金坤科创智慧人防监测综合管理系统软件V1.0

源代码

北京金坤科创技术有限公司

2017-08-24

后30页

package com.ajax;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.ApInfo;

import com.beans.ProjInfo;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.UsedCache;

/\*\*

\* 人防基站信息管理,增删改查

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxAp")

public class AjaxAp {

@Autowired

private CommonService commonService;

// 添加基站信息

@SuppressWarnings("unchecked")

@RequestMapping("/addAp.do")

@ResponseBody

public JSONObject addAp(String baseNo, String baseName, String projName,

String ip, String ap\_id, Double pos\_x, Double pos\_y, Double pos\_z,

String remarks) {

JSONObject json = new JSONObject();

ApInfo apInfo = new ApInfo();

List<ApInfo> findByHql = commonService.findByHql(ApInfo.class,

"baseNo = '" + baseNo + "'", null);

try {

if (findByHql.isEmpty()) {

apInfo.setUserNo(Global.useNo);

apInfo.setBaseNo(baseNo);

apInfo.setBaseName(baseName);

apInfo.setProjName(projName);

apInfo.setIp(ip);

apInfo.setAp\_id(ap\_id);

apInfo.setPos\_x(pos\_x);

apInfo.setPos\_y(pos\_y);

apInfo.setPos\_z(pos\_z);

apInfo.setWhether\_use("1");

apInfo.setAp\_flag(0);

commonService.saveDate(apInfo);

json.put("state", "success");

} else {

json.put("state", "exist");

}

} catch (Exception e) {

json.put("state", "no");

}

return json;

}

// 查询基站信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getApList.do")

@ResponseBody

public List getApList(HttpSession session, HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String baseName = request.getParameter("baseName") == null ? ""

: request.getParameter("baseName").trim();// 工事名称

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(baseName)) {

subwhere += " and baseName like'%" + baseName + "%'";

}

List<ApInfo> apInfoList = commonService.findByHqlPage(ApInfo.class,

subwhere, "id desc", currentpage, pagesize);

List<ApInfo> list = commonService.findByHql(ApInfo.class, "", null);

// 把基站信息存入缓存，基站MAC为key

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

ApInfo apInfo = list.get(i);

UsedCache.m\_station\_map.put(apInfo.getAp\_id(), apInfo);

}

return apInfoList;

}

// 查询总条数

@RequestMapping("/getApCount.do")

@ResponseBody

public JSONObject getApCount(HttpSession session, HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String baseName = request.getParameter("baseName") == null ? ""

: request.getParameter("baseName").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(baseName)) {

subwhere += " and baseName like '%" + baseName + "%'";

}

long records = commonService.getResultCount(ApInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 工事信息查询

@SuppressWarnings("unchecked")

@RequestMapping("/findProjName.do")

@ResponseBody

public List<ProjInfo> findProjName() {

List<ProjInfo> projList = commonService.findByHql(ProjInfo.class, "",

"");

return projList;

}

// 删除单条数据

@RequestMapping("/deleteAp.do")

@ResponseBody

public JSONObject deleteAp(String id) {

JSONObject json = new JSONObject();

try {

commonService.deleteData(ApInfo.class, "id='" + id + "'");

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

}

return json;

}

// 更新数据

@RequestMapping("/updateAp.do")

@ResponseBody

public JSONObject updateAp(String baseNo, String baseName, String projName,

String ip, String ap\_id, Double pos\_x, Double pos\_y, Double pos\_z,

String whether\_use, Integer ap\_flag) {

JSONObject json = new JSONObject();

try {

commonService.updateDate(ApInfo.class,

new String[] {

"baseName='" + baseName + "'",

"projName='" + projName + "'", "ip='" + ip + "'",

"ap\_id='" + ap\_id + "'", "pos\_x='" + pos\_x + "'",

"pos\_y='" + pos\_y + "'", "pos\_z='" + pos\_z + "'"

},

"baseNo='" + baseNo + "'");

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

json.put("state", "no");

}

return json;

}

//查询基站报警信息

@RequestMapping("/getApAlarm.do")

@ResponseBody

public JSONObject getApAlarm(){

JSONObject json = new JSONObject();

json.put("alarmStation", UsedCache.m\_station\_error\_alarm\_map);

return json;

}

}

package com.ajax;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.UePosition;

import com.beans.UserInfo;

import com.service.CommonService;

import com.util.global.UsedCache;

/\*\*

\* 获取轨迹回放信息

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxPositionTrack")

public class AjaxPositionTrack {

@Autowired

CommonService commonService;

@SuppressWarnings("unchecked")

@RequestMapping("getPositionTrack.do")

@ResponseBody

public JSONObject getPositionTrack(HttpServletRequest request) {

String peopleNameMac = request.getParameter("peopleNameMac") == null ? ""

: request.getParameter("peopleNameMac").trim();// 获取页面传来MAC

String start = request.getParameter("start") == null ? "" : request

.getParameter("start").trim();// 获取页面传来开始时间

String end = request.getParameter("end") == null ? "" : request

.getParameter("end").trim();// 获取页面传来结束时间

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");// 日期时间格式.

