package com.sdocean.security;

import java.util.Collection;

import java.util.Iterator;

import org.springframework.security.access.AccessDecisionManager;

import org.springframework.security.access.AccessDeniedException;

import org.springframework.security.access.ConfigAttribute;

import org.springframework.security.authentication.InsufficientAuthenticationException;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.GrantedAuthority;

public class MyAccessDecisionManager implements AccessDecisionManager {

public void decide(Authentication authentication, Object object, Collection<ConfigAttribute> configAttributes) throws AccessDeniedException, InsufficientAuthenticationException {

Authentication authentication1= authentication;

if(configAttributes == null) {

return;

}

//所请求的资源拥有的权限(这里一个资源只对应一个权限)

Iterator<ConfigAttribute> iterator = configAttributes.iterator();

while(iterator.hasNext()) {

ConfigAttribute configAttribute = iterator.next();

//访问所请求资源所需要的权限

String needPermission = configAttribute.getAttribute();

//用户所拥有的权限authentication

for(GrantedAuthority ga : authentication.getAuthorities()) {

if(needPermission.equals(ga.getAuthority())) {

return;

}

}

}

//没有权限

throw new AccessDeniedException(" 没有权限访问！ ");

}

public boolean supports(ConfigAttribute attribute) {

// TODO Auto-generated method stub

return true;

}

public boolean supports(Class<?> clazz) {

// TODO Auto-generated method stub

return true;

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import org.springframework.security.access.SecurityMetadataSource;

import org.springframework.security.access.intercept.AbstractSecurityInterceptor;

import org.springframework.security.access.intercept.InterceptorStatusToken;

import org.springframework.security.web.FilterInvocation;

import org.springframework.security.web.access.intercept.FilterInvocationSecurityMetadataSource;

public class MySecurityFilter extends AbstractSecurityInterceptor implements Filter {

//与applicationContext-security.xml里的myFilter的属性securityMetadataSource对应，

//其他的两个组件，已经在AbstractSecurityInterceptor定义

private FilterInvocationSecurityMetadataSource securityMetadataSource;

@Override

public SecurityMetadataSource obtainSecurityMetadataSource() {

return this.securityMetadataSource;

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

FilterInvocation fi = new FilterInvocation(request, response, chain);

invoke(fi);

}

private void invoke(FilterInvocation fi) throws IOException, ServletException {

// object为FilterInvocation对象

//super.beforeInvocation(fi);源码

//1.获取请求资源的权限

//执行Collection<ConfigAttribute> attributes = SecurityMetadataSource.getAttributes(object);

//2.是否拥有权限

//this.accessDecisionManager.decide(authenticated, object, attributes);

InterceptorStatusToken token = super.beforeInvocation(fi);

try {

fi.getChain().doFilter(fi.getRequest(), fi.getResponse());

} finally {

super.afterInvocation(token, null);

}

}

public FilterInvocationSecurityMetadataSource getSecurityMetadataSource() {

return securityMetadataSource;

}

public void setSecurityMetadataSource(FilterInvocationSecurityMetadataSource securityMetadataSource) {

this.securityMetadataSource = securityMetadataSource;

}

public void init(FilterConfig arg0) throws ServletException {

// TODO Auto-generated method stub

}

public void destroy() {

// TODO Auto-generated method stub

}

@Override

public Class<? extends Object> getSecureObjectClass() {

//下面的MyAccessDecisionManager的supports方面必须放回true,否则会提醒类型错误

return FilterInvocation.class;

}

}

package com.sdocean.security;

import java.util.ArrayList;

import java.util.Collection;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.springframework.security.access.ConfigAttribute;

import org.springframework.security.access.SecurityConfig;

import org.springframework.security.web.FilterInvocation;

import org.springframework.security.web.access.intercept.FilterInvocationSecurityMetadataSource;

import com.sdocean.resources.dto.Resources;

import com.sdocean.resources.service.ResourcesManager;

//1 加载资源与权限的对应关系

public class MySecurityMetadataSource implements FilterInvocationSecurityMetadataSource {

//由spring调用

public MySecurityMetadataSource(ResourcesManager resourcesManager) {

this.resourcesManager = resourcesManager;

loadResourceDefine();

}

private ResourcesManager resourcesManager;

private static Map<String, Collection<ConfigAttribute>> resourceMap = null;

public void setResourcesDao(ResourcesManager resourcesManager) {

this.resourcesManager = resourcesManager;

}

public Collection<ConfigAttribute> getAllConfigAttributes() {

// TODO Auto-generated method stub

return null;

}

public boolean supports(Class<?> clazz) {

// TODO Auto-generated method stub

return true;

}

//加载所有资源与权限的关系

private void loadResourceDefine() {

if(resourceMap == null) {

resourceMap = new HashMap<String, Collection<ConfigAttribute>>();

List<Resources> resources = this.resourcesManager.findAll();

for (Resources resource : resources) {

Collection<ConfigAttribute> configAttributes = new ArrayList<ConfigAttribute>();

//以权限名封装为Spring的security Object

ConfigAttribute configAttribute = new SecurityConfig(resource.getAUTHKEY());

configAttributes.add(configAttribute);

resourceMap.put(resource.getResUrl(), configAttributes);

}

}

// Set<Entry<String, Collection<ConfigAttribute>>> resourceSet = resourceMap.entrySet();

// Iterator<Entry<String, Collection<ConfigAttribute>>> iterator = resourceSet.iterator();

}

//返回所请求资源所需要的权限

public Collection<ConfigAttribute> getAttributes(Object object) throws IllegalArgumentException {

String requestUrl = ((FilterInvocation) object).getRequestUrl();

if(resourceMap == null) {

loadResourceDefine();

}

return resourceMap.get(requestUrl);

}

}

package com.sdocean.security;

import java.util.Collection;

import java.util.HashSet;

import java.util.Set;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import com.sdocean.users.model.SysUser;

import com.sdocean.users.service.UsersManager;

public class MyUserDetailServiceImpl implements UserDetailsService {

private UsersManager usersManager;

public void setUsersManager(UsersManager usersManager) {

this.usersManager = usersManager;

}

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

SysUser users = this.usersManager.getUsersByAccount(username);

if(users == null) {

throw new UsernameNotFoundException(username);

}

Collection<GrantedAuthority> grantedAuths = obtionGrantedAuthorities(users);

boolean enables = true;

boolean accountNonExpired = true;

boolean credentialsNonExpired = true;

boolean accountNonLocked = true;

User userdetail = new User(users.getUserName(), users.getPassword(), enables, accountNonExpired, credentialsNonExpired, accountNonLocked, grantedAuths);

return userdetail;

}

//取得用户的权限

private Set<GrantedAuthority> obtionGrantedAuthorities(SysUser user) {

Set<GrantedAuthority> authSet = new HashSet<GrantedAuthority>();

/\*List<Authority> authoritysByUserAccount = this.authorityManager.getAuthoritysByUserAccount(user);

for (Authority authority : authoritysByUserAccount) {

authSet.add(new SimpleGrantedAuthority(authority.getAuthKey()));

}\*/

return authSet;

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.commons.lang.StringUtils;

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

import org.springframework.security.authentication.AuthenticationServiceException;

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.AuthenticationException;

import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.log.service.SysLoginLogService;

import com.sdocean.users.model.SysUser;

import com.sdocean.users.service.UsersManager;

public class MyUsernamePasswordAuthenticationFilter extends UsernamePasswordAuthenticationFilter{

public static final String VALIDATE\_CODE = "validateCode";

public static final String USERNAME = "username";

public static final String PASSWORD = "password";

private UsersManager usersManager;

public void setUsersManager(UsersManager usersManager) {

this.usersManager = usersManager;

}

private SysLoginLogService loginLogService;

public void setLoginLogService(SysLoginLogService loginLogService){

this.loginLogService = loginLogService;

}

@Override

public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response) throws AuthenticationException {

if (!request.getMethod().equals("POST")) {

throw new AuthenticationServiceException("Authentication method not supported: " + request.getMethod());

}

//检测验证码

// checkValidateCode(request);

String username = obtainUsername(request);

String password = obtainPassword(request);

//验证用户账号与密码是否对应

username = username.trim();

SysUser users = this.usersManager.getUsersByAccount(username);

if(users == null || !users.getPassword().equals(password)) {

throw new AuthenticationServiceException("用户名或者密码错误！");

}

//将用户信息存入到session中

HttpSession session = request.getSession();

session.setAttribute("user", users);

UsernamePasswordAuthenticationToken authRequest = new UsernamePasswordAuthenticationToken(username, password);

// 允许子类设置详细属性

setDetails(request, authRequest);

//loginLogService.saveSysLoginLog(request);

return this.getAuthenticationManager().authenticate(authRequest);

}

protected void checkValidateCode(HttpServletRequest request) {

HttpSession session = request.getSession();

String sessionValidateCode = obtainSessionValidateCode(session);

//让上一次的验证码失效

session.setAttribute(VALIDATE\_CODE, null);

String validateCodeParameter = obtainValidateCodeParameter(request);

if (StringUtils.isEmpty(validateCodeParameter) || !sessionValidateCode.equalsIgnoreCase(validateCodeParameter)) {

throw new AuthenticationServiceException("验证码错误！");

}

}

private String obtainValidateCodeParameter(HttpServletRequest request) {

Object obj = request.getParameter(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

protected String obtainSessionValidateCode(HttpSession session) {

Object obj = session.getAttribute(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainUsername(HttpServletRequest request) {

Object obj = request.getParameter(USERNAME);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainPassword(HttpServletRequest request) {

Object obj = request.getParameter(PASSWORD);

return null == obj ? "" : obj.toString();

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.Authentication;

import org.springframework.security.web.authentication.SavedRequestAwareAuthenticationSuccessHandler;

import org.springframework.security.web.savedrequest.RequestCache;

public class SavedRequestAwareAuthenticationSuccessHandlerText extends SavedRequestAwareAuthenticationSuccessHandler{

@Override

public void onAuthenticationSuccess(HttpServletRequest request,

HttpServletResponse response, Authentication authentication)

throws ServletException, IOException {

// TODO Auto-generated method stub

super.onAuthenticationSuccess(request, response, authentication);

}

@Override

public void setRequestCache(RequestCache requestCache) {

// TODO Auto-generated method stub

super.setRequestCache(requestCache);

}

}

package com.sdocean.users.model;

public class SysUser {

private Integer id;

private String userName;

private String password;

private String realName;

private int positionId;

private String positionName;

private String telephone;

private String phone;

private String email;

private String birthday;

private String picurl;

private String companyId;

private String companyName;

private Integer logintype;

private Integer isactive;

private String isactiveName;

