.add(new Error("12", "PromoCode", "Please Enter Promocode within 100 Characters"));

}

if (StringUtils.isBlank(req.getStartDate().toString())) {

errors.add(new Error("13", "Start Date", "Please Enter Start Date"));

}

if (req.getLandingPageLink() == null || StringUtils.isBlank(req.getLandingPageLink())) {

errors.add(new Error("14", "Landing Page Link", "Please Enter Landing Page Link"));

} else if (req.getLandingPageLink().length() > 1000) {

errors.add(

new Error("14", "Landing Page Link", "Please Enter Landing Page Link within 1000 Characters"));

}

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return errors;

}

return errors;

}

// Save Campaign Master

@Override

public CampaignSaveRes saveCampaignMaster(CampaignMasterSaveReq req) {

CampaignSaveRes res = new CampaignSaveRes();

ModelMapper mapper = new ModelMapper();

CampaignMaster entity = new CampaignMaster();

Date EntryDate = null;

try {

CampaignMasterId id = new CampaignMasterId();

id.setCampaignId(req.getCampaignId());

Optional<CampaignMaster> data = masterrepo.findById(id);

// update

if (data.isPresent()) {

CampaignMaster ent = mapper.map(req, CampaignMaster.class);

ent.setCampaignId(req.getCampaignId());

masterrepo.save(ent);

res.setCode(req.getCampaignId().toString());

res.setResponse("Updated Successful");

}

// Insert

else {

Long count = masterrepo.count();

Long Id = 1001 + count;

String campId = "Camp-" + Id;

CampaignMaster ent = mapper.map(req, CampaignMaster.class);

ent.setCampaignId(campId);

ent.setEntryDate(new Date());

masterrepo.save(ent);

res.setCode(campId.toString());

res.setResponse("Inserted Successful");

;

}

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return res;

}

// Get Campaign Master

@Override

public CampaignDetailsRes campaign(CampaignDetailsGetReq req) {

CampaignDetailsRes res = new CampaignDetailsRes();

ModelMapper mapper = new ModelMapper();

try {

Date today = new Date();

Calendar cal = new GregorianCalendar();

cal.setTime(today);

cal.set(Calendar.HOUR\_OF\_DAY, 0);

cal.set(Calendar.MINUTE, 0);

today = cal.getTime();

String status = "Y";

CampaignMaster data = masterrepo.findByCampaignId(req.getCampaignId());

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

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return res;

}

// Validation for Campaign Details

@Override

public List<Error> validateCampaignDetails(CampaignDetailsSaveReq req) {

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

try {

if (req.getClientName() == null || StringUtils.isBlank(req.getClientName())) {

errors.add(new Error("01", "Branch Code", "Please Enter Branch Code"));

} else if (req.getClientName().length() > 100) {

errors.add(new Error("01", "Client Name", "Please Enter Client Name within 100 Characters"));

}

/\*

\* if(req.getClientRefNo() ==null || StringUtils.isBlank(req.getClientRefNo()))

\* { errors.add(new Error("02","Client Ref No", "Please Enter Client Ref No"));

\* } else if(req.getClientRefNo().length()>100) { errors.add(new

\* Error("02","Client Ref No","Please Enter Client Ref No within 100 Characters"

\* )); } if(req.getCrno()==null || StringUtils.isBlank(req.getCrno())) {

\* errors.add(new Error("03","Crno", "Please Enter Crno")); } else

\* if(req.getCrno().length()>100) { errors.add(new

\* Error("03","Crno","Please Enter Crno within 100 Characters" )); }

\*

\*/

String regex = "^[a-zA-Z0-9\_+&\*-]+(?:\\." + "[a-zA-Z0-9\_+&\*-]+)\*@" + "(?:[a-zA-Z0-9-]+\\.)+[a-z" + "A-Z]{2,7}$";

Pattern pattern = Pattern.compile(regex);

if (req.getEmailId() == null || StringUtils.isBlank(req.getEmailId())) {

errors.add(new Error("04", "EmailId", "Please Enter EmailId"));

} else if (req.getEmailId().length() > 100) {

errors.add(new Error("04", "EmailId", "Please Enter EmailId within 100 Characters"));

}

else if(StringUtils.isNotBlank(req.getEmailId())) {

Matcher matcher3 = pattern.matcher(req.getEmailId());

if (!matcher3.matches())

errors.add(new Error("04", "Email Id", "please Enter Valid Email Id"));