JSONObject json = new JSONObject();

try {

long start1 = df.parse(start).getTime()+8\*3600\*1000;

long end1 = df.parse(end).getTime()+8\*3600\*1000;

// 根据获取名字查询绑定的MAC地址和职务

List<UserInfo> list = commonService.findByHql(UserInfo.class,

"mac='" + peopleNameMac + "'", "");

if (!list.isEmpty()) {

String peopleName = list.get(0).getUserName();

String duty = list.get(0).getDuty();

// 根据绑定的MAC地址、开始时间和结束时间查询人员这段时间的定位轨迹信息

List<UePosition> positonTrack = commonService.findByHql(

UePosition.class, "ueId='" + peopleNameMac

+ "' and (recordTime between '" + start1

+ "'and'" + end1 + "')", null);

if (!positonTrack.isEmpty()) {

json.put("start", start);

json.put("end", end);

json.put("mac\_id", peopleNameMac);

json.put("peopleName", peopleName);

json.put("duty", duty);

json.put("positonTrack", positonTrack);

}else{

json.put("kong", "kong");//定位历史信息没有此人信息

}

}else{

json.put("kong", "kong");//人员列表没有此人

}

} catch (ParseException e) {

e.printStackTrace();

}

return json;

}

/\*\*

\* 查找项目室内地图上所有人员以及数量

\*/

@RequestMapping("/selectUserName.do")

@ResponseBody

public String selectUserName() throws Exception {

JSONObject json = new JSONObject();

try {

json.put("userName", UsedCache.userInfoMacMap);//定位人员信息

} catch (Exception e) {

System.out.println(e);

}

return json.toString();

}

}

package com.ajax;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.InspInfo;

import com.beans.ProjInfo;

import com.beans.UserInfo;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.LocalTime;

/\*\*

\* 巡查人员管理增删改查

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxInsp")

public class AjaxInsp {

@Autowired

private CommonService commonService;

// 添加信息

@RequestMapping("/addInsp.do")

@ResponseBody

public JSONObject addInsp(String inspName, String inspPhone, String projName,

String inspOutcome, String inspAdvice) {

JSONObject json = new JSONObject();

InspInfo inspInfo = new InspInfo();

/\*List<InspInfo> findByHql = commonService.findByHql(InspInfo.class,

"inspName = '" + inspName + "'", null);\*/

try {

/\*if (findByHql.isEmpty()) {\*/

inspInfo.setUserNo(Global.useNo);

inspInfo.setInspName(inspName);

inspInfo.setInspPhone(inspPhone);

inspInfo.setProjName(projName);

inspInfo.setInspOutcome(inspOutcome);

inspInfo.setInspAdvice(inspAdvice);

inspInfo.setInspTime(LocalTime.getDateTime());

commonService.saveDate(inspInfo);

json.put("state", "success");

/\*} else {

json.put("state", "exist");

}\*/

} catch (Exception e) {

json.put("state", "no");

}

return json;

}

// 查询信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getInspList.do")

@ResponseBody

public List getInsp(HttpSession session, HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String inspName = request.getParameter("inspName") == null ? ""

: request.getParameter("inspName").trim();// 人员姓名

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(inspName)) {

subwhere += " and inspName like'%" + inspName + "%'";

}

List<InspInfo> inspList = commonService.findByHqlPage(InspInfo.class,

subwhere, "id desc", currentpage, pagesize);

return inspList;

}

//工事信息查询

@SuppressWarnings("unchecked")

@RequestMapping("/findProjName.do")

@ResponseBody

public List<ProjInfo> findProjName (){

List<ProjInfo> projList = commonService.findByHql(ProjInfo.class,"", "");

return projList;

}

//人员信息查询

@SuppressWarnings("unchecked")

@RequestMapping("/findUserName.do")

@ResponseBody

public List<UserInfo> findUserName (){

List<UserInfo> userList = commonService.findByHql(UserInfo.class,"", "");

return userList;

}

// 查询总条数

@RequestMapping("/getInspCount.do")

@ResponseBody

public JSONObject getInspCount(HttpSession session,

HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String inspName = request.getParameter("inspName") == null ? ""

: request.getParameter("inspName").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(inspName)) {

subwhere += " and inspName like '%" + inspName + "%'";

}

long records = commonService.getResultCount(InspInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 删除单条数据

@RequestMapping("/delInsp.do")

@ResponseBody

public JSONObject delInsp(String id) {

JSONObject json = new JSONObject();

try {

commonService.deleteData(InspInfo.class, "id='" + id + "'");

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

}

return json;

}

// 更新数据

@RequestMapping("/updInsp.do")

@ResponseBody

public JSONObject updInsp(String id, String inspName, String inspPhone,

String projName, String inspOutcome, String inspAdvice) {

JSONObject json = new JSONObject();

try {

commonService.updateDate(InspInfo.class, new String[] {

"inspName='" + inspName + "'", "inspPhone='" + inspPhone + "'",

"projName='" + projName + "'", "inspOutcome='" + inspOutcome + "'",

"inspAdvice='" + inspAdvice + "'"},

"id='" + id + "'");