//修改密码用

private String newPass;

private String confimPass;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Integer getLogintype() {

return logintype;

}

public void setLogintype(Integer logintype) {

this.logintype = logintype;

}

public Integer getIsactive() {

return isactive;

}

public void setIsactive(Integer isactive) {

this.isactive = isactive;

}

public String getNewPass() {

return newPass;

}

public void setNewPass(String newPass) {

this.newPass = newPass;

}

public String getConfimPass() {

return confimPass;

}

public void setConfimPass(String confimPass) {

this.confimPass = confimPass;

}

public String getBirthday() {

return birthday;

}

public void setBirthday(String birthday) {

this.birthday = birthday;

}

public String getRealName() {

return realName;

}

public void setRealName(String realName) {

this.realName = realName;

}

public String getPositionName() {

return positionName;

}

public void setPositionName(String positionName) {

this.positionName = positionName;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public int getPositionId() {

return positionId;

}

public void setPositionId(int positionId) {

this.positionId = positionId;

}

public String getTelephone() {

return telephone;

}

public void setTelephone(String telephone) {

this.telephone = telephone;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

/\*public Date getBirthday() {

return birthday;

}

public void setBirthday(Date birthday) {

this.birthday = new Date();

}\*/

public String getPicurl() {

return picurl;

}

public void setPicurl(String picurl) {

this.picurl = picurl;

}

public String getCompanyId() {

return companyId;

}

public void setCompanyId(String companyId) {

this.companyId = companyId;

}

}

package com.sdocean.users.model;

public class UserStationDevice {

private int userId;

private String userName;

private int stationId;

private String stationName;

private int deviceId;

private String deviceName;

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 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;

}

}

/\*\*

\* PersonService.java created on 2010-11-1

\*/

package com.sdocean.users.service;

import java.util.Set;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.frame.util.WebUtils;

import com.sdocean.users.model.SysUser;

@Service

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

public class UsersHelper {

@Resource

UsersManager usersManager;

public SysUser getCurUser(){

String account=WebUtils.getCurUserAccount();

if(account!=null){

return usersManager.getUsersByAccount(account);

}else{

return null;

}

}

public boolean hasAuthOfCurUser(String authKey){

Set<String> auths=WebUtils.getCurUserAuths();

if(auths!=null){

return auths.contains(authKey);

}

return false;

}

}

package com.sdocean.users.service;

import java.util.ArrayList;

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.ZTreeModel;

import com.sdocean.company.dao.CompanyDao;

import com.sdocean.page.model.NgColumn;

import com.sdocean.page.model.UiColumn;

import com.sdocean.role.model.RoleModel;

import com.sdocean.sms.model.SmsSettingModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.dao.UsersDao;

import com.sdocean.users.model.SysUser;

@Service

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

public class UsersManager {

@Resource

private UsersDao usersDao;

@Resource

private CompanyDao companyDao;

@Transactional(readOnly = true)

public SysUser getUsersByAccount(String account) {

SysUser usersByAccount = this.usersDao.getUserByName(account);

return usersByAccount;

}

/\*

\* 为人员管理的查询结果添加表头

\*/

public List<UiColumn> getCols4UserList(){

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

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

UiColumn col2 = new UiColumn("用户名", "userName", true, "\*");

UiColumn col13 = new UiColumn("用户名", "realName", true, "\*");

UiColumn col3 = new UiColumn("手机号", "telephone", true, "\*");

UiColumn col4 = new UiColumn("职业id", "positionId", false, "\*");

UiColumn col5 = new UiColumn("职业", "positionName", true, "\*");

UiColumn col6 = new UiColumn("电话", "phone", true, "\*");

UiColumn col7 = new UiColumn("生日", "birthday", true, "\*");

UiColumn col8 = new UiColumn("文件路径", "picurl", true, "\*");

UiColumn col9 = new UiColumn("公司ID", "companyId", false, "\*");

UiColumn col10 = new UiColumn("单位名称", "companyName", true, "\*");

UiColumn col11 = new UiColumn("isactive", "isactive", false, "\*");

UiColumn col12 = new UiColumn("状态", "isactiveName", true, "\*");

UiColumn col14 = new UiColumn("邮箱", "email", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col13);

cols.add(col3);

cols.add(col4);

cols.add(col5);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col14);

cols.add(col10);

cols.add(col11);

cols.add(col12);

return cols;

}

/\*public List<UiColumn> getCols4UserList(){

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

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

UiColumn col2 = new UiColumn("用户名", "userName", true, "\*");

UiColumn col13 = new UiColumn("用户名", "realName", true, "\*");

UiColumn col3 = new UiColumn("手机号", "telephone", true, "\*");

UiColumn col4 = new UiColumn("职业id", "positionId", false, "\*");

UiColumn col5 = new UiColumn("职业", "positionName", true, "\*");

UiColumn col6 = new UiColumn("电话", "phone", true, "\*");

UiColumn col7 = new UiColumn("生日", "birthday", true, "\*");

UiColumn col8 = new UiColumn("文件路径", "picurl", true, "\*");

UiColumn col9 = new UiColumn("公司ID", "companyId", false, "\*");

UiColumn col10 = new UiColumn("公司名称", "companyName", true, "\*");

UiColumn col11 = new UiColumn("isactive", "isactive", false, "\*");

UiColumn col12 = new UiColumn("状态", "isactiveName", true, "\*");

UiColumn col14 = new UiColumn("邮箱", "email", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col13);

cols.add(col3);

cols.add(col4);

cols.add(col5);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col14);

cols.add(col10);

cols.add(col11);

cols.add(col12);

return cols;

}\*/

/\*

\* 为人员管理查询结果

\*/

public List<SysUser> getRows4UserList(SysUser model){

return usersDao.getRows4UserList(model);

}

/\*

\* 修改人员信息

\*/

public Result saveUserChange(SysUser user){

return usersDao.saveUserChange(user);

}

/\*

\* 新增人员信息

\*

\*/

public Result saveNewUser(SysUser user){

return usersDao.saveNewUser(user);

}

/\*

\* 获得所有有效公司下的所有有效人员的列表

\* 根据角色,将角色权限内的人物选定

\* 公司按照公司CODE排序,

\* 人员按照职位CODE排序

\*/

public List<ZTreeModel> getCompUsers4ZTree(RoleModel role){

//定义总的返回树

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

//初始化第一列数据

ZTreeModel com = new ZTreeModel("C0000", "0", "人员", null, true, 0, null, null, null, true);

//得到公司列表集合

List<ZTreeModel> comptrees = companyDao.getComList4ZTree();

//得到人员列表集合

List<ZTreeModel> usertrees = usersDao.getCompUsers4ZTree(role);

trees.add(com);

trees.addAll(comptrees);

trees.addAll(usertrees);

return trees;

}

/\*

\* 修改用户的人员信息

\*/

public Result saveUserNowChange(SysUser model){

return usersDao.saveUserNowChange(model);

}

/\*

\* 用户修改密码

\*/

public Result saveUserPassword(SysUser model){

return usersDao.saveUserPassword(model);

}

/\*

\* 获得拥有该站点权限下的所有用户的列表

\*/

public List<SysUser> getUsersByStation(StationModel station){

return usersDao.getUsersByStation(station);

}

/\*

\* 获得所有有效公司下的所有有效人员的列表

\* 根据角色,将角色权限内的人物选定

\* 公司按照公司CODE排序,

\* 人员按照职位CODE排序

\*/

public List<ZTreeModel> getCompUsers4StationSmsZTree(SmsSettingModel model){

//定义总的返回树

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

//初始化第一列数据

ZTreeModel com = new ZTreeModel("C0000", "0", "人员", null, true, 0, null, null, null, true);

//得到公司列表集合

List<ZTreeModel> comptrees = companyDao.getComList4ZTree();

//得到人员列表集合

List<ZTreeModel> usertrees = usersDao.getCompUsers4StatinSmsZTree(model);

trees.add(com);

trees.addAll(comptrees);

trees.addAll(usertrees);

return trees;

}

}

package com.sdocean.warn.action;

import java.text.DateFormat;

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 javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

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.dictionary.model.PublicModel;

import com.sdocean.dictionary.service.PublicService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.StationService;

import com.sdocean.users.model.SysUser;

import com.sdocean.warn.model.WarnModel;

import com.sdocean.warn.model.WarnValueModel;

import com.sdocean.warn.service.WarnService;

@Controller

public class WarnAction {

@Resource

WarnService warnService;

@Resource

OperationLogService logService;

@Resource

StationService stationService;

@Resource

PublicService publicService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_warnsetting.do")

public ModelAndView handleRequestInternal(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/warn/warnSettingInfo");

return mav;

}

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

@ResponseBody

public String showWarnSetting(@ModelAttribute("model") WarnModel model,HttpServletRequest request,

HttpServletResponse response){

//获得当前的人员信息

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

PageResult result = new PageResult();

//为查询结果添加表头

List<UiColumn> cols = warnService.getCols4List();

result.setCols(cols);

//为查询结果添加结果

List<WarnModel> rows = warnService.getWarnList(model, user);

result.setRows(rows);

return JsonUtil.toJson(result);

}

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

@ResponseBody

public String saveWarnChange(@ModelAttribute("model") WarnModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = warnService.saveWarnChange(model);

//保存操作记录

logService.saveOperationLog(result, request);

return result.getMessage();

}

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

@ResponseBody

public String saveNewWarn(@ModelAttribute("model") WarnModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = warnService.saveNewWarn(model);

//保存操作记录

logService.saveOperationLog(result, request);

return result.getMessage();

}

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

@ResponseBody

public String deleWarnSetting(@ModelAttribute("model") WarnModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = warnService.deleWarnSetting(model);

//保存操作记录

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 跳转到预警告警信息页面

\*/

@RequestMapping("info\_warnvalue.do")

public ModelAndView info\_warnvalue(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getPageVision()+"/warn/warnValueInfo");

return mav;

}

/\*

\* 初始化预警告警信息页面

\*/

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

@ResponseBody

public String init\_warnvalue(HttpServletRequest request,

HttpServletResponse response){

//获得当前的人员信息

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

WarnValueModel model = new WarnValueModel();

//根据人员权限初始化站点列表

List<StationModel> stationList = stationService.getStations4User(user);

List<StationModel> stations = new ArrayList<StationModel>();

StationModel allStation = new StationModel();

allStation.setId(0);

allStation.setTitle("全部站点");

stations.add(allStation);

stations.addAll(stationList);

//初始化类型列表

List<PublicModel> publics = publicService.getPublicsByParent("0008");

