**Pagination**

**Pagination Request Class**

package com.maan.crm.req;

import java.io.Serializable;

import lombok.\*;

import javax.persistence.\*;

import com.fasterxml.jackson.annotation.JsonProperty;

@Getter

@Setter

@NoArgsConstructor

@AllArgsConstructor

@ToString

public class CrmBranchMasterPaginationReq {

@JsonProperty("Limit")

private String limit;

@JsonProperty("Offset")

private String offset;

}

**In Controller**

////////Pagination

@PostMapping("/master/getbranchmaster")

**public** ResponseEntity<CommonCrmRes> getAllPaginationCrmBranchMaster(@RequestBody CrmBranchMasterPaginationReq req) {

CommonCrmRes data = **new** CommonCrmRes();

// Get All

List<CrmBranchMasterRes> res = entityService.getAllPaginationCrmBranchMaster(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***);

}

}

**In Service Class**

List<CrmBranchMasterRes> getAllPaginationCrmBranchMaster(CrmBranchMasterPaginationReq req);

**In service Implementation Class**

/////// Pagination

@Override

**public** List<CrmBranchMasterRes> getAllPaginationCrmBranchMaster(CrmBranchMasterPaginationReq req) {

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

ModelMapper mapper = **new** ModelMapper();

**try** {

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

Pageable paging = PageRequest.*of*(Integer.*valueOf*(req.getLimit()), Integer.*valueOf*(req.getOffset()), Sort.*by*("entryDate").descending());

Page<CrmBranchMaster> crmbranchmaster = repository.findAll(paging);

// Page<CrmBranchMaster> crmbranchmaster = repository.findByStatus(paging , "Y");

**for** (CrmBranchMaster data : crmbranchmaster) {

CrmBranchMasterRes res = **new** CrmBranchMasterRes();

res.setEffectiveDate(data.getEffectiveDate()==**null**?"":sdf.format( data.getEffectiveDate()));

res.setAddress1(data.getAddress1()== **null** ?"": data.getAddress1());

res.setAddress2(data.getAddress2() == **null** ?"": data.getAddress2());;

res.setAddress3(data.getAddress3() == **null** ?"": data.getAddress3());

res.setBelongingBranch(data.getBelongingBranch()==**null** ? "" : data.getBelongingBranch());

res.setBelongingType(data.getBelongingType() ==**null** ? "" : data.getBelongingType());

res.setBranchCode(data.getBranchCode()== **null** ?"" : decimalFormat.format(data.getBranchCode()));

res.setBranchName(data.getBranchName()== **null** ?"" : data.getBranchName());

res.setBranchPrefix(data.getBranchPrefix() == **null** ?"": data.getBranchPrefix());

res.setCity(data.getCity() == **null** ?"":data.getCity());

res.setCoreAppCode(data.getCoreAppCode() == **null**?"": data.getCoreAppCode());

res.setCountry(data.getCountry()==**null**?"":data.getCountry());

res.setCurrencyAbbreviation(data.getCurrencyAbbreviation()==**null** ?"": data.getCurrencyAbbreviation());

res.setCurrencyAcronym(data.getCurrencyAcronym()==**null** ?"": data.getCurrencyAcronym());

res.setCurrencyDecimalDigit(data.getCurrencyDecimalDigit()== **null** ?"": decimalFormat.format(data.getCurrencyDecimalDigit()));

res.setCurrencyDecimalName(data.getCurrencyDecimalName() == **null** ?"": data.getCurrencyDecimalName());

res.setCurrencyName(data.getCurrencyName()== **null** ?"": data.getCurrencyName());

res.setDecimalPlaces(data.getDecimalPlaces()== **null** ?"":decimalFormat.format(data.getDecimalPlaces()));

res.setDeptCode(data.getDeptCode()==**null** ?"": data.getDeptCode());

res.setDestinationCountryId(data.getDestinationCountryId() == **null** ?"":decimalFormat.format(data.getDestinationCountryId()));

res.setEmail(data.getEmail()==**null** ?"": data.getEmail());

res.setEntryDate(data.getEntryDate()==**null** ?"": sdf.format(data.getEntryDate()));

res.setExchangeId(data.getExchangeId()==**null** ?"":decimalFormat.format(data.getExchangeId()));

res.setFax(data.getFax()==**null** ?"":decimalFormat.format(data.getFax()));

res.setFooterImg(data.getFooterImg()==**null** ?"": data.getFooterImg());

res.setHeaderImg(data.getHeaderImg()==**null** ?"": data.getHeaderImg());

res.setInsCompanyId(data.getInsCompanyId()==**null**?"":decimalFormat.format(data.getInsCompanyId()));

res.setLongitudeYn(data.getLongitudeYn()==**null** ?"": data.getLongitudeYn());

res.setOriginationCountryId(data.getOriginationCountryId()==**null**?"" :decimalFormat.format(data.getOriginationCountryId()));

res.setPckeyCloseTrn(data.getPckeyCloseTrn()==**null** ?"": data.getPckeyCloseTrn());

res.setPhone(data.getPhone()==**null** ?"": data.getPhone());

res.setRegionCode(data.getRegionCode() == **null** ?"": data.getRegionCode());

res.setRemarks(data.getRemarks()==**null** ?"": data.getRemarks());

res.setSignImg(data.getSignImg()==**null** ?"": data.getSignImg());

res.setStamp(data.getStamp()== **null** ?"": data.getStamp());

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

res.setTax(data.getTax()== **null** ?"": decimalFormat.format(data.getTax()));

resList.add(res);

}

} **catch** (Exception e) {

e.printStackTrace();

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

**return** **null**;

}

**return** resList;

}

}