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

json.put("state", "no");

}

return json;

}

}

package com.ajax;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.ProjInfo;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.UsedCache;

/\*\*

\* 人防工事信息管理,增删改查

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxProj")

public class AjaxProj {

@Autowired

private CommonService commonService;

// 添加工事信息

@SuppressWarnings("unchecked")

@RequestMapping("/addProj.do")

@ResponseBody

public JSONObject addProj(HttpSession session, String projNo, String projName,

String wartimeUse, String isUse, String maintenance,

String attribute, String resistance, String defenseChemic,

String unit, String address) {

JSONObject json = new JSONObject();

ProjInfo projInfo = new ProjInfo();

List<ProjInfo> findByHql = commonService.findByHql(ProjInfo.class,

"projNo = '" + projNo + "'", null);

try {

if (findByHql.isEmpty() && 2==UsedCache.loginInfoMap.get(session.getId()).getpId()) {

projInfo.setUserNo(Global.useNo);

projInfo.setProjNo(projNo);

projInfo.setProjName(projName);

projInfo.setWartimeUse(wartimeUse);

projInfo.setIsUse(isUse);

projInfo.setMaintenance(maintenance);

projInfo.setAttribute(attribute);

projInfo.setResistance(resistance);

projInfo.setDefenseChemic(defenseChemic);

projInfo.setUnit(unit);

projInfo.setAddress(address);

projInfo.setEntryTime(new Date());

commonService.saveDate(projInfo);

json.put("state", "success");

} else {

json.put("state", "exist");

}

} catch (Exception e) {

json.put("state", "no");

}

return json;

}

// 查询工事信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getProjList.do")

@ResponseBody

public List getProjList(HttpSession session, HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String projName = request.getParameter("projName") == null ? ""

: request.getParameter("projName").trim();// 工事名称

String unit = request.getParameter("unit") == null ? "" : request

.getParameter("unit").trim();// 管理单位

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(projName)) {

subwhere += " and projName like'%" + projName + "%'";

}

if (!"".equals(unit)) {

subwhere += " and unit like '%" + unit + "%'";

}

List<ProjInfo> projList = commonService.findByHqlPage(ProjInfo.class,

subwhere, "id desc", currentpage, pagesize);

return projList;

}

// 查询总条数

@RequestMapping("/getProjCount.do")

@ResponseBody

public JSONObject getProjCount(HttpSession session,

HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String projName = request.getParameter("projName") == null ? ""

: request.getParameter("projName").trim();

String unit = request.getParameter("unit") == null ? "" : request

.getParameter("unit").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(projName)) {

subwhere += " and projName like '%" + projName + "%'";

}

if (!"".equals(unit)) {

subwhere += " and unit like '%" + unit + "%'";

}

long records = commonService.getResultCount(ProjInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 删除单条数据

@RequestMapping("/deleteProj.do")

@ResponseBody

public JSONObject deleteProj(HttpSession session, String id) {

JSONObject json = new JSONObject();

try {

if(2==UsedCache.loginInfoMap.get(session.getId()).getpId()){

commonService.deleteData(ProjInfo.class, "id='" + id + "'");

json.put("state", "success");

}else{

json.put("state", "exits");

}

} catch (Exception e) {

e.printStackTrace();

}

return json;

}

// 更新数据

@RequestMapping("/updateProj.do")

@ResponseBody

public JSONObject updateProj(String projNo, String projName,

String wartimeUse, String isUse, String maintenance,

String attribute, String resistance, String defenseChemic,

String unit, String address) {

JSONObject json = new JSONObject();

try {

commonService.updateDate(ProjInfo.class, new String[] {

"projName='" + projName + "'",

"wartimeUse='" + wartimeUse + "'", "isUse='" + isUse + "'",

"maintenance='" + maintenance + "'",

"attribute='" + attribute + "'",

"resistance='" + resistance + "'",

"defenseChemic='" + defenseChemic + "'",

"unit='" + unit + "'", "address='" + address + "'" },

"projNo='" + projNo + "'");

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

json.put("state", "no");

}

return json;

}

}

package com.ajax;

import net.sf.json.JSONObject;

import org.springframework.stereotype.Controller;

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

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

import com.util.global.UsedCache;

/\*\*

\* 从map缓存获取温湿度

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxSensor")

public class AjaxSensor {

@RequestMapping("getTemHumList.do")

@ResponseBody

public JSONObject getTemHumList() {

JSONObject json = new JSONObject();

// 把温湿度放入JSON

json.put("map", UsedCache.temHumInfoMap);

return json;

}

}

package com.ajax;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.StationErrorInfo;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

/\*\*

\* 查看温湿度历史信息

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxStationAlarmHistory")

public class AjaxStationAlarmHistory {

@Autowired

CommonService commonService;

//分页查询

@RequestMapping("getStationAlarmCount.do")