List<PublicModel> typeList = new ArrayList<PublicModel>();

PublicModel type = new PublicModel();

type.setClassId(0);

type.setValue("全部");

typeList.add(type);

typeList.addAll(publics);

//初始化查询时间

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

Calendar calendar = Calendar.getInstance();

calendar.setTime(new Date());

//设置结束时间

String endDate = beginDf.format(calendar.getTime());

//设置开始时间

calendar.add(Calendar.MONTH, -1);

String beginDate = beginDf.format(calendar.getTime());

model.setStationList(stations);

model.setTypeList(typeList);

model.setBeginDate(beginDate);

model.setEndDate(endDate);

return JsonUtil.toJson(model);

}

/\*

\* 展示预警告警信息

\*/

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

@ResponseBody

public String showWarnValues(@ModelAttribute("model") WarnValueModel model,HttpServletRequest request,

HttpServletResponse response){

//获得当前的人员信息

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

PageResult result = new PageResult();

//为查询结果添加表头

List<UiColumn> cols = warnService.getCols4WarnValue();

result.setCols(cols);

//为查询结果添加结果

List<WarnValueModel> rows = warnService.getWarnValueRows(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存操作信息

\*/

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

@ResponseBody

public String saveOperaValue(@ModelAttribute("model") WarnValueModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = warnService.saveOperaValue(model);

//保存操作记录

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 预警告警信息批量修改操作信息

\*/

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

@ResponseBody

public String updateAllOperaValue(@ModelAttribute("model") WarnValueModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = warnService.updateAllOperaValue(model);

//保存操作记录

logService.saveOperationLog(result, request);

return result.getMessage();

}

}

package com.sdocean.warn.dao;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

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.station.dao.StationDao;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

import com.sdocean.warn.model.Warn4FirstModel;

import com.sdocean.warn.model.WarnModel;

import com.sdocean.warn.model.WarnValueModel;

@Component

public class WarnDao extends OracleEngine {

@Resource

StationDao stationDao;

/\*

\* 为首页读取多个站点的预警信息

\*/

public Warn4FirstModel getWarn4First4All(Long warnType,List<StationModel> watchPoints,String beginDate){

Warn4FirstModel warn = new Warn4FirstModel();

String typeSql = "";

String warnTypeString = "";

if(warnType==0){//如果为0则表示预警信息

warnTypeString = "预警信息";

typeSql = " a.max\_value as upper,a.min\_value as lower";

}else{ //如果为1则表示告警信息

warnTypeString = "告警信息";

typeSql = " a.max\_value as upper,a.min\_value as lower";

}

List<WarnModel> warnModels = null;

for(StationModel watchPoint:watchPoints){

StringBuffer sql = new StringBuffer("");

sql.append(" select a.collect\_time as collecttime,c.title as wpName,c.id as wpId,d.title,e.logo as unitname,a.value as data,");

sql.append(" if(a.operate=0,'未处理','已处理') as operate,");

sql.append(typeSql);

sql.append(" from aiot\_warn\_alarm\_value a,dm\_indicator d,g\_unit e,aiot\_watch\_point c");

sql.append(" where a.indicatorCode = d.code and d.unitid = e.id");

sql.append(" and a.wpid = c.id");

sql.append(" and a.collect\_time >='").append(beginDate).append("'");

sql.append(" and a.wpId =").append(watchPoint.getId());

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

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

sql.append(" limit 0,3");

List<WarnModel> warnModel = null;

warnModel = this.queryObjectList(sql.toString(), WarnModel.class);

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

warnModels.addAll(warnModel);

}

}

warn.setWarnType(warnTypeString);

warn.setWarnModels(warnModels);

return warn;

}

/\*

\* 为首页读取预警信息

\*/

public Warn4FirstModel getWarn4First(Long warnType,StationModel watchPoint,String beginDate){

Warn4FirstModel warn = new Warn4FirstModel();

String typeSql = "";

String warnTypeString = "";

if(warnType==0){//如果为0则表示预警信息

warnTypeString = "预警信息";

typeSql = " a.max\_value as upper,a.min\_value as lower";

}else{ //如果为1则表示告警信息

warnTypeString = "告警信息";

typeSql = " a.max\_value as upper,a.min\_value as lower";

}

//开始拼接SQL语句

StringBuffer sql = new StringBuffer("");

sql.append(" select a.collect\_time as collecttime,d.title,e.logo as unitname,a.value as data,");

sql.append(" if(a.operate=0,'未处理','已处理') as operate,");

sql.append(typeSql);

sql.append(" from aiot\_warn\_alarm\_value a,dm\_indicator d,g\_unit e");

sql.append(" where a.indicatorCode = d.code and d.unitid = e.id");

sql.append(" and a.collect\_time >='").append(beginDate).append("'");

sql.append(" and a.wpId =").append(watchPoint.getId());

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

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

sql.append(" limit 0,3");

List<WarnModel> warnModels = null;

warnModels = this.queryObjectList(sql.toString(), WarnModel.class);

warn.setWarnType(warnTypeString);

warn.setWarnModels(warnModels);

return warn;

}

/\*

\* 根据条件查询出预警告警配置列表

\* 根据当前登录人员查询权限内的站点

\*/

public List<WarnModel> getWarnList(WarnModel model,SysUser user){

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

//获得该人员权限下的所有站点的集合

List<StationModel> stations = stationDao.getStations4User(user);

//拼接INSQL语句

StringBuffer insql = new StringBuffer("0");

for(StationModel station:stations){

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

}

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.wp\_id as wpId,b.title as wpName,a.indicator\_code as indicatorCode,c.title as indicatorName,");

sql.append(" a.warn\_lower as warnLower,a.warn\_upper as warnUpper,a.alarm\_lower as alarmLower,a.alarm\_upper as alarmUpper");

sql.append(" from aiot\_warn\_alarm\_config a,aiot\_watch\_point b,dm\_indicator c");

sql.append(" where a.wp\_id = b.id and b.isactive = 1");

sql.append(" and a.indicator\_code = c.code and c.isactive = 1");

sql.append(" and a.wp\_id in (").append(insql).append(")");

//添加查询条件

if(model!=null&&model.getWpId()>10000){

sql.append(" and a.wp\_id =").append(model.getWpId());

}

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

return list;

}

/\*

\* 修改预警告警配置

\*/

public Result saveWarnChange(WarnModel model){

Result result = new Result();

//初始化返回结果

result.setDotype(result.UPDATE);

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

result.setResult(result.SUCCESS);

result.setMessage("修改成功");

//判断该条配置是否冲突

StringBuffer checksql = new StringBuffer();

checksql.append(" select count(1) from aiot\_warn\_alarm\_config where wp\_id =? and indicator\_code =? and id <> ?");

Object[] checkParams = new Object[]{

model.getWpId(),model.getIndicatorCode(),model.getId()

};

int count = 0;

try {

count = this.queryForInt(checksql.toString(), checkParams);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("验证唯一性时失败");

return result;

}

if(count>0){

result.setResult(result.FAILED);

result.setMessage("已经包含同样的配置");

return result;

}

StringBuffer sql = new StringBuffer("");

sql.append(" update aiot\_warn\_alarm\_config set wp\_id=?,indicator\_code=?,warn\_lower=?,warn\_upper=?,alarm\_lower=?,alarm\_upper=? where id=?");

Object[] params = new Object[]{

model.getWpId(),model.getIndicatorCode(),model.getWarnLower(),

model.getWarnUpper(),model.getAlarmLower(),model.getAlarmUpper(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("修改失败,请重试");

}

return result;

}

/\*

\* 保存新增预警告警配置

\*/

public Result saveNewWarn(WarnModel model){

Result result = new Result();

//初始化返回结果

result.setDotype(result.ADD);

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

result.setResult(result.SUCCESS);

result.setMessage("新增成功");

StringBuffer sql = new StringBuffer("");

sql.append(" insert into aiot\_warn\_alarm\_config(wp\_id,indicator\_code,warn\_lower,warn\_upper,alarm\_lower,alarm\_upper)");

sql.append(" values(?,?,?,?,?,?) on duplicate key update warn\_lower=values(warn\_lower),warn\_upper=values(warn\_upper),alarm\_lower=values(alarm\_lower),alarm\_upper=values(alarm\_upper)");

Object[] params = new Object[]{

model.getWpId(),model.getIndicatorCode(),model.getWarnLower(),

model.getWarnUpper(),model.getAlarmLower(),model.getAlarmUpper()

};

int res = 0;

try {

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

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("新增失败,请重试");

}

return result;

}

/\*

\* 删除选中的预警告警配置

\*/

public Result deleWarnSetting(WarnModel 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");

//拼接删除语句

StringBuffer sql = new StringBuffer("");

sql.append(" delete from aiot\_warn\_alarm\_config where id in (").append(model.getIds()).append(")");

int res = 0 ;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("删除失败");

}

return result;

}

/\*

\* 展示条件内的预警告警信息

\*/

public List<WarnValueModel> getWarnValueRows(WarnValueModel model){

System.out.println(JsonUtil.toJson(model));

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.type,e.value as typeName,a.value,a.collect\_time,a.operate,case a.operate when 0 then '未操作' when 1 then '已操作' end as operateName,");

sql.append(" a.flag,a.remark,a.wpId,b.title as wpName,a.deviceId,c.name as deviceName,");

sql.append(" a.indicatorCode,d.title as indicatorName,a.min\_value as minConfig,a.max\_value as maxConfig,");

sql.append(" g.logo as unitName");

sql.append(" from aiot\_warn\_alarm\_value a,aiot\_watch\_point b,device\_catalog c,sys\_public e,");

sql.append(" dm\_indicator d left join g\_unit g on d.unitId = g.id and g.isactive = 1");

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

sql.append(" and a.deviceId = c.id and a.indicatorCode = d.code and d.isactive = 1");

sql.append(" and a.type = e.classid and e.parentcode = '0008'");

//添加查询条件

if(model!=null){

//添加站点条件

if(model.getWpId()>10000){

sql.append(" and a.wpid = ").append(model.getWpId());

}

//添加类型条件

if(model.getType()>0){

sql.append(" and a.type = ").append(model.getType());

}

//添加时间条件

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

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

}

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

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

}

if(model.getOperate()>=0){

sql.append(" and a.operate = ").append(model.getOperate());

}

}

//增加排序

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

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

return list;

}

/\*

\* 为预警告警信息添加操作以及备注

\*/

public Result saveOperaValue(WarnValueModel 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("操作成功");

StringBuffer sql = new StringBuffer();

sql.append(" update aiot\_warn\_alarm\_value set operate=?,flag=?,remark=? where id =?");

