package com.sdocean.common.service;

import java.util.ArrayList;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.SynthQueryDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.device.dao.DeviceDao;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.dictionary.dao.PublicDao;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.dao.SysLoginLogDao;

import com.sdocean.log.model.SysLoginLogModel;

import com.sdocean.page.model.NgColumn;

import com.sdocean.page.model.UiColumn;

import com.sdocean.position.dao.SysPositionDao;

import com.sdocean.position.model.SysPosition;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class CommonService {

@Autowired

private DeviceDao deviceDao;

/\*

\* 将 deviceid#indicatorid格式转换为

\* list<deviceModel> 格式

\*/

public List<DeviceModel> indicatoridsToDevices(String indicatorIds){

String[] ids = indicatorIds.split(",");

Map<String, String> deids = new HashMap<String, String>();

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

for(String id:ids){

String indicatorid = id.substring(0, id.indexOf("#"));

String indicatorname = id.substring(id.indexOf("#")+1,id.length());

if(indicatorid!=null&&!indicatorid.equals("0")){

DeviceModel device = new DeviceModel();

device.setId(Integer.parseInt(indicatorid));

device.setName(indicatorname);

list.add(device);

}

}

return list;

}

}

package com.sdocean.common.service;

import java.util.ArrayList;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.SynthQueryDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.device.dao.DeviceDao;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.dictionary.dao.PublicDao;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.dao.SysLoginLogDao;

import com.sdocean.log.model.SysLoginLogModel;

import com.sdocean.page.model.NgColumn;

import com.sdocean.page.model.UiColumn;

import com.sdocean.position.dao.SysPositionDao;

import com.sdocean.position.model.SysPosition;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class CommonService {

@Autowired

private DeviceDao deviceDao;

/\*

\* 将 deviceid#indicatorid格式转换为

\* list<deviceModel> 格式

\*/

public List<DeviceModel> indicatoridsToDevices(String indicatorIds){

String[] ids = indicatorIds.split(",");

Map<String, String> deids = new HashMap<String, String>();

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

for(String id:ids){

String indicatorid = id.substring(0, id.indexOf("#"));

String indicatorname = id.substring(id.indexOf("#")+1,id.length());

if(indicatorid!=null&&!indicatorid.equals("0")){

DeviceModel device = new DeviceModel();

device.setId(Integer.parseInt(indicatorid));

device.setName(indicatorname);

list.add(device);

}

}

return list;

}

}

package com.sdocean.company.model;

public class CompanyModel {

private int id;

private String code;

private String pcode;

private String name;

private String shortName;

private int level;

private int isactive;

private String isactiveName;

private String orderCode;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getPcode() {

return pcode;

}

public void setPcode(String pcode) {

this.pcode = pcode;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getShortName() {

return shortName;

}

public void setShortName(String shortName) {

this.shortName = shortName;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

@Override

public boolean equals(Object obj) {

CompanyModel comp = (CompanyModel) obj;

if(this.getCode().equals(comp.getCode())){

return true;

}else{

return false;

}

// TODO Auto-generated method stub

//return super.equals(obj);

}

}

package com.sdocean.company.model;

public class SysCompanyModel {

private int id;

private String code;

private String name;

private String shortName;

private int regionId;

private String regionName;

private int isactive;

private String isactiveName;

private String orderCode;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getShortName() {

return shortName;

}

public void setShortName(String shortName) {

this.shortName = shortName;

}

public int getRegionId() {

return regionId;

}

public void setRegionId(int regionId) {

this.regionId = regionId;

}

public String getRegionName() {

return regionName;

}

public void setRegionName(String regionName) {

this.regionName = regionName;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

}

package com.sdocean.dataQuery.dao;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import java.util.Map;

import javax.annotation.Resource;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.model.AiotModel;

import com.sdocean.dataQuery.model.DataChangeModel;

import com.sdocean.dataQuery.model.DataImportModel;

import com.sdocean.dataQuery.model.DataQueryDataModel;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.firstpage.model.Ddata;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.firstpage.model.MetaData4FirstPage;

import com.sdocean.firstpage.model.MetaFirstPage;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.indicator.dao.IndicatorDao;

import com.sdocean.indicator.model.IndicatorModel;

import com.sdocean.metadata.dao.MetadataDao;

import com.sdocean.metadata.dao.MetadataTableDao;

import com.sdocean.metadata.model.HalfHour;

import com.sdocean.metadata.model.MetadataModel;

import com.sdocean.metadata.model.MetadataTable;

import com.sdocean.metadata.model.SyndataModel;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.dao.StationDao;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Component

public class DataQueryDao extends OracleEngine{

@Resource

MetadataTableDao tableDao;

@Resource

IndicatorDao indicatorDao;

@Resource

StationDao stationDao;

@Resource

MetadataDao metaDataDao;

/\*

\* 为实时数据添加表头

\*/

public List<UiColumn> getCols4DataQuery(DataQueryModel model){

List<UiColumn> cols = new ArrayList<UiColumn>();

StringBuffer sql = new StringBuffer("");

sql.append(" select '时间' as displayName,'collect\_time' as field,'true' as visible,'\*' as width,'' as cellFilter");

sql.append(" union all ");

sql.append(" select distinct case when b.logo is null or length(b.logo) = 0 then a.title else concat(a.title,'(',b.logo,')') end as displayName,");

sql.append(" a.code as field,'true' as visible,'\*' as width,concat('number:',c.pointNum) as cellFilter");

sql.append(" from dm\_indicator a left join g\_unit b ");

sql.append(" on a.unitid = b.id and b.isactive = 1");

sql.append(" ,view\_stationid\_deviceid\_indicatorid c ");

sql.append(" where a.id = c.indicatorid");

sql.append(" and a.isactive = 1");

sql.append(" and c.deviceid =").append(model.getDeviceId());

sql.append(" and c.stationid =").append(model.getStationId());

cols = this.queryObjectList(sql.toString(), UiColumn.class);

return cols;

}

/\*

\* 为实时数据查询提供结果

\*/

public List<Map<String, Object>> getRows4DataQuery(DataQueryModel model){

List<Map<String, Object>> rows = null;

StationModel station = new StationModel();

station.setId(model.getStationId());

//根据起始时间判断出需要查询的表名的集合

List<MetadataTable> tables = tableDao.getTables4Meta(station,model.getBeginDate(), model.getEndDate(), 1);

//查询出当前设备下的参数列表

DeviceModel device = new DeviceModel();

device.setId(model.getDeviceId());

List<IndicatorModel> indicators = indicatorDao.getIndicators4Deivce(device);

//定义总的SQL语句

StringBuffer sql = new StringBuffer("");

//定义查询参数

StringBuffer indicatorsql = new StringBuffer("");

StringBuffer indicatorGroupBySql = new StringBuffer("");

indicatorGroupBySql.append(" group by collect\_time having ( 0= 1");

for(IndicatorModel indicator:indicators){

indicatorsql.append(" ,sum(if(indicator\_code='").append(indicator.getCode()).append("',data,0)) as ").append(indicator.getCode());

indicatorGroupBySql.append(" ||sum(if(indicator\_code='").append(indicator.getCode()).append("',data,0)) <> 0");

}

indicatorGroupBySql.append(" )");

//定义where查询条件

StringBuffer wheresql = new StringBuffer(" where ");

wheresql.append("wpid = ").append(model.getStationId()).append(" and deviceid =").append(model.getDeviceId());

//增加时间参数.

if(model.getBeginDate()!=null){

wheresql.append(" and collect\_time >= '").append(model.getBeginDate()).append("'");

}

if(model.getEndDate()!=null){

wheresql.append(" and collect\_time <= '").append(model.getEndDate()).append("'");

}

//定义排序语句

StringBuffer ordersql = new StringBuffer(" order by collect\_time desc");

for(int i=0;i<tables.size();i++){

StringBuffer selectSql = new StringBuffer("select collect\_time");

MetadataTable table = tables.get(i);

String tableName = table.getTableName();

StringBuffer fromSql = new StringBuffer(" from ");

fromSql.append(tableName);

//生成每个表的查询语句

selectSql.append(indicatorsql).append(fromSql).append(wheresql).append(indicatorGroupBySql);

//将每个表的sql语句添加到总语句中

sql.append(selectSql);

if(i<tables.size()-1){

sql.append(" union all ");

}

}

sql.append(ordersql);

rows = this.queryForList(sql.toString());

return rows;

}

/\*

\* 查询出当前站点信息的最后实时数据

\* 为首页展示

\*/

public List<MetaFirstPage> getData4FirstPage(StationModel station){

//初始化返回结果

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

//获得站点下的1级设备列表

List<DeviceModel> devices1 = new ArrayList<DeviceModel>();

StringBuffer device1Sql = new StringBuffer("");

device1Sql.append(" select b.id,b.name");

device1Sql.append(" from map\_awp\_device\_catalog a,device\_catalog b");

device1Sql.append(" where a.device\_catalog\_id = b.id and a.aiot\_watch\_point\_id = ").append(station.getId());

devices1 = this.queryObjectList(device1Sql.toString(), DeviceModel.class);

List<DeviceModel> devices = new ArrayList<DeviceModel>();

//获得一级设备下的2级设备列表

for(DeviceModel dev:devices1){

List<DeviceModel> devices2 = new ArrayList<DeviceModel>();

StringBuffer device2Sql= new StringBuffer("");

device2Sql.append(" select b.id,b.name");

device2Sql.append(" from map\_device\_device a,device\_catalog b");

device2Sql.append(" where a.cdeviceid = b.id and a.pdeviceid = ").append(dev.getId());

devices2 = this.queryObjectList(device2Sql.toString(), DeviceModel.class);

if(devices2!=null&&devices2.size()>0){

devices.addAll(devices2);

}

}

devices1.addAll(devices);

for(DeviceModel dev:devices1){

List<MetaFirstPage> meta = new ArrayList<MetaFirstPage>();

StringBuffer sql = new StringBuffer();

sql.append(" select '").append(station.getTitle()).append("' as stationname,indicatorcode,collecttime,data,unit");

sql.append(" from aiot\_device\_data");

sql.append(" where deviceid = ").append(dev.getId());

meta = this.queryObjectList(sql.toString(), MetaFirstPage.class);

list.addAll(meta);

}

//根据设备列表得到实时数据

return list;

}

//通过站点,起始时间,类型,设备,以及参数查询出实时数据

public Ddata getDataByWDTDI(StationModel station,String nowDate,int type,LastMetaData device,MetaData4FirstPage indicator){

Ddata ddata = null;

MetadataTable table = tableDao.getOneTable(station, nowDate, type);

String tableName = table.getTableName();

StringBuffer dsql = new StringBuffer("");

dsql.append("select collect\_time as lastTime, data from ").append(tableName);

dsql.append(" where wpid = ").append(station.getId()).append(" and indicator\_code ='").append(indicator.getIndicatorCode()).append("'");

dsql.append(" and data is not null and data <> 88888");

dsql.append(" and deviceid = ").append(device.getDeviceId());

dsql.append(" order by collect\_time desc limit 1");

ddata = this.queryObject(dsql.toString(), Ddata.class);

return ddata;

}

/\*

\* 为数据修改的查询提供结果

\*/

public List<MetadataModel> getResult4DataChangeshow(DataChangeModel model){

List<MetadataModel> rows = new ArrayList<MetadataModel>();

//通过站点以及查询的开始以及结束时间,判断要查询的表

List<MetadataTable> tables = tableDao.getTables4Meta(model.getStation(),model.getBeginDate(), model.getEndDate(), 1);

StringBuffer sql = new StringBuffer(""); //定义总的SQL语句

StringBuffer selectSql = new StringBuffer("");

selectSql.append(" select id,collect\_time,data ");

StringBuffer whereSql = new StringBuffer("");

whereSql.append(" where wpid = 10007");

whereSql.append(" and deviceid = 44");

whereSql.append(" and indicator\_Code = 'WaterTemp'");

whereSql.append(" and collect\_time >= '").append(model.getBeginDate()).append("'");

whereSql.append(" and collect\_time <= '").append(model.getEndDate()).append("'");

StringBuffer orderSql = new StringBuffer("");

orderSql.append(" order by collect\_time desc");

for(int i=0;i<tables.size();i++){

StringBuffer fromSql = new StringBuffer(" from ");

fromSql.append(" aiot\_metadata\_10007\_16\_4 ");

sql.append(selectSql).append(fromSql).append(whereSql);

if(i<tables.size()-1){

sql.append(" union all ");

}

}

sql.append(orderSql);

rows = this.queryObjectList(sql.toString(), MetadataModel.class);

return rows;

}

/\*

\* 保存修改

\*/

public Result saveChangeData(DataChangeModel model){

//初始化返回结果

Result result = new Result();

result.setDotype(Result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setMessage("修改成功");

result.setResult(Result.SUCCESS);

//根据站点ID获得站点信息

StationModel station = stationDao.getStationById(model.getStationId());