}

CampaignDetails search = repo.findByMobileNumberAndCampaignId(req.getMobileNumber(),req.getCampaignId());

if (req.getMobileNumber() == null || StringUtils.isBlank(req.getMobileNumber())) {

errors.add(new Error("05", "MobileNumber", "Please Enter MobileNumber"));

} else if (req.getMobileNumber().length() > 10) {

errors.add(new Error("05", "MobileNumber", "Please Enter MobileNumber within 10 Characters"));

}

else if(req.getMobileNumber().equalsIgnoreCase(search.getMobileNumber())){

errors.add(new Error("05","Mobile Number", "This mobile Number is already register for this campaign"));

}

if (req.getOccupation() == null || StringUtils.isBlank(req.getOccupation())) {

errors.add(new Error("06", "Occupation", "Please Enter Occupation"));

} else if (req.getOccupation().length() > 100) {

errors.add(new Error("06", "Occupation", "Please Enter Occupation within 100 Characters"));

}

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return errors;

}

return errors;

}

// Save Campaign Details

@Override

public CampaignMasterSaveRes saveCampaignDetails(CampaignDetailsSaveReq req) {

CampaignMasterSaveRes res = new CampaignMasterSaveRes();

ModelMapper mapper = new ModelMapper();

CampaignDetails entity = new CampaignDetails();

try {

String clientRefNo = "";

ClientDetails data = clientrepo.findByMobileNumber(req.getMobileNumber());

CampaignMaster campdata = masterrepo.findByCampaignId(req.getCampaignId());

//CLient Save

if (data == null) {

Long newId = clientrepo.count() + 1001;

Random rnd = new Random();

int number = rnd.nextInt(100);

String randomNo = String.format("%02d", number);

clientRefNo = "C/" + newId + "/" + randomNo;

ClientDetails ent = mapper.map(req, ClientDetails.class);

ent.setBranchCode(campdata.getBranchCode());

ent.setInsCompanyId(campdata.getInsCompanyId());

ent.setCreatedBy(campdata.getCreatedBy());

ent.setClientRefNo(clientRefNo);

ent.setCrno(req.getCrno());

clientrepo.save(ent);

res.setClientRefNo(clientRefNo);

res.setResponse("Inserted Successful");

} else {

clientRefNo = data.getClientRefNo();

res.setClientRefNo(clientRefNo);

}

// Lead Save

Long count = leadrepo.count();

Random rnd1 = new Random();

int number1 = rnd1.nextInt(100);

String randomNo1 = String.format("%02d", number1);

String leadId = "100";

Long Id =count +100;

leadId = Id.toString()+ randomNo1;

LeadDetails lead = mapper.map(campdata, LeadDetails.class);

lead.setLeadId(Integer.valueOf(leadId));

lead.setClientRefNo(clientRefNo);

leadrepo.save(lead);

res.setLeadId(leadId);

res.setResponse("Inserted Successful");

// Campaign Save

CampaignDetailsId id = new CampaignDetailsId();

String customerId = "";

Long newId2 = repo.count() + 1001;

customerId = newId2.toString() ;

CampaignDetails ent = mapper.map(req, CampaignDetails.class);

ent.setCustomerId(customerId);

ent.setClientRefNo(clientRefNo);

ent.setLeadId(Integer.valueOf(leadId));;

repo.save(ent);

res.setCampaigncode(customerId);

res.setResponse("Inserted Successful");

}catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return res;

}

// Get Campaign Details

@Override

public CampaignRes getcampaign(CampaignGetReq req) {

CampaignRes res = new CampaignRes();

ModelMapper mapper = new ModelMapper();

try {

CampaignDetails data = repo.findByMobileNumber(req.getMobileNumber());

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

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return res;

}

// Get All Campaign Details

@Override

public List<CampaignRes> getallcampaign(CampaignDetailsGetAllReq req) {

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

ModelMapper mapper = new ModelMapper();

try {

List<CampaignDetails> campaigndetails = repo.findByCampaignId(req.getCampaignId());

for (CampaignDetails data : campaigndetails) {

CampaignRes res = new CampaignRes();

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

resList.add(res);

}

} catch (Exception e) {

e.printStackTrace();

log.info("Log Details", e.getMessage());

return null;

}

return resList;

}

}