Object[] params = new Object[]{

model.getOperate(),model.getFlag(),model.getRemark(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("操作失败");

}

return result;

}

/\*

\* 读取预警告警中的未操作的

\*/

public Integer getNum4WarnValue(StationModel model){

Integer num = 0;

StringBuffer sql = new StringBuffer();

sql.append(" select count(1) from aiot\_warn\_alarm\_value where operate = 0");

num = this.queryForInt(sql.toString(), null);

return num;

}

/\*

\* 预警告警信息中批量操作

\*/

public Result updateAllOperaValue(WarnValueModel 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("批量修改成功");

//开始执行操作

StringBuffer sql = new StringBuffer();

sql.append(" update aiot\_warn\_alarm\_value set");

sql.append(" operate=").append(model.getOperate()).append(",");

sql.append(" flag='").append(model.getFlag()).append("',");

sql.append(" remark='").append(model.getRemark()).append("'");

sql.append(" where id in (").append(model.getIds()).append(")");

int res = 0;

try {

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

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("批量修改失败");

}

return result;

}

}

package com.sdocean.warn.model;

import java.util.List;

import javax.persistence.Entity;

@Entity

public class Warn4FirstModel {

private String warnType;

private List<WarnModel> warnModels;

public String getWarnType() {

return warnType;

}

public void setWarnType(String warnType) {

this.warnType = warnType;

}

public List<WarnModel> getWarnModels() {

return warnModels;

}

public void setWarnModels(List<WarnModel> warnModels) {

this.warnModels = warnModels;

}

}

package com.sdocean.warn.model;

public class WarnModel {

private int id;

private int wpId;

private String wpName;

private String indicatorCode;

private String indicatorName;

private Double warnLower; //预警低值

private Double warnUpper; //预警高值

private Double alarmLower; //告警低值

private Double alarmUpper; //告警高值

private String ids;

private String collecttime;

private String title;

private String data;

private String unitname;

private String upper;

private String lower;

public String getCollecttime() {

return collecttime;

}

public void setCollecttime(String collecttime) {

this.collecttime = collecttime;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getData() {

return data;

}

public void setData(String data) {

this.data = data;

}

public String getUnitname() {

return unitname;

}

public void setUnitname(String unitname) {

this.unitname = unitname;

}

public String getUpper() {

return upper;

}

public void setUpper(String upper) {

this.upper = upper;

}

public String getLower() {

return lower;

}

public void setLower(String lower) {

this.lower = lower;

}

public String getIds() {

return ids;

}

public void setIds(String ids) {

this.ids = ids;

}

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 getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

public String getIndicatorName() {

return indicatorName;

}

public void setIndicatorName(String indicatorName) {

this.indicatorName = indicatorName;

}

public Double getWarnLower() {

return warnLower;

}

public void setWarnLower(Double warnLower) {

this.warnLower = warnLower;

}

public Double getWarnUpper() {

return warnUpper;

}

public void setWarnUpper(Double warnUpper) {

this.warnUpper = warnUpper;

}

public Double getAlarmLower() {

return alarmLower;

}

public void setAlarmLower(Double alarmLower) {

this.alarmLower = alarmLower;

}

public Double getAlarmUpper() {

return alarmUpper;

}

public void setAlarmUpper(Double alarmUpper) {

this.alarmUpper = alarmUpper;

}

}

package com.sdocean.warn.model;

import java.util.List;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.station.model.StationModel;

public class WarnValueModel {

private int id;

private int wpId;

private String wpName;

private int deviceId;

private String deviceName;

private String indicatorCode;

private String indicatorName;

private String beginDate;

private String endDate;

private int type;

private String typeName;

private Double value;

private String collect\_time;

private int operate; //0:未操作 1:已操作

private String operateName;

private String flag; //备注

private String remark;

private Double minConfig; //保留配置中的最低下限

private Double maxConfig; //保留配置中的最高上限

private String unitName;

private String ids;

//

private List<StationModel> stationList;

private List<PublicModel> typeList;

public String getIds() {

return ids;

}

public void setIds(String ids) {

this.ids = ids;

}

public List<StationModel> getStationList() {

return stationList;

}

public void setStationList(List<StationModel> stationList) {

this.stationList = stationList;

}

public List<PublicModel> getTypeList() {

return typeList;

}

public void setTypeList(List<PublicModel> typeList) {

this.typeList = typeList;

}

public String getOperateName() {

return operateName;

}

public void setOperateName(String operateName) {

this.operateName = operateName;

}

public String getUnitName() {

return unitName;

}

public void setUnitName(String unitName) {

this.unitName = unitName;

}

public String getWpName() {

return wpName;

}

public void setWpName(String wpName) {

this.wpName = wpName;

}

public String getDeviceName() {

return deviceName;

}

public void setDeviceName(String deviceName) {

this.deviceName = deviceName;

}

public String getIndicatorName() {

return indicatorName;

}

public void setIndicatorName(String indicatorName) {

this.indicatorName = indicatorName;

}

public String getTypeName() {

return typeName;

}

public void setTypeName(String typeName) {

this.typeName = typeName;

}

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 int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

public String getBeginDate() {

return beginDate;

}

public void setBeginDate(String beginDate) {

this.beginDate = beginDate;

}

public String getEndDate() {

return endDate;

}

public void setEndDate(String endDate) {

this.endDate = endDate;

}

public int getType() {

return type;

}

public void setType(int type) {

this.type = type;

}

public Double getValue() {

return value;

}

public void setValue(Double value) {

this.value = value;

}

public String getCollect\_time() {

return collect\_time;

}

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

this.collect\_time = collect\_time;

}

public int getOperate() {

return operate;

}

public void setOperate(int operate) {

this.operate = operate;

}

public String getFlag() {

return flag;

}

public void setFlag(String flag) {

this.flag = flag;

}

public String getRemark() {

return remark;

}

public void setRemark(String remark) {

this.remark = remark;

}

public Double getMinConfig() {

return minConfig;

}

public void setMinConfig(Double minConfig) {

this.minConfig = minConfig;

}

public Double getMaxConfig() {

return maxConfig;

}

public void setMaxConfig(Double maxConfig) {

this.maxConfig = maxConfig;

}

}

package com.sdocean.warn.service;

import java.util.ArrayList;

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.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

import com.sdocean.warn.dao.WarnDao;

import com.sdocean.warn.model.Warn4FirstModel;

import com.sdocean.warn.model.WarnModel;

import com.sdocean.warn.model.WarnValueModel;

@Service

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

public class WarnService {

@Resource

WarnDao warnDao;

/\*

\* 为首页读取预警信息

\*/

public Warn4FirstModel getWarn4First(Long warnType,StationModel watchPoint,String beginDate){

return warnDao.getWarn4First(warnType, watchPoint, beginDate);

}

/\*

\* 为首页读取多个站点的预警信息

\*/

public Warn4FirstModel getWarn4First4All(Long warnType,List<StationModel> watchPoints,String beginDate){

return warnDao.getWarn4First4All(warnType, watchPoints, beginDate);

}

/\*

\* 根据条件查询出预警告警配置列表

\* 根据当前登录人员查询权限内的站点

\*/

public List<WarnModel> getWarnList(WarnModel model,SysUser user){

return warnDao.getWarnList(model, user);

}

public List<UiColumn> getCols4List() {

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

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

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

UiColumn col3 = new UiColumn("站点名称", "wpName", true, "\*");

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

UiColumn col5 = new UiColumn("参数名称", "indicatorName", true, "\*");

UiColumn col6 = new UiColumn("预警低值", "warnLower", true, "\*");

UiColumn col7 = new UiColumn("预警高值", "warnUpper", true, "\*");

UiColumn col8 = new UiColumn("告警低值", "alarmLower", true, "\*");

UiColumn col9 = new UiColumn("告警高值", "alarmUpper", 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);

cols.add(col9);

return cols;

}

/\*

\* 修改预警告警配置

\*/

public Result saveWarnChange(WarnModel model){

return warnDao.saveWarnChange(model);

}

/\*

\* 保存新增预警告警配置

\*/

public Result saveNewWarn(WarnModel model){

return warnDao.saveNewWarn(model);

}

/\*

\* 删除选中的预警告警配置

\*/

public Result deleWarnSetting(WarnModel model){

return warnDao.deleWarnSetting(model);

}

/\*

\* 为预警告警信息展示添加表头

\*/

public List<UiColumn> getCols4WarnValue() {

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

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

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

UiColumn col3 = new UiColumn("站点名称", "wpName", true, "\*");

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

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

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

UiColumn col7 = new UiColumn("参数名称", "indicatorName", true, "\*");

UiColumn col8 = new UiColumn("type", "type", false, "\*");

UiColumn col9 = new UiColumn("类型", "typeName", true, "\*");

UiColumn col10 = new UiColumn("数值", "value", true, "\*");

UiColumn col11 = new UiColumn("采集时间", "collect\_time", true, "\*");

UiColumn col12 = new UiColumn("operate", "operate", false, "\*");

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

UiColumn col14 = new UiColumn("备注", "flag", true, "\*");

UiColumn col15 = new UiColumn("说明", "remark", true, "\*");

UiColumn col16 = new UiColumn("配置下限", "minConfig", true, "\*");

UiColumn col17 = new UiColumn("配置上限 ", "maxConfig", true, "\*");

UiColumn col18 = new UiColumn("单位 ", "unitName", 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);

cols.add(col9);

cols.add(col10);

cols.add(col18);

cols.add(col11);

cols.add(col12);

cols.add(col13);

cols.add(col14);

cols.add(col15);

cols.add(col16);

cols.add(col17);

return cols;

}

/\*

\* 为预警告警信息展示添加结果集

\*/

public List<WarnValueModel> getWarnValueRows(WarnValueModel model){

return warnDao.getWarnValueRows(model);

}

/\*

\* 为预警告警信息添加操作以及备注

\*/

public Result saveOperaValue(WarnValueModel model){

return warnDao.saveOperaValue(model);

}

/\*

\* 读取预警告警中的未操作的

\*/

public Integer getNum4WarnValue(StationModel model){

return warnDao.getNum4WarnValue(model);

}

/\*

\* 预警告警信息中批量操作

\*/

public Result updateAllOperaValue(WarnValueModel model){

return warnDao.updateAllOperaValue(model);

}

}

package com.sdocean.tool.action;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import javax.annotation.Resource;

import javax.servlet.ServletOutputStream;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellStyle;

import org.apache.poi.ss.usermodel.IndexedColors;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.xssf.streaming.SXSSFWorkbook;

import org.springframework.stereotype.Controller;