//更改元数据内容

StringBuffer msql = new StringBuffer("");

//获得该元数据存放的表

MetadataTable mtable = tableDao.getOneTable(station, model.getCollect\_time(), 1);

msql.append("insert into ").append(mtable.getTableName()).append(" (collect\_time,collect\_type,wpid,indicator\_code,data,deviceid)");

msql.append(" values(?,?,?,?,?,?) on duplicate key update data=values(data)");

Object[] params = new Object[]{

model.getCollect\_time(),1,model.getStationId(),model.getIndicatorCode(),model.getNewData(),model.getDeviceId()

};

int mres = 0;

/\*try {

mres = this.update(msql.toString(), params);

} catch (Exception e) {

// TODO: handle exception

result.setMessage("修改元数据失败");

result.setResult(Result.FAILED);

return result;

}\*/

//修改综合元数据

//获得该数据的综合元数据的表

MetadataTable stable = tableDao.getOneTable(station, model.getCollect\_time(), 2);

//获得该条记录在综合元数据中对应的时间

HalfHour hh = metaDataDao.getSynDateByMdate(model.getCollect\_time());

//根据时间以及各个参数,查询出在综合元数据表中的记录

SyndataModel syn = new SyndataModel();

StringBuffer csql = new StringBuffer("");

csql.append(" select collect\_time,indicator\_code as indicatorCode,data,wpid,collect\_type,deviceid");

csql.append(" from ").append(stable.getTableName());

csql.append(" where collect\_time ='").append(hh.getCollectTime()).append("'");

csql.append(" and wpid = ").append(model.getStationId()).append(" and deviceid = ").append(model.getDeviceId());

csql.append(" and indicator\_code = '").append(model.getIndicatorCode()).append("'");

try {

syn = this.queryObject(csql.toString(), SyndataModel.class);

} catch (Exception e) {

// TODO: handle exception

result.setMessage("匹配综合元数据失败");

result.setResult(Result.FAILED);

return result;

}

if(syn==null||(syn.getData()==model.getOldData())){

StringBuffer isql = new StringBuffer("");

isql.append(" insert into ").append(stable.getTableName()).append("(collect\_time,indicator\_code,data,wpid,collect\_type,deviceid)");

isql.append(" values(?,?,?,?,?,?) on duplicate key update data=values(data)");

Object[] iparams = new Object[]{

hh.getCollectTime(),model.getIndicatorCode(),

model.getNewData(),model.getStationId(),1,model.getDeviceId()

};

int sres = 0;

/\*try {

sres = this.update(isql.toString(), iparams);

} catch (Exception e) {

// TODO: handle exception

result.setMessage("更新综合元数据失败");

result.setResult(Result.FAILED);

}\*/

}

return result;

}

/\*

\* 将导入数据保存到数据库中

\*/

public Result saveImportData(List<DataImportModel> list,SysUser user){

//初始化返回结果

Result result = new Result();

//生成版本信息号

SimpleDateFormat sdf = new SimpleDateFormat("yyMMddHHmmss");

Date now = new Date();

String nowString = sdf.format(now);

String visionData = "T2"+"t"+nowString;

StringBuffer vision =new StringBuffer("");

vision.append(" insert into aiot\_meta\_datavision(code,userid)");

vision.append(" values('").append(visionData).append("',").append(user.getId()).append(")");

try {

this.update(vision.toString(), null);

} catch (Exception e) {

// TODO: handle exception

result.setMessage("保存版本号错误");

return result;

}

for(DataImportModel data:list){

//通过站点ID以及采集时间,确定需要查询的表

StationModel station = new StationModel();

station.setId(data.getStationId());

MetadataTable metaTable = tableDao.getOneTable(station, data.getCollect\_time(), 1);

MetadataTable synTable = tableDao.getOneTable(station, data.getCollect\_time(), 2);

//

}

return result;

}

/\*

\* 测试mysql的存储过程

\*/

public void queryData(){

List<Map<String, Object>> result = this.procForList("call queryData('pH','2016-07-01','2016-07-02');");

}

/\*

\* 获得某个设备的表头

\*/

public String getCols4DataQuery(int deviceid){

String cols = "";

StringBuffer sql = new StringBuffer("");

sql.append("select data from map\_device\_data where type = 1 and deviceid = ").append(deviceid).append(" limit 1");

cols = this.queryForString(sql.toString(), null);

return cols;

}

/\*

\* 获得某个设备的内容

\*/

public List<Map<String, Object>> getRows4DataQuery(int deviceid){

List<Map<String, Object>> rows = null;

StringBuffer sql = new StringBuffer("");

sql.append(" select id,deviceid,indicatorcode,collecttime,data,unit ");

sql.append(" from aiot\_device\_data where deviceid = ").append(deviceid);

rows = this.queryForList(sql.toString());

return rows;

}

/\*

\* 获得某个设备的图片合集

\*/

public List<DataQueryDataModel> getImages4DataQuery(int deviceid){

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

StringBuffer sql= new StringBuffer();

sql.append("select data from map\_device\_data where type = 3 and deviceid = ").append(deviceid);

list = this.queryObjectList(sql.toString(), DataQueryDataModel.class);

return list;

}

/\*

\*

\*/

public List<AiotModel> getAiotDatas(int deviceid){

List<AiotModel> data = new ArrayList<AiotModel>();

StringBuffer sql= new StringBuffer();

sql.append(" select id,deviceid,indicatorcode,collecttime,data,unit ");

sql.append(" from aiot\_device\_data where deviceid = ").append(deviceid);

data = this.queryObjectList(sql.toString(), AiotModel.class);

return data;

}

}

package com.sdocean.dataQuery.dao;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.LinkedList;

import java.util.List;

import java.util.Map;

import javax.annotation.Resource;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Echarts;

import com.sdocean.common.model.PlotLine;

import com.sdocean.common.model.YAxis;

import com.sdocean.dataQuery.model.GraphModel;

import com.sdocean.device.dao.DeviceDao;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.dictionary.action.WaterQualityStandardAction;

import com.sdocean.dictionary.dao.WaterQualityStandardDao;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.indicator.dao.IndicatorDao;

import com.sdocean.indicator.model.IndicatorModel;

import com.sdocean.metadata.dao.MetadataTableDao;

import com.sdocean.metadata.model.MetadataTable;

import com.sdocean.station.model.StationModel;

@Component

public class GraphQueryDao extends OracleEngine{

@Resource

MetadataTableDao tableDao;

@Resource

IndicatorDao indicatorDao;

@Resource

DeviceDao deviceDao;

@Resource

WaterQualityStandardDao wasDao;

public Echarts getEcharts4Graph(StationModel station,GraphModel model){

Map<String, Object> indicatorNameMap = new HashMap<String, Object>();

indicatorNameMap = indicatorDao.getIndicatorTitleMap();

Map<String, Object> indicatorUnitMap = new HashMap<String, Object>();

indicatorUnitMap = indicatorDao.getIndicatorUnitMap();

Echarts echarts = new Echarts();

//根据起始时间判断出需要查询的表名的集合

List<MetadataTable> tables = tableDao.getTables4Meta(station,model.getBeginDate(), model.getEndDate(), 2);

List<DeviceModel> devices = model.getDevices();

//遍历设备列表,得到模板设备

//初始化模板设备

DeviceModel device = new DeviceModel();

Integer interal = 0;

for(DeviceModel dev:devices){

dev = deviceDao.getInterval4device(station.getId(), dev);

int last = dev.getInterval();

if(last >= interal){

interal = last;

device = dev;

}

}

//定义总的sql语句

StringBuffer sql = new StringBuffer("");

//根据表名遍历

for(int i=0;i<tables.size();i++){

//得到需要查询的表名

String tableName = tables.get(i).getTableName();

//得到别名

String deviceName = "d"+device.getId();

//定义最外层的select部分

StringBuffer allseleSql = new StringBuffer("");

//定义最外层的from部分

StringBuffer allfromsql = new StringBuffer("");

//定义最外层的group部分

StringBuffer allGroupSql = new StringBuffer("");

allGroupSql.append(" group by ").append(deviceName).append(".collect\_time");

allfromsql.append(" from (");

allseleSql.append("select ").append(deviceName).append(".collect\_time");

//定义最外层的select中的其他部分

StringBuffer selectSql = new StringBuffer(""); //定义select部分

StringBuffer fromSql = new StringBuffer(""); //定义from部分

StringBuffer whereSql = new StringBuffer(""); //定义where部分

StringBuffer groupSql = new StringBuffer(""); //定义group部分

StringBuffer orderSql = new StringBuffer(""); //定义排序部分

selectSql.append(" select collect\_time");

fromSql.append(" from ").append(tableName);

whereSql.append(" where 1 =1");

if(model!=null&&model.getBeginDate()!=null&&model.getBeginDate().length()>0){

whereSql.append(" and collect\_time >= '").append(model.getBeginDate()).append("'");

}

if(model!=null&&model.getEndDate()!=null&&model.getEndDate().length()>0){

whereSql.append(" and collect\_time <= '").append(model.getEndDate()).append("'");

}

whereSql.append(" and data <> 0 and data <> 88888 and data <> 9999");

whereSql.append(" and wpid = ").append(station.getId());

//设备条件

StringBuffer whereDeviceSql = new StringBuffer("");

whereDeviceSql.append(" and deviceid =").append(device.getId());

groupSql.append(" group by collect\_time");

orderSql.append(" order by collect\_time desc");

//

//遍历模板设备下的参数

List<IndicatorModel> indicators = device.getIndicators();

//

//定义select条件查询

StringBuffer sumsql = new StringBuffer("");

//定义indicatorcode in 语句

StringBuffer insql = new StringBuffer(" and indicator\_code in ('0'");

for(IndicatorModel indicator:indicators){

sumsql.append(",sum(if(indicator\_code='").append(indicator.getCode()).append("',data,0)) as ").append(indicator.getCode());

insql.append(",'").append(indicator.getCode()).append("'");

allseleSql.append(",").append(deviceName).append(".").append(indicator.getCode());

}

insql.append(")");

//整合第一个子查询

StringBuffer csql = new StringBuffer("");

csql.append(selectSql).append(sumsql).append(fromSql).append(whereSql).append(whereDeviceSql).append(insql).append(groupSql);

//将第一个子查询整合到allfrom中

allfromsql.append(csql).append(") ").append(deviceName);

//遍历设备列表中的其他列表

for(DeviceModel dev:devices){

//去除当做模板的设备

if(dev.getId()!=device.getId()&&dev.getIndicators().size()>0){

//得到该设备的别名

String devname ="d"+dev.getId();

allseleSql.append(",max(").append(devname).append(".collect\_time) as ").append(devname).append("time");

//设备条件

StringBuffer whereDevSql = new StringBuffer("");

whereDevSql.append(" and deviceid = ").append(dev.getId());

//定义select条件查询

StringBuffer csumsql = new StringBuffer("");

//定义indicatorcode in 语句

StringBuffer cinsql = new StringBuffer(" and indicator\_code in ('0'");

//遍历该设备下的参数

List<IndicatorModel> cindicators = dev.getIndicators();

for(IndicatorModel indicator:cindicators){

csumsql.append(",sum(if(indicator\_code='").append(indicator.getCode()).append("',data,0)) as ").append(indicator.getCode());

cinsql.append(",'").append(indicator.getCode()).append("'");

allseleSql.append(", case when ").append(devname).append(".").append(indicator.getCode()).append(" is null then 0 else ").append(devname).append(".").append(indicator.getCode());

allseleSql.append(" end as ").append(indicator.getCode());

}

cinsql.append(")");

//定义该设备的子查询部分

StringBuffer cdevSql = new StringBuffer("");

cdevSql.append(" left join (");

cdevSql.append(selectSql).append(csumsql).append(fromSql).append(whereSql).append(whereDevSql).append(cinsql).append(groupSql);

cdevSql.append(orderSql).append(") ").append(devname).append(" on ").append(deviceName).append(".collect\_time >=").append(devname).append(".collect\_time");

//将该设备下的子查询部分整合到allfrom中

allfromsql.append(cdevSql);

}

}

sql.append(allseleSql).append(allfromsql).append(allGroupSql);

if(i<tables.size()-1){

sql.append(" union all ");

}

}

sql.append(" order by collect\_time");

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

rows = this.queryForList(sql.toString());

//定义四个YAxis备用

List<Object> xAxis = new ArrayList<Object>();

YAxis YAxis1 = new YAxis();

YAxis YAxis2 = new YAxis();

YAxis YAxis3 = new YAxis();

YAxis YAxis4 = new YAxis();

//循环遍历list

for(Map<String,Object> resmap:rows){

int i=0;

for(String key:resmap.keySet()){

if(i==0){

xAxis.add(resmap.get(key));

i++;

}else{

String fieldName = (String) indicatorNameMap.get(key);