@ResponseBody

public JSONObject getTemHumCount(HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String stationName = request.getParameter("stationName") == null ? ""

: request.getParameter("stationName").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(stationName)) {

subwhere += " and stationName like '%" + stationName + "%'";

}

long records = commonService.getResultCount(

StationErrorInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 查询温度历史信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getStationAlarmHistoryList.do")

@ResponseBody

public List getBaseStaList(HttpSession session, HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String stationName = request.getParameter("stationName") == null ? ""

: request.getParameter("stationName").trim();//设备地址

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(stationName)) {

subwhere += " and stationName like'%" + stationName + "%'";

}

List<StationErrorInfo> stationErrorInfo = commonService.findByHqlPage(

StationErrorInfo.class, subwhere, "id desc", currentpage, pagesize);

return stationErrorInfo;

}

}

package com.ajax;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.UserInfo;

import com.beans.UserManager;

import com.common.util.MD5;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.LocalTime;

import com.util.global.UsedCache;

/\*\*

\* 人员管理增删改查

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxUser")

public class AjaxUser {

@Autowired

private CommonService commonService;

// 添加信息

@SuppressWarnings("unchecked")

@RequestMapping("/addUser.do")

@ResponseBody

public JSONObject addUser(HttpSession session, String userName, String sex,

String age, String duty, String number, String address, String mac) {

JSONObject json = new JSONObject();

UserInfo userInfo = new UserInfo();

List<UserInfo> list1 = commonService.findByHql(UserInfo.class,

"userName = '" + userName + "'", null);

List<UserInfo> list2 = commonService.findByHql(UserInfo.class,

"number = '" + number + "'", null);

UserManager userManager = new UserManager();

try {

if (list1.isEmpty()&& list2.isEmpty()

&& 2 == UsedCache.loginInfoMap.get(session.getId()).getpId()) {

userInfo.setUserNo(Global.useNo);

userInfo.setUserName(userName);

userInfo.setSex(sex);

userInfo.setAge(age);

userInfo.setDuty(duty);

userInfo.setNumber(number);

userInfo.setAddress(address);

userInfo.setMac(mac);

userInfo.setEntryTime(LocalTime.getDateTime());

// 保存人员信息

commonService.saveDate(userInfo);

UsedCache.userInfoMap.put(userName, userInfo);

userManager.setUserNo(number);

userManager.setPassword(MD5.md5Encode("12345678"));

userManager.setManagerName(duty);

userManager.setUserRule(UsedCache.loginInfoMap.get(

session.getId()).getUserNo());

userManager.setpId(3);

commonService.saveDate(userManager);

json.put("state", "success");

} else {

json.put("state", "exist");

}

} catch (Exception e) {

json.put("state", "no");

}

return json;

}

// 查询信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getUserList.do")

@ResponseBody

public List getUserList(HttpSession session, HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String userName = request.getParameter("userName") == null ? ""

: request.getParameter("userName").trim();// 人员姓名

String subwhere = "userNo like'%" + Global.useNo + "%'";// 获取当前登录账号

if (!"".equals(userName)) {

subwhere += " and userName like'%" + userName + "%'";

}

List<UserInfo> userList = commonService.findByHqlPage(UserInfo.class,

subwhere, "id desc", currentpage, pagesize);

// 查询最新用户信息放入Map

selectUser();

return userList;

}

// 查询总条数

@RequestMapping("/getUserCount.do")

@ResponseBody

public JSONObject getUserCount(HttpSession session,

HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String userName = request.getParameter("userName") == null ? ""

: request.getParameter("userName").trim();

/\* String subwhere = "userName <>'admin'"; \*/

String subwhere = "userNo like'%" + Global.useNo + "%'";// 获取当前登录账号

if (!"".equals(userName)) {

subwhere += " and userName like '%" + userName + "%'";

}

long records = commonService.getResultCount(UserInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 删除单条数据

@SuppressWarnings("unchecked")

@RequestMapping("/delUser.do")

@ResponseBody

public JSONObject delUser(HttpSession session, String id) {

JSONObject json = new JSONObject();

try {

if (2 == UsedCache.loginInfoMap.get(session.getId()).getpId()) {

commonService.deleteData(UserInfo.class, "id='" + id + "'");

List<UserInfo> findByHql = commonService.findByHql(

UserInfo.class, "id='" + id + "'", null);

if (null != findByHql.get(0).getNumber()) {

commonService.deleteData(UserManager.class, "userNo='"

+ findByHql.get(0).getNumber() + "'");

}

json.put("state", "success");

} else {

json.put("state", "exits");

}

} catch (Exception e) {

e.printStackTrace();

}

// 查询删除后用户数据

selectUser();

return json;

}

// 更新数据

@SuppressWarnings("unchecked")

@RequestMapping("/updUser.do")

@ResponseBody

public JSONObject updUser(String id, String userName, String sex,

String age, String duty, String number, String address, String mac) {

JSONObject json = new JSONObject();

try {

List<UserInfo> findByHql = commonService.findByHql(UserInfo.class,

"id = '" + id + "'", null);