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

import com.sdocean.log.service.OperationLogService;

import com.sdocean.station.service.StationService;

@Controller

public class ExcelExportAction {

@Resource

StationService stationService;

@Resource

OperationLogService logService;

@RequestMapping("excelExportinfo.do")

public void excelExportinfo(HttpServletRequest request,

HttpServletResponse response)throws Exception{

request.setCharacterEncoding("utf-8");//注意编码

String fileName = request.getParameter("fileName");

String content = request.getParameter("content");

String[] array = content.split("[\\n]");

SXSSFWorkbook wb = new SXSSFWorkbook(100); // keep 100 rows in memory,

Sheet sh = wb.createSheet();

int row = 0;

int h = 0;

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

String rowString = array[i];

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

Row header = sh.createRow(row);

row++;

String[] cells = rowString.split(",");

for(int j=0;j<cells.length;j++){

Cell cell = header.createCell(j);

if(cells[j].contains("\"")){

cells[j] = cells[j].replaceAll("\"", "");

cell.setCellValue(cells[j]);

}else{

if(isDouble(cells[j].trim())||isNum(cells[j].trim())){

CellStyle styleFormat = wb.createCellStyle();

styleFormat.setFillForegroundColor(IndexedColors.TAN.getIndex());// #F2DEDE

styleFormat.setFillPattern(CellStyle.SOLID\_FOREGROUND);

cell.setCellValue(Double.parseDouble(cells[j]));

}else{

cell.setCellValue(cells[j]);

}

}

}

}

}

response.reset();

response.setContentType("application/x-msdownload");

String pName = fileName;

try {

response.setHeader("Content-Disposition", "attachment; filename="

+ new String(pName.getBytes("gb2312"), "ISO-8859-1")

+ ".xlsx");

} catch (UnsupportedEncodingException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

ServletOutputStream outStream = null;

try {

outStream = response.getOutputStream();

wb.write(outStream);

wb.close();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

outStream.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public static boolean isDouble(String value) {

try {

Double.parseDouble(value);

if (value.contains("."))

return true;

return false;

} catch (NumberFormatException e) {

return false;

}

}

public static boolean isNum(String value){

boolean result=value.matches("[0-9]+");

return result;

}

}

package com.sdocean.tool.action;

import java.awt.image.BufferedImage;

import java.io.File;

import java.util.Enumeration;

import java.util.HashMap;

import java.util.Map;

import javax.annotation.Resource;

import javax.imageio.ImageIO;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

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

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

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

import com.baidu.ueditor.ActionEnter;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.tool.model.UeditorModel;

import com.sdocean.tool.service.UeditorService;

@Controller

public class UeditorAction {

@Resource

UeditorService ueditorService;

@Resource

ConfigInfo info;

@RequestMapping("ueditorControll.do")

public void ueditorControll(HttpServletRequest request,

HttpServletResponse response)throws Exception{

Map map = new HashMap();

Enumeration paramNames = request.getParameterNames();

while (paramNames.hasMoreElements()) {

String paramName = (String) paramNames.nextElement();

String[] paramValues = request.getParameterValues(paramName);

if (paramValues.length == 1) {

String paramValue = paramValues[0];

if (paramValue.length() != 0) {

}

}

}

request.setCharacterEncoding( "utf-8" );

response.setHeader("Content-Type" , "text/html");

String rootPath = getClass().getResource("/").getFile().toString();

response.getWriter().write( new ActionEnter( request, rootPath ).exec() );

}

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

@ResponseBody

public String imageUpload(HttpServletRequest request,

HttpServletResponse response)throws Exception{

UeditorModel edit =ueditorService.imageUpload(request, response,info);

return JsonUtil.toJson(edit);

}

@RequestMapping("imageShow.do")

public void imageShow(HttpServletRequest request,

HttpServletResponse response)throws Exception{

String fileName = (String) request.getParameter("fileName");

String path = info.getImagePath()+"/"+fileName;

File file = new File(path);

BufferedImage image = ImageIO.read(file);

ImageIO.write(image, "jpg", response.getOutputStream());

}

}

package com.sdocean.station.dao;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.PlatformModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class PlatformDao extends OracleEngine {

/\*

\* 获得符合查询条件的平台的列表

\*/

public List<PlatformModel> getPlatformList(PlatformModel model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select id,title,pic,detail,ordercode,isactive,case isactive when 1 then '启用' else '禁用' end as isactivename");

sql.append(" from g\_platform where 1=1");

//增加查询条件

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

sql.append(" and title like '%").append(model.getTitle()).append("%'");

}

if(model!=null&&model.getIsactive()<2){

sql.append(" and isactive = ").append(model.getIsactive());

}

//增加排序

sql.append(" order by ordercode");

System.out.println(sql.toString());

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

return list;

}

/\*

\* 新增平台

\*/

public Result addPlatform(PlatformModel 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");

StringBuffer sql = new StringBuffer("");

sql.append(" insert into g\_platform(title,pic,detail,ordercode,isactive)");

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

Object[] params = new Object[]{

model.getTitle(),model.getPic(),model.getDetail(),model.getOrderCode(),model.getIsactive()

};

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("新增失败");

}

return result;

}

/\*

\* 修改平台

\*/

public Result savePlatform(PlatformModel 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 sql = new StringBuffer("");

sql.append(" update g\_platform set title=?,pic=?,detail=?,ordercode=?,isactive=?");

sql.append(" where id=?");

Object[] params = new Object[]{

model.getTitle(),model.getPic(),model.getDetail(),model.getOrderCode(),model.getIsactive(),model.getId()

};

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("修改失败");

}

return result;

}

/\*

\* 删除平台

\*/

public Result delePlatform(PlatformModel 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(" update g\_platform set isactive=0 ");

sql.append(" where id=?");

Object[] params = new Object[]{

model.getId()

};

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("删除失败");

}

return result;

}

}

package com.sdocean.station.dao;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationDeviceComm;

import com.sdocean.station.model.StationInfo;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationCommDao extends OracleEngine {

/\*

\* 查询符合条件的站点配置

\*/

public List<StationDeviceComm> getStationCommList(StationDeviceComm model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.ip,a.port,a.cronExp,a.stationid,b.title as stationname,a.betweenTime,");

sql.append(" a.lastDataTime,a.functionname,a.writeTimeOut,a.readTimeOut,c.lastMetaDate,");

sql.append(" m.ProtocalVersion,m.TemplateFile,a.deviceid,d.name as deviceName");

sql.append(" from aiot\_station\_device\_comm a left join");

sql.append(" (select wpid,max(endtime) as lastMetaDate from aiot\_meta\_table where type = 1 and isactive = 1 group by wpid) c");

sql.append(" on a.stationid = c.wpid");

sql.append(" left join aiot\_metadata\_converter\_template m on a.stationId = m.stationId,");

sql.append(" aiot\_watch\_point b ,device\_catalog d");

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

sql.append(" and a.deviceid = d.id");

/\*sql.append(" select a.id,a.ip,a.port,a.cronExp,a.stationId,b.title as stationName,");

sql.append(" a.lastDataTime,a.FunctionName,a.writeTimeOut,a.readTimeOut,a.mainCount,c.lastMetaDate,");

sql.append(" m.ProtocalVersion,m.TemplateFile,group\_concat(n.name,'|') as DataItem,a.ifsms,case a.ifsms when 0 then '不发送' else '发送' end as ifsmsname");

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

sql.append(" (select wpid,max(endtime) as lastMetaDate from aiot\_meta\_table where type = 1 and isactive = 1 group by wpid) c ");

sql.append(" on a.stationId = c.wpid");

sql.append(" left join aiot\_metadata\_converter\_template m on a.stationId = m.stationId");

sql.append(" left join (select j.stationid,k.name from aiot\_station\_device\_comm j,device\_catalog k where j.deviceid = k.id) n on a.stationid = n.stationid");

sql.append(" ,aiot\_watch\_point b");

sql.append(" where a.stationId = b.id and b.isactive = 1");\*/

//添加查询条件

if(model!=null&&model.getStationId()>10000){

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

}

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

return list;

}

/\*

\* 保存站点配置修改

\*/

public Result saveStationCommChange(StationDeviceComm 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");

//判断是否违背站点唯一原则

StringBuffer checkSql = new StringBuffer("");

checkSql.append("select count(1) from aiot\_station\_device\_comm where stationid=").append(model.getStationId()).append(" and deviceid=").append(model.getDeviceId()).append(" and id<>").append(model.getId());

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("检查站点唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("当前站点已有配置");

return result;

}

//保存基本配置

StringBuffer sql = new StringBuffer("");

sql.append(" update aiot\_station\_device\_comm set stationid=?,deviceid=?,betweentime=?,lastdatatime=?,ip=?,port=?,cronExp=?,functionName=?,writetimeout=?,readtimeOut=?");

sql.append(" where id=?");

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getBetweenTime(),

model.getLastDataTime(),model.getIp(),model.getPort(),model.getCronExp(),

model.getFunctionName(),model.getWriteTimeOut(),model.getReadTimeOut(),

model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("保存基本配置时失败");

return result;

}

return result;

}

/\*

\* 保存新站点配置

\*/

public Result saveNewStationComm(StationDeviceComm 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("保存成功");

//判断是否违背站点唯一性原则

StringBuffer checkSql = new StringBuffer("");

checkSql.append("select count(1) from aiot\_station\_device\_comm where stationid=").append(model.getStationId()).append(" and deviceid=").append(model.getDeviceId());

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("检查站点唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("当前站点已有配置");

return result;

}

StringBuffer sql = new StringBuffer("");

sql.append(" insert into aiot\_station\_device\_comm(stationid,deviceid,betweentime,lastdatatime,ip,port,cronExp,functionName,writetimeout,readtimeOut)");

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

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getBetweenTime(),

model.getLastDataTime(),model.getIp(),model.getPort(),model.getCronExp(),

model.getFunctionName(),model.getWriteTimeOut(),model.getReadTimeOut()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存基本信息时失败");

}

return result;

}

/\*

\* 得到该站点下的设备列表,并选中已经查询的设备

\*/

public List<SelectTree> getDevciesByStation4Tree(StationModel model){

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

//得到第一层数据

SelectTree first = new SelectTree();

first.setName("站点列表");

first.setIsExpanded(true);

first.setIsActive(true);

//读取该站点已经查询的数据

StringBuffer msql = new StringBuffer("");

msql.append(" select deviceid as id from aiot\_station\_device\_comm where stationid = ").append(model.getId());

List<DeviceModel> devices = this.queryObjectList(msql.toString(), DeviceModel.class);