List<PlotLine> plotLines = wasDao.getPlotLines(station, key);

if(fieldName!=null&&fieldName.length()>0){

if(i==1){ //将数据插入到YAxis1中

YAxis1.fieldName=(String) indicatorNameMap.get(key);

YAxis1.unit=(String) indicatorUnitMap.get(key);

YAxis1.yAxis2.add(resmap.get(key));

YAxis1.plotLines = plotLines;

}else if(i==2){ //将数据插入到YAxis2中

YAxis2.fieldName=(String) indicatorNameMap.get(key);

YAxis2.unit=(String) indicatorUnitMap.get(key);

YAxis2.yAxis2.add(resmap.get(key));

YAxis2.plotLines = plotLines;

}else if(i==3){ //将数据插入到YAxis3中

YAxis3.fieldName=(String) indicatorNameMap.get(key);

YAxis3.unit=(String) indicatorUnitMap.get(key);

YAxis3.yAxis2.add(resmap.get(key));

YAxis3.plotLines = plotLines;;

}else if(i==4){ //将数据插入到YAxis4中

YAxis4.fieldName=(String) indicatorNameMap.get(key);

YAxis4.unit=(String) indicatorUnitMap.get(key);

YAxis4.yAxis2.add(resmap.get(key));

YAxis4.plotLines = plotLines;

}

i++;

}

}

}

}

echarts.setxAxis(xAxis);

List<YAxis> ya = new ArrayList<>();

if(YAxis1.fieldName!=null&&YAxis1.fieldName.length()>0){

ya.add(YAxis1);

}

if(YAxis2.fieldName!=null&&YAxis2.fieldName.length()>0){

ya.add(YAxis2);

}

if(YAxis3.fieldName!=null&&YAxis3.fieldName.length()>0){

ya.add(YAxis3);

}

if(YAxis4.fieldName!=null&&YAxis4.fieldName.length()>0){

ya.add(YAxis4);

}

echarts.setyAxis(ya);

return echarts;

}

}

package com.sdocean.log.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.log.model.SysLoginLogModel;

@Component

public class SysLoginLogDao extends OracleEngine{

public void saveSysLoginLog(SysLoginLogModel model){

StringBuffer sql = new StringBuffer("");

Object[] qaram = new Object[]{model.getUserId(),model.getLoginTime(),model.getLoginType(),

model.getIpAddress(),model.getSystemCode(),model.getSystemName()};

sql.append("insert into sys\_login\_log(userid,logintime,logintype,ipaddress,systemcode,systemname) values(?,?,?,?,?,?)");

this.update(sql.toString(), new Object[]{model.getUserId(),model.getLoginTime(),model.getLoginType(),

model.getIpAddress(),model.getSystemCode(),model.getSystemName()});

}

/\*

\* 查询一定时间范围内的用户登录日志

\*/

public List<SysLoginLogModel> getLoginLogList(SysLoginLogModel model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.userid,b.realname as username,a.logintime as logtime,a.logintype,c.value as logintypename,");

sql.append(" a.ipaddress,a.systemcode,a.systemname");

sql.append(" from sys\_login\_log a,sys\_user b,sys\_public c");

sql.append(" where a.userid = b.id");

sql.append(" and c.parentcode = '0001' and a.logintype = c.classid");

//增加查询条件

if(model!=null&&model.getBeginTime()!=null&&model.getBeginTime().length()>0){

sql.append(" and a.logintime >= '").append(model.getBeginTime()).append("'");

}

if(model!=null&&model.getEndTime()!=null&&model.getEndTime().length()>0){

sql.append(" and a.logintime <= '").append(model.getEndTime()).append("'");

}

//增加排序语句

sql.append(" order by a.logintime desc");

list = this.queryObjectList(sql.toString(), SysLoginLogModel.class);

return list;

}

}

package com.sdocean.log.model;

import java.util.Date;

public class OperationLogModel {

public static final int SUCCESS = 1;

public static final int FAILED = 0;

public static final int ADD = 1;

public static final int UPDATE = 2;

public static final int DELETE = 3;

private int id;

private int userId;

private String userName;

private String menuname;

private String url;

private int dotype;

private String dotypeName;

private String model;

private int result;

private String resultName;

private String message;

private String errorcode;

private Date operationtime;

private String operaTime;

private String beginTime;

private String endTime;

public String getResultName() {

return resultName;

}

public void setResultName(String resultName) {

this.resultName = resultName;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getDotypeName() {

return dotypeName;

}

public void setDotypeName(String dotypeName) {

this.dotypeName = dotypeName;

}

public String getOperaTime() {

return operaTime;

}

public void setOperaTime(String operaTime) {

this.operaTime = operaTime;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getMenuname() {

return menuname;

}

public void setMenuname(String menuname) {

this.menuname = menuname;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public int getDotype() {

return dotype;

}

public void setDotype(int dotype) {

this.dotype = dotype;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public int getResult() {

return result;

}

public void setResult(int result) {

this.result = result;

}

public String getErrorcode() {

return errorcode;

}

public void setErrorcode(String errorcode) {

this.errorcode = errorcode;

}

public Date getOperationtime() {

return operationtime;

}

public void setOperationtime(Date operationtime) {

this.operationtime = operationtime;

}

}

package com.sdocean.log.model;

import java.util.List;

import com.sdocean.station.model.StationModel;

public class SysLog {

private int id;

private int wpId;

private String wpName;

private String beginTime;

private String endTime;

private List<StationModel> stations;

public List<StationModel> getStations() {

return stations;

}

public void setStations(List<StationModel> stations) {

this.stations = stations;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getWpId() {

return wpId;

}

public void setWpId(int wpId) {

this.wpId = wpId;

}

public String getWpName() {

return wpName;

}

public void setWpName(String wpName) {

this.wpName = wpName;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

}

package com.sdocean.log.model;

import java.util.Date;

public class SysLoginLogModel {

private int id;

private int userId;

private String userName;

private Date loginTime;

private String ipAddress;

private int loginType; //登录方式 1:用户名密码登录 2:CA认证登录

private String loginTypeName;

private String systemCode;

private String systemName;

private String logTime;

private String beginTime;

private String endTime;

public String getLoginTypeName() {

return loginTypeName;

}

public void setLoginTypeName(String loginTypeName) {

this.loginTypeName = loginTypeName;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getLogTime() {

return logTime;

}

public void setLogTime(String logTime) {

this.logTime = logTime;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public String getSystemCode() {

return systemCode;

}

public void setSystemCode(String systemCode) {

this.systemCode = systemCode;

}

public int getLoginType() {

return loginType;

}

public void setLoginType(int loginType) {

this.loginType = loginType;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public Date getLoginTime() {

return loginTime;

}

public void setLoginTime(Date loginTime) {

this.loginTime = loginTime;

}

public String getIpAddress() {

return ipAddress;

}

public void setIpAddress(String ipAddress) {

this.ipAddress = ipAddress;

}

public String getSystemName() {

return systemName;

}

public void setSystemName(String systemName) {

this.systemName = systemName;

}

}

package com.sdocean.main.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.main.model.AiotMainConfigModel;

import com.sdocean.main.model.StationDeviceMainModel;

import com.sdocean.station.model.StationModel;

@Component

public class AiotMainConfigDao extends OracleEngine{

/\*

\* 获得例行维护分类列表

\*/

public List<AiotMainConfigModel> getAiotMainConfigList(AiotMainConfigModel model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.code,a.name,a.how,a.ordercode ");

sql.append(" from aiot\_main\_config a");

sql.append(" where 1= 1 ");

//添加查询条件

if(model!=null&&model.getCode()!=null&&model.getCode().length()>0){

sql.append(" and code like '%").append(model.getCode()).append("%'");

}

//添加排序

sql.append(" order by ordercode");

list = this.queryObjectList(sql.toString(), AiotMainConfigModel.class);

return list;

}

/\*

\* 新增例行维护分类

\*/

public Result saveNewMainConfig(AiotMainConfigModel model){

//初始化返回结果

Result result = new Result();

result.setDotype(Result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("新增成功");

//判断CODE是否重复

int res = 0;

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from aiot\_main\_config where code = '").append(model.getCode()).append("'");

try {

res = this.queryForInt(csql.toString(), null);

} catch (Exception e) {

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("唯一验证时失败");

return result;

}

if(res >0){

result.setResult(Result.FAILED);

result.setMessage("已有相同的CODE");

return result;

}

//插入数据

StringBuffer sql = new StringBuffer("");

sql.append(" insert into aiot\_main\_config(code,name,how,ordercode)");

sql.append(" values(?,?,?,?)");

Object[] params = new Object[]{

model.getCode(),model.getName(),model.getHow(),

model.getOrderCode()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("插入数据失败");

return result;

}

return result;

}

/\*

\* 修改例行维护分类

\*/

public Result saveChangeMainConfig(AiotMainConfigModel model){

//初始化返回结果

Result result = new Result();

result.setDotype(Result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("修改成功");

//判断CODE是否重复

int res = 0;

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from aiot\_main\_config where code = '").append(model.getCode()).append("' and id <> ").append(model.getId());

try {

res = this.queryForInt(csql.toString(), null);

} catch (Exception e) {

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("唯一验证时失败");

return result;

}

if(res >0){

result.setResult(Result.FAILED);

result.setMessage("已有相同的CODE");

return result;

}

//修改数据

StringBuffer sql = new StringBuffer("");

sql.append(" update aiot\_main\_config set code=?,name=?,how=?,ordercode=? where id=?");

Object[] params = new Object[]{

model.getCode(),model.getName(),model.getHow(),

model.getOrderCode(),model.getId()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("修改数据失败");

return result;

}

return result;

}

/\*

\* 删除例行维护分类

\*/

public Result deleMainConfig(AiotMainConfigModel model){

//初始化返回结果

Result result = new Result();

result.setDotype(Result.DELETE);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("删除成功");

StringBuffer sql = new StringBuffer("");

sql.append(" delete from aiot\_main\_config where id =").append(model.getId());

try {

this.update(sql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage(" 数据失败");

return result;

}

return result;

}

}

package com.sdocean.main.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.main.model.ErrorTenance;

import com.sdocean.station.model.StationModel;

@Component

public class ErrorTenanceDao extends OracleEngine {

/\*

\* 查询出查询条件下的异常维护列表

\*/

public List<ErrorTenance> getErrorsByStation(ErrorTenance model,List<StationModel> stations){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,GROUP\_CONCAT(k.deviceid) as deviceids,GROUP\_CONCAT(k.devicename) as devicenames,");

sql.append(" a.stationid,b.title as stationName,a.collecttime,");

sql.append(" a.userId,c.realname as userName,a.error,a.reason,a.state,");

sql.append(" d.value as statename,a.begintime,a.endtime,a.material,a.result");

sql.append(" from aiot\_errortenance a left join (");

sql.append(" select m.errorid,m.deviceid,n.name as devicename");

sql.append(" from aiot\_errortenance\_device m,device\_catalog n");

sql.append(" where m.deviceid = n.id");

sql.append(" ) k on a.id = k.errorid,");

sql.append(" aiot\_watch\_point b,sys\_user c,sys\_public d");

sql.append(" where a.stationid = b.id and a.userid = c.id");

sql.append(" and a.state = d.classid and d.parentcode = '0016'");

//增加查询条件

//增加站点查询条件

if(model!=null&&model.getStationId()>0){

sql.append(" and a.stationid=").append(model.getStationId());

}else{

sql.append(" and a.stationid in (0");

for(StationModel station:stations){

sql.append(",").append(station.getId());

}

sql.append(")");

}

//增加时间查询条件

if(model!=null&&model.getBeginTime()!=null&&model.getBeginTime().length()>0){

sql.append(" and a.begintime >= '").append(model.getBeginTime()).append("'");

}

if(model!=null&&model.getEndTime()!=null&&model.getEndTime().length()>0){

sql.append(" and a.begintime <= '").append(model.getEndTime()).append("'");

}

//增加状态查询条件

if(model!=null&&model.getState()>0){

sql.append(" and a.state =").append(model.getState());

}

sql.append(" group by a.id");

//增加排序

sql.append(" order by a.begintime desc");

list = this.queryObjectList(sql.toString(), ErrorTenance.class);

return list;

}

/\*

\* 增加新的异常维护记录

\*/

public Result saveNewErrorTenance(ErrorTenance model){

//初始化查询结果

Result result = new Result();

result.setDotype(Result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("新增成功");

//插入数据,并获得刚插入的ID

StringBuffer isql = new StringBuffer("");

isql.append(" insert into aiot\_errortenance(stationid,userid,error,reason,state,begintime,endtime,material,result)");

isql.append(" values(?,?,?,?,?,?,?,?,?)");

Object[] params = new Object[]{

model.getStationId(),model.getUserId(),model.getError(),

model.getReason(),model.getState(),model.getBeginTime(),

model.getEndTime(),model.getMaterial(),model.getResult()