//判断电话是否修改，如修改则更新权限列表同时初始化密码

if (!number.equals(findByHql.get(0).getNumber())) {

List<UserManager> findByHql2 = commonService.findByHql(

UserManager.class, "userNo='"

+ findByHql.get(0).getNumber() + "'", null);

commonService.updateDate(UserManager.class,new String[] {

"userNo='" + number + "'",

"password='" + MD5.md5Encode("12345678") + "'" },

"id = '" + findByHql2.get(0).getId() + "'");

}

commonService.updateDate(UserInfo.class, new String[] {

"userName='" + userName + "'", "sex='" + sex + "'",

"age='" + age + "'", "duty='" + duty + "'",

"number='" + number + "'", "address='" + address + "'",

"mac='" + mac + "'" }, "id='" + id + "'");

// 查询更新后用户数据

selectUser();

json.put("state", "success");

} catch (Exception e) {

e.printStackTrace();

json.put("state", "no");

}

return json;

}

// 查询最新用户信息并以MAC为key放入MAP

@SuppressWarnings("unchecked")

public void selectUser() {

List<UserInfo> userInfo = commonService.findByHql(UserInfo.class, "",

null);

UsedCache.userInfoMacMap.clear();

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

// MAC不为空则以MAC为key

if (null != userInfo.get(i).getMac()) {

UsedCache.userInfoMacMap.put(userInfo.get(i).getMac(),

userInfo.get(i));

}

}

}

}

package com.ajax;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.WaterInfo;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.LocalTime;

/\*\*

\* 查看水浸状况历史信息

\*

\* @author XY

\*

\*/

@Controller

@RequestMapping("ajaxWaterHistory")

public class AjaxWaterHistory {

@Autowired

CommonService commonService;

// 分页查询

@RequestMapping("getWaterHistoryCount.do")

@ResponseBody

public JSONObject getWaterHistoryCount(HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String deviceAddress = request.getParameter("deviceAddress") == null ? ""

: request.getParameter("deviceAddress").trim();

String waterStatus = request.getParameter("waterStatus") == null ? ""

: request.getParameter("waterStatus").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(deviceAddress)) {

subwhere += " and deviceAddress like '%" + deviceAddress + "%'";

}

if (!"".equals(waterStatus)) {

subwhere += " and waterStatus like '%" + waterStatus + "%'";

}

long records = commonService.getResultCount(WaterInfo.class, subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 查询水浸历史信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getWaterHistoryList.do")

@ResponseBody

public List getWaterHistoryList(HttpSession session,

HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String deviceAddress = request.getParameter("deviceAddress") == null ? ""

: request.getParameter("deviceAddress").trim();// 设备地址

String waterStatus = request.getParameter("waterStatus") == null ? ""

: request.getParameter("waterStatus").trim();// 浸水状态

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(deviceAddress)) {

subwhere += " and deviceAddress like'%" + deviceAddress + "%'";

}

if (!"".equals(waterStatus)) {

subwhere += " and waterStatus like '%" + waterStatus + "%'";

}

List<WaterInfo> waterHistoryList = commonService.findByHqlPage(

WaterInfo.class, subwhere, "id desc", currentpage, pagesize);

return waterHistoryList;

}

// 查询水浸状况最新信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getWaterNewList.do")

@ResponseBody

public List getWaterNewList(HttpServletRequest request) {

String showDevice = request.getParameter("showDevice") == null ? ""

: request.getParameter("showDevice").trim();// 设备地址

String time\_water = request.getParameter("time\_water") == null ? ""

: request.getParameter("time\_water").trim();

int graph = Integer.parseInt(time\_water);// 转为整数型以进行条件判断

String now = LocalTime.getDateTime();

SimpleDateFormat datetimeFormat = new SimpleDateFormat(

"yyyy-MM-dd HH:mm:ss");// 日期时间格式

Calendar c = Calendar.getInstance();

List<WaterInfo> list = new ArrayList();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(showDevice)) {

subwhere += " and nodeName ='" + showDevice + "'";

}

if (graph == 1) {

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

String one\_month = datetimeFormat.format(c.getTime());// 一个月前时间

subwhere += " and entryTime between '" + one\_month + "'and'" + now

+ "' ";

list = commonService.findByHql(WaterInfo.class, subwhere, null);

} else if (graph == 2) {

c.add(Calendar.MONTH, -3);

String three\_months = datetimeFormat.format(c.getTime());// 三个月前时间

subwhere += " and entryTime between '" + three\_months + "'and'"

+ now + "'";

list = commonService.findByHql(WaterInfo.class, subwhere, null);

} else if (graph == 3) {

c.add(Calendar.MONTH, -6);

String half\_year = datetimeFormat.format(c.getTime());// 半年前时间

subwhere += " and entryTime between '" + half\_year + "'and'" + now

+ "'";

list = commonService.findByHql(WaterInfo.class, subwhere, null);

} else if (graph == 4) {

c.add(Calendar.YEAR, -1);