String item = "0";

for(DeviceModel device:devices){

item = item + ',' + device.getId();

}

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id as id ,a.name,'true' as isExpanded,");

sql.append(" case when a.id in (").append(item).append(") then 'true' else 'false' end as selected");

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

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

sql.append(" and b.aiot\_watch\_point\_id =").append(model.getId());

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

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

first.setChildren(childlist);

list.add(first);

return list;

}

/\*

\* 根据站点读取当前站点的信息

\*/

public StationInfo getStationInfoByWpId(StationInfo model){

StationInfo result = new StationInfo();

StringBuffer sql = new StringBuffer("");

sql.append("select id,stationid,infomation from aiot\_station\_info where stationid = ").append(model.getStationId());

result = this.queryObject(sql.toString(), StationInfo.class);

if(result==null){

result = new StationInfo();

result.setStationId(model.getStationId());

}

return result;

}

/\*

\* 保存站点的信息展示

\*/

public Result saveStationInfomation(StationInfo model){

//初始化返回结果

Result result = new Result();

result.setDotype(result.UPDATE);

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

result.setMessage("保存成功");

result.setResult(result.SUCCESS);

StringBuffer sql = new StringBuffer();

sql.append("insert into aiot\_station\_info(stationid,infomation) values(?,?) on DUPLICATE KEY UPDATE infomation=values(infomation)");

Object[] params = new Object[]{

model.getStationId(),model.getInfomation()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setMessage("保存失败");

result.setResult(result.FAILED);

}

return result;

}

/\*

\* 删除站点信息

\*/

public Result deleStationComm(StationDeviceComm model){

//初始化返回参数

Result result = new Result();

result.setDotype(Result.DELETE);

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

result.setResult(Result.SUCCESS);

result.setMessage("删除成功");

//删除操作

StringBuffer sql = new StringBuffer("");

sql.append("delete from aiot\_station\_device\_comm where id = ").append(model.getId());

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("删除失败");

}

return result;

}

}

package com.sdocean.station.dao;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationDeviceComm;

import com.sdocean.station.model.StationInfo;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationCommDao extends OracleEngine {

/\*

\* 查询符合条件的站点配置

\*/

public List<StationDeviceComm> getStationCommList(StationDeviceComm model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.ip,a.port,a.cronExp,a.stationid,b.title as stationname,a.betweenTime,");

sql.append(" a.lastDataTime,a.functionname,a.writeTimeOut,a.readTimeOut,c.lastMetaDate,");

sql.append(" m.ProtocalVersion,m.TemplateFile,a.deviceid,d.name as deviceName");

sql.append(" from aiot\_station\_device\_comm a left join");

sql.append(" (select wpid,max(endtime) as lastMetaDate from aiot\_meta\_table where type = 1 and isactive = 1 group by wpid) c");

sql.append(" on a.stationid = c.wpid");

sql.append(" left join aiot\_metadata\_converter\_template m on a.stationId = m.stationId,");

sql.append(" aiot\_watch\_point b ,device\_catalog d");

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

sql.append(" and a.deviceid = d.id");

/\*sql.append(" select a.id,a.ip,a.port,a.cronExp,a.stationId,b.title as stationName,");

sql.append(" a.lastDataTime,a.FunctionName,a.writeTimeOut,a.readTimeOut,a.mainCount,c.lastMetaDate,");

sql.append(" m.ProtocalVersion,m.TemplateFile,group\_concat(n.name,'|') as DataItem,a.ifsms,case a.ifsms when 0 then '不发送' else '发送' end as ifsmsname");

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

sql.append(" (select wpid,max(endtime) as lastMetaDate from aiot\_meta\_table where type = 1 and isactive = 1 group by wpid) c ");

sql.append(" on a.stationId = c.wpid");

sql.append(" left join aiot\_metadata\_converter\_template m on a.stationId = m.stationId");

sql.append(" left join (select j.stationid,k.name from aiot\_station\_device\_comm j,device\_catalog k where j.deviceid = k.id) n on a.stationid = n.stationid");

sql.append(" ,aiot\_watch\_point b");

sql.append(" where a.stationId = b.id and b.isactive = 1");\*/

//添加查询条件

if(model!=null&&model.getStationId()>10000){

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

}

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

return list;

}

/\*

\* 保存站点配置修改

\*/

public Result saveStationCommChange(StationDeviceComm 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");

//判断是否违背站点唯一原则

StringBuffer checkSql = new StringBuffer("");

checkSql.append("select count(1) from aiot\_station\_device\_comm where stationid=").append(model.getStationId()).append(" and deviceid=").append(model.getDeviceId()).append(" and id<>").append(model.getId());

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("检查站点唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("当前站点已有配置");

return result;

}

//保存基本配置

StringBuffer sql = new StringBuffer("");

sql.append(" update aiot\_station\_device\_comm set stationid=?,deviceid=?,betweentime=?,lastdatatime=?,ip=?,port=?,cronExp=?,functionName=?,writetimeout=?,readtimeOut=?");

sql.append(" where id=?");

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getBetweenTime(),

model.getLastDataTime(),model.getIp(),model.getPort(),model.getCronExp(),

model.getFunctionName(),model.getWriteTimeOut(),model.getReadTimeOut(),

model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("保存基本配置时失败");

return result;

}

return result;

}

/\*

\* 保存新站点配置

\*/

public Result saveNewStationComm(StationDeviceComm 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("保存成功");

//判断是否违背站点唯一性原则

StringBuffer checkSql = new StringBuffer("");

checkSql.append("select count(1) from aiot\_station\_device\_comm where stationid=").append(model.getStationId()).append(" and deviceid=").append(model.getDeviceId());

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("检查站点唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("当前站点已有配置");

return result;

}

StringBuffer sql = new StringBuffer("");

sql.append(" insert into aiot\_station\_device\_comm(stationid,deviceid,betweentime,lastdatatime,ip,port,cronExp,functionName,writetimeout,readtimeOut)");

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

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getBetweenTime(),

model.getLastDataTime(),model.getIp(),model.getPort(),model.getCronExp(),

model.getFunctionName(),model.getWriteTimeOut(),model.getReadTimeOut()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存基本信息时失败");

}

return result;

}

/\*

\* 得到该站点下的设备列表,并选中已经查询的设备

\*/

public List<SelectTree> getDevciesByStation4Tree(StationModel model){

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

//得到第一层数据

SelectTree first = new SelectTree();

first.setName("站点列表");

first.setIsExpanded(true);

first.setIsActive(true);

//读取该站点已经查询的数据

StringBuffer msql = new StringBuffer("");

msql.append(" select deviceid as id from aiot\_station\_device\_comm where stationid = ").append(model.getId());

List<DeviceModel> devices = this.queryObjectList(msql.toString(), DeviceModel.class);

String item = "0";

for(DeviceModel device:devices){

item = item + ',' + device.getId();

}

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id as id ,a.name,'true' as isExpanded,");

sql.append(" case when a.id in (").append(item).append(") then 'true' else 'false' end as selected");

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

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

sql.append(" and b.aiot\_watch\_point\_id =").append(model.getId());

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

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

first.setChildren(childlist);

list.add(first);

return list;

}

/\*

\* 根据站点读取当前站点的信息

\*/

public StationInfo getStationInfoByWpId(StationInfo model){

StationInfo result = new StationInfo();

StringBuffer sql = new StringBuffer("");

sql.append("select id,stationid,infomation from aiot\_station\_info where stationid = ").append(model.getStationId());

result = this.queryObject(sql.toString(), StationInfo.class);

if(result==null){

result = new StationInfo();

result.setStationId(model.getStationId());

}

return result;

}

/\*

\* 保存站点的信息展示

\*/

public Result saveStationInfomation(StationInfo model){

//初始化返回结果

Result result = new Result();

result.setDotype(result.UPDATE);

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

result.setMessage("保存成功");

result.setResult(result.SUCCESS);

StringBuffer sql = new StringBuffer();

sql.append("insert into aiot\_station\_info(stationid,infomation) values(?,?) on DUPLICATE KEY UPDATE infomation=values(infomation)");

Object[] params = new Object[]{

model.getStationId(),model.getInfomation()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setMessage("保存失败");

result.setResult(result.FAILED);

}

return result;

}

/\*

\* 删除站点信息

\*/

public Result deleStationComm(StationDeviceComm model){

//初始化返回参数

Result result = new Result();

result.setDotype(Result.DELETE);

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

result.setResult(Result.SUCCESS);

result.setMessage("删除成功");

//删除操作

StringBuffer sql = new StringBuffer("");

sql.append("delete from aiot\_station\_device\_comm where id = ").append(model.getId());

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("删除失败");

}

return result;

}

}

package com.sdocean.station.dao;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationDeviceComm;

import com.sdocean.station.model.StationDeviceModel;

import com.sdocean.station.model.StationInfo;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationDeviceDao extends OracleEngine {

/\*

\* 读取站点设备配置列表

\*/

public List<StationDeviceModel> getStationDeviceList(StationDeviceModel model,List<StationModel> stations){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,b.id as stationId,b.title as stationName,c.id as deviceId,c.name as deviceName,");

sql.append(" a.pointNum,a.createtime,a.ordercode");

sql.append(" from map\_awp\_device\_catalog a,aiot\_watch\_point b,device\_catalog c");

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

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

//添加查询条件

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

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

}

//增加排序

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

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

return list;

}

/\*

\* 添加站点设备配置

\*/

public Result saveNewStationDevice(StationDeviceModel 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");

//判断是否违法唯一性原则

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId());

csql.append(" and device\_catalog\_id = ").append(model.getDeviceId());

int cres = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("唯一性检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//添加数据

StringBuffer sql = new StringBuffer("");

sql.append(" insert into map\_awp\_device\_catalog(aiot\_watch\_point\_id,device\_catalog\_id,pointnum,ordercode,createtime)");

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

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getPointNum(),model.getOrderCode(),model.getCreateTime()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("新增失败");

return result;

}

return result;

}

/\*

\* 修改站点设备配置

\*/

public Result saveChangeStationDevice(StationDeviceModel 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");

//判断是否违法唯一性原则

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId());

csql.append(" and device\_catalog\_id = ").append(model.getDeviceId());

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

int cres = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("唯一性检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//添加数据

StringBuffer sql = new StringBuffer("");

sql.append(" update map\_awp\_device\_catalog set aiot\_watch\_point\_id=?,device\_catalog\_id=?,pointnum=?,ordercode=?,createtime=? where id=?");

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getPointNum(),model.getOrderCode(),model.getCreateTime(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("修改失败");

return result;

}

return result;