};

try {

this.update(isql.toString(), params);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();;

result.setResult(Result.FAILED);

result.setMessage("插入表失败");

return result;

}

//得到刚插入的维护记录的ID值

int id = 0;

String csql="select last\_insert\_id()";

id = this.queryForInt(csql, null);

model.setId(id);

//插入涉及到的deviceids

if(model!=null&&model.getDeviceIds()!=null&&model.getDeviceIds().length()>0){

String deviceids[] = model.getDeviceIds().split(",");

StringBuffer msql = new StringBuffer();

msql.append(" insert into aiot\_errortenance\_device(errorid,deviceid) values (0,0)");

for(int i=0;i<deviceids.length;i++){

msql.append(",(").append(id).append(",").append(deviceids[i]).append(")");

}

msql.append(" on duplicate key update deviceid=values(deviceid)");

try {

this.update(msql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("插入设备附表失败");

}

}

return result;

}

/\*

\* 修改异常维护上报信息

\*/

public Result saveChangeErrorTenance(ErrorTenance model){

//初始化返回结果

Result result = new Result();

result.setDotype(Result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("修改成功");

StringBuffer usql = new StringBuffer("");

usql.append(" update aiot\_errortenance set error=?,reason=?,state=?,begintime=?,endtime=?,material=?,result=? where id=?");

Object[] params = new Object[]{

model.getError(),model.getReason(),model.getState(),

model.getBeginTime(),model.getEndTime(),model.getMaterial(),

model.getResult(),model.getId()

};

try {

this.update(usql.toString(), params);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("修改失败");

return result;

}

//修改涉及的设备列表

StringBuffer dsql = new StringBuffer("");

dsql.append(" delete from aiot\_errortenance\_file where errorid = ").append(model.getId());

try {

this.update(dsql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("删除原配置失败");

return result;

}

//更新设备列表

if(model!=null&&model.getDeviceIds()!=null&&model.getDeviceIds().length()>0){

String deviceids[] = model.getDeviceIds().split(",");

StringBuffer msql = new StringBuffer();

msql.append(" insert into aiot\_errortenance\_device(errorid,deviceid) values (0,0)");

for(int i=0;i<deviceids.length;i++){

msql.append(",(").append(model.getId()).append(",").append(deviceids[i]).append(")");

}

msql.append(" on duplicate key update deviceid=values(deviceid)");

try {

this.update(msql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("插入设备附表失败");

}

}

return result;

}

/\*

\* 为异常维护记录提供设备列表

\*/

public List<SelectTree> getDeviceList4Error(ErrorTenance model){

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

//添加设备列表的首层

SelectTree first = new SelectTree();

first.setId("0");

first.setName("设备");

first.setIsExpanded(true);

first.setIsActive(true);

//获得所有有效的设备的列表

List<SelectTree> children = new ArrayList<SelectTree>();

StringBuffer sql = new StringBuffer("");

sql.append(" select b.id,b.name,case when k.deviceid is null then 'false' else 'true' end as selected");

sql.append(" from map\_awp\_device\_catalog a left join (");

sql.append(" select m.stationid,n.deviceid");

sql.append(" from aiot\_errortenance m,aiot\_errortenance\_device n");

sql.append(" where m.id = n.errorid and m.stationid =").append(model.getStationId());

sql.append(" and m.id = ").append(model.getId());

sql.append(" ) k on a.aiot\_watch\_point\_id = k.stationid ");

sql.append(" and a.device\_catalog\_id = k.deviceid,");

sql.append(" device\_catalog b");

sql.append(" where a.device\_catalog\_id = b.id");

sql.append(" and a.aiot\_watch\_point\_id =").append(model.getStationId());

children = this.queryObjectList(sql.toString(), SelectTree.class);

//将设备列表添加的首层的child中

first.setChildren(children);

//将首层添加到结果集中

list.add(first);

return list;

}

}

package com.sdocean.main.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.main.model.AiotMainConfigModel;

import com.sdocean.main.model.StationDeviceMainModel;

import com.sdocean.station.model.StationModel;

@Component

public class StationDeviceMainDao extends OracleEngine{

/\*

\* 查询条件范围内的站点设备维护配置列表

\*/

public List<StationDeviceMainModel> getStationDeviceMainList(StationDeviceMainModel model,List<StationModel> stations){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.madcid,c.id as stationid,c.title as stationName,");

sql.append(" d.id as deviceid,d.name as deviceName,b.createtime,");

sql.append(" a.amcid as amconfigid,e.code as mainConfigCode,e.name as mainConfigName,");

sql.append(" a.mainnum,e.how");

sql.append(" from aiot\_madc\_amc a,map\_awp\_device\_catalog b,");

sql.append(" aiot\_watch\_point c,device\_catalog d,aiot\_main\_config e");

sql.append(" where a.madcid = b.id and a.amcid = e.id");

sql.append(" and b.aiot\_watch\_point\_id = c.id ");

sql.append(" and b.device\_catalog\_id = d.id");

//添加查询条件

if(model!=null&&model.getStationId()>0){

sql.append(" and c.id = ").append(model.getStationId());

}else{

StringBuffer ssql = new StringBuffer("");

ssql.append(" (0");

for(StationModel station:stations){

ssql.append(",").append(station.getId());

}

ssql.append(")");

sql.append(" and c.id in ").append(ssql);

}

sql.append(" order by c.id,b.orderCode");

list = this.queryObjectList(sql.toString(), StationDeviceMainModel.class);

return list;

}

/\*

\* 保存新增的站点设备维护配置

\*/

public Result saveStationDeviceMain(StationDeviceMainModel model){

//初始化返回列表

Result result = new Result();

result.setDotype(Result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("新增成功");

//判断唯一性原则

StringBuffer csql = new StringBuffer("");

int cres = 0;

csql.append(" select count(1) from aiot\_madc\_amc where madcid = ").append(model.getMadcId()).append(" and amcid =").append(model.getAmconfigId());

try {

cres = this.queryForInt(csql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("唯一性原则检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//开始添加站点设备维护配置

StringBuffer sql = new StringBuffer("");

sql.append(" insert into aiot\_madc\_amc(madcid,amcid,mainnum) values(?,?,?)");

Object[] params = new Object[]{

model.getMadcId(),model.getAmconfigId(),model.getMainNum()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("新增失败");

return result;

}

return result;

}

/\*

\* 保存修改的站点设备维护配置

\*/

public Result saveChangeStationDeviceMain(StationDeviceMainModel model){

//初始化返回列表

Result result = new Result();

result.setDotype(Result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("修改成功");

//判断唯一性原则

StringBuffer csql = new StringBuffer("");

int cres = 0;

csql.append(" select count(1) from aiot\_madc\_amc where madcid = ").append(model.getMadcId()).append(" and amcid =").append(model.getAmconfigId());

csql.append(" and id <> ").append(model.getId());

try {

cres = this.queryForInt(csql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("唯一性原则检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//开始添加站点设备维护配置

StringBuffer sql = new StringBuffer("");

sql.append(" update aiot\_madc\_amc set madcid=?,amcid=?,mainnum=?, where id=?");

Object[] params = new Object[]{

model.getMadcId(),model.getAmconfigId(),model.getMainNum(),model.getId()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("修改失败");

return result;

}

return result;

}

/\*

\* 删除选中的站点设备维护配置

\*/

public Result deleStationDeviceMain(StationDeviceMainModel model){

//初始化返回列表

Result result = new Result();

result.setDotype(Result.DELETE);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.SUCCESS);

result.setMessage("删除成功");

StringBuffer sql = new StringBuffer("");

sql.append("delete from aiot\_madc\_amc where id = ").append(model.getId());

int res = 0;

try {

res = this.update(sql.toString(), null);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("删除失败");

}

return result;

}

/\*

\* 根据站点和设备获得有效的例行维护种类的列表

\*/

public List<AiotMainConfigModel> getAiotMainConfigListByStationDevice(StationDeviceMainModel model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select b.id,b.code,b.name,b.how,b.ordercode");

sql.append(" from aiot\_madc\_amc a,aiot\_main\_config b,map\_awp\_device\_catalog c");

sql.append(" where a.madcid = c.id and a.amcid = b.id ");

sql.append(" and c.aiot\_watch\_point\_id =").append(model.getStationId()).append(" and c.device\_catalog\_id = ").append(model.getDeviceId());

sql.append(" order by b.ordercode");

list = this.queryObjectList(sql.toString(), AiotMainConfigModel.class);

return list;

}

}

package com.sdocean.main.model;

public class AiotMainConfigModel {

private int id;

private String code;

private String name;

private String how;

private String orderCode;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getHow() {

return how;

}

public void setHow(String how) {

this.how = how;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

}

package com.sdocean.main.model;

import java.util.List;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.station.model.StationModel;

public class ErrorTenance {

private int id;

private int stationId;

private String stationName;

private String collectTime;

private int userId;

private String userName;

private String error;

private String reason;

private int state;

private String stateName;

private String beginTime;

private String endTime;

private String material;

private String result;

private List<StationModel> stations; //查询页面初始化的站点权限列表

private List<DeviceModel> devices; //异常维护牵扯到的设备列表

private String deviceIds;

private String deviceNames;

public String getDeviceNames() {

return deviceNames;

}

public void setDeviceNames(String deviceNames) {

this.deviceNames = deviceNames;

}

public String getDeviceIds() {

return deviceIds;

}

public void setDeviceIds(String deviceIds) {

this.deviceIds = deviceIds;

}

public List<DeviceModel> getDevices() {

return devices;

}

public void setDevices(List<DeviceModel> devices) {

this.devices = devices;

}

public List<StationModel> getStations() {

return stations;

}

public void setStations(List<StationModel> stations) {

this.stations = stations;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public String getStationName() {

return stationName;

}

public void setStationName(String stationName) {

this.stationName = stationName;

}

public String getCollectTime() {

return collectTime;

}

public void setCollectTime(String collectTime) {

this.collectTime = collectTime;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getError() {

return error;

}

public void setError(String error) {

this.error = error;

}

public String getReason() {

return reason;

}

public void setReason(String reason) {

this.reason = reason;

}

public int getState() {

return state;

}

public void setState(int state) {

this.state = state;

}

public String getStateName() {

return stateName;

}

public void setStateName(String stateName) {

this.stateName = stateName;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public String getMaterial() {

return material;

}

public void setMaterial(String material) {

this.material = material;

}

public String getResult() {

return result;

}

public void setResult(String result) {

this.result = result;

}

}

package com.sdocean.main.model;

import java.util.List;

import com.sdocean.station.model.StationModel;

public class MainTenance {

private int id;

private int stationId;

private String stationName;

private int deviceId;

private String deviceIds;

private String deviceName;

private String beginTime;

private String endTime;

private int state;

private String stateName;

private int userId;

private String userName;

private String collectTime;

private int configId;

private String configName;

private String reason;

private String result;

private String lastMainTime;

private String planTime;

private int mainnum; //例行维护时间间隔

private String material; //维护耗费的材料

private List<StationModel> stations;

private int mtype ; //操作类型

private String createTime; //设备的安装时间

private int madcId ; //站点包含设备的表的id

private int amcId;

private String amcName;

public int getAmcId() {

return amcId;

}

public void setAmcId(int amcId) {

this.amcId = amcId;

}

public String getAmcName() {

return amcName;

}

public void setAmcName(String amcName) {

this.amcName = amcName;

}

public int getMadcId() {

return madcId;

}

public void setMadcId(int madcId) {

this.madcId = madcId;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public int getMtype() {

return mtype;

}

public void setMtype(int mtype) {

this.mtype = mtype;

}

public List<StationModel> getStations() {

return stations;

}

public void setStations(List<StationModel> stations) {

this.stations = stations;

}

public String getMaterial() {

return material;

}

public void setMaterial(String material) {

this.material = material;

}

public int getMainnum() {

return mainnum;

}

public void setMainnum(int mainnum) {

this.mainnum = mainnum;

}

public String getLastMainTime() {

return lastMainTime;

}

public void setLastMainTime(String lastMainTime) {

this.lastMainTime = lastMainTime;

}

public String getPlanTime() {

return planTime;

}

public void setPlanTime(String planTime) {

this.planTime = planTime;

}

public String getDeviceIds() {

return deviceIds;

}

public void setDeviceIds(String deviceIds) {

this.deviceIds = deviceIds;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public String getStationName() {

return stationName;

}

public void setStationName(String stationName) {

this.stationName = stationName;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getDeviceName() {

return deviceName;

}

public void setDeviceName(String deviceName) {

this.deviceName = deviceName;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public int getState() {

return state;

}

public void setState(int state) {

this.state = state;

}

public String getStateName() {

return stateName;

}

public void setStateName(String stateName) {

this.stateName = stateName;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getCollectTime() {

return collectTime;

}

public void setCollectTime(String collectTime) {

this.collectTime = collectTime;

}

public int getConfigId() {

return configId;

}

public void setConfigId(int configId) {

this.configId = configId;