String one\_year = datetimeFormat.format(c.getTime());// 一年前时间

subwhere += " and entryTime between '" + one\_year + "'and'" + now

+ "'";

list = commonService.findByHql(WaterInfo.class, subwhere, null);

} else {

list = commonService.findByHql(WaterInfo.class, subwhere, null);

int i = 0;

int len = list.size();

String entryTime = list.get(len-1).getEntryTime();

String day = entryTime.split(" ")[0];

for (; i < len; i++) {

if (!day.equals(list.get(i).getEntryTime().split(" ")[0])) {

list.remove(i);

i = i - 1;

len = len - 1;

}

}

}

return list; // 获取最新一天数据列表

}

// 设备查询

@SuppressWarnings("unchecked")

@RequestMapping("/findDevice.do")

@ResponseBody

public List<String> findDevice() {

List<String> deviceList = commonService.findByHqlGetColumn(

WaterInfo.class, "", "nodeName","distinct nodeName");

return deviceList;

}

}

package com.ajax;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.DistanceInfo;

import com.common.dao.CommonDao;

import com.common.util.PageUtil;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.LocalTime;

/\*\*

\* 查看激光测距历史信息

\*

\* @author YYJ

\*

\*/

@Controller

@RequestMapping("ajaxDistanceHistory")

public class AjaxDistanceHistory {

@Autowired

CommonService commonService;

CommonDao commonDao;

// 分页查询

@RequestMapping("getDistanceHistoryCount.do")

@ResponseBody

public JSONObject getDistanceHistoryCount(HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String deviceAddress = request.getParameter("deviceAddress") == null ? ""

: request.getParameter("deviceAddress").trim();

String alertStatus = request.getParameter("alertStatus") == null ? ""

: request.getParameter("alertStatus").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(deviceAddress)) {

subwhere += " and deviceAddress like '%" + deviceAddress + "%'";

}

if (!"".equals(alertStatus)) {

subwhere += " and alertStatus like '%" + alertStatus + "%'";

}

long records = commonService.getResultCount(DistanceInfo.class,

subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 查询激光测距历史信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getDistanceHistoryList.do")

@ResponseBody

public List getDistanceHistoryList(HttpSession session,

HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String deviceAddress = request.getParameter("deviceAddress") == null ? ""

: request.getParameter("deviceAddress").trim();// 设备地址

String alertStatus = request.getParameter("alertStatus") == null ? ""

: request.getParameter("alertStatus").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(deviceAddress)) {

subwhere += " and deviceAddress like'%" + deviceAddress + "%'";

}

if (!"".equals(alertStatus)) {

subwhere += " and alertStatus like '%" + alertStatus + "%'";

}

List<DistanceInfo> distanceHistoryList = commonService.findByHqlPage(

DistanceInfo.class, subwhere, "id desc", currentpage, pagesize);

return distanceHistoryList;

}

// 查询激光测距最新信息集合

@SuppressWarnings({ "unchecked", "rawtypes" })

@RequestMapping("/getDistanceNewList.do")

@ResponseBody

public List getDistanceNewList(HttpServletRequest request) {

String showDevice = request.getParameter("showDevice") == null ? ""

: request.getParameter("showDevice").trim();// 设备地址

String time\_distance = request.getParameter("time\_distance") == null ? ""

: request.getParameter("time\_distance").trim();

int graph = Integer.parseInt(time\_distance);// 转为整数型以进行条件判断

String now = LocalTime.getDateTime();

SimpleDateFormat datetimeFormat = new SimpleDateFormat(

"yyyy-MM-dd HH:mm:ss");// 日期时间格式

Calendar c = Calendar.getInstance();

List<DistanceInfo> list = new ArrayList();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(showDevice)) {

subwhere += " and nodeName ='" + showDevice + "'";

}

if (graph == 1) {

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

String one\_month = datetimeFormat.format(c.getTime());// 一个月前时间

subwhere += " and entryTime between '" + one\_month + "'and'" + now

+ "' ";

list = commonService.findByHql(DistanceInfo.class, subwhere, null);

} else if (graph == 2) {

c.add(Calendar.MONTH, -3);

String three\_months = datetimeFormat.format(c.getTime());// 三个月前时间

subwhere += " and entryTime between '" + three\_months + "'and'"

+ now + "'";

list = commonService.findByHql(DistanceInfo.class, subwhere, null);

} else if (graph == 3) {

c.add(Calendar.MONTH, -6);

String half\_year = datetimeFormat.format(c.getTime());// 半年前时间

subwhere += " and entryTime between '" + half\_year + "'and'" + now

+ "'";

list = commonService.findByHql(DistanceInfo.class, subwhere, null);

} else if (graph == 4) {

c.add(Calendar.YEAR, -1);

String one\_year = datetimeFormat.format(c.getTime());// 一年前时间

subwhere += " and entryTime between '" + one\_year + "'and'" + now

+ "'";

list = commonService.findByHql(DistanceInfo.class, subwhere, null);

} else {

list = commonService.findByHql(DistanceInfo.class, subwhere, null);

int i = 0;

int len = list.size();

String entryTime = list.get(len - 1).getEntryTime();