}

/\*

\* 删除站点设备配置

\*/

public Result deleteStationDevice(StationDeviceModel 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 map\_awp\_device\_catalog 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Id以及设备Id获得站点设备配置

\*/

public StationDeviceModel getStationDeviceByWidDid(StationDeviceModel model){

StationDeviceModel stationDevice = new StationDeviceModel();

StringBuffer sql = new StringBuffer("");

sql.append(" select id,aiot\_watch\_point\_id as stationid,device\_catalog\_id as deviceId,pointnum,ordercode,createtime");

sql.append(" from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId()).append(" and device\_catalog\_id =").append(model.getDeviceId());

sql.append(" limit 1");

stationDevice = this.queryObject(sql.toString(), StationDeviceModel.class);

return stationDevice;

}

}

package com.sdocean.station.dao;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationDeviceComm;

import com.sdocean.station.model.StationDeviceModel;

import com.sdocean.station.model.StationInfo;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationDeviceDao extends OracleEngine {

/\*

\* 读取站点设备配置列表

\*/

public List<StationDeviceModel> getStationDeviceList(StationDeviceModel model,List<StationModel> stations){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,b.id as stationId,b.title as stationName,c.id as deviceId,c.name as deviceName,");

sql.append(" a.pointNum,a.createtime,a.ordercode");

sql.append(" from map\_awp\_device\_catalog a,aiot\_watch\_point b,device\_catalog c");

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

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

//添加查询条件

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

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

}

//增加排序

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

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

return list;

}

/\*

\* 添加站点设备配置

\*/

public Result saveNewStationDevice(StationDeviceModel 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");

//判断是否违法唯一性原则

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId());

csql.append(" and device\_catalog\_id = ").append(model.getDeviceId());

int cres = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("唯一性检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//添加数据

StringBuffer sql = new StringBuffer("");

sql.append(" insert into map\_awp\_device\_catalog(aiot\_watch\_point\_id,device\_catalog\_id,pointnum,ordercode,createtime)");

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

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getPointNum(),model.getOrderCode(),model.getCreateTime()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("新增失败");

return result;

}

return result;

}

/\*

\* 修改站点设备配置

\*/

public Result saveChangeStationDevice(StationDeviceModel 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");

//判断是否违法唯一性原则

StringBuffer csql = new StringBuffer("");

csql.append(" select count(1) from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId());

csql.append(" and device\_catalog\_id = ").append(model.getDeviceId());

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

int cres = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("唯一性检查时失败");

return result;

}

if(cres>0){

result.setResult(Result.FAILED);

result.setMessage("违法唯一性原则");

return result;

}

//添加数据

StringBuffer sql = new StringBuffer("");

sql.append(" update map\_awp\_device\_catalog set aiot\_watch\_point\_id=?,device\_catalog\_id=?,pointnum=?,ordercode=?,createtime=? where id=?");

Object[] params = new Object[]{

model.getStationId(),model.getDeviceId(),model.getPointNum(),model.getOrderCode(),model.getCreateTime(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("修改失败");

return result;

}

return result;

}

/\*

\* 删除站点设备配置

\*/

public Result deleteStationDevice(StationDeviceModel 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 map\_awp\_device\_catalog 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Id以及设备Id获得站点设备配置

\*/

public StationDeviceModel getStationDeviceByWidDid(StationDeviceModel model){

StationDeviceModel stationDevice = new StationDeviceModel();

StringBuffer sql = new StringBuffer("");

sql.append(" select id,aiot\_watch\_point\_id as stationid,device\_catalog\_id as deviceId,pointnum,ordercode,createtime");

sql.append(" from map\_awp\_device\_catalog where aiot\_watch\_point\_id =").append(model.getStationId()).append(" and device\_catalog\_id =").append(model.getDeviceId());

sql.append(" limit 1");

stationDevice = this.queryObject(sql.toString(), StationDeviceModel.class);

return stationDevice;

}

}

package com.sdocean.station.dao;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.springframework.security.core.userdetails.User;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationDeviceIndicator;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationDeviceIndicatorDao extends OracleEngine {

/\*

\* 根据站点,以ztree的形式展示有显示权限的设备以及关联的参数

\*/

public List<ZTreeModel> getDeiviceIndicatorsByStation(int stationId){

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

//添加首行记录

ZTreeModel first = new ZTreeModel();

first.setId("0");

first.setName("设备/参数");

first.setOpen(true);

alllist.add(first);

for(ZTreeModel all:alllist){

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

//获得当前站点下的设备列表

StringBuffer deviceSql = new StringBuffer("");

deviceSql.append(" select a.id,0 as pid,a.name as name,");

deviceSql.append(" case when m.deviceid is null then 'false' else 'true' end as checked");

deviceSql.append(" from device\_catalog a");

deviceSql.append(" left join (select distinct deviceid from aiot\_station\_device\_indicator ");

deviceSql.append(" where stationid =").append(stationId).append(" ) m ");

deviceSql.append(" on a.id = m.deviceid,");

deviceSql.append(" map\_awp\_device\_catalog b");

deviceSql.append(" where b.aiot\_watch\_point\_id = ").append(stationId);

deviceSql.append(" and a.id = b.device\_catalog\_id");

deviceSql.append(" order by a.ordercode");

list = this.queryObjectList(deviceSql.toString(), ZTreeModel.class);

//遍历设备列表,获得设备下的所属的参数列表

for(ZTreeModel device:list){

StringBuffer indicatorSql = new StringBuffer("");

indicatorSql.append(" select concat(").append(device.getId()).append(",'#',a.id) as id,").append(device.getId()).append(" as pid,a.title as name,");

indicatorSql.append(" case when m.indicatorid is null then 'false' else 'true' end as checked");

indicatorSql.append(" from dm\_indicator a left join (select distinct indicatorid from aiot\_station\_device\_indicator ");

indicatorSql.append(" where stationid =").append(stationId).append(" and deviceid =").append(device.getId()).append(" ) m");

indicatorSql.append(" on a.id = m.indicatorid,");

indicatorSql.append(" device\_catalog\_indicator b");

indicatorSql.append(" where a.isactive = 1");

indicatorSql.append(" and a.id = b.indicatorid");

indicatorSql.append(" and b.catalogid = ").append(device.getId());

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

children = this.queryObjectList(indicatorSql.toString(), ZTreeModel.class);

device.setChildren(children);

}

all.setChildren(list);

}

return alllist;

}

/\*

\* 根据模糊查询条件,查询出该用户角色下的所有站点列表

\*/

public List<StationModel> getStation4DeviceIndicator(int userid,String code){

List<StationModel> stations = new ArrayList<StationModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.title,b.name as companyName");

sql.append(" from view\_role\_user\_station a, g\_company b");

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

//增加条件

sql.append(" and a.userid = ").append(userid);

//增加模糊查询条件

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

sql.append(" and (a.title like '%").append(code).append("%'");

sql.append(" or b.name like '%").append(code).append("%' )");

}

//增加排序

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

stations = this.queryObjectList(sql.toString(), StationModel.class);

return stations;

}

/\*

\* 保存站点设备参数

\*/

public Result saveStationDeviceIndicator(StationDeviceIndicator 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("保存成功");

//删除该站点的原有数据

StringBuffer deleSql = new StringBuffer("");

deleSql.append("delete from aiot\_station\_device\_indicator where stationid =").append(model.getStationId());

int deleRes = 0;

try {

deleRes = this.update(deleSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("删除原有数据时失败");

return result;

}

//开始保存数据

//定义总体sql

StringBuffer sql = new StringBuffer("");

sql.append("insert into aiot\_station\_device\_indicator(stationid,deviceid,indicatorid) values");

StringBuffer valueSql = new StringBuffer("");

valueSql.append("(0,0,0)");

//获得参数集合

String indicatorIds = model.getIndicatorIds();

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

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

String code= codes[i];

String deviceId = code.substring(0, code.indexOf("#"));

String indicatorId = code.substring(code.indexOf("#")+1, code.length());

valueSql.append(",(").append(model.getStationId()).append(",").append(deviceId).append(",").append(indicatorId).append(")");

}

sql.append(valueSql).append(" on duplicate key update indicatorid=values(indicatorid)");

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存失败");

}

return result;

}

}

package com.sdocean.station.dao;

import org.springframework.stereotype.Component;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.station.model.DevicePage;

import com.sdocean.station.model.StationDevicePage;

@Component

public class StationMapDao extends OracleEngine {

public StationDevicePage getStationDevicePageByIds(String stationId,String pdeviceid,String cdeviceid){

StationDevicePage page = new StationDevicePage();

StringBuffer sql = new StringBuffer("");

sql.append("select stationid,pdeviceid,cdeviceid,page ");

sql.append(" from map\_station\_device\_page");

sql.append(" where stationid = ").append(stationId);

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

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

page = this.queryObject(sql.toString(), StationDevicePage.class);

return page;

}

public DevicePage getDevicePageById(String deviceid){

DevicePage page = new DevicePage();

StringBuffer sql = new StringBuffer();

sql.append(" select deviceid,page from map\_device\_page where deviceid=").append(deviceid);

page = this.queryObject(sql.toString(), DevicePage.class);

return page;

}

}

package com.sdocean.station.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.ZTreeModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationTypeModel;

import com.sdocean.users.model.SysUser;

@Component

public class StationTypeDao extends OracleEngine {

/\*

\* 为站点类型展示结果查询结果集

\*/

public List<StationTypeModel> getTypeList(StationTypeModel type){

List<StationTypeModel> res = new ArrayList<StationTypeModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.code,a.name,a.remark,a.isactive,b.value as iactiveName,icon");

sql.append(" from g\_station\_type a ");

sql.append(" left join sys\_public b on a.isactive = b.classid and b.parentcode = '0004'");

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

//添加查询条件

//code 或 name 模糊查询

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

sql.append(" and (code like '%").append(type.getCode()).append("%' or name like '%").append(type.getCode()).append("%')");

}

if(type!=null&&type.getIsactive()<2){

sql.append(" and a.isactive = ").append(type.getIsactive());

}

res = this.queryObjectList(sql.toString(), StationTypeModel.class);

return res;

}

/\*

\* 保存站点类型修改

\*/

public Result saveTypeChange(StationTypeModel 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("保存成功!");

//判断code是否重复

StringBuffer check = new StringBuffer("");

check.append(" select count(1) from g\_station\_type");

check.append(" where code = '").append(model.getCode()).append("'");

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

int count = this.queryForInt(check.toString(), null);

if(count>0){

result.setResult(result.FAILED);

result.setMessage("保存失败,类型名称重复.");

return result;