}

public String getConfigName() {

return configName;

}

public void setConfigName(String configName) {

this.configName = configName;

}

public String getReason() {

return reason;

}

public void setReason(String reason) {

this.reason = reason;

}

public String getResult() {

return result;

}

public void setResult(String result) {

this.result = result;

}

}

package com.sdocean.main.model;

public class MainTenanceFile {

private int id;

private int mainId;

private String realName;

private String pathName;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getMainId() {

return mainId;

}

public void setMainId(int mainId) {

this.mainId = mainId;

}

public String getRealName() {

return realName;

}

public void setRealName(String realName) {

this.realName = realName;

}

public String getPathName() {

return pathName;

}

public void setPathName(String pathName) {

this.pathName = pathName;

}

}

package com.sdocean.main.model;

public class StationDeviceMainModel {

private int id;

private int madcId;

private int stationId;

private String stationName;

private int deviceId;

private String deviceName;

private String createTime;

private int amconfigId;

private String mainConfigCode;

private String mainConfigName;

private String how;

private int mainNum;

public int getMainNum() {

return mainNum;

}

public void setMainNum(int mainNum) {

this.mainNum = mainNum;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getMadcId() {

return madcId;

}

public void setMadcId(int madcId) {

this.madcId = madcId;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public String getStationName() {

return stationName;

}

public void setStationName(String stationName) {

this.stationName = stationName;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getDeviceName() {

return deviceName;

}

public void setDeviceName(String deviceName) {

this.deviceName = deviceName;

}

public int getAmconfigId() {

return amconfigId;

}

public void setAmconfigId(int amconfigId) {

this.amconfigId = amconfigId;

}

public String getMainConfigCode() {

return mainConfigCode;

}

public void setMainConfigCode(String mainConfigCode) {

this.mainConfigCode = mainConfigCode;

}

public String getMainConfigName() {

return mainConfigName;

}

public void setMainConfigName(String mainConfigName) {

this.mainConfigName = mainConfigName;

}

public String getHow() {

return how;

}

public void setHow(String how) {

this.how = how;

}

}

package com.sdocean.main.service;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.main.dao.ErrorTenanceDao;

import com.sdocean.main.dao.MainTenanceDao;

import com.sdocean.main.model.ErrorTenance;

import com.sdocean.main.model.MainTenance;

import com.sdocean.main.model.MainTenanceFile;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor = Exception.class, propagation = Propagation.REQUIRED)

public class ErrorTenanceService {

@Resource

private ErrorTenanceDao errorDao;

public List<UiColumn> getCols4MainEditList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col0 = new UiColumn("id", "id", false, "\*");

UiColumn col1 = new UiColumn("stationId", "stationId", false, "\*");

UiColumn col2 = new UiColumn("站点名称", "stationName", true, "\*");

UiColumn col3 = new UiColumn("deviceId", "deviceIds", false, "\*");

UiColumn col4 = new UiColumn("设备名称", "deviceNames", true, "\*");

UiColumn col6 = new UiColumn("设备状态", "stateName", true, "\*");

UiColumn col8 = new UiColumn("维护开始时间", "beginTime", true, "\*");

UiColumn col9 = new UiColumn("维护结束时间", "endTime", true, "\*");

UiColumn col10 = new UiColumn("推测原因", "reason", true, "\*");

UiColumn col7 = new UiColumn("异常问题", "error", true, "\*");

UiColumn col11 = new UiColumn("维护结果", "result", true, "\*");

UiColumn col12 = new UiColumn("需求材料", "material", true, "\*");

UiColumn col13 = new UiColumn("userId", "userId", false, "\*");

UiColumn col14 = new UiColumn("上报人", "userName", true, "\*");

UiColumn col15 = new UiColumn("上报时间", "collectTime", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col4);

cols.add(col6);

cols.add(col8);

cols.add(col9);

cols.add(col7);

cols.add(col10);

cols.add(col11);

cols.add(col12);

cols.add(col13);

cols.add(col14);

cols.add(col15);

return cols;

}

/\*

\* 查询出查询条件下的异常维护列表

\*/

public List<ErrorTenance> getErrorsByStation(ErrorTenance model,List<StationModel> stations){

return errorDao.getErrorsByStation(model, stations);

}

/\*

\* 增加新的异常维护记录

\*/

public Result saveNewErrorTenance(ErrorTenance model){

return errorDao.saveNewErrorTenance(model);

}

/\*

\* 修改异常维护上报信息

\*/

public Result saveChangeErrorTenance(ErrorTenance model){

return errorDao.saveChangeErrorTenance(model);

}

/\*

\* 为异常维护记录提供设备列表

\*/

public List<SelectTree> getDeviceList4Error(ErrorTenance model){

return errorDao.getDeviceList4Error(model);

}

}

package com.sdocean.main.service;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.main.dao.MainTenanceDao;

import com.sdocean.main.model.MainTenance;

import com.sdocean.main.model.MainTenanceFile;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor = Exception.class, propagation = Propagation.REQUIRED)

public class MainTenanceService {

@Resource

private MainTenanceDao mainDao;

public List<UiColumn> getCols4MainEditList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col0 = new UiColumn("id", "id", false, "\*");

UiColumn col1 = new UiColumn("wpId", "stationId", false, "\*");

UiColumn col2 = new UiColumn("站点名称", "stationName", true, "\*");

UiColumn col3 = new UiColumn("deviceId", "deviceId", false, "\*");

UiColumn col4 = new UiColumn("设备名称", "deviceName", true, "\*");

UiColumn col6 = new UiColumn("设备状态", "stateName", true, "\*");

UiColumn col60 = new UiColumn("维护类型", "configId", false, "\*");

UiColumn col7 = new UiColumn("维护类型", "configName", true, "\*");

UiColumn col8 = new UiColumn("维护开始时间", "beginTime", true, "\*");

UiColumn col9 = new UiColumn("维护结束时间", "endTime", true, "\*");

UiColumn col10 = new UiColumn("维护原因", "reason", true, "\*");

UiColumn col11 = new UiColumn("维护结果", "result", true, "\*");

UiColumn col12 = new UiColumn("需求材料", "material", true, "\*");

UiColumn col13 = new UiColumn("userId", "userId", false, "\*");

UiColumn col14 = new UiColumn("上报人", "userName", true, "\*");

UiColumn col15 = new UiColumn("上报时间", "collectTime", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col4);

cols.add(col6);

cols.add(col60);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col10);

cols.add(col11);

cols.add(col12);

cols.add(col13);

cols.add(col14);

cols.add(col15);

return cols;

}

/\*

\* 为站点维护查询提供表头

\*/

public List<UiColumn> getCols4MainShowList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("wpId", "wpId", false, "\*");

UiColumn col2 = new UiColumn("站点名称", "wpName", true, "\*");

UiColumn col3 = new UiColumn("deviceId", "deviceId", false, "\*");

UiColumn col4 = new UiColumn("设备名称", "deviceName", true, "\*");

UiColumn col6 = new UiColumn("设备状态", "stateName", true, "\*");

UiColumn col60 = new UiColumn("维护类型", "configId", false, "\*");

UiColumn col7 = new UiColumn("维护类型", "configName", true, "\*");

UiColumn col8 = new UiColumn("维护开始时间", "beginTime", true, "\*");

UiColumn col9 = new UiColumn("维护结束时间", "endTime", true, "\*");

UiColumn col10 = new UiColumn("维护原因", "reason", true, "\*");

UiColumn col11 = new UiColumn("维护结果", "result", true, "\*");

UiColumn col12 = new UiColumn("需求材料", "material", true, "\*");

UiColumn col13 = new UiColumn("userId", "userId", true, "\*");

UiColumn col14 = new UiColumn("上报人", "userName", true, "\*");

UiColumn col15 = new UiColumn("上报时间", "collectTime", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col4);

cols.add(col6);

cols.add(col60);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col10);

cols.add(col11);

cols.add(col12);

cols.add(col13);

cols.add(col14);

cols.add(col15);

return cols;

}

/\*

\* 为站点维护上报提供查询结果集

\*/

public List<MainTenance> getMainList4StationMainEdit(MainTenance model,List<StationModel> stations){

return mainDao.getMainList4StationMainEdit(model, stations);

}

/\*

\* 获得新增的站点设备维护上报并返回新增的ID值

\*/

public Result saveNewMainEdit(MainTenance model){

return mainDao.saveNewMainEdit(model);

}

/\*

\* 保存修改维护上报,

\*/

public Result saveChangeMainEdit(MainTenance model){

return mainDao.saveChangeMainEdit(model);

}

/\*

\* 根据站点设备维护ID,以及文件列表,保存文件数据到附表

\*/

public void saveFileByMain(MainTenance main,List<MainTenanceFile> files){

mainDao.saveFileByMain(main, files);

}

/\*

\* 根据站点设备维护ID,得到该设备维护记录上传的图片列表

\*/

public List<MainTenanceFile> getFileListByMain(MainTenance model){

return mainDao.getFileListByMain(model);

}

/\*

\* 为首页展示系统维护时间提供数据

\*/

public List<MainTenance> getPlanMain4FirstPage(StationModel station){

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

SimpleDateFormat sd = new SimpleDateFormat("yyyy-MM-dd");

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

//得到该站点下需要例行维护的设备的列表,以及该设备的创建时间

list = mainDao.getDeviceList4MainByStation(station);

//根据站点和设备遍历其例行维护类型,得到时间最靠前的时间,以及类型名称

for(MainTenance main:list){

String createTime = main.getCreateTime();

Date planTime = null;

String stateName = ""; //例行维护类型

try {

planTime = sdf.parse("2099-12-21 00:00:01");

} catch (ParseException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

//根据站点和设备得到有效的例行维护类型列表,以及间隔时间

List<MainTenance> configs = mainDao.getMainConfigListByStationDevice(main);

//根据站点\设备和例行维护类型,以及该类型的间隔时间,得到预计的下次维护时间,

for(MainTenance con:configs){

String endTime = "";

//该站点 设备和例行维护类型的间隔维护时间

int mainNum = con.getMainnum();

MainTenance res = mainDao.getLastMainByStationDeviceConfig(con);

if(res!=null&&res.getEndTime()!=null&&res.getEndTime().length()>0){

endTime = res.getEndTime();

}else{

endTime = createTime;

}

Date end = null;

try {

end = sdf.parse(endTime);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Calendar calendar=Calendar.getInstance();

calendar.setTime(end);

calendar.add(Calendar.DATE, mainNum);

end = calendar.getTime();

if(end.getTime()<planTime.getTime()){

planTime = end;

stateName = con.getAmcName();

}

}

String plan = sd.format(planTime);

main.setAmcName(stateName);

main.setPlanTime(plan);

}

return list;

}

/\*

\* 以表格的形式显示设备维护提醒

\*/

public PageResult getPlanMain4FirstPage2(StationModel station){

PageResult page = new PageResult();

//初始化表头

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col0 = new UiColumn("设备1", "fubiao1", true, "\*");

UiColumn col1 = new UiColumn("设备2", "fubiao2", true, "\*");

UiColumn col2 = new UiColumn("设备3", "danghang", true, "\*");

UiColumn col4 = new UiColumn("设备4", "yidong", true, "\*");

UiColumn col6 = new UiColumn("设备5", "qianbiao", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col4);

cols.add(col6);

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

Map row = new HashMap<>();

row.put("fubiao1", "计划2016-11-11 更换电池");

row.put("fubiao2", "计划2016-11-11 更换电池");

row.put("danghang", "计划2016-11-11 更换电池");

row.put("yidong", "计划2016-11-11 更换电池");

row.put("qianbiao", "计划2016-11-11 更换电池");

rows.add(row);

page.setCols(cols);

page.setRows(rows);

return page;

}

public PageResult getPlanMain4allFirstPage(StationModel station){

PageResult page = new PageResult();

//初始化表头

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col0 = new UiColumn("浮标观测平台1", "fubiao1", true, "\*");

UiColumn col1 = new UiColumn("浮标观测平台2", "fubiao2", true, "\*");

UiColumn col2 = new UiColumn("当行观测平台", "danghang", true, "\*");

UiColumn col4 = new UiColumn("移动观测平台", "yidong", true, "\*");

UiColumn col6 = new UiColumn("潜标观测平台", "qianbiao", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col4);

cols.add(col6);

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

Map row = new HashMap<>();

row.put("fubiao1", "计划2016-11-11 更换电池");

row.put("fubiao2", "计划2016-11-11 更换电池");

row.put("danghang", "计划2016-11-11 更换电池");

row.put("yidong", "计划2016-11-11 更换电池");

row.put("qianbiao", "计划2016-11-11 更换电池");

rows.add(row);

page.setCols(cols);

page.setRows(rows);

return page;