String day = entryTime.split(" ")[0];

for (; i < len; i++) {

if (!day.equals(list.get(i).getEntryTime().split(" ")[0])) {

list.remove(i);

i = i - 1;

len = len - 1;

}

}

}

return list; // 获取最新一天数据列表

}

// 设备查询

@SuppressWarnings("unchecked")

@RequestMapping("/findDevice.do")

@ResponseBody

public List<String> findDevice() {

List<String> deviceList = commonService.findByHqlGetColumn(

DistanceInfo.class, "", "nodeName","distinct nodeName");

return deviceList;

}

}

package com.ajax;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import net.sf.json.JSONObject;

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

import org.springframework.stereotype.Controller;

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

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

import com.beans.AttendanceInfo;

import com.common.util.BufferDataInit;

import com.common.util.PageUtil;

import com.form.PositionBean;

import com.service.CommonService;

import com.util.global.Global;

import com.util.global.LocalTime;

import com.util.global.UsedCache;

/\*\*

\* 查看出入工事信息

\*

\* @author ZHUZHIYONG

\*

\*/

@Controller

@RequestMapping("ajaxAttendance")

public class AjaxAttendance {

@Autowired

CommonService commonService;

// 分页查询

@RequestMapping("getAttendanceCount.do")

@ResponseBody

public JSONObject getAttendanceCount(HttpServletRequest request) {

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String peopleName = request.getParameter("peopleName") == null ? ""

: request.getParameter("peopleName").trim();

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(peopleName)) {

subwhere += " and peopleName like '%" + peopleName + "%'";

}

long records = commonService.getResultCount(AttendanceInfo.class,

subwhere);

long pageCount = PageUtil.getPageCount(records, pagesize);

JSONObject json = new JSONObject();

json.put("pageCount", pageCount);

return json;

}

// 查询出入人防工事信息集合

@SuppressWarnings({ "unchecked" })

@RequestMapping("/getAttendanceList.do")

@ResponseBody

public List<AttendanceInfo> getBaseStaList(HttpSession session,

HttpServletRequest request) {

Integer currentpage = request.getParameter("currentpage") == null ? PageUtil.currentpage

: Integer.parseInt(request.getParameter("currentpage").trim());// 当前页

Integer pagesize = request.getParameter("pagesize") == null ? PageUtil.pagesize

: Integer.parseInt(request.getParameter("pagesize").trim());// 页行数

String peopleName = request.getParameter("peopleName") == null ? ""

: request.getParameter("peopleName").trim();// 设备地址

String subwhere = "userNo like'%"+Global.useNo+"%'";//获取当前登录账号

if (!"".equals(peopleName)) {

subwhere += " and peopleName like'%" + peopleName + "%'";

}

List<AttendanceInfo> attendanceInfo = commonService.findByHqlPage(

AttendanceInfo.class, subwhere, "id desc", currentpage, pagesize);

return attendanceInfo;

}

/\*\*

\* 进出人防工事打卡

\*/

@SuppressWarnings("unchecked")

public void searchAttendance() {

// 遍历map，对出入人防工事自动打卡登记

for (Map.Entry<String, PositionBean> entry : UsedCache.positionBeanMap

.entrySet()) {

String mac = entry.getValue().getMac();

AttendanceInfo attendanceInfo = new AttendanceInfo();

List<AttendanceInfo> list = BufferDataInit.commonService.findByHql(

AttendanceInfo.class, "mac='" + mac

+ "' and startTime like '%" + LocalTime.getDate()

+ "%' ", null);

// 判断人员打卡记录是否为空，为空则进行打卡

if (list.isEmpty()) {

attendanceInfo.setPeopleName(UsedCache.userInfoMacMap.get(mac)

.getUserName());

attendanceInfo.setMac(mac);

attendanceInfo.setStartTime(LocalTime.getDateTime());

attendanceInfo.setUserNo(Global.useNo);//添加操作管理员账号

BufferDataInit.commonService.saveDate(attendanceInfo);

}

SimpleDateFormat sd = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

try {

long currentTime = sd.parse(LocalTime.getDateTime()).getTime();

long positon\_Time = sd.parse(entry.getValue().getPosTime())

.getTime();

// 判断人员是否离开工事，离开则进行打卡登记

if ((currentTime - positon\_Time) > 10 \* 1000) {// 相差10秒

BufferDataInit.commonService.updateDate(

AttendanceInfo.class, new String[] { "endTime='"

+ LocalTime.getDateTime() + "'" }, "mac='"

+ mac + "' and startTime='"

+ list.get(0).getStartTime() + "'");

}

} catch (ParseException e) {

e.printStackTrace();

}

}

}

}

package com.service.impl;

import java.util.List;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.common.dao.CommonDao;

import com.service.CommonService;

/\*\*

\* 通用service

\* @author ZHUZHIYONG

\*

\*/

@Service

@Transactional

public class CommonServiceImpl implements CommonService{

@Autowired

private CommonDao commonDao;

public CommonDao getCommonDao() {

return commonDao;