}

//将结果保存到数据库中

StringBuffer sql = new StringBuffer();

sql.append(" update g\_station\_type set code = ?,name =?,remark=?,isactive =?,icon=? where id =?");

Object[] params = new Object[]{

model.getCode(),model.getName(),model.getRemark(),model.getIsactive(),model.getIcon(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存失败,请重试");

return result;

}

return result;

}

/\*

\* 新增站点类型保存

\*/

public Result saveNewType(StationTypeModel model){

//初始化返回结果

Result result = new Result();

result.setDotype(result.ADD);

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

result.setMessage("新增成功");

result.setResult(result.SUCCESS);

//判断CODE是否重复

StringBuffer check = new StringBuffer("");

check.append(" select count(1) from g\_station\_type where code = '").append(model.getCode()).append("'");

int count = this.queryForInt(check.toString(), null);

if(count>0){

result.setResult(result.FAILED);

result.setMessage("新增失败,类型名称重复");

return result;

}

//将结果保存到数据库中

StringBuffer sql = new StringBuffer("");

sql.append(" insert into g\_station\_type(code,name,isactive,remark,icon) values(?,?,?,?,?)");

Object[] params = new Object[]{

model.getCode(),model.getName(),model.getIsactive(),model.getRemark(),model.getIcon()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("新增失败,请重试");

return result;

}

return result;

}

}

package com.sdocean.station.model;

public class DevicePage {

private int id;

private int deviceid;

private String page;

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 getPage() {

return page;

}

public void setPage(String page) {

this.page = page;

}

}

package com.sdocean.station.model;

public class PlatformModel {

private int id;

private String title;

private String pic;

private String detail;

private String orderCode;

private int isactive;

private String isactiveName;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getPic() {

return pic;

}

public void setPic(String pic) {

this.pic = pic;

}

public String getDetail() {

return detail;

}

public void setDetail(String detail) {

this.detail = detail;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

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;

}

}

package com.sdocean.station.model;

public class StationDeviceComm {

//基础部分

private int stationId;

private String stationName;

private int deviceId;

private String deviceName;

//读取数据部分

private String protocalVersion; //数据版本

private String templateFile; //模板

private String dataItem; //读取数据集合

//通信部分

private int id;

private String ip;

private String port;

private String cronExp;

private String lastDataTime;

private String functionName;

private int writeTimeOut;

private int readTimeOut;

//生成元数据表

private String lastMetaDate;

//是否发送短信

private int ifSms;

private String ifSmsName;

//数据监测间隔时间

private int betweenTime; //以分钟为单位

public int getBetweenTime() {

return betweenTime;

}

public void setBetweenTime(int betweenTime) {

this.betweenTime = betweenTime;

}

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 getIfSms() {

return ifSms;

}

public void setIfSms(int ifSms) {

this.ifSms = ifSms;

}

public String getIfSmsName() {

return ifSmsName;

}

public void setIfSmsName(String ifSmsName) {

this.ifSmsName = ifSmsName;

}

public String getProtocalVersion() {

return protocalVersion;

}

public void setProtocalVersion(String protocalVersion) {

this.protocalVersion = protocalVersion;

}

public String getTemplateFile() {

return templateFile;

}

public void setTemplateFile(String templateFile) {

this.templateFile = templateFile;

}

public String getDataItem() {

return dataItem;

}

public void setDataItem(String dataItem) {

this.dataItem = dataItem;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCronExp() {

return cronExp;

}

public void setCronExp(String cronExp) {

this.cronExp = cronExp;

}

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 getIp() {

return ip;

}

public void setIp(String ip) {

this.ip = ip;

}

public String getPort() {

return port;

}

public void setPort(String port) {

this.port = port;

}

public String getLastDataTime() {

return lastDataTime;

}

public void setLastDataTime(String lastDataTime) {

this.lastDataTime = lastDataTime;

}

public String getFunctionName() {

return functionName;

}

public void setFunctionName(String functionName) {

this.functionName = functionName;

}

public int getWriteTimeOut() {

return writeTimeOut;

}

public void setWriteTimeOut(int writeTimeOut) {

this.writeTimeOut = writeTimeOut;

}

public int getReadTimeOut() {

return readTimeOut;

}

public void setReadTimeOut(int readTimeOut) {

this.readTimeOut = readTimeOut;

}

public String getLastMetaDate() {

return lastMetaDate;

}

public void setLastMetaDate(String lastMetaDate) {

this.lastMetaDate = lastMetaDate;

}

}

package com.sdocean.station.model;

public class StationDeviceIndicator {

private int id;

private int stationId;

private String stationName;

private String companyName;

private int deviceId;

private int indicatorId;

private String indicatorIds;

public String getIndicatorIds() {

return indicatorIds;

}

public void setIndicatorIds(String indicatorIds) {

this.indicatorIds = indicatorIds;

}

public String getStationName() {

return stationName;

}

public void setStationName(String stationName) {

this.stationName = stationName;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

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 int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public int getIndicatorId() {

return indicatorId;

}

public void setIndicatorId(int indicatorId) {

this.indicatorId = indicatorId;

}

}

package com.sdocean.station.model;

public class StationDeviceModel {

private int id;

private int stationId;

private String stationName;

private int deviceId;

private String deviceName;

private int pointNum; //小数位数

private String createTime; //设备安装时间

private String orderCode; //排序编码

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 int getPointNum() {

return pointNum;

}

public void setPointNum(int pointNum) {

this.pointNum = pointNum;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

}

package com.sdocean.station.model;

public class StationDevicePage {

private int id;

private int stationId;

private int pdeviceId;

private int cdeviceId;

private String page;

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 int getPdeviceId() {

return pdeviceId;

}

public void setPdeviceId(int pdeviceId) {

this.pdeviceId = pdeviceId;

}

public int getCdeviceId() {

return cdeviceId;

}

public void setCdeviceId(int cdeviceId) {

this.cdeviceId = cdeviceId;

}

public String getPage() {

return page;

}

public void setPage(String page) {

this.page = page;

}

}

package com.sdocean.station.model;

public class StationInfo {

private int id;

private int stationId;

private String stationName;

private String infomation;

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 getInfomation() {

return infomation;

}

public void setInfomation(String infomation) {

this.infomation = infomation;

}

}

package com.sdocean.station.model;

import com.sdocean.common.model.SelectTree;

public class StationModel {

private int id;

private String code;

private String title;

private double latitude; //维度

private double longitude; //经度

private String station\_gateway; //站点地址

private int stationtype\_id; //站点类型

private String stationTypeName; //站点类型姓名

private int region\_id; //站点归属地区

private String regionName; //地区名称

private String companyId; //站点归属单位

private String companyName; //站点归属单位名称

private String brief; //站点描述

private String detail; //站点细节;

private String pic; //站点图片

private int isactive;

private String isactiveName;

private int waterType; //站点水的类型 sys\_public 0005

private String waterTypeName;

private String orderCode;

private SelectTree region;

private String deviceIds;

private String deviceNames;

private int ifsms;

private String ifsmsName;

private String lastMetaDate;

private String url;

private String icon;

//从属平台

private int platformId;

private String platformName;

// a.code,a.title,a.latitude,a.longitude,a.station\_gateway,a.stationtype\_id,a.dictregion\_id,a.brief,a.detail

// a.pic,a.isactive,a.waterType

public int getId() {

return id;

}

public int getPlatformId() {

return platformId;

}

public void setPlatformId(int platformId) {

this.platformId = platformId;

}

public String getPlatformName() {

return platformName;

}

public void setPlatformName(String platformName) {

this.platformName = platformName;

}

public String getIcon() {

return icon;

}

public void setIcon(String icon) {

this.icon = icon;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getLastMetaDate() {

return lastMetaDate;

}

public void setLastMetaDate(String lastMetaDate) {

this.lastMetaDate = lastMetaDate;

}

public int getIfsms() {

return ifsms;

}

public void setIfsms(int ifsms) {

this.ifsms = ifsms;

}

public String getIfsmsName() {

return ifsmsName;

}

public void setIfsmsName(String ifsmsName) {

this.ifsmsName = ifsmsName;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getCompanyId() {

return companyId;

}

public void setCompanyId(String companyId) {

this.companyId = companyId;

}

public String getDeviceIds() {

return deviceIds;

}

public void setDeviceIds(String deviceIds) {

this.deviceIds = deviceIds;

}

public String getDeviceNames() {

return deviceNames;

}

public void setDeviceNames(String deviceNames) {

this.deviceNames = deviceNames;

}

public SelectTree getRegion() {

return region;

}

public void setRegion(SelectTree region) {

this.region = region;

}

public String getStationTypeName() {

return stationTypeName;

}

public void setStationTypeName(String stationTypeName) {

this.stationTypeName = stationTypeName;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public double getLatitude() {

return latitude;

}

public void setLatitude(double latitude) {

this.latitude = latitude;

}

public double getLongitude() {

return longitude;

}

public void setLongitude(double longitude) {

this.longitude = longitude;

}

public String getStation\_gateway() {

return station\_gateway;

}

public void setStation\_gateway(String station\_gateway) {

this.station\_gateway = station\_gateway;

}

public int getStationtype\_id() {

return stationtype\_id;

}

public void setStationtype\_id(int stationtype\_id) {

this.stationtype\_id = stationtype\_id;

}

public int getRegion\_id() {

return region\_id;

}

public void setRegion\_id(int region\_id) {

this.region\_id = region\_id;

}

public String getRegionName() {

return regionName;

}

public void setRegionName(String regionName) {

this.regionName = regionName;

}

public String getBrief() {

return brief;

}

public void setBrief(String brief) {

this.brief = brief;

}

public String getDetail() {

return detail;

}

public void setDetail(String detail) {

this.detail = detail;

}

public String getPic() {

return pic;

}

public void setPic(String pic) {

this.pic = pic;

}

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 int getWaterType() {

return waterType;

}

public void setWaterType(int waterType) {

this.waterType = waterType;

}

public String getWaterTypeName() {

return waterTypeName;

}

public void setWaterTypeName(String waterTypeName) {

this.waterTypeName = waterTypeName;

}

}

package com.sdocean.station.model;

public class StationTypeModel {

private int id;

private String code;

private String name;

private String remark;

private int isactive;

private String iactiveName;

private String icon;

public String getIcon() {

return icon;

}

public void setIcon(String icon) {

this.icon = icon;

}

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 getRemark() {

return remark;

}

public void setRemark(String remark) {

this.remark = remark;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIactiveName() {

return iactiveName;

}

public void setIactiveName(String iactiveName) {

this.iactiveName = iactiveName;

}

}