}

}

package com.sdocean.main.service;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.main.dao.StationDeviceMainDao;

import com.sdocean.main.model.AiotMainConfigModel;

import com.sdocean.main.model.StationDeviceMainModel;

import com.sdocean.menu.dao.SysMenuDao;

import com.sdocean.menu.model.CurrMenu;

import com.sdocean.menu.model.SysMenu;

import com.sdocean.page.model.UiColumn;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class StationDeviceMainService {

@Autowired

StationDeviceMainDao stationDeviceMainDao;

/\*

\* 为查询登录日志提供表头信息

\*/

public List<UiColumn> getCols4MainList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*");

UiColumn col2 = new UiColumn("stationId", "stationId", false, "\*");

UiColumn col3 = new UiColumn("站点", "stationName", true, "\*");

UiColumn col4 = new UiColumn("deviceId", "deviceId", false, "\*");

UiColumn col5 = new UiColumn("设备", "deviceName", true, "\*");

UiColumn col6 = new UiColumn("amconfigId", "amconfigId", false, "\*");

UiColumn col7 = new UiColumn("例行维护", "mainConfigName", true, "\*");

UiColumn col8 = new UiColumn("维护周期", "mainNum", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col4);

cols.add(col5);

cols.add(col6);

cols.add(col7);

cols.add(col8);

return cols;

}

/\*

\* 查询条件范围内的站点设备维护配置列表

\*/

public List<StationDeviceMainModel> getStationDeviceMainList(StationDeviceMainModel model,List<StationModel> stations){

return stationDeviceMainDao.getStationDeviceMainList(model, stations);

}

/\*

\* 保存新增的站点设备维护配置

\*/

public Result saveStationDeviceMain(StationDeviceMainModel model){

return stationDeviceMainDao.saveStationDeviceMain(model);

}

/\*

\* 保存修改的站点设备维护配置

\*/

public Result saveChangeStationDeviceMain(StationDeviceMainModel model){

return stationDeviceMainDao.saveChangeStationDeviceMain(model);

}

/\*

\* 删除选中的站点设备维护配置

\*/

public Result deleStationDeviceMain(StationDeviceMainModel model){

return stationDeviceMainDao.deleStationDeviceMain(model);

}

/\*

\* 根据站点和设备获得有效的例行维护种类的列表

\*/

public List<AiotMainConfigModel> getAiotMainConfigListByStationDevice(StationDeviceMainModel model){

return stationDeviceMainDao.getAiotMainConfigListByStationDevice(model);

}

}

package com.sdocean.menu.action;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

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

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

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

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

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.menu.service.SysMenuService;

import com.sdocean.role.model.RoleModel;

@Controller

public class MenuAction {

@Resource

SysMenuService menuService;

/\*

\* 获得某角色下的菜单列表,以TREE的形式显示

\*/

@RequestMapping(value="getMenus4Tree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getMenus4Tree(@ModelAttribute("role") RoleModel role,HttpServletRequest request,

HttpServletResponse response){

List<ZTreeModel> tree = menuService.getMenus4Tree(role);

return JsonUtil.toJson(tree);

}

/\*

\* 获得某角色下的首页菜单列表,以TREE的形式显示

\*/

@RequestMapping(value="getFirstMenus4Tree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getFirstMenus4Tree(@ModelAttribute("role") RoleModel role,HttpServletRequest request,

HttpServletResponse response){

List<ZTreeModel> tree = menuService.getFirstMenus4Tree(role,role.getMenuType());

return JsonUtil.toJson(tree);

}

}

package com.sdocean.menu.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.model.SysLoginLogModel;

import com.sdocean.menu.model.CurrMenu;

import com.sdocean.menu.model.SysMenu;

import com.sdocean.role.model.RoleModel;

import com.sdocean.users.model.SysUser;

@Component

public class SysMenuDao extends OracleEngine{

/\*

\* 通过用户,获得该用户拥有的所有的权限内的的3级菜单

\* parentCode代表的2级菜单的CODE

\* 10000是后台菜单

\* 60000是前台菜单

\*/

public List<SysMenu> getMenuListByUser(SysUser user,String parentCode){

List<SysMenu> menuList = new ArrayList<SysMenu>();

StringBuffer sql = new StringBuffer("");

sql.append(" select distinct a.code,a.pcode,a.type,a.name,a.level,a.url,a.picture,a.isactive,a.ordercode ");

sql.append(" from sys\_menu a,sys\_role\_menu b,sys\_role\_user c ,sys\_role d");

sql.append(" where d.type = 1 and d.id = c.role\_id "); //d.type=1 代表的是菜单角色

sql.append(" and c.role\_id = b.role\_id");

sql.append(" and b.menu\_id = a.code and a.level = 3 and a.isactive = 1"); //a.level=3 代表菜单等级

sql.append(" and c.user\_id = ").append(user.getId());

sql.append(" and a.pcode = '").append(parentCode).append("'");

sql.append(" order by orderCode ");

menuList = this.queryObjectList(sql.toString(), SysMenu.class);

return menuList;

}

/\*

\* 通过父类菜单,获得该用户权限内的所有子菜单

\*/

public SysMenu getMenuByPmenu(SysMenu pmenu,SysUser user){

List<SysMenu> childList = new ArrayList<SysMenu>();

//开始拼接SQL语句

StringBuffer sql = new StringBuffer("");

sql.append(" select distinct a.code,a.pcode,a.type,a.name,a.level,a.url,a.picture,a.isactive,a.ordercode");

sql.append(" from sys\_menu a,sys\_role\_menu b,sys\_role\_user c,sys\_role d");

sql.append(" where d.type = 1 and d.isactive = 1 and c.role\_id = d.id");

sql.append(" and b.role\_id = c.role\_id and a.code = b.menu\_id");

sql.append(" and a.isactive = 1 and a.pcode ='").append(pmenu.getCode()).append("'");

sql.append(" and c.user\_id =").append(user.getId());

sql.append(" order by ordercode ");

childList = this.queryObjectList(sql.toString(), SysMenu.class);

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

for(SysMenu child:childList){

child = this.getMenuByPmenu(child,user);

}

}

pmenu.setChildMenu(childList); //将子菜单列表存入到父菜单中

return pmenu;

}

/\*

\* 通过子类id获得菜单信息以及父类菜单信息

\*/

public CurrMenu getCurrMenuById(String menuId){

CurrMenu menu = new CurrMenu();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.code as cMenuId,a.name as cMenuName,a.url as curl,a.pCode as pMenuId,b.name as pMenuName");

sql.append(" from sys\_menu a ,sys\_menu b where a.pcode = b.code");

sql.append(" and a.code = '").append(menuId).append("'");

menu = this.queryObject(sql.toString(), CurrMenu.class);

return menu;

}

/\*

\* 查询当前角色的菜单列表,并通过TREE的形式展现

\*/

public List<ZTreeModel> getMenus4Tree(RoleModel role){

List<ZTreeModel> tree = new ArrayList<ZTreeModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.code as id,a.pcode as pId,a.name,case when b.menu\_id is null then 'false' else 'true' end as checked,case level when 3 then 'true' else 'true' end as open");

sql.append(" from sys\_menu a ");

sql.append(" left join sys\_role\_menu b on a.code = b.menu\_id and b.role\_id = ").append(role.getId());

tree = this.queryObjectList(sql.toString(), ZTreeModel.class);

return tree;

}

/\*

\* 查询当前菜单的首页菜单权限 ,并通过TREE的形式展示

\*/

public List<ZTreeModel> getFirstMenus4Tree(RoleModel role,String menuType){

List<ZTreeModel> tree = new ArrayList<ZTreeModel>();

//获得总的菜单选项

StringBuffer sql = new StringBuffer();

sql.append(" select a.code as id,a.pcode as pId,a.name,case when b.menu\_id is null then 'false' else 'true' end as checked,case level when 3 then 'true' else 'true' end as open,");

sql.append(" case when length(a.url) > 0 and type =1 then 'false' else 'true' end as nocheck ");

sql.append(" from sys\_menu a ");

sql.append(" left join sys\_role\_firstmenu b on a.code = b.menu\_id and b.role\_id = ").append(role.getId());

sql.append(" where a.code = '000001'");

tree = this.queryObjectList(sql.toString(), ZTreeModel.class);

//获得总的前台/后台总菜单

StringBuffer psql = new StringBuffer("");

psql.append(" select a.code as id,a.pcode as pId,a.name,case when b.menu\_id is null then 'false' else 'true' end as checked,case level when 3 then 'true' else 'true' end as open,");

psql.append(" case when length(a.url) > 0 and type =1 then 'false' else 'true' end as nocheck ");

psql.append(" from sys\_menu a ");

psql.append(" left join sys\_role\_firstmenu b on a.code = b.menu\_id and b.role\_id = ").append(role.getId());

psql.append(" where a.code = '").append(menuType).append("'");

List<ZTreeModel> ptree = new ArrayList<ZTreeModel>();

ptree = this.queryObjectList(psql.toString(), ZTreeModel.class);

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

tree.addAll(ptree);

}

//根据总的前台/后台总菜单,获得他们的所有的子菜单

for(ZTreeModel ztree:ptree){

List<ZTreeModel> ctree = new ArrayList<ZTreeModel>();

ctree = this.getChildByPcode(ztree.getId(), role);

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

tree.addAll(ctree);

}

}

//将总的结果去重

List<ZTreeModel> mtree = new ArrayList<ZTreeModel>();

for(ZTreeModel t:tree){

if(!mtree.contains(t)){

mtree.add(t);

}

}

return mtree;

}

/\*

\* 通过父菜单的code获得所有所有子菜单

\* 并选中在角色中选中的菜单

\*/

public List<ZTreeModel> getChildByPcode(String code,RoleModel role){

List<ZTreeModel> tree = new ArrayList<ZTreeModel>();

StringBuffer sql = new StringBuffer();

sql.append(" select a.code as id,a.pcode as pId,a.name,case when b.menu\_id is null then 'false' else 'true' end as checked,case level when 3 then 'true' else 'true' end as open,");

sql.append(" case when length(a.url) > 0 and type =1 then 'false' else 'true' end as nocheck ");

sql.append(" from sys\_menu a ");

sql.append(" left join sys\_role\_firstmenu b on a.code = b.menu\_id and b.role\_id = ").append(role.getId());

sql.append(" where a.pcode = '").append(code).append("'");

tree = this.queryObjectList(sql.toString(), ZTreeModel.class);

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

List<ZTreeModel> ctree = new ArrayList<ZTreeModel>();

for(ZTreeModel ztree:tree){

ctree.addAll(this.getChildByPcode(ztree.getId(), role));

}

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

tree.addAll(ctree);

}

}

return tree;

}

/\*

\* 查询当前用户的首页菜单权限

\*/

public SysMenu getFirstMenu(SysUser user,String roleType){

SysMenu menu = new SysMenu();

StringBuffer sql = new StringBuffer("");

sql.append(" select distinct a.code,a.pcode,a.type,a.name,a.level,a.url,a.picture,a.isactive,a.ordercode");

sql.append(" from sys\_menu a,sys\_role\_firstmenu b,sys\_role c,sys\_role\_user d");

sql.append(" where a.code = b.menu\_id");

sql.append(" and b.role\_id = c.id and c.isactive = 1");

sql.append(" and c.id = d.role\_id and c.type = ").append(roleType);

sql.append(" and d.user\_id =").append(user.getId());

sql.append(" limit 1");

menu = this.queryObject(sql.toString(), SysMenu.class);

return menu;

}

/\*

\* 根据菜单的CODE获得菜单的2级,

\* 判断该菜单数据前台还是后台

\*/

public SysMenu getLevel2MentByChildCode(String code){

SysMenu menu = new SysMenu();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.code,a.pcode,a.type,a.name,a.level,");

sql.append(" a.url,a.picture,a.isactive,a.ordercode");

sql.append(" from sys\_menu a,sys\_menu b");

sql.append(" where a.code = b.pcode");

sql.append(" and a.isactive = 1 and b.isactive = 1");

sql.append(" and b.code = '").append(code).append("'");

menu = this.queryObject(sql.toString(), SysMenu.class);

if(menu!=null&&menu.getLevel()!=2){

menu = this.getLevel2MentByChildCode(menu.getCode());

}

return menu;

}

}

package com.sdocean.menu.model;

public class CurrMenu {

private String pMenuId;

private String pMenuName;

private String cMenuId;

private String cMenuName;

private String curl;

public String getCurl() {

return curl;

}

public void setCurl(String curl) {

this.curl = curl;

}

public String getpMenuName() {

return pMenuName;

}

public void setpMenuName(String pMenuName) {

this.pMenuName = pMenuName;

}

public String getpMenuId() {

return pMenuId;

}

public void setpMenuId(String pMenuId) {

this.pMenuId = pMenuId;

}

public String getcMenuId() {

return cMenuId;

}

public void setcMenuId(String cMenuId) {

this.cMenuId = cMenuId;

}

public String getcMenuName() {

return cMenuName;

}

public void setcMenuName(String cMenuName) {

this.cMenuName = cMenuName;