}

public void setCommonDao(CommonDao commonDao) {

this.commonDao = commonDao;

}

@SuppressWarnings("rawtypes")

@Override

public long getResultCount(Class className, String subwhere) {

return commonDao.getCountByHql(className, subwhere);

}

@SuppressWarnings("rawtypes")

@Override

public List findByHql(Class className, String subwhere, String orderby) {

return commonDao.findByHql(className, subwhere, orderby);

}

@SuppressWarnings("rawtypes")

@Override

public List findByHqlPage(Class className, String subwhere, String orderby,

Integer currentPage, Integer pageSize) {

return commonDao.findByHqlPage(className, subwhere,orderby, currentPage, pageSize);

}

@SuppressWarnings("rawtypes")

@Override

public Object[] getResult(Class className, String subwhere, String orderby,

int current, int pagesize) {

//创建数据

Object[] obj = new Object[2];

//获取总数

obj[0] = getResultCount(className, subwhere);

//获取数据集

obj[1] = findByHqlPage(className, subwhere,orderby, current, pagesize);

return obj;

}

@SuppressWarnings("rawtypes")

@Override

public void deleteData(Class className, String subwhere) {

commonDao.deleteTableMoreData(className, subwhere);

}

@SuppressWarnings("rawtypes")

@Override

public void updateDate(Class className, String[] setHql, String subwhere) {

String hql="";

//添加修改语句

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

hql += setHql[i]+" ,";

}

if(!"".equals(hql))hql = hql.substring(0,hql.length()-1);

commonDao.updateTableMoreData(className, hql, subwhere);

}

@Override

public void saveDate(Object object) {

commonDao.saveTable(object);

}

@SuppressWarnings("rawtypes")

@Override

public List findByHqlGetColumn(Class className, String subwhere,

String orderby, String column) {

return commonDao.findByHqlGetColumn(className, subwhere, orderby, column);

}

@Override

public Object executeSql(String sql){

return commonDao.executeSql(sql);

}

}

**package** com.service;

**import** java.util.List;

/\*\*

\*

\* service操作

\* **@author** zhangbin

\*

\*/

@SuppressWarnings("rawtypes")

**public** **interface** CommonService {

//hql分页查询数量

**public** **long** getResultCount(Class className,String subwhere);

//根据hql语句查询数据

**public** List findByHql(Class className,String subwhere,String orderby);

//根据hql语句分页查询数据

**public** List findByHqlPage(Class className,String subwhere,String orderby,Integer currentPage,Integer pageSize);

//hql分页

**public** Object[] getResult(Class className,String subwhere,String orderby,**int** current,**int** pagesize);

//批量删除数据

**public** **void** deleteData(Class className,String subwhere);

//批量修改数据

**public** **void** updateDate(Class className,String[] setHql,String subwhere);

//添加数据库信息

**void** saveDate(Object object);

**public** List findByHqlGetColumn(Class className,String subwhere,String orderby,String column);

Object executeSql(String sql);

}

/\*\*

\* 定位标签第三个按键事件 服务取消

\* @author ZHUZHIYONG

\*/

package com.ps;

import com.ps.Host\_ds\_def.HOST\_MSG\_TYPE;

public class Station\_status\_event\_msg extends Host\_protocol\_base {

public Station\_status\_event\_msg(HOST\_MSG\_TYPE msg\_type) {

super(msg\_type);

// TODO Auto-generated constructor stub

}

@Override

public boolean accept(Host\_protocol\_visitor visitor) {

// TODO Auto-generated method stub

return visitor.visit(this);

}

@Override

public Host\_protocol\_base clone() {

// TODO Auto-generated method stub

return new Station\_status\_event\_msg(HOST\_MSG\_TYPE.STATION\_STATUS\_EVENT);

}

}

**package** com.ps;

**import** com.socket.PositionServerUtils;

**public** **class** Host\_protocol\_decoder **extends** Host\_protocol\_visitor {

**private** String received\_data;

**public** Host\_protocol\_decoder(String received\_data) {

// **TODO** Auto-generated constructor stub

**this**.received\_data = received\_data;

}

/\*\*

\* 定位标签第一个按键报警解析 紧急求助

\* **@author** ZHUZHIYONG

\*/

@Override

**public** **boolean** visit(Target\_button1\_event\_msg target\_button1\_event\_msg) {

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

PositionServerUtils.*button1\_alarm\_handler*(received\_data);

**return** **true**;

}

/\*\*

\* 定位标签第二个按键解析 服务签到

\* **@author** ZHUZHIYONG

\*/

@Override

**public** **boolean** visit(Target\_button2\_event\_msg target\_button2\_event\_msg) {

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

PositionServerUtils.*button2\_alarm\_handler*(received\_data);

**return** **true**;

}

/\*\*

\* 基站事件

\* **@author** ZHUZHIYONG

\*/

@Override

**public** **boolean** visit(Station\_status\_event\_msg station\_status\_event\_msg) {

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

PositionServerUtils.*station\_alarm\_handler*(received\_data);

**return** **true**;

}

}