}

}

package com.sdocean.menu.model;

import java.util.List;

public class SysMenu {

private String code;

private String pcode;

private int type; //是否节点

private String typeName; //是否节点

private String name; //菜单名称

private String url; //菜单链接

private String picture; //图片链接地址

private int isactive; //启用标志

private int order; //排序标志

private int level; //菜单级别

private List<SysMenu> childMenu;

private int isopen;

private int iscurr;

public int getIsopen() {

return isopen;

}

public void setIsopen(int isopen) {

this.isopen = isopen;

}

public int getIscurr() {

return iscurr;

}

public void setIscurr(int iscurr) {

this.iscurr = iscurr;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public List<SysMenu> getChildMenu() {

return childMenu;

}

public void setChildMenu(List<SysMenu> childMenu) {

this.childMenu = childMenu;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getPcode() {

return pcode;

}

public void setPcode(String pcode) {

this.pcode = pcode;

}

public int getType() {

return type;

}

public void setType(int type) {

this.type = type;

}

public String getTypeName() {

return typeName;

}

public void setTypeName(String typeName) {

this.typeName = typeName;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getPicture() {

return picture;

}

public void setPicture(String picture) {

this.picture = picture;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public int getOrder() {

return order;

}

public void setOrder(int order) {

this.order = order;

}

}

package com.sdocean.menu.service;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.menu.dao.SysMenuDao;

import com.sdocean.menu.model.CurrMenu;

import com.sdocean.menu.model.SysMenu;

import com.sdocean.role.model.RoleModel;

import com.sdocean.users.model.SysUser;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class SysMenuService {

@Autowired

SysMenuDao menuDao;

//获取该用户权限内的所有的父菜单的列表

public List<SysMenu> getMenuListByUser(SysUser user,String parentCode){

List<SysMenu> menuList = new ArrayList<SysMenu>();

menuList = menuDao.getMenuListByUser(user,parentCode);

//遍历所有父菜单列表,得到所有的有权限的有效的子菜单

for(SysMenu menu:menuList){

menu = menuDao.getMenuByPmenu(menu,user);

}

return menuList;

}

/\*

\* 通过子类ID获得menu信息

\*/

public CurrMenu getCurrMenuById(String menuId){

return menuDao.getCurrMenuById(menuId);

}

/\*

\* 查询当前角色的菜单列表,并通过TREE的形式展现

\*/

public List<ZTreeModel> getMenus4Tree(RoleModel role){

return menuDao.getMenus4Tree(role);

}

/\*

\* 查询当前菜单的首页菜单权限 ,并通过TREE的形式展示

\*/

public List<ZTreeModel> getFirstMenus4Tree(RoleModel role,String menuType){

return menuDao.getFirstMenus4Tree(role,menuType);

}

/\*

\* 查询当前用户的首页菜单权限

\*/

public SysMenu getFirstMenu(SysUser user,String roleType){

return menuDao.getFirstMenu(user,roleType);

}

/\*

\* 将当前用户的菜单保存到SESSION中

\*/

public void saveCurrMenu(CurrMenu menu,HttpServletRequest request){

HttpSession session = request.getSession();

session.setAttribute("currMenu", menu);

@SuppressWarnings("unchecked")

List<SysMenu> menuList = (List<SysMenu>) session.getAttribute("menuList");

for(SysMenu m:menuList){

m.setIsopen(0);

if(m.getCode().equals(menu.getpMenuId())){

m.setIsopen(1);

for(SysMenu c:m.getChildMenu()){

c.setIscurr(0);

if(c.getCode().equals(menu.getcMenuId())){

c.setIscurr(1);

}

}

}

}

session.setAttribute("menuList", menuList);

}

/\*

\* 将当前用户的后台菜单保存到SESSION中

\*/

public void saveManageCurrMenu(CurrMenu menu,HttpServletRequest request){

HttpSession session = request.getSession();

session.setAttribute("manageCurrMenu", menu);

@SuppressWarnings("unchecked")

List<SysMenu> menuList = (List<SysMenu>) session.getAttribute("manageMenuList");

for(SysMenu m:menuList){

m.setIsopen(0);

if(m.getCode().equals(menu.getpMenuId())){

m.setIsopen(1);

for(SysMenu c:m.getChildMenu()){

c.setIscurr(0);

if(c.getCode().equals(menu.getcMenuId())){

c.setIscurr(1);

}

}

}

}

session.setAttribute("manageMenuList", menuList);

}

/\*

\* 根据菜单的CODE获得菜单的2级,

\* 判断该菜单数据前台还是后台

\*/

public SysMenu getLevel2MentByChildCode(String code){

return menuDao.getLevel2MentByChildCode(code);

}

}

package com.sdocean.metadata.action;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

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

import org.springframework.stereotype.Controller;

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.metadata.service.MetadataTableService;

import com.sdocean.page.model.NgColumn;

import com.sdocean.page.model.PageResult;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

import com.sdocean.role.model.RoleModel;

import com.sdocean.role.service.RoleService;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Controller

public class MetaDataTableAction {

private static Logger log = Logger.getLogger(MetaDataTableAction.class);

@Autowired

private MetadataTableService metaService;

@Resource

OperationLogService logService;

/\*

\* 获得角色的列表

\*/

@RequestMapping(value="makeMetaTable4Station.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String makeMetaTable4Station(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = metaService.makeMetaTable4Station(model, 4);

return result.getMessage();

}

}

package com.sdocean.metadata.dao;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.dataQuery.model.DataImportModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.metadata.model.DeviceCollectModel;

import com.sdocean.metadata.model.HalfHour;

import com.sdocean.metadata.model.MetadataModel;

@Component

public class MetadataDao extends OracleEngine{

/\*

\* 将通过CSV导入的数据转换成metadata格式

\*/

public List<MetadataModel> changeMetadata(List<DataImportModel> datas,int collect\_type,String markCode){

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

for(DataImportModel data:datas){

MetadataModel mModel = new MetadataModel();

mModel.setCollect\_time(data.getCollect\_time().replace("/", "-"));

mModel.setCollect\_type(collect\_type);

mModel.setWpId(data.getStationId());

mModel.setDeviceId(data.getDeviceId());

mModel.setIndicator\_code(data.getIndicatorCode());

mModel.setData(data.getData());

mModel.setMarkCode(markCode);

list.add(mModel);

}

return list;

}

/\*

\* 整合数据的采集时间

\*/

public MetadataModel changeNewMetadata(MetadataModel model){

DeviceCollectModel dcm = this.getCollectTimeByDevice(model.getWpId(),model.getDeviceId(), model.getCollect\_time());

String collectTime = dcm.getCollectTime();

String createTime = dcm.getCreateTime();

model.setCollect\_time(collectTime);

model.setCreateTime(createTime);

return model;

}

/\*

\* 根据设备ID,以及传感器的得到结果的时间,得到采集时间

\*/

public DeviceCollectModel getCollectTimeByDevice(int stationId,int deviceId,String createTime){

DeviceCollectModel dcm = null;

StringBuffer sql = new StringBuffer();

sql.append(" select id,deviceid,starttime,endtime,collecttime,daynum");

sql.append(" from aiot\_device\_collect");

sql.append(" where deviceid = ").append(deviceId);

sql.append(" and stationId = ").append(stationId);

sql.append(" and starttime <= date\_format('").append(createTime).append("','%H:%i:%s')");

sql.append(" and endtime > date\_format('").append(createTime).append("','%H:%i:%s')");

sql.append(" limit 1");

dcm = this.queryObject(sql.toString(), DeviceCollectModel.class);

if(dcm!=null&&dcm.getCollectTime()!=null&&dcm.getCollectTime().length()>0){

String day = createTime.substring(0, 11)+dcm.getCollectTime();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date collect = null;

try {

//对采集时间进行时间运算

collect = new Date(sdf.parse(day).getTime()+dcm.getDaynum()\*24\*60\*60\*1000);

String collectTime = sdf.format(collect);

dcm.setCollectTime(collectTime);

dcm.setCreateTime(createTime);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}else{

dcm = new DeviceCollectModel();

dcm.setCollectTime(createTime);

dcm.setCreateTime(createTime);

}

return dcm;

}

/\*

\* 将metadata存入到基础元数据表中

\*/

public void saveMetaData(String tableName,MetadataModel bean){

//保存数据到元数据表中

StringBuffer sql = new StringBuffer("");

sql.append("insert into ").append(tableName).append("(collect\_time, collect\_type, sensor\_type\_code, wpid, indicator\_code, mark\_code, data, createtime,deviceid,dataversion)");

sql.append("values (?,?,?,?,?,?,?,?,?,?)").append(" ON DUPLICATE KEY UPDATE data=VALUES(data)");

Object[] metaParams = new Object[]{

bean.getCollect\_time(),bean.getCollect\_type(),bean.getSensorTypeCode(),

bean.getWpId(),bean.getIndicator\_code(),bean.getMarkCode(),bean.getData(),

bean.getCreateTime(),bean.getDeviceId(),bean.getDataversion()

};

this.update(sql.toString(), metaParams);

}

/\*

\* 根据给出的元数据的时间,判断出在综合元数据中对应的时间

\*/

public HalfHour getSynDateByMdate(String mdate){

HalfHour hh = new HalfHour();

StringBuffer sql = new StringBuffer("");

sql.append(" select id,starttime,endtime,collecttime,daynum");

sql.append(" from aiot\_halfhour\_config");

sql.append(" where starttime <date\_format('").append(mdate).append("','%H:%i:%s')");

sql.append(" and endtime >= date\_format('").append(mdate).append("','%H:%i:%s')");

sql.append(" limit 1");

/\*sql.append("select starttime,endtime from aiot\_halfhour ");

sql.append(" where starttime <'").append(mdate).append("'");

sql.append(" and endtime >='").append(mdate).append("'");\*/

hh = this.queryObject(sql.toString(), HalfHour.class);

String collectTime = mdate.substring(0, 11)+hh.getCollectTime();

hh.setCollectTime(collectTime);

return hh;

}

/\*

\* 将metadata存入到综合元数据表中

\*/

public void saveSynData(String tableName,MetadataModel bean){

StringBuffer syninsertsql = new StringBuffer("");

//根据给出的元数据时间,判断出综合元数据中对应的时间

HalfHour hh = new HalfHour();

hh=this.getSynDateByMdate(bean.getCollect\_time());

//开始进阶SQL语句

syninsertsql.append(" insert into ").append(tableName).append("(collect\_time,data,indicator\_code,wpid,collect\_type,deviceid)");

syninsertsql.append(" values('").append(hh.getCollectTime()).append("',").append(bean.getData()).append(",'").append(bean.getIndicator\_code());

syninsertsql.append("',").append(bean.getWpId()).append(",").append(bean.getCollect\_type()).append(",").append(bean.getDeviceId()).append(")");

syninsertsql.append(" ON DUPLICATE KEY UPDATE data=VALUES(data) ");

/\*syninsertsql.append(" select endtime,").append(bean.getData()).append(",'").append(bean.getIndicator\_code());

syninsertsql.append("',").append(bean.getWpId()).append(",").append(bean.getCollect\_type()).append(",").append(bean.getDeviceId());

syninsertsql.append(" from aiot\_halfhour");

syninsertsql.append(" where '").append(bean.getCollect\_time()).append("' <= endtime and '") .append(bean.getCollect\_time()).append("' > starttime");

syninsertsql.append(" ON DUPLICATE KEY UPDATE data=VALUES(data) ");\*/

this.execute(syninsertsql.toString());

}

}

package com.sdocean.metadata.dao;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.metadata.model.HalfHour;

import com.sdocean.metadata.model.MetadataTable;

import com.sdocean.station.model.StationModel;

@Component

public class MetadataTableDao extends OracleEngine{

/\*

\* 通过起止时间获得需要查询的表名的集合

\*/

public List<MetadataTable> getTables4Meta(StationModel station,String beginTime,String endTime,int type){

List<MetadataTable> tables = new ArrayList<>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.type,a.tablename,a.wpid,a.begintime,a.endtime,a.isactive");

sql.append(" from aiot\_meta\_table a");

sql.append(" where a.begintime <= '").append(endTime).append("' and a.endtime > '").append(beginTime).append("'");

sql.append(" and isactive = 1 and type = ").append(type);

sql.append(" and wpid = ").append(station.getId());

tables = this.queryObjectList(sql.toString(), MetadataTable.class);

return tables;

}

/\*

\* 通过站点,以及查询时间\类型,得到需要查询的元数据表

\*/

public MetadataTable getOneTable(StationModel station,String collect\_time,int type){

MetadataTable table = new MetadataTable();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.type,a.tablename,a.wpid,a.begintime,a.endtime,a.isactive from aiot\_meta\_table a");

sql.append(" where wpid =").append(station.getId());

sql.append(" and begintime <= '").append(collect\_time).append("'");

sql.append(" and endtime > '").append(collect\_time).append("'");

sql.append(" and type = ").append(type);

sql.append(" and isactive = 1");

table = this.queryObject(sql.toString(), MetadataTable.class);

return table;

}

/\*

\* 获得某站点最后的元数据表

\*/

public MetadataTable getLastTable4Station(StationModel model,int type){

StringBuffer lastMetaSql = new StringBuffer("");

lastMetaSql.append(" select id,type,tablename,wpid,begintime,endtime,isactive from aiot\_meta\_table where isactive = 1 and wpid =").append(model.getId());

lastMetaSql.append(" and type = ").append(type);

lastMetaSql.append(" order by endtime desc limit 1");

MetadataTable lastmetaTable = this.queryObject(lastMetaSql.toString(), MetadataTable.class);

return lastmetaTable;

}

//通过上个元数据表,获得下个元数据表

public MetadataTable getNextMetaTable(MetadataTable lastmetaTable,String metaName,StationModel station,int type){

//定义时间格式

DateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

MetadataTable nowMetaTable = new MetadataTable();

//获得上次的截止时间作为这次的起始时间

String bgmeta = lastmetaTable.getEndTime();

Date nDate = null;

try {

nDate = df.parse(bgmeta);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Calendar calendar = Calendar.getInstance();

calendar.setTime(nDate);

//获得一个季度后(三个月)的时间

calendar.add(calendar.MONTH, 3);

String edmeta = df.format(calendar.getTime()).substring(0,10)+" 00:00:00";

int sid = 0;

//除了辽河站点外,其他站点中间为自己的ID

if(station.getId()==10001){

sid = 4;

}else{

sid = station.getId();

}

//获得起始时间的年份简写

int myear = nDate.getYear()-100;

//获得编号的简写

int mid = (nDate.getMonth()+3)/3;

metaName = metaName +"\_"+sid+"\_"+myear+"\_"+mid;

nowMetaTable.setType(type);

nowMetaTable.setWpId(station.getId());

nowMetaTable.setTableName(metaName);

nowMetaTable.setBeginTime(bgmeta);

nowMetaTable.setEndTime(edmeta);

nowMetaTable.setIsactive(1);

return nowMetaTable;

}

/\*

\* 新建元数据表

\*/

public String saveMetaTable(MetadataTable model,int type){

String result = "";

//将数据存入到表中

StringBuffer insertSql = new StringBuffer("");

insertSql.append(" insert into aiot\_meta\_table(type,tablename,wpid,begintime,endtime,isactive)");

insertSql.append(" values(?,?,?,?,?,?) on duplicate key update tablename=values(tablename)");

Object[] insertParams = new Object[]{

type,model.getTableName(),model.getWpId(),

model.getBeginTime(),model.getEndTime(),model.getIsactive()

};

int insert = 0;

try {

insert = this.update(insertSql.toString(), insertParams);

} catch (Exception e) {

result = "保存到元数据表中失败";

return result;

}

//生成元数据表

String liketable = "";

if(type==1){

liketable = "aiot\_metadata";

}else if(type==2){

liketable = "aiot\_syndata";

}

//判断是否存在同名表

String checkSql = "select 1 from "+model.getTableName();

int check = 0;

try {

check = this.queryForInt(checkSql, null);

} catch (Exception e) {

check = -1; //当check=-1时表示没有改表

}

if(check<0){

StringBuffer createtable = new StringBuffer("");

createtable.append("create table if not exists ").append(model.getTableName()).append(" like ").append(liketable);

try {

this.execute(createtable.toString());

} catch (Exception e) {

result = "生成元数据表失败";

return result;

}

}

return result;

}

}

package com.sdocean.metadata.model;

public class DeviceCollectModel {

private int id;

private int stationId;

private int deviceId;

private String startTime;

private String endTime;

private String collectTime;

private String createTime;

private int daynum;

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getStartTime() {

return startTime;

}

public void setStartTime(String startTime) {

this.startTime = startTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public String getCollectTime() {

return collectTime;

}

public void setCollectTime(String collectTime) {

this.collectTime = collectTime;

}

public int getDaynum() {

return daynum;

}

public void setDaynum(int daynum) {

this.daynum = daynum;

}

}

package com.sdocean.metadata.model;

public class HalfHour {

private int id;

private String startTime;

private String endTime;

private String collectTime;

private int daynum;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCollectTime() {

return collectTime;

}

public void setCollectTime(String collectTime) {

this.collectTime = collectTime;

}

public int getDaynum() {

return daynum;

}

public void setDaynum(int daynum) {

this.daynum = daynum;

}

public String getStartTime() {

return startTime;

}

public void setStartTime(String startTime) {

this.startTime = startTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

}

package com.sdocean.metadata.model;

public class MetadataModel {

private int id;

private String collect\_time;

private int collect\_type;

private int wpId;

private String indicator\_code;

private double data;

private String createTime;

private String dataversion;

private int deviceId;

private String markCode;

private String sensorTypeCode;

public String getSensorTypeCode() {

return sensorTypeCode;

}

public void setSensorTypeCode(String sensorTypeCode) {

this.sensorTypeCode = sensorTypeCode;

}

public String getMarkCode() {

return markCode;

}

public void setMarkCode(String markCode) {

this.markCode = markCode;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCollect\_time() {

return collect\_time;

}

public void setCollect\_time(String collect\_time) {

this.collect\_time = collect\_time;

}

public int getCollect\_type() {

return collect\_type;

}

public void setCollect\_type(int collect\_type) {

this.collect\_type = collect\_type;

}

public int getWpId() {

return wpId;

}

public void setWpId(int wpId) {

this.wpId = wpId;

}

public String getIndicator\_code() {

return indicator\_code;

}

public void setIndicator\_code(String indicator\_code) {

this.indicator\_code = indicator\_code;

}

public double getData() {

return data;

}

public void setData(double data) {

this.data = data;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public String getDataversion() {

return dataversion;

}

public void setDataversion(String dataversion) {

this.dataversion = dataversion;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

}

package com.sdocean.metadata.model;

public class MetadataTable {

private int id;

private int type;

private String tableName;

private int wpId;

private String beginTime;

private String endTime;

private int isactive;

public int getWpId() {

return wpId;

}

public void setWpId(int wpId) {

this.wpId = wpId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getType() {

return type;

}

public void setType(int type) {

this.type = type;

}

public String getTableName() {

return tableName;

}

public void setTableName(String tableName) {

this.tableName = tableName;

}

public String getBeginTime() {

return beginTime;

}

public void setBeginTime(String beginTime) {

this.beginTime = beginTime;

}

public String getEndTime() {

return endTime;

}

public void setEndTime(String endTime) {

this.endTime = endTime;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

}

package com.sdocean.metadata.model;

public class SyndataModel {

private String collect\_time;

private String indicatorCode;

private double data;

private int wpid;

private int collect\_type;

private int deviceId;

public String getCollect\_time() {

return collect\_time;

}

public void setCollect\_time(String collect\_time) {

this.collect\_time = collect\_time;

}

public String getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

public double getData() {

return data;

}

public void setData(double data) {

this.data = data;

}

public int getWpid() {

return wpid;

}

public void setWpid(int wpid) {

this.wpid = wpid;

}

public int getCollect\_type() {

return collect\_type;

}

public void setCollect\_type(int collect\_type) {

this.collect\_type = collect\_type;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

}

<%@ page contentType="text/html; charset=UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>

<!DOCTYPE html>

<html >

<c:set var="ctx" value="${pageContext.request.contextPath}"/>

<head>

<meta charset="utf-8" />

<title>${system.systemName }</title>

<meta name="keywords" content="入海污染源" />

<style>

.kongzhimianban-img{

width:1000px;

height:530px;

background-image:url(images/station/adcp/mapground.png);

background-size: cover;

}

</style>

</head>

<body ng-app="myApp" ng-controller="customersCtrl">

<%@ include file="../common/top.jsp" %>

<!-- angularjs select tree -->

<link rel="stylesheet" href="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.css" />

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.js"></script>

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.tpl.js"></script>

<script type="text/javascript" src="${ctx }/yantai/js/station/stationGuijiInfo.js"></script>

<div class="main-container" id="main-container">

<script type="text/javascript">

try{ace.settings.check('main-container' , 'fixed')}catch(e){}

</script>

<div class="main-container-inner">

<%@ include file="../common/leftMenu4guiji.jsp" %>

<div class="main-content" >

<div class="breadcrumbs" id="breadcrumbs">

<script type="text/javascript">

try{ace.settings.check('breadcrumbs' , 'fixed')}catch(e){}

</script>

<ul class="breadcrumb">

<li>

<i class="icon-home home-icon"></i>

<a href="#">${currMenu.pMenuName }</a>

</li>

<li>

<a href="#">${currMenu.cMenuName }</a>

</li>

</ul><!-- .breadcrumb -->

</div>

<div style="word-wrap:break-word; overflow:hidden;">

<ul class="seachform">

<li><label>站点列表</label>

<div class="vocation">

<select ng-model="u.stationId" class="select3"

ng-options="option.id as option.title for option in stationList">

</select>

</div>

</li>

<!-- <li><label>参数类型</label> -->

<li><label>状态</label>

<div class="vocation">

<select ng-model="u.isactive" class="select3"

ng-change="query()"

ng-options="option.value as option.name for option in activelist">

</select>

</div>

</li>

<li><label>&nbsp;</label><input name="" type="button" class="scbtn" ng-click="query()" value="查询"/></li>

<li><label>移动目标位置 经度:</label>

<div class="vocation">

<input type="number" ng-model="u.code1" placeholder="目标经度" >

</div>

</li>

<li><label> 维度:</label>

<div class="vocation">

<input type="number" ng-model="u.code2" placeholder="目标维度" >

</div>

</li>

<li><label>&nbsp;</label><input name="" type="button" class="scbtn" value="移动"/></li>

</ul>

</div>

<div class="container-fluid">

<div class="row-fluid">

<div class="col-xs-12 col-sm-8 " style="padding-left:100px">

<!-- <div id="datashow"></div> -->

<div class="widget-body" style=" height: 100%; margin: 0 auto">

<div class="widget-main no-padding kongzhimianban-img" >

<img src="images/station/adcp/10007.png" id="door\_left" style="margin-left:200px;margin-top:257px;width:560px;height:330px" ng-click="showMubiao2()"/>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<script type="text/ng-template" id="popupTmpl.html">

<div id="editdiv" class="page-content" style="margin-top:-13px;margin-bottom:-13px;margin-left:-13px;margin-right:-13px">

<div class="page-header">

<strong>

{{station.title}}

<i class="icon-double-angle-right"></i>

</strong>

</div><!-- /.page-header -->

<div class="row">

<div class="col-xs-12" id ="ppdiv">

<form class="form-horizontal" role="form" name="myForm" novalidate>

<input type="hidden" id="form-field-1" ng-model="station.id" />

<div class="form-group">

<label class="col-sm-3 control-label no-padding-right" for="form-field-1"> 目标经度 </label>

<div class="col-sm-9">

<input type="text" ng-model="station.jingdu" name="code" readonly class="col-xs-10 col-sm-5" maxlength="10"

value="100" />

</div>

</div>

<div class="space-4"></div>

<div class="form-group">

<label class="col-sm-3 control-label no-padding-right" for="form-field-1"> 目标维度 </label>

<div class="col-sm-9">

<input type="text" ng-model="station.weidu" name="code" readonly class="col-xs-10 col-sm-5" maxlength="10"

value="200" />

</div>

</div>

<div class="space-4"></div>

<div class="space-4"></div>

<div class="clearfix form-actions">

<div class="col-md-offset-3 col-md-9">

<button class="btn btn-info" type="button" ng-click="save()" >

<i class="icon-ok bigger-110"></i>

保存

</button>

&nbsp; &nbsp; &nbsp;

</div>

</div>

</form>

</div>

<div>

</div>

</script>

<script type="text/ng-template" id="popupTmpl2.html">

<div id="editdiv" class="page-content" style="margin-top:-13px;margin-bottom:-13px;margin-left:-13px;margin-right:-13px">

<div class="page-header">

<strong>

{{station.title}}

<i class="icon-double-angle-right"></i>

</strong>

</div><!-- /.page-header -->

<div class="row">

<div class="col-xs-12" id ="ppdiv">

<form class="form-horizontal" role="form" name="myForm" novalidate>

<div class="form-group">

<img src="images/station/qiqiu.png" id="door\_left" style="margin-left:0px;margin-top:0px;width:100%"/>

</div>

<div class="space-4"></div>

<div class="space-4"></div>

<div class="clearfix form-actions">

<div class="col-md-offset-3 col-md-9">

<button class="btn btn-info" type="button" ng-click="save()" >

<i class="icon-ok bigger-110"></i>

保存

</button>

&nbsp; &nbsp; &nbsp;

</div>

</div>

</form>

</div>

<div>

</div>

</script>

</body>

</html>